

Photorealistic Rendering for Augmented Reality: A Global Illumination and BRDF Solution

Saulo Pessoa*

Guilherme Moura†

João Lima‡

Veronica Teichrieb§

Judith Kelner¶

Virtual Reality and Multimedia Research Group
Informatics Center, Federal University of Pernambuco, Brazil

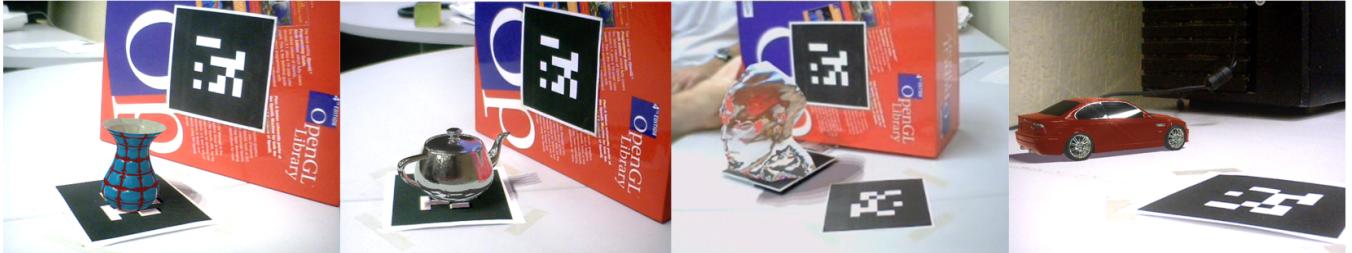


Figure 1: Virtual objects being coherently inserted into a real scene. From left to right, a porcelain vase with consistent illumination, a chrome teapot reflecting surrounding real objects, a glass bust refracting real objects behind it, and a car model exhibiting various materials.

ABSTRACT

This paper presents a solution for the photorealistic rendering of synthetic objects into dynamic real scenes, in Augmented Reality applications. In order to achieve this goal, an Image Based Lighting approach is used, where environment maps with different levels of glossiness are generated for each virtual object in the scene at every frame. Due to this, illumination effects, such as color bleeding and specular reflections, can be simulated for virtual objects in a consistent way. A unifying sampling method for the spherical harmonics transformation pass is also used. It is independent of map format and does not need to apply different weights for each sample. The developed technique is combined with an extended version of Lafourte Spatial BRDF, featuring Fresnel effect and an innovative tangent rotation parameterization. The solution is evaluated in various Augmented Reality case studies, where other features like shadowing and lens effects are also exploited.

Index Terms: H.5.1 [Information Interfaces and Presentation]: Multimedia Information Systems—Artificial, Augmented, and Virtual Realities; I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Color, Shading, Shadowing, and Texture

1 INTRODUCTION

Realism is a desired feature in any Computer Graphics application. According to Ferwerda [12], three types of realism may be considered in Computer Graphics: physical realism, where the aspect and behavior of the elements of a scene are precise with respect to the laws of Physics; photorealism, where the scene is visually compelling, although not necessarily correct in terms of Physics; and functional realism, where the main concern is providing useful

information about the task represented by the scene. In the Augmented Reality (AR) domain, while functional realism is more suitable to some systems, such as industrial ones [16], many application domains take profit from photorealism. Two examples are interior design [14] and advertising [4].

Illumination is one of the key aspects that require careful consideration by photorealistic AR. There are two approaches for handling scene illumination: local, where spotlights, directional or point light sources are used; and global, where environment maps are used. Global methods are usually more complex than local ones, but they can handle indirect illumination, therefore generating more realistic results. Another important factor in photorealistic AR is the surface reflectance modeling. Bidirectional Reflectance Distribution Functions (BRDFs) [32] are often used to perform this task, since some of them allow the modeling of several objects made from different materials in a very realistic manner.

This paper contributes investigating Computer Graphics techniques that are suitable to solve the photorealistic AR problem. These techniques are placed together in an all-in-one solution in order to achieve seamlessly insertion of virtual objects. Some extensions to the adopted Image Based Lighting (IBL) and BRDF approaches are also proposed. King's IBL [23] was adopted and extended to generate two additional levels of blurred environment map. These are later used by the BRDF in order to render materials that exhibit glossy appearance. It is worth reminding that, differently from McAllister [27], every environment map here is blurred on the fly. This last feature is what support dynamic specular and diffuse (light bleeding) reflections. The utilization of a different map sampling scheme for the spherical harmonics transformation is also proposed. It has the advantages of being both simpler and more general. In addition, instead of using a simple Lambertian BRDF, the Lafourte Spatial BRDF (SBRDF) [27] was chosen. This SBRDF was extended to support tangent rotation parameterization and the Fresnel effect. Although McAllister results suggest support to tangent rotation, the lack details motivated a custom solution proposed in this paper. Figure 1 exhibits some of the obtained results.

This paper is organized as follows. Section 2 presents some previous work about the topics hereby mentioned and explains how the contributions of this work are related to them. Section 3 de-

*e-mail:sap@cin.ufpe.br

†e-mail:gsm@cin.ufpe.br

‡e-mail:jpsml@cin.ufpe.br

§e-mail:vt@cin.ufpe.br

¶e-mail:jk@cin.ufpe.br

tails the novel implemented features regarding global illumination and BRDF. Section 4 describes the entire photorealistic AR solution developed as part of this effort. The results obtained from the implemented novel features, as well as the use of some AR case studies based on the proposed solution, are highlighted in Section 5. Finally, Section 6 draws some conclusions and outlines future work.

2 RELATED WORK

There are plenty of issues involved in photorealistic AR. In order to justify the use, or lack of use, of photorealistic AR, a number of researches on human perception were undertaken. The general consensus that emerged is that while visual realism may not be directly related to productivity in virtual environments, it does nonetheless increase the virtual world's sense of presence [41]. Certain realistic rendering effects, such as shadows, have been studied in particular to demonstrate their importance in determining the presence of virtual objects [37]. However, research on perceptually based rendering has shown that there are certain visual effects that humans are insensitive to [13]. Clearly, there is still a need for further studies about the effects of realism on user's perception of the AR domain. On the other hand, there is a significant amount of evidence suggesting that enhanced realism often gives the user a greater sense of presence of virtual objects within the real world, hence justifying time and efforts spent in its research.

Seamlessly inserting virtual objects into real scenes is a topic that has attracted researchers' attention since the last decade. Although, due to graphics hardware limitations, research focus was predominantly on non real-time solutions. Debevec and Malik have significantly contributed to this area, showing how to recover High Dynamic Range (HDR) radiance maps from photographs [8]. Debevec has composed real and virtual scenes using a Differential Rendering (DR) technique [7], and has shown how the IBL technique may be used to illuminate synthetic objects with measurements of real light [6]. Since IBL techniques utilize an accurate measurement of real light, Debevec successfully made virtual objects appear as if they were actually in a real-world scene. However, the lack of a real-time solution suitable to AR applications remained.

Nowadays, modern GPU processing capacity and its new programming features made feasible, at interactive rates, techniques that used to be suitable only for non real-time applications. IBL is an example of such techniques that can approximate a global illumination effect through the use of HDR environment maps. Ramamoorthi and Hanrahan showed an efficient representation for irradiance environment maps (also called diffuse environment maps) using spherical harmonics [30]. In order to boost performance, King [23] and Kautz et al. [22] have ported that representation to GPU. King used a sampling scheme that is dependent on the environment map format, which requires a weighting procedure during the spherical harmonics transformation pass. This is not only complex but also depends on the environment map format. King merely used the environment map itself and the generated diffuse environment map, thus the simulation was confined to only completely diffuse or specular objects, as well as an average effect blending both maps. Kautz et al. generated an additional glossy environment map using a box filter as a way to expand the range of supported materials. This paper takes a step further by incorporating a more representative reflectance scheme.

Kanbara and Yokoya [21] integrated in an AR system a technique for geometric and photometric registration. Photometric registration was made to identify real light sources reflected on a small black sphere placed over a marker. The small black sphere was used in order to handle the limitations of working with a Low Dynamic Range (LDR) video stream. Using this approach, Kanbara achieved a consistent diffuse illumination, including shadowing, between real and virtual worlds. Supan et al. [38] obtained sim-

ilar results using two different setups in order to acquire incident lightning: placing a perfect mirrored sphere at the scene or using a camera with fisheye lens. Their advantage is that they can also exhibit specular reflections since incident lightning is acquired in a higher resolution. Madsen and Laursen [26] have also incorporated photometric registration in an AR system using pre-acquired HDR environment maps. Grosch et al. [15], on the other hand, use a HDR camera to capture the environment in real-time. Their solution, however, limits the results to simulate indirect lighting in a controlled Cornell Box. Although exhibiting pleasant results, these implementations lack a more sophisticated way to simulate surfaces reflectance properties.

Lafortune et al. proposed a simple, yet powerful, and general BRDF representation [24], although running only on non real-time applications. McAllister et al. have added to the Lafortune's representation the capability of spatially varying BRDF parameters (SBRDF) and have ported it to GPU [27]. This approach also supported both local and global illumination models although the latter only handled static environment maps, since they were pre-filtered. [25] used a similar approach for fitting real photographs of captured materials by a specific parametric function. However, they chose Phong's model which is much simpler than Lafortune's.

On of this paper's contribution lies in extending the work presented in [27] to support dynamic environments in order to make it suitable for AR applications. This is achieved by filtering environment maps during application execution. This feature allows some effects to be simulated, like light bleeding between virtual objects themselves and between virtual and real objects may be simulated. In addition, some new features were incorporated into the SBRDF, namely, a Fresnel approximation term and support for tangent rotation parameterization. A different environment map sampling approach was also used, allowing some simplification and generalizing the way different environment maps are handled during spherical harmonics transformation pass. Heymann et al. in [20] also achieved light bleeding effect although their approach do not take benefits from spherical harmonics. He performed environmental map filtering within the spatial domain, which is prone to performance penalties.

3 NOVEL FEATURES

This section presents the innovative work that takes a step beyond both [23] and [27].

3.1 SBRDF Extension

Lafortune's BRDF representation [24] is a compact and general way to represent surface reflectance properties. It is suitable to both local and global illumination models. Although McAllister et al. had already improved such representation in order to support spatial variation, the work presented in this paper extends their ideas to achieve more representative and realistic results. Such gain is accomplished through the improvement of two existing features: anisotropic reflection and the Fresnel factor. McAllister et al. approximate space-varying direction of anisotropy and the Fresnel effect through the use of multiple lobes per material. This work proposes different methods to achieve these effects, using a single lobe and increasing the quality of the final result.

The first extension, i.e., the tangent rotation map, is necessary due to the presence of anisotropic materials where the direction of anisotropy is frequently associated to the tangent vector. Therefore, a tangent rotation map (see Figure 2a) was conceived as a monochromatic texture where the color stored in a given position represents the angle through which the tangent vector of a point on the object surface should rotate around the normal vector. The values stored in the texture are mapped to the range $[0, 2\pi]$. This approach allows per-pixel tangent variation, expanding the range of

materials reproduced by this solution, e.g., that of a circular brushed metal. This type of material is illustrated in Figure 2b.

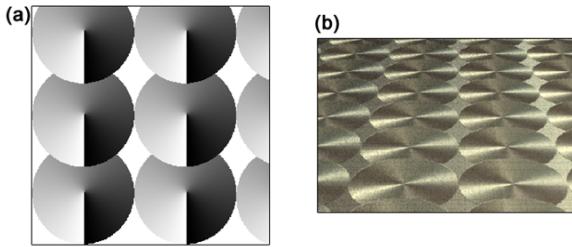


Figure 2: Tangent rotation map (a) and an example of circular brushed metal material (b).

It is important to notice that bilinear filtering of this map, when it has discrepant neighbor texels values, may produce undesirable artifacts. The results from their combination do not necessarily represent a valid interpolated rotation.

The other important extension added to the SBRDF presented in [27] is the improvement of the Fresnel factor approximation. This attempts to imitate the perfect reflections that arise when incident light occurs at grazing angles.

For the current solution, only one specular lobe is being used, thus the original SBRDF representation was simplified to the following:

$$L_r(R) = \rho_d D(N) + \rho_s S(p(R), n) g(R), \quad (1)$$

where $L_r(R)$ is the radiance coming from direction R , ρ_d and ρ_s represent respectively diffuse and specular albedo, $D(N)$ is the irradiance over normal direction N (it is sampled from diffuse environment map), $S(p(R), n)$ represents the specular environment maps that will be sampled according to the glossiness exponent n (0 for completely diffuse appearance and 1 for a perfect mirror). $p(R)$ is the peak vector calculated by:

$$p(R) = \begin{bmatrix} C_x & 0 & 0 \\ 0 & C_y & 0 \\ 0 & 0 & C_z \end{bmatrix} \cdot \begin{bmatrix} R_x \\ R_y \\ R_z \end{bmatrix}, \quad (2)$$

where C_x , C_y , and C_z are the Lafortune SBRDF parameters [27]. Finally, $g(R)$ is the term that avoids radiance from below the surface from being included in $L_r(R)$ when low value glossiness exponents are used. This last term is referred to as the irradiance falloff, and is given by:

$$g(R) = N \cdot p(R). \quad (3)$$

In order to include the Fresnel term in Equation 1, Schlick's approximation [33] was extended to allow a parameterization of the factor, resulting in the following Fresnel contribution:

$$f(R) = (1 - (R \cdot N))^{\left(\frac{1}{\varphi} - 1\right)}, \quad (4)$$

where $f(R)$ is the resultant Fresnel contribution ranging in the interval $[0, 1]$ (0 indicates no contribution and 1 indicates maximum contribution) and φ is the falloff parameter ranging in the interval $(0, 1]$ (close to 0 values tend to give contribution only at grazing angles regions whereas close to 1 values tend to spread the contribution towards no grazing angles regions). The result of Equation 4 is then used to generate new terms for the specular albedo ρ_s , the glossiness exponent n , and the irradiance falloff $g(R)$, giving respectively:

$$\rho_{s(new)} = \rho_s + (1 - \rho_s)f(R), \quad (5)$$

$$n_{(new)} = n + (1 - n)f(R), \quad (6)$$

and

$$g(R)_{(new)} = g(R) + (1 - g(R))f(R). \quad (7)$$

Since the glossiness exponent increases due to the Fresnel contribution, the irradiance falloff has to be softened accordingly, as expressed in Equation 7. At the end, replacing the old terms by these new ones in Equation 1 will give the final SBRDF expression.

3.2 IBL Sampling

In the proposed approach, the way that King [23] samples the environment map was changed in order to make easier the spherical harmonic transformation pass. King samples texel by texel, for example, if it is a cubic environment map, then he spends six full screen quad passes to sample the whole map (one for each face). When it is a dual paraboloid map, then King's strategy spends two passes. The main difficulty encountered by such sampling scheme is that each sample will be subtended by a different solid angle, which consequently implies in using a different weight for each sample. Furthermore, samples weights will also vary according to each environment map format, for example, the place with the highest density of samples in a cubic environment map is close to the cube edges, whereas in a latitude longitude environment map it is close to the poles. In order to avoid such difficulties and generalize the sampling operation, a uniform spaced sampling scheme was used. Using such scheme, the discrete spherical harmonics transformation becomes:

$$L_l^m = \sum_{\Omega} L(\omega) y_l^m(\omega) \frac{4\pi}{n}, \quad (8)$$

where L_l^m is a coefficient that will be generated in the intervals $0 \leq l \leq 15$ (since 256 coefficients are needed as shown in Section 3.3), and $-l \leq m \leq l$. Ω is the set of every direction in the 3D space, ω is a direction used to sample the HDR environment map and a spherical harmonic base function, L is the HDR environment map, y_l^m is a spherical harmonic base function, n is the number of samples, and the factor $\frac{4\pi}{n}$ is a constant weight since the amount of samples n is previously known. In order to generate the evenly spaced directions ω , a mapping from uniformly spaced 2D coordinates into two angles was used. These two angles represent a direction into spherical coordinates and the mapping is performed using the following expression:

$$(x, y) \rightarrow (2 \arccos(\sqrt{1-x}), 2\pi y) \rightarrow (\theta, \phi), \quad (9)$$

where (x, y) are evenly spaced 2D points over the unit square, and (θ, ϕ) represents the direction into a 3D space [35]. This process generates directions similar to the intersections points between parallels and meridians lines in the geographic coordinate system, but with a sparser distribution of directions close to the poles (see Figure 3(b)). As the proposed sampling scheme does not guarantee that every environment map texel will be sampled, some aliasing may occur depending on the amount of samples performed. In order to reduce this, stratified sampling may be done. This is achieved by randomly jittering each evenly spaced 2D points into their respective bounding regions. These bounding regions are same sized and non overlapping regions that surround each 2D point (see Figure 3(a)). Using such idea, part of the aliasing that would appear is replaced by some noise that is less unpleasant to human eyes. An alternative way to generate evenly spaced direction could be using an isocube representation [40], however, without any significant visual quality gain and performance penalties due to its bigger expression.

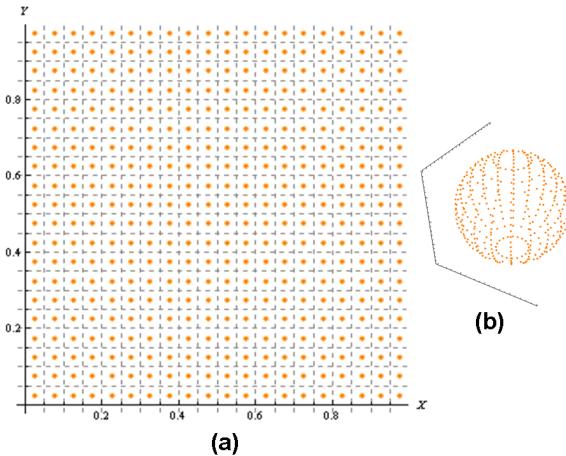


Figure 3: The mapping from 2D to 3D coordinate systems in order to generate the evenly spaced directions. 400 evenly spaced 2D points with their respective bounding regions in (a) and the resultant mapping using the proposed sampling scheme in (b).

3.3 Dynamic Map Generation

McAllister et al. were able to make Lafortune’s BRDF feasible in real-time using GPUs although environment maps should be static in their nature. Nonetheless, in AR applications, virtual and real objects may move independently, hence virtual objects appearance may also change because of the surrounding scene changes. These changes are usually seen as effects like light bleeding and dynamic specular reflections. In order to support such effects, a new environment map is captured from the center of each virtual object, so each one will have its own environment map. This environment map is acquired by hiding its owner and rendering the surrounding scene from the owner’s center. The surrounding scene is composed by the other virtual objects, the phantom objects, and the skybox with the real environment map. A phantom object is a pre-modeled representation of a real object. It is used to capture interactions between real and virtual objects, such as shadows, interreflections, and occlusions. Six passes are required in order to generate a cubic map although a dual paraboloid map [18] may also be used, requiring only two passes.

The generated environment map can only support the rendering of perfectly reflexive objects, so, in order to extend it some new maps are first generated. Actually, one diffuse and two glossy environment maps are also generated, hence, four environment maps will finally be used to illuminate virtual objects. These maps are interpolated in order to simulate a large variety of materials, for example, different levels of glossy materials are simulated using their glossiness exponent as a parameter to weight each generated environment map.

By exchanging the sampling scheme with the one suggested in last section, King’s technique was used to generate the diffuse environment map. According to Ramamoorthi and Hanrahan [30], 9 spherical harmonics coefficients (degree 2) are enough to generate a diffuse environment map. By extending this idea, the first glossy environment map was also generated, but now using 256 coefficients (degree 15). In order to boost performance, the spherical harmonics basis functions were precomputed in Wolfram Mathematica and stored into a texture. Figure 4 shows the arrangement used to store them. Since the diffuse and the first glossy environment maps tend to be very blurred, they could be well and efficiently made in the frequency domain, although the same does not happen again to the second glossy environment map. Since it has relatively higher frequency information, its convolution in the fre-

quency space would be hard and generates bad quality results. It is therefore preferably carried out in the spatial domain through a brute force convolution. In this case, a Summed-Area Table (SAT) technique [19] was also used in an attempt to enhance the overall performance. However, because of the overhead needed to generate the table at every frame and the undesirable artifacts generated by the box filtering, this approach was abandoned. The size of each generated environment map may also be determined based on its type. As the diffuse and the first glossy environment maps are more blurred than the second glossy and the specular maps, the first ones may have smaller sizes. Although this reduction boosts performance, it also makes texels’ bounds clearly visible. Texture bilinear filtering is then used to correct this drawback. Figure 5 shows the generated environment maps.

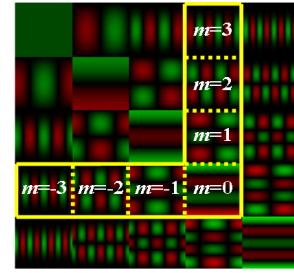


Figure 4: Organization into a texture of the preprocessed spherical harmonics basis functions. This figure shows functions until fifth degree ($l = 4$), where the fourth ($l = 3$) is highlighted. The order m ranging from $-l = -3$ to $l = 3$ is also shown. Functions are in latitude longitude format and each one occupies one tile. Since they are normalized, values stored range from -1 to 1 . It is important to highlight that this texture may not be bilinear sampled since padding was not used.

In addition to the trivial idea of putting each generated environment map into a distinct texture, there is also the possibility of storing each one into a different MIP-MAP level of the same texture. These levels may also be interpolated using built-in derivative texture sampling functions present on modern GPU shader languages.

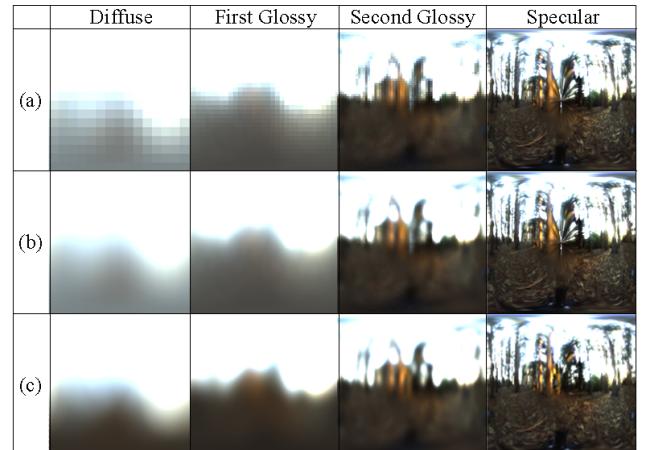


Figure 5: In (a), the generated diffuse, first glossy, second glossy, and specular environment maps. Their sizes are respectively 16×16 , 32×32 , 64×64 , and 128×128 . In (b), the resultant images when maps in (a) are bilinear sampled. In (c), the same environment maps generated with the HDR Shop.

4 PROPOSED SOLUTION

This paper presents a solution where many different approaches to realistic rendering are combined in a single pipeline. Such feat was achieved using a composition of shaders which do most of the hard calculations on GPU, leaving only the program creation to the CPU, in short terms. The final goal is to build a more realistic rendered object mixed seamlessly with the real world environment. The proposed and implemented technique explored two important aspects related to realism.

The first challenge to overcome was the need for global illumination. In order to fit into the real world illumination, the virtual object should receive light from the same sources as that of the real objects present in the scene. This feature was accomplished through the use of IBL, employing the improvements discussed in Section 3. This way, images from the room where the camera would shoot were previously captured in HDR, and then applied to a skybox in order to illuminate virtual objects (Figure 6(a)). The drawback of this solution is that virtual objects will only be consistently illuminated when the user is close to the place where the real environment map was captured. In order to solve this issue, it would be necessary to implement a real-time HDR acquisition procedure capable of continuously update the environment map during user interaction, as described by [39].

The IBL implementation used cannot generate shadows, both projected and self-shadowing. Therefore, the standard shadow mapping technique implemented in OGRE 3D [5] was used as a short-term solution to generate the projected shadows. The virtual light sources were manually positioned trying to mimic the position of the most significant lights in the real world (Figure 6(b)). Currently there is no self-shadowing generation in this solution; however, an Ambient Occlusion [34] technique is under development, as explained in Section 6.

Additionally, in the real world, intense light sources usually generate effects, e.g., Bloom and Glare, upon interaction with the camera lens or the human eye [36]. These effects can also appear on the object surface, when it sufficiently reflects the surrounding environment. This way, in order to appear as much realistic as possible, virtual objects should reproduce the same effects. The generation of such effects was implemented in a post-processing step and was based on Kawase's work [3]. The integration of these effects in the final solution accomplished a better merging between the virtual object and the real environment, since the object's silhouette is better smoothed through the use of this technique.

Another important feature that was implemented in this solution is the ability to adjust scene exposure based on its radiance. This technique is often called Exposure Control, and is currently implemented in every camera software, as well as being a natural ability of human eyes. This effect is accomplished through the use of a tone mapping algorithm [31], which tends to darken the scene when it is too bright.

However, in conjunction with King's IBL technique, it was only possible to reproduce a small set of objects, with regard to their appearance, i.e., completely diffuse, mirrors, and plastic-like objects. In order to expand the set of objects that could be realistically rendered by this solution, it was necessary to add support for a more complex BRDF representation. Consequently, the SBRDF presented in [27] and the extensions mentioned in Section 3 were implemented. Furthermore, other parameters were included in the material color computation. It is possible, for instance, to decrease the object opacity value and create transparent materials, which can also be distorted by a refraction index and a color dispersion value. Moreover, the material surface can be detailed by a normal map passed as a parameter alongside with a weight value.

Finally, in order to assemble this complex pipeline into an AR application, a technique called DR was implemented, as described by Debevec [7].

The final composition of these techniques is an AR solution for applications that require realistic objects inserted in the real world. It was implemented in C++ using OGRE 3D resources. The shading language was Cg. Marker tracking is performed using [10], which allows easily changing the tracking technique applied.

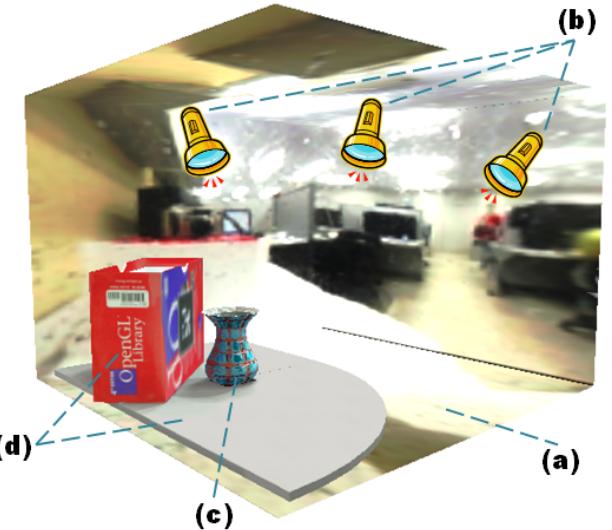


Figure 6: Illustration of the internal setup used to generate porcelain vase scene in Figure 1. (a) is the previously captured environment map applied to a skybox, (b) are the virtual light sources manually positioned trying to mimic the position of the most significant lights in the real world, (c) is the virtual porcelain vase, and (d) are the phantom objects relative to the table and OpenGL red box.

5 RESULTS

The results presented in this section were captured using a resolution of 1024×768 pixels directly from the AR system running in real-time. Hardware used was a PC equipped with an AMD Athlon 64 3200+ processor, 2GB of RAM memory, and a GeForce 8800 GTX graphics card with 768MB of memory. Some third parties solutions were used to build the AR system, like ARToolKitPlus [2] for camera and objects pose tracking and OGRE 3D rendering engine. Real ambient lighting was previously probed through a photographic camera and a chrome sphere. The chrome sphere was photographed with different exposure levels in order to generate an HDR image in the HDR Shop tool [1], as depicted in Figure 7.

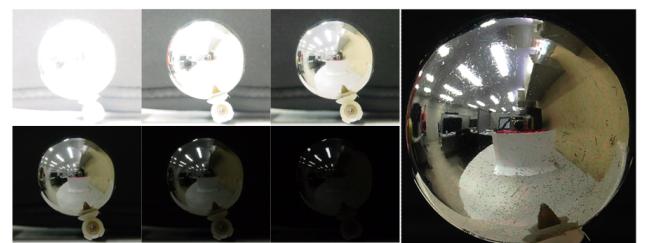


Figure 7: On the left, the chrome sphere shot with different exposure levels, and on the right the resultant HDR spherical environment map generated with HDR Shop.

As explained in Section 3, the addition of these new features to the SBRDF turned it more general and greatly improved the realism of the virtual objects inserted in the real world. Figure 8 illustrates

the results of the application of a tangent map to the object's mesh, in this case, a metallic table. This type of material can be easily represented by the extended SBRDF through the use an extra texture (see Figure 8).

In addition, as a subtle improvement, the Fresnel factor helps the object to blend realistically with the scene background. This effect can better represent specific types of materials, such as marble or glossy surfaces. As one can see from Figure 9, the object specularity increases at grazing angles, reflecting the red box on the top of the teapot and the gray table beneath its bottom part.

In addition to these new features, some effects from third parties (mentioned in Section 4) completed the range of objects that can be represented by this solution. Some of these effects, however, have a special appeal when used in the AR context, e.g., transparency, as seen in Figure 10. This implementation uses the camera stream in order to give a realistic shading to the virtual object. Although, when a virtual object is close to the screen border or when the refraction index is very high, the refracted ray may point towards locations outside the screen bounds, where there is no camera stream information. This leads to some strange shading effects that may be alleviated by enabling clamping in the camera stream texture.

Another type of material available in the proposed solution that is often encountered in the real world is the class of retro-reflective objects, e.g., traffic signs or safety clothing. Figure 11 shows an example of this kind of material generated by the SBRDF. It can be noticed here, that when the camera rises, some parts of the object begin to glow, since the camera moves towards the same direction of the most relevant light sources. Thus, the major part of light from the environment returns in the same direction, reaching the camera lens, instead of reflecting in the opposite direction.

Another feature implemented in this solution is a normal mapping technique (as shown in Figure 12), which increases the object's level of detail without needing any additional geometry.

A subtle effect that is usually observed in the real world happens when color bleeds from objects to others. This effect was achieved in the AR solution through the generation of environment maps for each virtual object. Notice that the real red box presented in Figure 13 has also a marker on it in order for its respective phantom object to be coherently registered. Figure 13 illustrates this effect, where the real red box bleeds its color over the virtual statue model. The inverse bleeding would also be achieved generating environment maps for the OpenGL red box phantom object and proceeding with the DR technique. The interaction between real and virtual objects is the most relevant byproduct of DR. Every change introduced in the real world due to virtual objects addition is achieved applying this technique, including shadows.

In addition to color bleeding, realistic mirror like reflections and occlusion were also simulated (see Figure 14). In the reflection case, the virtual teapot reflected real objects surrounding it, such as the red box and the table where objects were placed. In the occlusion case the virtual statue and its shadows are occluded by the red box. Occlusion was achieved through the rendering of the phantom objects to the depth buffer as described in [17]. In both cases a phantom of each real object was used.

A more complex virtual object is shown in Figure 15. It exhibits a bigger variety of materials, like chrome metal (front up wire), brushed metal (wheels), glass, plastic (front down wire), rubber (tires), polished (painting), and emissive material (headlights). Glare and Bloom are exhibited in the car headlights since a HDR texture with high texel values was mapped onto it. The car mesh has 9319 triangular faces and 10795 vertices. This example, like the previous ones, executed at around 14 fps. It happens due to fact that the proposed solution cost depends mostly on the number of convoluted environment maps (or the number of objects), and not on the geometry complexity. More precise performance evaluation of the environmental maps generation may be found in [28].

6 CONCLUSIONS AND FUTURE WORK

The implemented photorealistic AR solution allows consistent rendering of dynamic scenes and sophisticated materials. In addition, features such as lens effects and shadowing are also supported, contributing to a richer final result. The visual aspect of the generated scenes was satisfactory, as well as the frame rates achieved, which were suitable for AR applications.

However, there is a need for a more tailored approach for shadowing. A Screen-Space Ambient Occlusion (SSAO) technique [34] is currently under development, in order to give support to self-shadowing. The implementation of a shadow mapping technique that makes use of environment maps, such as [11], is also planned.

With regard to environment map generation, geometry shaders could be used to generate a cubic map in a single pass. In addition, a technique for building the environment map of the real world in real-time [9] could be adapted to generate HDR maps.

In order to facilitate the development of photorealistic AR applications, an Application Programming Interface (API) based on OGRE is under development. Since environment tracking will be abstracted by the [10] library, it will be possible to employ different techniques for registering virtual and real worlds, like markerless tracking [29]. A material editor is also under development, where the designer/artist will be able to set all the parameters of the object material through a graphical interface, visualize the results in real-time and export the project to a configuration file that will be later used by the programmer to create the AR application.

REFERENCES

- [1] Hdr shop home. HDR Shop, June 2009. [online] <http://gl.ict.usc.edu/HDRShop/>.
- [2] Home page - artoolkitplus. ARToolKitPlus, June 2009. [online] http://studierstube.icg.tu-graz.ac.at/handheld_ar/artoolkitplus.php.
- [3] Home page - masaki kawase. M. Kawase, June 2009. [online] <http://www.daionet.gr.jp/~masa/>.
- [4] Interactive kiosk. TotalImmersion, June 2009. [online] <http://www.t-immersion.com>.
- [5] Ogre: Open source graphics engine. OGRE 3D, June 2009. [online] <http://www.ogre3d.org/>.
- [6] P.Debevec. Image-based lighting. In *SIGGRAPH '06: ACM SIGGRAPH 2006 Courses*, page 4, New York, NY, USA, 2006. ACM.
- [7] P.Debevec. Rendering synthetic objects into real scenes: bridging traditional and image-based graphics with global illumination and high dynamic range photography. In *SIGGRAPH '08: ACM SIGGRAPH 2008 classes*, pages 1–10, New York, NY, USA, 2008. ACM.
- [8] P. E. Debevec and J. Malik. Recovering high dynamic range radiance maps from photographs. In *SIGGRAPH '08: ACM SIGGRAPH 2008 classes*, pages 1–10, New York, NY, USA, 2008. ACM.
- [9] S. DiVerdi, J. Wither, and T. Hllerer. All around the map: Online spherical panorama construction. *Computers & Graphics*, 33(1):73 – 84, 2009.
- [10] T. Farias, J. Lima, V. Teichrieb, and J. Kelner. Ogrear: Construction of augmented reality applications using high-level libraries. Technical report, Federal University of Pernambuco, 2007.
- [11] R. Fernando. Percentage-closer soft shadows. In *SIGGRAPH '05: ACM SIGGRAPH 2005 Sketches*, page 35, New York, NY, USA, 2005. ACM.
- [12] J. A. Ferwerda. Three varieties of realism in computer graphics. In B. E. Rogowitz and T. N. Pappas, editors, *Proceedings of Human Vision and Electronic Imaging VIII (January 21, 2003, Santa Clara, California, USA)*, volume 5007 of *SPIE Proceedings Series*, pages 290–297, Bellingham, Washington, 2003. SPIE/IS&T.
- [13] J. A. Ferwerda, P. Shirley, S. N. Pattanaik, and D. P. Greenberg. A model of visual masking for computer graphics. In *SIGGRAPH '97: Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pages 143–152, New York, NY, USA, 1997. ACM Press/Addison-Wesley Publishing Co.
- [14] S. Gibson, J. Cook, T. Howard, and R. Hubbold. Rapid shadow generation in real-world lighting environments. In *EGRW '03: Proceedings*

- of the 14th Eurographics workshop on Rendering*, pages 219–229, Aire-la-Ville, Switzerland, Switzerland, 2003. Eurographics Association.
- [15] T. Grosch, T. Eble, and S. Mueller. Consistent interactive augmentation of live camera images with correct near-field illumination. In *VRST '07: Proceedings of the 2007 ACM symposium on Virtual reality software and technology*, pages 125–132, New York, NY, USA, 2007. ACM.
 - [16] M. Haller. Photorealism or/and non-photorealism in augmented reality. In *VRCAI '04: Proceedings of the 2004 ACM SIGGRAPH international conference on Virtual Reality continuum and its applications in industry*, pages 189–196, New York, NY, USA, 2004. ACM.
 - [17] M. Haller, S. Drab, and W. Hartmann. A real-time shadow approach for an augmented reality application using shadow volumes. In *VRST '03: Proceedings of the ACM symposium on Virtual reality software and technology*, pages 56–65, New York, NY, USA, 2003. ACM.
 - [18] W. Heidrich and H.-P. Seidel. View-independent environment maps. In *HWWS '98: Proceedings of the ACM SIGGRAPH/EUROGRAPHICS workshop on Graphics hardware*, pages 39–ff., New York, NY, USA, 1998. ACM.
 - [19] J. Hensley, T. Scheuermann, G. Coombe, M. Singh, and A. Lastra. Fast summed-area table generation and its applications. *Computer Graphics Forum*, 24:547–555, 2005.
 - [20] S. Heymann, A. Smolic, K. Müller, and B. Froehlich. Illumination reconstruction from real-time video for interactive augmented reality. In *Proc. of International Workshop on Image Analysis for Multimedia Interactive Services*, 2005.
 - [21] M. Kanbara and N. Yokoya. Real-time estimation of light source environment for photorealistic augmented reality. In *ICPR '04: Proceedings of the Pattern Recognition, 17th International Conference on (ICPR'04) Volume 2*, pages 911–914, Washington, DC, USA, 2004. IEEE Computer Society.
 - [22] J. Kautz, K. Daubert, and H.-P. Seidel. Advanced environment mapping in vr applications. *Computers and Graphics*, 28(1):99 – 104, 2004.
 - [23] G. King. *GPU Gems 2: Programming Techniques for High-Performance Graphics and General-Purpose Computation*, chapter 10, pages 167–176. Addison-Wesley Professional, 2005.
 - [24] E. P. F. Lafontaine, S.-C. Foo, K. E. Torrance, and D. P. Greenberg. Non-linear approximation of reflectance functions. In *SIGGRAPH '97: Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pages 117–126, New York, NY, USA, 1997. ACM Press/Addison-Wesley Publishing Co.
 - [25] W.-C. Ma, S.-H. Chao, B.-Y. Chen, C.-F. Chang, M. Ouhyoung, and T. Nishita. An efficient representation of complex materials for real-time rendering. In *VRST '04: Proceedings of the ACM symposium on Virtual reality software and technology*, pages 150–153, New York, NY, USA, 2004. ACM.
 - [26] C. B. Madsen and R. E. Laursen. A scalable gpu-based approach to shading and shadowing for photorealistic real-time augmented reality. In *GRAPP (GM/R)*, pages 252–261, 2007.
 - [27] D. K. McAllister, A. Lastra, and W. Heidrich. Efficient rendering of spatial bi-directional reflectance distribution functions. In *HWWS '02: Proceedings of the ACM SIGGRAPH/EUROGRAPHICS conference on Graphics hardware*, pages 79–88, Aire-la-Ville, Switzerland, Switzerland, 2002. Eurographics Association.
 - [28] S. A. Pessoa, E. L. Apolinário, G. de S. Moura, J. P. S. do M. Lima, M. A. S. Bueno, V. Teichrieb, and J. Kelner. Illumination techniques for photorealistic rendering in augmented reality. In *Proc. Symposium on Virtual and Augmented Reality*, pages 223–232. SBC, 2008.
 - [29] J. Pilet, A. Geiger, P. Lagger, V. Lepetit, and P. Fua. An all-in-one solution to geometric and photometric calibration. In *ISMAR '06*, pages 69–78, Washington, DC, USA, 2006. IEEE Computer Society.
 - [30] R. Ramamoorthi and P. Hanrahan. An efficient representation for irradiance environment maps. In *SIGGRAPH '01: Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 497–500, New York, NY, USA, 2001. ACM.
 - [31] E. Reinhard, M. Stark, P. Shirley, and J. Ferwerda. Photographic tone reproduction for digital images. *ACM Trans. Graph.*, 21(3):267–276, 2002.
 - [32] S. Rusinkiewicz. A survey of brdf representation for computer graphics. Available: Szymon Rusinkiewicz's home page. URL: <http://www.cs.princeton.edu/smrc348c-97/surveypaper.html>, 1997.
 - [33] C. Schlick. An inexpensive brdf model for physically-based rendering. *Computer Graphics Forum*, 13:233–246, 1994.
 - [34] P. Shanmugam and O. Arikán. Hardware accelerated ambient occlusion techniques on gpus. In *I3D '07: Proceedings of the 2007 symposium on Interactive 3D graphics and games*, pages 73–80, New York, NY, USA, 2007. ACM.
 - [35] M. P. B. Slomp, M. M. Oliveira, and D. I. Patrício. A gentle introduction to precomputed radiance transfer. *RITA*, 13(2):131–160, 2006.
 - [36] G. Spencer, P. Shirley, K. Zimmerman, and D. P. Greenberg. Physically-based glare effects for digital images. In *SIGGRAPH '95: Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 325–334, New York, NY, USA, 1995. ACM.
 - [37] N. Sugano, H. Kato, and K. Tachibana. The effects of shadow representation of virtual objects in augmented reality. In *ISMAR '03: Proceedings of the 2nd IEEE/ACM International Symposium on Mixed and Augmented Reality*, page 76, Washington, DC, USA, 2003. IEEE Computer Society.
 - [38] P. Supan, I. Stuppacher, and M. Haller. Image based shadowing in real-time augmented reality. *IJVR*, 5(3):1–7, 2006.
 - [39] J. Waese and P. Debevec. A real-time high dynamic range light probe. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques: Conference Abstracts and Applications*, page 247. ACM Press/Addison-Wesley Publishing Co., 2002.
 - [40] L. Wan, T.-T. Wong, and C.-S. Leung. Isocube: Exploiting the cube-map hardware. *IEEE Transactions on Visualization and Computer Graphics*, 13(4):720–731, 2007.
 - [41] R. Welch, T. Blackmon, A. Liu, B. Mellers, and L. Stark. The effects of pictorial realism, delay of visual feedback, and observer interactivity on the subject sense of presence. *Presence*, 5(3):263–273, 1996.

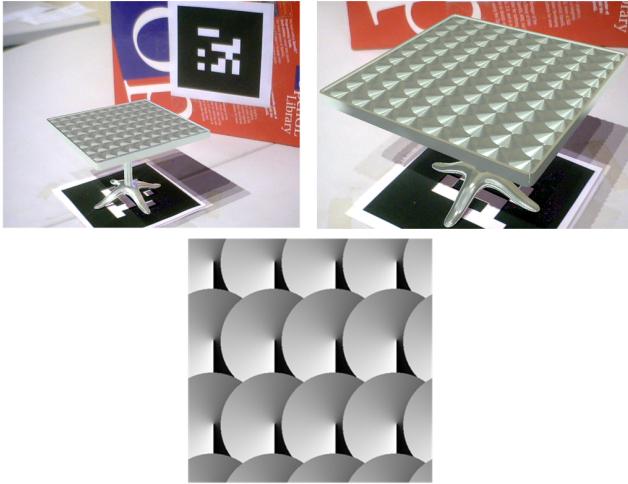


Figure 8: Virtual table inserted into the real world (top-left); surface details showing the tangent rotation (top-right) and the tangent map used (bottom).



Figure 9: Virtual object without the Fresnel term (left) and with the Fresnel term (right).



Figure 10: A transparent elephant object without refraction (left) and with refraction (right). The snapshot with refraction also exhibits some light dispersion.



Figure 11: Example of retro-reflective material.



Figure 12: Normal map disabled (left) and enabled (right).



Figure 13: Color bleeding example.



Figure 14: Interactive reflection (left) and occlusion (right).



Figure 15: A more complex example of virtual object inserted into a real scene.