Image Based Lighting with Near Light Sources

Shiho Furuya, Takayuki Itoh

Graduate School of Humanitics and Sciences, Ochanomizu University

E-mail: {shiho, itot}@itolab.is.ocha.ac.jp

Abstract

Recent some researches on realistic computer graphics (CG) focus on synthesis of photographs and virtual objects. Image Based Lighting (IBL) is one of the techniques to generate naturally illuminated images by using photographs. This paper reports a new experiment to render virtual objects into real scenes with near light sources, which have not been considered in original IBL. Our experiment shows that we can generate more realistic images by faithfully constructing near light source environment.

1. Introduction

We have a long history of research on realistic computer graphics (CG). We also have a long history of geometric modeling of artificial objects, such as automobile and building, by using computer aided design (CAD) systems. Generally we need high-level skill and a long time for very realistic CG image creation, if we manually design all objects in a scene. Recent CG researchers have focused on synthesis of photographs and manually designed artificial objects, as a reasonable technique to generate more realistic images.

Let us say the scene in the photograph as "real scene", and artificial objects to be synthesized to the photograph as "virtual object". We should balance two kinds of light sources, where one is the virtual light source illuminating the virtual objects, and the other is the real light source illuminating the real scene in the photograph. Originally design of light sources needs a lot of skill and time, and moreover, the reality of CG image strongly depends on the design of light sources, even when we do not synthesis with the photograph. Balance of two kinds of light sources is a further difficult problem.

Image Based Lighting (IBL) is a novel technique to realize natural illumination by using photographs as light sources [1]. IBL uses an omni-directional photograph taken at the position where virtual objects locate, as the light source for global illumination. Designers do not need to finely set up virtual light sources in IBL. Therefore, even non-expert designers can create realistic naturally illuminated scenes by applying IBL.

However, existing IBL techniques usually assume light sources infinitely far away from virtual objects. This is one of the problems of IBL, because illumination of the real world cannot be limited to distant light sources. We often have to set up light sources locating near virtual objects to generate realistic synthesized images, but original IBL does not suppose such situation.

We think this problem is important in some use cases; for example, let us suppose to use IBL for production of pamphlets or promotional films in the manufacturing industry. Especially we think IBL is useful to synthesize unreleased new product designed using CAD systems onto real scenes. In this situation we can easily assume the scenes including light sources near the position where we would like to put the virtual objects. We think it is important to reconstruct the near light sources in the photograph to realistically illuminate the virtual object.

This paper reports a new experiment to render virtual objects into real scenes, with near light sources, which have not been considered in the original IBL. The technique presented in this paper reconstructs the near light sources, which are in the real scene and nearby the virtual objects. It calculates illumination using the reconstructed near light sources, in addition to the omni-directional photograph as distant light sources, such as a fluorescent light on a ceiling of the room.

2. Related works

The technique presented in this paper is an extension of IBL proposed by Debevec [1]. IBL requires following three items to generate synthesized image:

- **Virtual objects** are defined as three dimensional (3D) geometric models designed using CG or CAD systems.
- **Local image** is defined as a photograph taken at an optional viewpoint. IBL generates an image synthesizing the local image and the virtual objects.
- **Distant image** is defined as a photograph of omni-directional scene taken at the position where virtual objects locate.

Figure 1 denotes the procedure of IBL. Here, IBL assumes a floor, where virtual objects locate around the origin of a three-dimensional coordinate system. It also assumes walls which surround virtual objects from far away, where a distant image is mapped on the walls. It supposes pixels of the distant image as light source, to illuminate the virtual objects. It finally synthesizes the local image and the virtual objects considering of the light distribution computed by the above process.

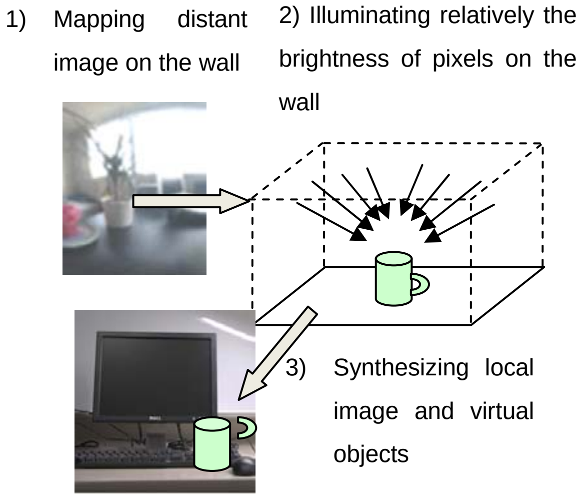
1) Mapping distant image on the wall
2) Illuminating relatively the brightness of pixels on the wall
3) Synthesizing local image and virtual objects

Figure 1: Procedure of IBL.

IBL simplifies the process of lighting design for global illumination, because the only we need to prepare is an omni-directional photograph. However, the original IBL [1] limits the irradiation of the light to virtual objects only light sources in the distant image. We therefore should assume that there are no near light sources while using the original IBL. Conversely, when light sources are shot in the local image and nearby virtual objects, it may be inconsistent between the calculation result of global illumination and the lighting in the local image. It is possible to take a photograph for distant image so that it contains the near light sources, in order to solve this problem. However, we do not think it is the fundamental solution for this problem, because one reason of the inconsistency is difference of degree of incidence of the illumination between distant and near light sources. Unless we set up independent near light sources in addition to the distant image, we cannot consider of the difference of irradiation effects between distant and near light sources.

Sato et al. also presented an IBL-like technique [2] which determines light environments by shooting omni-directional photographs of the real world using a fish-eye lens. The light environments measured from the omni-directional photograph are approximated as three dimensional triangular elements, whose vertices are mainly locate at the portions those intensities are high. The technique then calculates the illumination and shade of virtual objects using the light environments, and finally synthesizes them to a photograph. The technique is applicable for both indoor and outdoor scenes; however, it does not import near light sources.

Okabe et al presented an image analysis technique under near light sources, and an image composition technique as an application of the analysis [3]. However, the technique requires several images under various lighting conditions.

3. Rendering virtual objects into real scenes

3.1 Overview

Similar to the original IBL, the technique presented in this paper also requires three items, virtual objects, a local image, and a distant image, as explained in Section 2. The technique realistically synthesizes the virtual objects into the real scene, by using the three items and implementing three steps shown in Figure 1. Our technique requires HDR format for local and distant images, similar to Debevec’s technique [1].

Figure 2 shows the processing flow of our technique. It first constructs the light source environment based on the illumination distribution obtained from the distant image, and near light sources obtained from the local image, as Step 1 in Figure 2. It considers the result of Step1 as initial condition of light energy, and then calculates the stable state of light energy using radiosity method, as Step 2 in Figure 2. It finally generates synthetic image referring the computed light energy, as Step 3 in Figure 2.

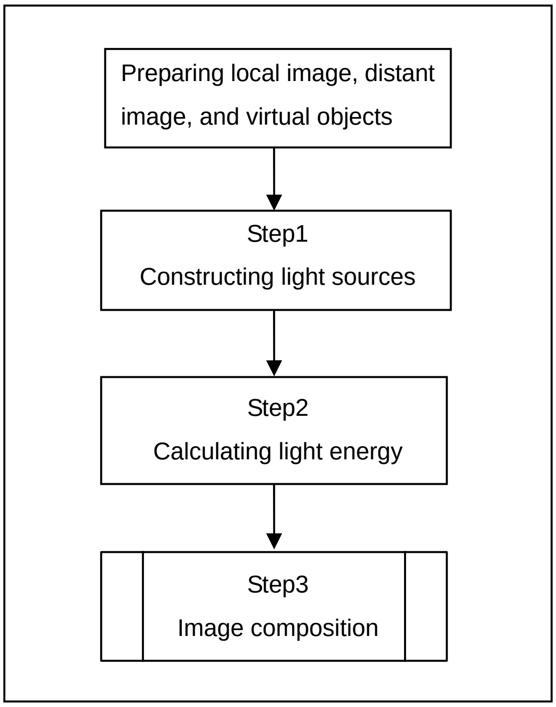
Preparing local image, distant image, and virtual objects
Step1
Constructing light sources
Step2
Calculating light energy
Step3
Image composition

Figure 2: Processing flow of the presented technique.

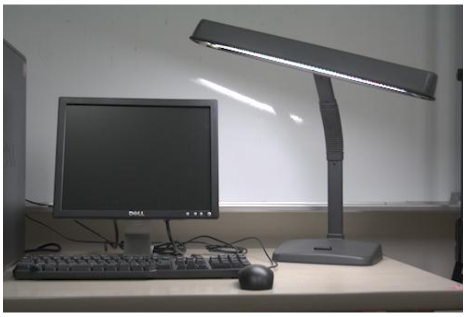

Figure 3: An example of the local image.

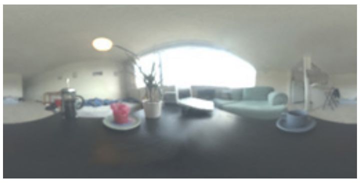

Figure 4: An example of the distant image.

Figures 3 and 4 show local and distant images we used in our experiments. We assume that a local image shoots near light sources. We use the photograph of scenery reflected to the chromium ball (specular surface ball) as well as described in [1], as a distant image.

We assume to use High Dynamic Range (HDR) format for distant and local images. HDR format assigns more than 8bits for RGB values of each pixel, and therefore it can finely represent the distribution of illumination.

Figure 5 shows an example of virtual objects we used in our experiments. We assume that 3D geometry of virtual objects is given as triangular mesh.

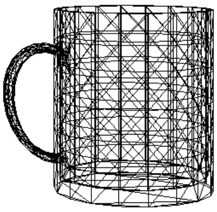

Figure 5: An example of virtual objects (mug) modeled as triangular mesh.

3.2 Step1: Constructing light sources

This section explains the step for constructing light sources. Figure 6 shows an illustration which denotes how we construct light source environment.

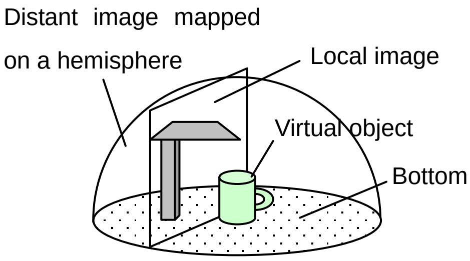

Figure 6: Constructing the light source environment with two HDR images.

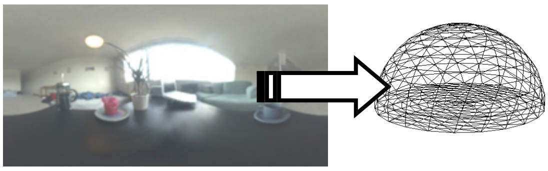

Figure 7: Mapping distant image onto a hemisphere.

In this technique, we suppose that virtual objects locate around the center of a circular bottom in a space surrounded by a hemispherical wall from far away. Moreover, looking from an optional view point, we suppose to locate a local image in the back of virtual objects. Also, the technique constructs distant light sources by mapping a distant image onto the hemispherical wall modeled as triangular mesh, which enclose all virtual objects and the local image, as shown in Figure 7. At this moment, the technique calculates energy E_1 of every triangles of the hemisphere, by averaging RGB values of pixels of the distant image inside each triangle. Here, E_1 denotes the initial energy of light which is needed in Step 2. Also, it supposes E_1 of triangular elements of the floor as 0.

The technique supposes that users design the geometry of near light sources as triangular meshes as well as the hemispherical wall. We assume that the geometry consists of light and non-light parts (i.e. plastic stand). Currently we manually design rough 3D geometry of near light sources using CG software, as shown in Figure 8. Naturally, it is preferable to automatically reconstruct their 3D geometry from the local images. Here, there are many techniques on reconstruction of 3D geometry from two or more images, in the field of stereo vision. Moreover, there are several techniques on manual reconstruction of 3D geometry, such as mapping primitive geometry to objects in the photograph [4], in the field of Image Based Rendering. Application of such techniques will be one of our future works.

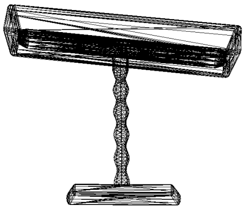

Figure 8: Simple 3D geometry data of a near light source designed using CG software.

After designing the near light sources as triangular meshes, the technique assigns the values proportional to RGB values of lights in the local image, to E_1 of the triangular elements corresponding to the light parts. Also, it assigns E_1 as 0, to the other triangular elements corresponding to the non-light parts.

In order to appropriately balance the optical energy between the distant and near light sources, we have to assign the weights to the light sources, because their actual light energy do not always proportional to the RGB values in distant and local images. Currently we explore a preferable weight value by try-and-error; however, we think that we should explore a technique to automatically compute the preferable weight value.

3.3 Step2: Calculating light energy

After assigning the initial energy E_1 to the triangles of hemisphere and near light sources as described in Section 3.2, we calculate the stable values of the energy E_2 of each triangle. Current our implementation applies the radiosity method for calculating the energy. The technique calculates E_2 twice: one is for the scene with the virtual objects, and the other is for the scene without them.

Figures 9 and 10 show the results of distributions of light energy calculated by radiosity method. Current our implementation provides an OpenGL-based viewer to show the radiosity results, which supports interactive viewing operation for manual calibration between the local image and the radiosity results. If users have an environment to shoot local images with marking multiple positions, we can automate the manual calibration.

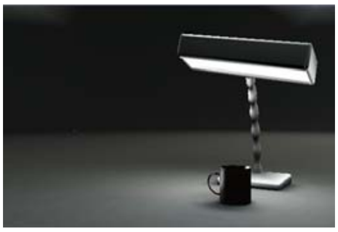

Figure 9: Distribution of light energy when a virtual object is in the scene.

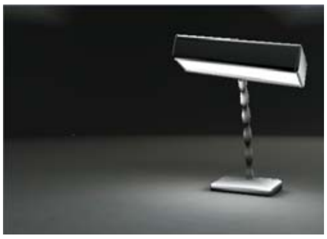

Figure 10: the distribution of light intensity when a virtual object is not in the scene.

3.4 Step3: Image composition

This section explains how to generate a synthetic image. At this step, new RGB values are computed based on RGB values of the local image, and energy E_2 of each triangle computed by the procedure described in Section 3.3. This step assumes that virtual objects locate in front of real objects shot in the local image, and supposes that the virtual objects are not partially occluded by the real objects.

Similar to the original IBL described in [1], our technique determines whether local image or virtual objects is rendered, for each pixel of the synthesized image. Here, current our implementation generates a mask image, as shown in Figure 11, where black pixels denote they are not occluded by the virtual objects. Simultaneously, it captures RGB values of each pixel of two images shown in Figures 9 and 10 by using the function glReadPixels().

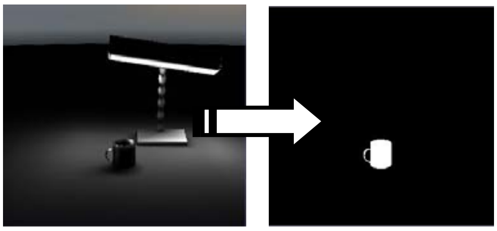

Figure 11: Mask image for determination of each pixel for image composition.

Our implementation calculates RGB values of pixels, where are black in the mask image, from the RGB values of the local image. In this case, it calculates the RGB values by multiplying the ration of energy before and after locating the virtual objects by the RGB values of the local image, as the following equation:

$$\begin{pmatrix} R & G & B \end{pmatrix} = \begin{pmatrix} R' & G' & B' \end{pmatrix} \begin{pmatrix} I_R \\ I_G \\ I_B \end{pmatrix}$$

$\begin{pmatrix} R & G & B \end{pmatrix}$: RGB values of a synthetic image

$\begin{pmatrix} R' & G' & B' \end{pmatrix}$: RGB values of an original image

$\begin{pmatrix} I_R \\ I_G \\ I_B \end{pmatrix}$: The ratio of energy before and after locating the virtual objects.

The calculation reflects the optical effects of virtual objects, such as reflection of light energy from the virtual objects to the floor, or shadows on the floor made by the virtual objects.

Our implementation calculates RGB values of pixels, where are white in the mask image, by applying well-known shading models; however, currently it considers only diffuse reflection. As future works, we have to treat not only to diffuse reflection but to specular reflection etc., and to express the virtual object to compound with reality by applying well-known shading models such as BRDF (Bidirectional Reflectance Distribution Function).

4. Experiments

This section shows our experiments. We implemented and executed all steps on Windows XP and GNU gcc 3.4. We created HDR images by unifying the brightness distribution of three or more images taken under different exposure to prepare distant image and local image. Moreover, we designed 3D geometric models of virtual objects and a near light source using 3D Studio Max, and prepared by saving this as triangular mesh data in VRML format. After saving the image generated by Step 3 in HDR format, we converted the image into JPEG format by using commercial software and applying tone mapping.

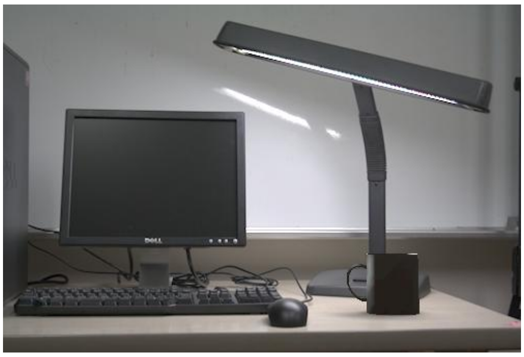

Figure 12: Example of an image with consideration of the near light source.

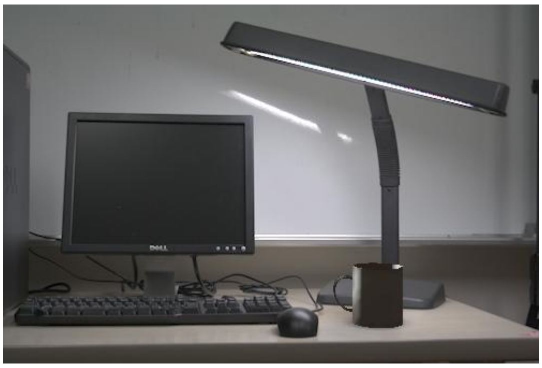

Figure 13: Example of an image without consideration of the near light source.

Figure 12 shows example of an image generated by our technique. We can observe the shadow of cup caused by the illumination of stand light. Fig.12 shows that consideration of near light source is effective in IBL.

We also generated a synthetic image without taking a near light source into consideration by using existing method, as shown in Figure 13. While generating both images in Figures 12 and 13, fluorescent lights of a ceiling of a room as distant light sources illuminated a mug. Processing flows of image synthesis in Figures 12 and 13 were same, except E_1 values of near light sources were 0 while generating the image shown in Figure 13.

Moreover, we actually placed the mug near the stand light, and shot a photograph, as shown in Figure 14.

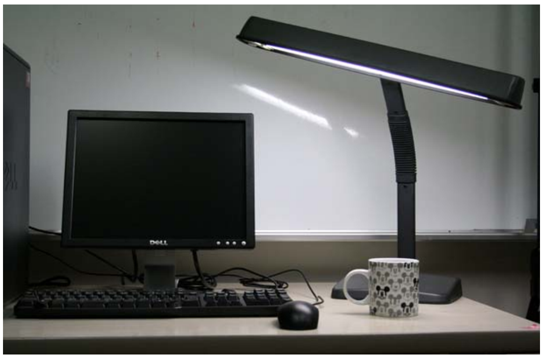

Figure 14: Photograph which actually placed mug.

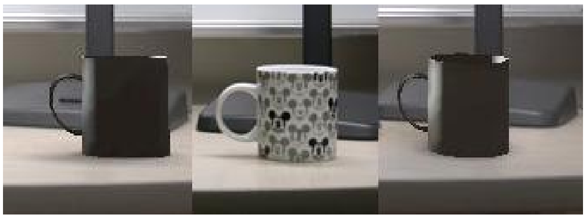

Figure 15: Comparison of synthetic portions. (left) From Figure 12. (middle) From Figure 14. (right) From Figure 13.

Figure 15 shows the zoomed portion of the synthesized virtual object and the real object. Considering of the near light source, our technique represented the shadow near the virtual object very similar to the real shadow observed in Figure 14. Thus, we emphasize constructing light source environments of the scene more faithfully in order to generate a natural synthetic image.

5. Conclusion and future works

In this paper, we presented how to render virtual objects into real scenes with near light sources using IBL. The following points are our future works.

- Semi-automatic generation of the 3D geometry of near light sources to ease a burden of a users' input.
- Re-examination about the consistency of the brightness when users use distant image and local image which were shot and edited under different conditions

The purpose of this report was to generate a real synthetic "still" image; however, we think our concept in this paper can be applied to PRT (Pre-computed Radiance Transfer) [5] for real-time realistic image synthesis.

References

[1] P. Debevec, "Rendering Synthetic Objects into Real Scenes: Bridging Traditional and Image-based Graphics with Global Illumination and High Dynamic Range Photography", SIGGRAPH 1998, pp. 189-198, 1998.

[2] I. Sato, Y. Sato, K. Ikeuchi, "Seamless Integration of Computer Generated Objects into a Real Scene Based on a Real Illumination Distribution", Journal of the Institute of Electronics, Information, and Communication Engineers, J81-D-II, No. 5, pp. 861-871, 1998.

[3] T. Okabe, Y. Sato, "Effects of Image Segmentation for Approximation Object Appearance under Near Lighting", MIRU 2005, pp. 80-87, 2005. 80-87, 2005.

[4] P. Debevec, C. Taylor, J. Malik, "Modeling and Rendering Architecture from Photographs", ACM SIGGRAPH '96, pp. 11-20, 1996.

[5] P. Sloan, J. Kautz, J. Snyder, "Precomputed Radiance Transfer for Real-Time Rendering in Dynamic, Low-Frequency Lighting Environments", SIGGRAPH 2002, July, pp. 527-536, 2002