Photorealistic rendering for augmented reality using environment illumination

Kusuma Agusanto, Li Li, Zhu Chuangui, Ng Wan Sing
Nanyang Technological University
Singapore 627798
{ PF379843, PG02363551, mcgzhu, mwsng } @ntu.edu.sg

Abstract

Mixing 3D computer-generated images with real-scene images seamlessly in augmented reality has many desirable and wide areas of applications such as entertainment, cinematography, design visualization and medical trainings. The challenging task is to make virtual objects blend harmoniously into the real scene and appear as if they are like real. Apart from constructing detailed geometric 3D model representation and obtaining accurate surface properties for virtual objects, adopting real scene lighting information to render virtual objects is another important factor to achieve photorealistic rendering. Such a factor not only improves visual complexity of virtual objects, but also determines the consistency of illumination between the virtual objects and the surrounding real objects in the scene. Conventional rendering techniques such as ray tracing, and radiosity require intensive computation and data preparation to solve the lighting transport equation. Hence, they are less practical for rendering virtual objects in augmented reality, which demands a real-time performance. This work explores an image-based and hardware-based approach to improve photorealism for rendering synthetic objects in augmented reality. It uses a recent technique of image-based lighting, environment illumination maps, and a simple yet practical multi-pass rendering framework for augmented reality.

1 Introduction

A seamless integration between virtual objects and real environment in real time is one of the desired goals of research and development in the AR field. This simply means synthetic objects are perfectly blended with real image background. User perceives computer-generated imagery indistinguishable from real ones as well as a part of surrounding real scenery. Besides matching geometric characteristic of virtual objects with real camera parameters, one important factor to achieve seamless blending is when virtual objects appear with a high level of photorealism as if they are formed by a mean of photography. Photorealistic rendering can offer substantial insights to achieve this.

A common illumination for virtual objects and real surrounding objects is the first necessary step if photorealistic quality is to be achieved. It gives not only consistent looks between synthetic and real surrounding objects but also visual subtleties such as inter-reflections and shadows. The illumination could be inter-reflected among virtual objects, or between virtual objects and real surrounding objects. This also applies to shadowing effects as well. As a result, a sense of natural perception can be developed from these visual cues, that would usually further lead to intuitive interaction towards virtual objects. For example, with cast of shadows, an AR viewer can judge spatial relationship of virtual objects easier than without them [1]. Thus, the challenge to apply photorealistic rendering in augmented reality is slightly more difficult. This is because not only the virtual objects must look real but also *visually consistent* and *coherent* with respect to real environment, and yet, the rendering process must fit within real time requirement.

Drawing a parallel in the works of compositing photorealistic computer generated imagery element in cinematography, ensuring this element be consistently lit with real lighting in the scene has been an indispensable task to digital visual effects artists. Normally, it requires manually surveying the field where footage is taken, so that lighting condition can be determined, and then used to set virtual lightings to be as close as possible to the scene lightings. This includes positions, intensities, colors, and numbers of the lighting in the scene. Another approach is to photograph a reference object in the scene where the synthetic object is to be rendered, and use its appearance as a qualitative guide in manually configuring the lighting environment. These methods typically require considerable labor intensive hand-refinement by digital artists. Hence, this suggests us an idea that photorealistic rendering in augmented reality work may give vital contributions for visual effects domain.

Proceedings of the Second IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR '03)
0-7695-2006-5/03 $17.00 © 2003 IEEE

There are many new fields outside the entertainment arena that could also potentially benefit from augmented reality with photorealistic virtual objects. Aesthetic industrial design visualization is one of them [2]. In this field, a framework of collaboration between multi-users is possible, and AR could provide an interactive way of visualizing the design under realistic illuminations. Many other interesting applications such as in [3, 4, 5, 6] can truly benefit from applying photorealistic rendering in AR as well.

The remainder of this paper is organized as follows. Previous related works are related in Section 2. Section 3 will explain the scene radiance measurement. The methods to process the result of the acquired radiance data are discussed in Section 4. A multi-pass rendering framework for AR that makes use of the acquired environment illuminations is discussed in Section 5. In Section 6, the working augmented reality system we used will be introduced. Finally, some experiment results are demonstrated and a conclusion will be drawn.

2 Previous works

Overlaying virtual objects and real background images with real illumination was pioneered by Fournier et al. [7]. This work tried to overlay synthetic objects in video sequences of real background images and render them physically accurate. Progressive radiosity [8] was used to compute the solution to global illumination by initially merging the real scene objects represented as relatively large blocks, and its result was then transferred on the real image. Drettakis et al. [9] extended this work by more efficiently solving global illumination with hierarchical radiosity [10], and modeling the real scene, which is required for radiosity calculation. It is known that radiosity requires a complete scene description, which is difficult to acquire, in order to run accurately. One of the difficult tasks is in assigning the reflectance for all surfaces in the scene. Both of these works use a few heuristics and approximations to estimate the reflectance. Furthermore, the latter work required about two to three seconds per frame for a simple environment to update illumination whenever dynamic objects, including real and synthetic ones, move.

Sato et al. [11] worked out the algorithms to capture environment illumination and to model the distant parts of the environment respectively by using omni-directional stereo cameras. The illumination was represented as a set of images, containing scene radiance distributions, which were estimated from the brightness of the omnidirectional images. Finally, the synthetic objects were superimposed onto a real scene with a ray-casting algorithm. Another similar approach of using scene radiance in image synthesis was the recent work done by Gibson et al. [12] that was able to achieve interactive frame rates at about ten frames per second. Unlike the previous works, this higher frame rate is possible as recent hardware-based rendering techniques [13], such as shadow mapping and use of accumulation buffer, were utilized.

Another recent work with different methods from what have been described above is an approach in estimating illumination parameters using ellipsoidal models [14] for compositing synthetic objects and video broadcast industry application. Yet another one deals with photometric image-based rendering that extracts illumination information from set of photographs of the scenes [15]. Such information is then used to render the new scene with different lighting conditions.

Photorealistic effects produced by global illumination methods, such as shadows, specular reflections, caustics, refractions, and transparency, can also be simulated in hardware-based rendering method via multi-pass technique [16]. Though being visual approximation, they appear realistic at interactive frame rates. Furthermore, the use of texture mapping to improve details of realism has also been implemented in graphics hardware. Texture mapping offers myriad possible effects in improving photorealism. It can be applied to simulate shadows [17], diffuse and specular reflections, reflection mapping as well as bump mapping [18, 19].

Complex lighting is difficult to achieve with graphics hardware so the empirically based Phong illumination model [20] is commonly adopted. Miller et al. [21] introduced a way to render objects not only with Phong's shading but also with the addition of reflected environment details through an illumination map, which consists of a diffuse and a specular maps, otherwise the effect is achieved using distributed ray tracing. Diffuse and specular maps contain diffuse and specular illuminations respectively, which are obtained by prefiltering an environment map.

3 Scene radiance acquisitions

In this section, a technique in capturing real-world illumination, needed for our rendering methods is discussed. The technique introduced by P. Debevec [22, 23], is followed. Such a technique is known as image-based lighting [24]. We will briefly introduce such a technique here. A concept on image-based lighting is firstly introduced; then, the method for capturing real-world illumination via HDR photography and for creating the HDR image is elaborated. Some issues regarding this technique for AR application is also discussed.

Proceedings of the Second IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR '03)
0-7695-2006-5/03 $17.00 © 2003 IEEE

3.1 Image-Based Lighting

Image-based lighting is meant for integrating computer-generated imagery with photography realistically, that use measurements of real-world lighting to illuminate computer-generated objects. This method had been successfully applied specifically for computer animations such as: in Fiat Lux [25], it has great potential for real-time applications, such as augmented reality. In principle, image-based lighting objects can be understood as two-dimensional images, having properties of: high dynamic range images, and omnidirectional images.

The dynamic range property measures the brightness variation of a scene image. It is often measured in terms of a contrast ratio between the brightest and darkest parts of the scene in the image. A high dynamic range (HDR) image theoretically can contain large variation of brightness of the scene in it and the pixel image values are proportional to the amount of light in the world corresponding to that pixel, unlike most regular images whose pixel values are nonlinearly encoded. For this reason, HDR images are also called radiance maps as they contain a spatial distribution of true radiance values in a scene. In addition, high dynamic range pixels are commonly represented by floating-point numbers for each color channel, unlike those of low dynamic range, which use integer values from 0 to 255. Some examples of common HDR image formats are 12-bytes per pixel portable floatmap, 16 bit or floating-point TIFF, Greg Ward's LogLuv TIFF - based on human color perception [26], NewTek flexible image format, and 32-bits per pixel RADIANCE image format [27]. In addition, *HDRShop* software developed by Debevec et al. [28] is a powerful HDR image processing and manipulation tool. It can be used to combine acquired set of low dynamic range images into a single HDR image.

Omnidirectional images are images that capture very wide view angle of a scene, in which each direction of incoming radiance towards the image plane of an imaging sensor is geometrically known. Often it assumes orthographic projection to derive such geometric relationship, which means the reflected rays towards the sensor are parallel to the viewing direction. Several types of omnidirectional image representations are such as cubic environment or cubic map [29], sphere map [30], and dual parabolic map [31].

3.2 Methods of measurement

There are three commonly used methods to acquire scene radiance in image-based lighting approach such as: using fish-eye lens cameras [11]Spheron-VR, omnidirectional vision sensors [32] or light probe with HDR photography [22]. Only the latter approach will be covered here. In principle, they use either mirrors with a standard camera with adjustable shutter speed or special lens camera with wide-angle view of the scene. Such cameras are utilized to measure the scene radiance by taking pictures of the particular scene of interest using sequential exposure image technique [33], which sequentially captures multiple images of the same scene using a camera with varying exposures, controlled either by changing the aperture size or F-number of the imaging optics system or the shutter speed of the image detector. For a live video camera, like omnidirectional vision sensors, one way to change the exposure is by changing its aperture size (or often known also as F-stop) or by using neutral density filters, which is a type of photographic filters to reduce amount light received by a film.

In this work, HDR photography and a light probe, which is a mirror ball that can be made of an aluminum sphere coated with chromium, were used to acquire scene radiance with image-based lighting technique. This method is neither difficult nor expensive. Firstly, we placed the light probe in the scene, where the virtual objects are to be placed and illuminated. And, by following the specified exposures in [23], we captured the scene with multiple exposures. The resulting images were low dynamic range sphere maps of the scene. This set of images was then assembled into a single HDR image by using HDRShop software as described in [28].

3.3 Discussions

In the scene radiance measurement, we are more interested in its spatial distribution values in the form of an image-based representation and less emphasis on its true values. With regard to adopting the light probe and HDR photography approach, some assumptions are made, concerning the acquired HDR image of the scene. The lighting conditions in our AR applications must be similar to those of the measured scene. Otherwise, the illumination effects on the virtual objects may appear inconsistent with respect to its surrounding objects. It is assumed that dynamic movements of distant surrounding objects with respect to the virtual objects have no effects in the appearance of the virtual objects. This would not be true when either the distant objects or the virtual objects have mirror-like surfaces. In addition, it is also assumed that each of HDR images, which are sphere maps, covers some workspace that allows the virtual objects to move within this space. The exact size of this workspace would not be determined quantitatively but in principle we use an observation if the appearance of the virtual objects is somewhat less convincing, the objects have been out of range of measured working space. In such a case, a new map corresponding to that workspace is then needed. This is because of the view-dependent nature of a sphere map. It is impossible to measure every point within the workspace of the virtual objects if the virtual objects are dynamic in the AR applications. Thus, several pre-determined locations in the scene that would cover the workspace of the application will be measured.

Another issue is that ideally the HDR image needs to be aligned to the viewer's orientation of the augmented reality system if the viewing direction is not the same with that of the camera taking the HDR picture. This is to ensure correct reflections for the virtual objects. The matching operation is done by image warping, which is an expensive operation in term of computation if it is done in real-time. One approach is by taking pictures of the light probe from different viewing angles. These viewing angles are determined from the applications.

The aforementioned assumptions imply that the appearance of the virtual objects illuminated with the measured real lighting would not give very accurate reflections of the scene, unlike those from the corresponding real objects if they were present in the scene. However, such differences are often unnoticeable by our visual perception.

4 Environment illuminations maps

A set of high dynamic range radiance map acquired by image-based lighting contains the reflections from all the incoming rays of a scene as seen from the measured points. Each pixel value in these maps contains direct illuminations from the light sources and also indirect illuminations as contributed by inter-reflections between the scene objects. These facts suggest that the radiance maps can be utilized to illuminate three dimensional computer generated imageries with real scene lighting in computer graphics rendering. In short, they are used for *environment illumination maps*.

There are three common methods to make use of the radiance maps. The first method is to use them via chrome-mapping [30]. Another one is to use them as a light-based model in global illumination rendering [22]. The third one is to use them as illumination maps in hardware-based rendering, which basically we employ for rendering in AR.

4.1 Methods of creating illumination maps

There are several methods to create an illumination map. A straightforward method is by taking high dynamic range photographs of a spherical object made of a particular material with which the virtual object has also a similar property. For example, if the object is to be modeled with a glossy look, the sphere could be made of aluminum-alloy with some degree of surface roughness. The map created would be similar to a glossy map. This approach is, however, not flexible as it requires us to fabricate each sphere with the desired material. We instead adopt two methods to derive illumination maps from the radiance map. The first method uses global illumination to synthesize them, while the other achieves this by prefiltering an environment map. In the first method, the light-based model is built from the radiance map. It is basically a virtual environment, which is made of a simple box textured with cubic mapping of the radiance map. A synthetic sphere is then placed at the centre of it. The size of the synthetic sphere is relatively very small with respect to that of the box, having a ratio of its diameter to the box's width between 1:50 to 1:500. The material property of the synthetic sphere is set according to the desired illumination map. For example, a diffuse sphere will create a diffuse map. Similarly a low shiny sphere with some degree of surface roughness will create a glossy map. The position of the viewer can be set arbitrarily as long as the final image shows reasonable size of the sphere. In this case, the diameter of the sphere, which is the size of illumination map itself, is ideally from 128 to 256 pixels. The direction of the viewer is, however, always set towards the center of the sphere. This implies that the resulting illumination map can have a different viewpoint with the one from which the light probe was taken. Once the synthetic sphere and the viewer have been determined, the illumination map can be synthesized by running the global illumination calculation, in particular using ray tracing. We used RADIANCE [34] to achieve this. It is noted that no light sources are defined in this ray tracing calculation. The faces of the box act as if they are area light sources. Based on the number of ray samples and depth-level defined in the calculation, rays are shot from each face towards the objects inside the box and bounce back towards any of the faces once they hit an object. The intersection between the ray samples and the faces are texture-mapped polygons from which the illumination intensity is calculated.

4.2 Environment map prefilterings

Environment map prefiltering is generally a method to calculate diffuse and specular component from an environment map by solving the radiance equation below:

$$\mathbf{L}_o(x,\omega_o) = \mathbf{L}_e + \int_\Omega f_r(x,\omega_i,\omega_o)\mathbf{L}_i(x,\omega_o) cos\theta_i d\omega_i \quad (1)$$

The equation above states that the surface radiance $\mathbf{L}_o$ from an object is the sum of radiance $\mathbf{L}_e$ emitted from the object and the radiance $\mathbf{L}_i$ reflected from the object. The last term is an integral over the hemisphere Ω, which takes into account the fact that light is incident from all directions weighted by the BRDF of object, f_r and incident angle θ_i. In this approach, two important issues are to select the appropriate environment map types, and the BRDF types. The types of environment map determine how the pixel values in the image would be selected, while the BRDF types determine how such values would contribute to each reflection components in the reflection model.

Miller et al. [21] proposed the idea to use an environment map to illuminate a computer graphics model. The diffuse and specular maps were created by prefiltering a sphere map. Greene further applied such an approach onto cubic mapping [29]. Heidrich et al. [13] extended the concept of prefiltering into glossy maps and also onto dual paraboloid maps, which were implemented in graphics hardware. Cabral et al. introduces warping scheme in reflection space for prefiltered environment maps [35]. Recently, Kautz et al. have extensively contributed on the environment map prefiltering works. They introduced a general framework for prefiltering with arbitrary BRDF as well as very interesting ideas on hardware-accelerated prefiltering [36, 37, 38]. Ramamoorthi proposed a signal-processing framework for calculating environment irradiance, and used spherical harmonics coefficients to represent prefiltered environment maps [39, 40].

In our work, we used several approach to prefilter the environment maps. We adopted diffuse and glossy algorithms as in [36] with Phong's BRDF model [41] and HDRShop software.

5 Multi-pass rendering framework

Multi-pass rendering is used in our work to combine different photorealistic effects with the helps of graphics hardware in the augmented reality system. The idea here is that different special effects, which are usually required specific, rendering techniques, can be achieved in multiple-passes. First, we introduce the general framework for multi-pass rendering for rendering in AR as follows:

1. *For number of image samples*
2. *Jitter the camera*
3. *Enable stenciling / blending operation*
4. *For number of passes*
5. *Add effects on the polygon model*
6. *Render the polygon model*
7. *Do stencil / blending test*
8. *Perform accumulation buffer operation*

In such a framework, there are three important operations: accumulation buffer operation, adding effects based on stenciling operation, and adding effects based on blending operation. The first operation concerns with scene antialiasing. It is required only if a high quality of rendering is desired, since this operation usually slower the speed of rendering. Otherwise, the number of image samples can be set to one.

The second operation concerns with adding effects on the polygon model via operation on the stencil buffer. In particular, the effects that can be added include reflections and shadows. It is noted however that such effects are usually simulated as well as approximated, thus not as accurate as those obtained by global illumination calculations.

The last operation concerns more general hardware-based rendering techniques such as texture mapping, convolution operation - image blurring or sharpening, and bump mapping with textures [42], all of which are eventually combined together by the blending operation. Each rendering pass may consist of more than one rendering technique. For an example, a render-to-texture operation involves at least two rendering techniques, which are done in a single pass. The first is to render the scene into a texture map stored either in the frame buffer or as an image file. The texture is then applied to the polygon model via texture mapping operation.

From this general framework, an OpenGL implementation aimed for our augmented reality application was developed. In particular, the lines 3 to 7 are implemented from the framework into the pseudo codes below.

1. *Enable blending*
2. *For each of virtual objects*
3. *Transform the virtual object*
4. *Enable 2D texture mapping*
5. *Set blending function* (GL_ONE, GL_ZERO)
6. *Render the virtual object*
7. *Enable 2D environment mapping*
8. *Set environment mapping to* GL_DECAL *mode*
9. *Generate the sphere mapping texture coordinates*
10. *Set blending function* $(GL\_DST\_COLOR,$ $GL\_ONE\_MINUS\_SRC\_ALPHA)$
11. *Render the virtual object*

The implementation above focuses on the effects based on the blending operation for each individual virtual object. There are two passes of rendering and each pass adds one photorealistic effect. The first effect is to add a skin texture to the virtual object. A skin texture here is a 2D image pattern, associating with surface texture of the virtual object, like wood, fabrics, marble or tiles. The purpose is to add details on the virtual object. The second effect is to add a complex illumination on the virtual object based on the illumination maps we have derived in the previous section. With this effect, it is not required to define the light sources in the OpenGL virtual scene. To enforce this effect, setting the environment map into GL_DECAL mode is necessary. Here, it is assumed that the virtual object is represented by a polygon model. In addition, we have also added the soft shadowing effects [43] into the multi-pass rendering framework. We estimated the pose of the light sources for shadows from the acquired light probe image.

6 Working AR system

We use ARToolKit to build our video-based AR system in our work. ARToolKit is a C language software library that lets programmers easily develop Augmented Reality applications [44]. It is developed by Mark Billinghurst et al. from Human Interface Technology Laboratory (HITL) University of Washington together with Hirokazu Kato from Hiroshima City University. In this work, the AR system was setup in SGI O2+ 400 MHz machine running on SGI IRIX 6.5 operating system. An optional HMD Sony Glasstron for the display with an attached color camera was used. For the virtual objects in our AR works, we used 3D polygon models, which are well-supported in OpenGL.

7 Results and discussions

Several experimental results are presented in this section. These include results from image-based lighting, environment illumination maps, rendering from multi-pass algorithms, and rendering in ARToolKit with environment illuminations.

7.1 Image-Based Lighting

Some results low dynamic range images of an indoor scene taken in our Labs are shown in Figure 1. The pictures of the light probe were taken by HDR photography. They were taken with sequential exposures of: 1/200, 1/100, 1/30, 1/2, 1, and 2 seconds with F2.4 aperture size.

The digital image was set at 1280 x 960 pixels resolution. The set of low dynamic range images were assembled into a single high dynamic range image using HDRShop. The HDR image result however could not be reproduced via printing in this paper.

7.2 Environment illumination maps

The figures in Figure 2 show the result of diffuse and glossy environment maps acquired from the methods - using RADIANCE and environment map prefilterings, as mentioned in Section 4.

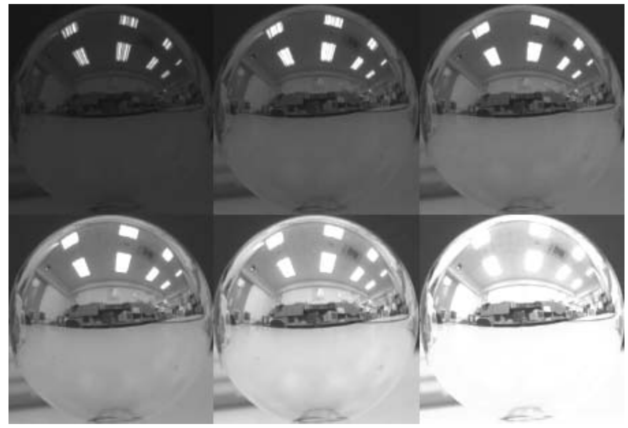

Figure 1. The result of HDR photography of the light probe in the indoor scene. The exposures from the top to bottom and left to right are 1/200, 1/100, 1/30, 1/2, 1 and 2 seconds with F2.4 aperture size.

In the first row from left to right, the figures are diffuse environment maps obtained from using RADIANCE, HDRShop, and our diffuse prefiltering algorithm, while in the second row from left to right, the figures are glossy environment maps acquired from RADIANCE, HDRShop, and our glossy prefiltering algorithm. It is noted here that all these figures are displayed as low dynamic range images in this printing.

7.3 Multi-pass rendering algorithm

We presented the results of rendering of a virtual object under environment illuminations with a real scene background from which the illuminations were measured in Figure 3. In Figure 3a, the virtual object is a 3D polygon model of an angel and it is rendered only with a diffuse illumination map. The other result is a rendering of a 3D polygon model of Happy Buddha, which is acquired from The Stanford Computer Graphics Laboratory, with its skin texture and glossy environment illuminations. Soft shadow effects are also included in such a result. It is noted here, the pose of both 3D models in such results were adjusted at interactive frame-rates with a fixed viewpoint.

7.4 Rendering in AR system

We implemented the OpenGL multi-pass rendering in the AR system. The result of such multi-pass renderings with the AR system is shown in Figure 4.

In such a figure, the virtual objects are 3D polygon models of statues rendered with skin textures together with en-

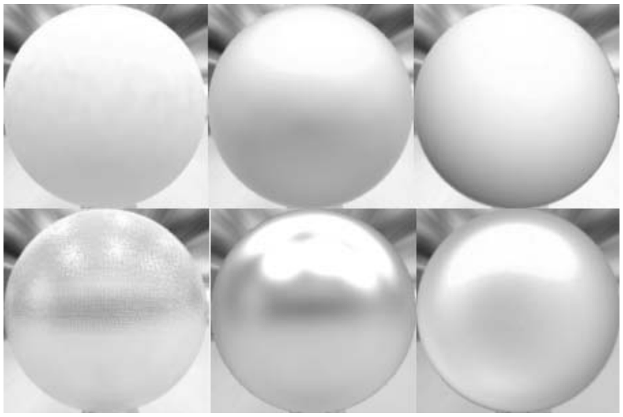

Figure 2. (From left to right) Diffuse environment maps (first row) and glossy environment maps (second row) acquired from RADIANCE, HDRShop, and our prefiltering algorithm.

Figure 3. (Left) A virtual model of an angel rendered with diffuse environment map only. (Right) A virtual model of Happy Buddha rendered with skin texture and glossy environment map with soft shadow effects. This model is from The Stanford Computer Graphics Labs. All renderings are done almost in real-time ($>$17 fps).

vironment illumination maps. The statute model on the left was rendered with a diffuse illumination map and ceramic skin texture, while the statue on the right was rendered with a glossy illumination map and gold skin texture. Both models were positioned slanted with respect of the ARToolKit markers, and they were rendered at interactive frame-rates while the viewpoints changed. The soft shadow effects were not yet included such results.

7.5 Discussions

With the pre-calculated real lighting data represented in HDR environment illumination maps, complex and consistent illuminations of the virtual object rendering in AR can be achieved as shown in the results. However, currently with this approach, the viewpoints in AR (the camera capturing the scene image) are limited to the number of acquired light probe images for rendering the virtual objects with glossy environment illumination map due to view-dependent nature of the sphere map. Thus, other environment map representations such as cubic mapping or dual paraboloid maps which are view-independent are desirable.

The multi-pass rendering framework as described before is practical and effective for improving photorealism of the virtual object rendering in augmented reality applications. It is practical because many rendering techniques to improve photorealism for interactive applications have been developed sophisticatedly, and mainly spurred by the gaming industry. Thus, the existing techniques can be found or even the new ones can be developed and then combined into the framework straightforwardly to achieve any particular photorealistic effects. It is effective because more and more new rendering algorithms are implemented in dedicated processors, which are often included in the release of new graphics hardware as API extensions, specifically OpenGL extensions. Thus, it is not difficult to adopt any extensions suited to the desired purpose into the framework just by treating them as similar to any OpenGL standard instructions.

8 Conclusions

To improve the renderings of the AR synthetic objects with hardware-based methods, real lighting information are taken into consideration. Such data are measured and pre-calculated beforehand. It is then processed into a suitable representation, which graphics hardware can take advantage of. We have described in this paper the methods to acquire the scene illumination with image-based lighting approach. We also discussed how to use such acquired radiance maps into environment illumination maps, which are suitable representation for hardware-based rendering. In addition, we proposed two methods to create such maps with global illumination approach and environment map prefilterings. Finally, a standard multi-pass rendering framework for photorealistic rendering with AR has also been discussed here.

The contributions in this work are not revolutionary ren-

Figure 4. (Left) A virtual model of a statue rendered with diffuse environment map and a ceramic skin texture. (Right) A virtual model of an angel rendered with glossy environment map and a gold skin texture. Both models were rendered with our augmented reality system nearly real-time ($\sim$17fps).

dering methods but simple and practical to integrate into augmented reality systems. Nevertheless, this work has offered insights on the photorealism aspect of renderings not only in augmented reality but also in real time graphics rendering applications in which photorealism is important.

References

[1] S. Pinker, *How the Mind Works*. W. W. Norton & Company, 1999.

[2] M. Fiorentino, R. Amicis, G. Monno, and A. Stork, "Spacedesign: A Mixed Reality Workspace for Aesthetic Industrial Design," in *Proc. of International Symposium on Mixed and Augmented Reality*, pp. 86–94, 2002.

[3] A. S. Carlin, H. Hoffman, and S. Weghorst, "Virtual reality and tactile augmentation in the treatment of spider phobia: A case study," *Behavior Research and Therapy*, vol. 35, no. 2, pp. 153–158, 1997.

[4] C. Stapleton, C. Hughes, M. Moshell, P. Micikevicius, and M. Altman, "Applying mixed reality to entertainment," *Computer Magazine*, vol. 35, no. 12, 2002.

[5] W. Broll, E. Meier, and T. Schardt, "The virtual round table - a collaborative augmented multi-user environment," in *The Third International Conference on Collaborative Virtual Environments*, (San Francisco, USA), Sept. 2000.

[6] W. Broll, E. Meier, and T. Schardt, "The Mixed Reality Stage," in *Proc. of SIGGRAPH*, 2000.

[7] A. Fournier, A. S. Gunawan, and C. Romanzin, "Common illumination between real and computer generated scenes," in *Proc. of Graphics Interface*, pp. 254–262, 1993.

[8] M. F. Cohen, S. F. Chen, J. R. Wallace, and D. Greenberg, "A progressive refinement approach to fast radiosity image generation," *Computer graphics*, no. 75-84, 1988.

[9] G. Drettakis, L. Robert, and S. Bougnoux, "Interactive common illumination for computer augmented reality," in *Eurographics Rendering Workshop*, pp. 45–56, 1997.

[10] P. Hanrahan, D. Salzman, and L. Aupeperle, "A rapid hierarchical radiosity algorithm," in *Proc. of SIGGRAPH*, pp. 197–206, 1991.

[11] I. Sato, Y. Sato, and K. Ikeuchi, "Acquiring a radiance distribution to superimpose virtual objects onto a real scene," *IEEE Trans. on Visualization and Computer Graphics*, pp. 1–12, January-March 1999.

[12] S. Gibson and A. Murta, "Interactive rendering with Real-World Illumination," in *Proc. of Eurographics*, 2000.

[13] W. Heidrich and H. P. Seidel, "Realistic, hardware-accelerated shading and lighting," in *Proc. of SIGGRAPH*, pp. 171–178, Aug. 1999.

[14] J. Stauder, "Augmented reality with automatic illumination control incorporating ellipsoidal models," *IEEE Trans. Multimedia*, vol. 1, no. 2, pp. 136–143, 1999.

[15] Y. Mukaigawa, S. Mihashi, and T. Shakunaga, "Photometric image-based rendering for virtual lighting image synthesis," in *Proc. of 2nd Int'l Workshop Augmented Reality*, (Los Alamitos, Calif.), pp. 115–124, IEEE CS Press, 1999.

[16] P. J. Diefenbach, *Pipeline Rendering: Interaction and Realism Through Hardware-based Multi-Pass Rendering*. PhD thesis, University of Pennsylvania, Jun. 1996.

[17] P. Heckbert and M. Herf, "Fast Soft Shadows," in *In Proceedings of SIGGRAPH '96*, p. 145, Aug. 1996.

[18] P. Heckbert, "Survey of texture mapping," *IEEE Computer Graphics and Applications*, pp. 56–67, Nov. 1986.

[19] A. Schilling, "Towards real-time photorealistic rendering: Challenges and solutions," in *Proc. of SIGGRAPH '97 / Eurographics workshop on graphics hardware*, (Los Angeles, California), pp. 7–15, Aug. 1997.

[20] B. T. Phong, "Illumination for computer-generated pictures," *Comm. ACM*, vol. 18, no. 6, 1975.

[21] G. S. Miller and C. R. Hoffman, "Illumination and reflection maps: simulated objects in simulated environments." SIGGRAPH '84 Course Notes, Jul. 1984.

[22] P. E. Debevec, "Rendering synthetic objects into real scenes: Bridging traditional and image-based graphics with global illumination and high dynamic range photography," in *Proc. of SIGGRAPH '98*, pp. 189–198, Jul. 1998.

[23] P. E. Debevec and J. Malik, "Recovering high dynamic range radiance maps from photographs," in *Proc. of SIGGRAPH '97*, pp. 369–378, Aug. 1997.

[24] P. Debevec and D. Lemmon, "Image-based lighting." SIGGRAPH 2001 Course Notes, Aug. 2001.

[25] C. Zandonella, "Let there be light," *NewScientist Magazine*, pp. 42–45, 30th Oct. 1999.

[26] G. W. Larson, "Overcoming Gamut and Dynamic Range Limitations in Digital Images," in *Proc. of the Sixth Color Imaging Conference*, Nov. 1998.

[27] G. W. Larson, "Real pixels," *Graphics Gems IV*, 1994.

[28] HDRShop, "High-dynamic range image processing tools." http://athens.ict.usc.edu/HDRShop/.

[29] N. Greene, "Environment mapping and other applications of world projections," *IEEE Computer Graphics and Applications*, vol. 6, no. 11, pp. 21–29, 1986.

[30] J. F. Blinn and M. E. Newell, "Texture and reflection in computer generated images," *Comm. ACM*, vol. 19, no. 10, pp. 362–367, 1976.

[31] W. Heidrich and H. P. Seidel, "View-independent environment maps," in *Eurographics / SIGGRAPH Workshop on Graphics Hardware*, pp. 27–45, 1998.

[32] H. Ishiguro, *Panoramic Vision: Sensors, Theory, and Applications*, ch. Development of low-cost compact omnidirectional vision sensors. Springer, 2001.

[33] S. K. Nayar and T. Mitsunaga, "High dynamic range imaging: Spatially varying pixel exposures," in *Proc. of IEEE Computer Vision and Pattern Recognition*, Jun. 2000.

[34] G. J. Ward, "The RADIANCE lighting simulation and rendering system," in *Proc. of SIGGRAPH '94*, pp. 459–472, Jul. 1994.

[35] B. Cabral, M. Olano, and P. Nemec, "Reflection space image based rendering," in *Proc. of SIGGRAPH '99*, pp. 165–170, Aug. 1999.

[36] J. Kautz, P. Vazquez, W. Heidrich, and H. P. Seidel, "A unified approach to prefiltered environment maps," in *Eurographics Rendering Workshop' 00*, pp. 185–196, 2000.

[37] J. Kautz and M. McCool, "Interactive rendering with arbitrary BRDFs using separable approximations," in *Eurographics Rendering Workshop'99*, pp. 281–292, Jun. 1999.

[38] J. Kautz and M. McCool, "Approximation of glossy reflection with prefiltered environment maps," in *Proc. of Graphics Interface*, pp. 119–126, May 2000.

[39] R. Ramamoorthi and P. Hanrahan, "An efficient representation for irradiance environment maps," in *Proc. of SIGGRAPH'01*, pp. 497–500, 2001.

[40] R. Ramamoorthi and P. Hanrahan, "Frequency space environment map rendering," in *Proc. of SIGGRAPH'02*, 2002.

[41] R. R. Lewis, "Making shaders more physically plausible," in *Fourth Eurographics Workshop on Rendering*, pp. 47–62, Jun. 1993.

[42] T. McReynolds and B. Blythe, "Advanced graphics programming techniques using opengl." SIGGRAPH'99 Course Notes, Aug. 1999.

[43] N. Robin, M. Kilgard, and P. Brian, "Opengl and window system integration." SIGGRAPH'97 Course Notes, Aug. 1997.

[44] "Shared-Space-ARToolKit." www.hitl.washington.edu/projects/shared_space.