A New Image Based Ligthing Method: Practical Shadow-Based Light Reconstruction

Jaemin Lee and Ergun Akleman*
Visualization Sciences Program
Texas A&M University

Abstract

In this paper we present a practical image based lighting method. Our method is based on a simple and easily constructible device: a square plate with a cylindrical stick. We have developed a user-guided system to approximately recover illumination information (i.e. orientations, colors, and intensities of light sources) from a photograph of this device. Our approach also helps to recover surface colors of real objects based on reconstructed lighting information.

1. Motivation

The compositing, which can be defined as the integration of computer-generated images (CG) and live-action photographs, has been an essential part of movie making. Currently, there exist a number of large special effect companies specialized mainly in compositing. For a succesful compositing it is crucial to seamlessly integrate real and CG components. It is well known that such a seamless integration requires solving a set of extremely difficult problems. Most of these problems are related to recovering the following real world information: **1.** Illumination information to simulate the same lighting environment for both CG and real objects, **2.** Camera positions and parameters, **3.** Shapes, material properties, and motion of real objects. The special effect companies solve these problems by employing a large number of computer graphics experts and mathematicians. Since most small special effect firms can not afford to employ large number of computer graphics experts and mathematicians, for such small companies there is a need for simple and effective methods to recover the real world information.

In this paper, we present a image based lighting method. Our method does not require any specific expertise in computer graphics or mathematics. The usability of our method was tested in a graduate level compositing course. Based on our method each student was able to seamlessly integrate CG characters into complicated real scenes. Figure 1 shows a frame from a composited animation created by one student. Based on this experience, we claim that our method can easily be used even by very small special effect firms and individuals.

*Corresponding Author: Address: 216 Langford Center, College Station, Texas 77843-3137. email: ergun@viz.tamu.edu. phone: +(409) 845-6599.

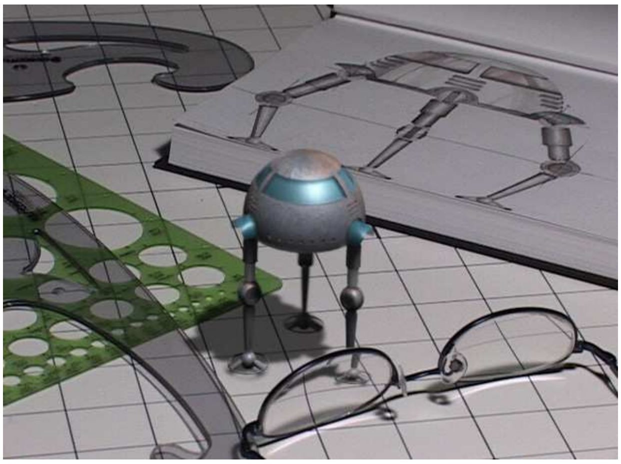

Figure 1. : A frame from a composited animation by Han Lei.

2. Introduction and Related Work

Image based techniques have recently been developed to achieve solutions to the problems addressed in the previous section. These techniques include camera calibration, image based lighting, and image based modeling. Among these techniques, we focuce on image based lighting that is used to recover illumination information from photographs of the real world. There are three major image based lighting approaches: **1.** Fisheye lens: Uses fisheye lense photographs that give a 180 degree field of view [9]. **2.** Gazing ball: Uses a photograph of a mirrored ball placed for catching the reflected view on its surface [1, 12]. (This method is generally used by special effect companies.) **2.** Shadow based: Uses shadows in the photographs [7, 8, 9].

Our method we present is shadow based. Shadow based methods have been proven to be reliable in recovering illumination information [7, 8, 9]. Shadow irradiance contains illuminant information such as intensities and orientations of light sources [11, 6].

Despite their power, existing shadow based methods are not very useful for practical applications. There are two major problems with these methods. **1.** For any given photograph, basic shapes and material properties of real objects have to be reconstructed by the users. **2.** The illumination information is computed by solving a large matrix. Solution of such a large matrix may be computationally expensive .

The main advantages of our method over previous shadow based approaches are the followings: **1.** We use photographs of a simple device that can easily be constructed. Since the shape of the device is simple, it is easy to create its CG model. Moreover we can use the very same CG model for different projects, because it is the only object that our method requires to retrieve illumination information. **2.** We observed that only a few characteristic pixels are sufficient to recover illumination information. The rest of the pixels in the image is redundant. We have developed a user-assisted software to choose the characteristic pixels. Using these characteristic pixels we can drastically reduce the size of the matrix to be inverted. **3.** Parameters of the shaders can be adjusted to create the same surface colors of objects.

The limitations of our method can be summarized as follows: **1.**The illumination information recovered by the proposed method does not include indirect illumination such as reflections from adjacent surfaces in a scene. **2.** Initial intensities of lights from our software can be too strong to use them without readjustment. These high intensities are caused by physical inaccuracies in widely used lighting and shading models in computer graphics. This paper also presents solutions to these limitations without sacrificing quality and practicality of our method.

3. Methodology

To provide a simple and practical image based lighting method, we first simplify the complexity of illumination by classifying illumination information into three categories: **1.** Key illumination: It represents main lighting sources in a scene (key lights). Key lights are the lights that produce visible shadows on floor surfaces. **2.** Direct illumination: It represents all the lights that do not create visible shadows. It also includes reflections coming from environment except adjacent surfaces. These lights are often called fill lights in computer graphics and movie industry. **3.** Indirect illumination: It includes reflections from adjacent surface.

In this work, we propose to recover both key and direct illumination from photographs of a simple device that is shown in Figure 2. This device consists of a wooden plate with a cylindrical stick in the center. Using color values of the pixels corresponding to the surface of the plate it is possible to derive a set of linear equations to estimate key and direct illumination as follows:

$$A.L = C \tag{1}$$

where A is a matrix whose coefficients are computed from the visibility of an incoming light source, L is a vector that includes the unknown values of light intensities for given set of light positions, and C is a vector that includes the color value of selected surface points [9].

Solving the equation 1 can be computationally difficult if the number of selected surface points are extremely large. For instance, the plate surface Figure 2 consists of approximately 75,000 pixels. Therefore, if we consider the entire region of this white plate, it will be difficult to solve the equation 1. Fortunately, most of these pixels are redundant and their number can easily be reduced.

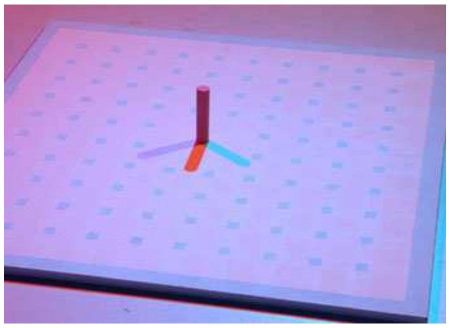

Figure 2. A photograph of the device under studio lighting.

To simplify the problem we only consider two types of regions on the plate[1]. **1.** Shadows cast by the stick on the painted white surface and **2.** the rest of the white surface of the plate. Notice that the colors in each one of these regions are almost constant in most cases. Thus, a few number of samples from each of shadow regions and the rest of the plate is enough to recover the lighting information. By using samples from only these particular regions, the size of the matrix A drastically reduces, and it becomes very easy to solve the equation 1. We further observe that **1.** key illumination can be recovered only from shadow colors cast by the stick and **2.** direct illumination can be recovered from the color of the rest of the plate.

[1]As seen in Figure 2, there are small blue squares painted on the white surface of our geometric model. We use these squares to identify a camera orientation relative to the device. They are not be used in collecting color samples.

In order to collect the samples (or in other words to choose characteristic pixels), we use a user-assisted approach. In our approach, the users select the ends of the shadows to compute the orientations, colors and intensities of key lights. We collect one color sample for each shadow and get one key light for each distinct shadow region. An additional color sample is collected from anywhere on the unshaded white surface to compute direct illumination, i.e. the intensity and color values of fill lights.

The orientation of a key light is identified based on a surface point selected in the end of a shadow for a color sample as shown in Figure 3. This surface point is used to create a line passing through both the surface point and the top of the stick. Then, an intersection point between the line and an imaginary hemisphere over the model is defined as the location of a key light. With this system, the distance of the key light can be adjusted by changing the size of the dome without affecting its direction.

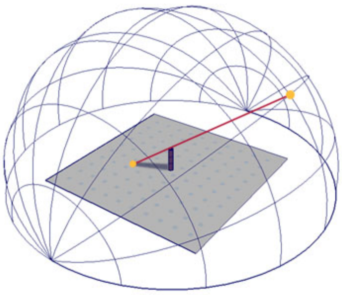

Figure 3. Computing the orientation of a key light from shadow positions.

Initial lighting information directly computed by solving the linear equation based on the color samples may not always work. This is because shader models that are used in 3D modeling and animation systems are generally crude approximations of the real physical behaviors. Our solution to work around this problem is to adjust colors of all the lights in a scene according to the ratios of pixel values on the white surfaces in a rendered image and a photograph. This process can be performed iteratively until the both color values are close enough. In addition to the intensity adjustment, fill lights need to be improved further for better results. Since our method does not include indirect illumination caused by adjacent surface reflectances, we add another set of lights to the scene on the ground level. These lights simulate indirect light rays bounding off from the adjacent surfaces. Then, the overall intensity of those lights is adjusted based on a sample photograph.

Under a reconstructed lighting environment, we can also recover material colors of real objects assuming that the recovered illumination is the same as real one in the photograph. In general we can define colors only in relative ways based on lighting environments, for instance white in daylight might look yellowish in incandescent lights. Therefore, if colors of synthetic objects need to be matched with real ones in a photograph, we have to find a common factor first, which affects color rendition under a specific lighting condition. This common factor is found from a reconstructed lighting condition with an arbitrary surface color, then it is used for color matching according to the following linear model. $c_p = a.c_o$

where, c_o is an unknown color value before being illuminated by the current lights, a is an illumination factor of the current lights, and c_p is a color after being illuminated by the current lights.

4. Procedure

We have developed a C++ program that be called *lightRecon* to compute locations and intensities of light sources based on pixel values selected by users from a given photograph. We have a N x N size nonsingular matrix from the data, and find its solution through matrix inversion using elementary operation. We also use a commercial software, namely Maya, to recover camera information by matching a virtual scene with a camera view from live footage. The virtual scene contains a CG model of our physical device and light sources that take lighting data from lightRecon to illuminate the scene. We also use Maya to render the scene with a recovered illumination information.

The overall process of our method consists of six steps.

1. Field record: This step consists of five stages: **1.1.** Survey data: It is to record measurements of a real scene, such as lighting condition, surface information of objects in the scene, camera height and angle, dimension of the device, distance between camera and the device, film format and size, lens size, and exposure information. **1.2.** Photograph of the real scene: This is an image without any reference objects in a scene. We use this photograph to composite with CG objects as final outputs. **1.3.** Photograph of a gazing ball: This is an image with a gazing ball. We use this photograph to have an additional information about environment. **1.4.** Photograph of the device: An image of the device is used to obtain pixel values of the white surface of the device as illumination data. And it also helps us to set up a camera view based on the picture. **1.5.** Photograph of material samples: We use this picture to find proper parameters of our shaders to match colors of synthetic and real objects. The Figures 4, 5 and 6 shows the images that are needed for a field record.

No shadows should be cast on the white surface of the device except ones from the cylindrical stick in the center of the plate. Multiple exposures are recommended to be able to choose ideal pictures.

2. Camera parameter reconstruction: We build a 3D replica of a real scene based on the survey data using a 3D application. A virtual model of the device is used to find a camera orientation. Note that the blue squares on the white surface of the device is designed to help us to identify the orientation of a camera.

Figure 4. An example of background.

3. Illumination reconstruction: Compute lighting information by using lightRecon. Our software allows to select pixels, constructs a matrix from selected pixel values and computes orientations and intensities of lights by using this matrix. T

4. Illumination adjustment: To improve initial light intensities, which are computed by lightRecon, we readjust them based on pixel values of both a rendered image and a background photograph. The overall process of this stage as follows: **4.1.** Render the virtual model of the device with initial light intensities. **4.2.** Open both the rendered image and the background photograph in any image prossing software and collect a pixel value from each image. **4.3.** Find a ratio of each color of two selected pixels. **4.4.** Multiply the initial intensities by the ratios to compute new intensities of the light sources. **4.5.** Render the scene with new illumination information. Go to 4.1 until the ratio values approach near to one.

5. Material colors: Parameters of shader colors for synthetic objects are estimated from the photographs of material samples taken in a real scene. The overall process of this stage is the following: **5.1.** Render a scene with synthetic objects using shader colors that are chosen from the photographs of material samples. **5.2.** Open the rendered image in a image processing program, then collect color values of the synthetic objects. We need these color values including initial shader colors as data in order to find an unknown factor value of the linear equation as we introduced in the methodology. This factor represents a lighting environment of the real scene, which affects the rendition of the material samples. **5.3.** Estimate an unknown factor of the linear equation with the given data, then find new parameters of the shader colors by applying the obtained factor values and the material colors in the photographs to the same equation. **5.4.** Rerender the same scene with the updated shader colors. If necessary, conduct the process iteratively until you get right colors.

Photograph of a gazing ball

Photograph of our device

Material samples

Figure 5. An example of field record based on the backgound image in the Figure 4.

5. Conclusion

Image based ligting is essential for realistic compositing. Although many image based ligting methods have been introduced, they require a substantial amount of computer graphics experience and mathematical knowledge. In this work, we present a simple method to overcome this disadvantage with existing techniques.

We tested our method in a graduate level compositing course. Based on our method each student was able to seamlessly integrate CG characters into complicated real scenes. Based on this experience, we claim that our method can easily be used even by small firms and individuals.

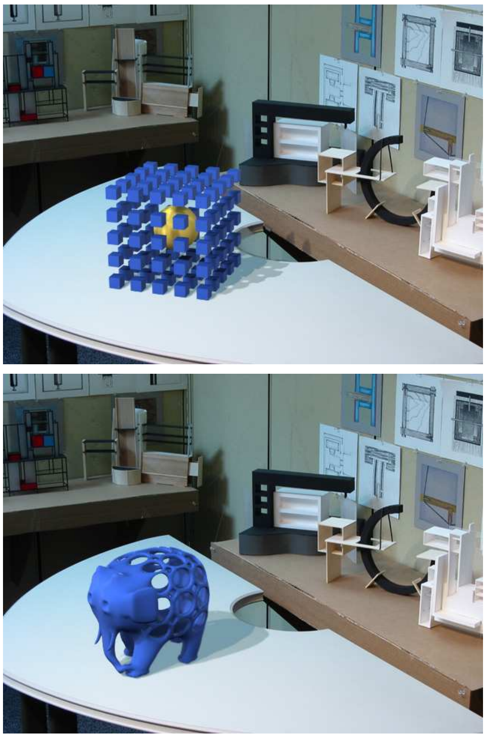

Figure 6. Two examples of compositing based the information given in Figure 5.

References

[1] P. E. Debevec, ”Rendering Synthetic Objects into Real Scenes: Bridging Traditional and Image-based Graphics with Global Illumination and High Dynamic Range Photography”, *Proc. SIGGRAPH 98*, pp. 189-198, July, 1998.

[2] A. Fournier, A. Gunawan and C. Romanzin, ” Common Illumination between Real and Computer Generated Scenes”, *Proc. Graphics Interface ’93*, pp. 254-262, 1993.

[3] B. K. P. Horn, ”Obtaining shape from shading information”, Chapter 4, *The Psychlogy of Computer Vision*, McGraw-Hill Book Co., New York, N. Y., 1975.

[4] B. K. P. Horn, ”Understanding image intensities”, *Artificial Intelligence*, 8(2), pp. 201-231, 1977.

[5] K. Ikeuchi and K. Sato, ”Determining Reflectance using Range and Brightness Images”, *Proc. IEEE Intl. Conference on Computer Vision 90*, pp. 12-20, 1990.

[6] J. R. Kender and E. M. Smith, ”Shape from Darkness: Deriving Surface Information from Dynamic Shadows”, *Proc. IEEE Intl. Conference on Computer Vision 87*, pp. 539-546, 1987.

[7] I. Sato, Y. Sato and K. Ikeuchi, ”Illumination Distribution from Shadows”, *Proc. IEEE Conference on Computer Vision and Pattern Recognition 99*, pp. 306-312, June 1999.

[8] I. Sato, Y. Sato and K. Ikeuchi, ”Illumination distribution from brightness in shadows: adaptive estimation of illumination distribution with unknown reflectance properties in shadow regions”, *Proc. IEEE Conference on Computer Vision and Pattern Recognition 99*, pp. 875-882, September 1999.

[9] I. Sato, Y. Sato and K. Ikeuchi, ”Acquiring a Radiance Distribution to Superimpose Virtual Objects onto a Real Scene ”, *IEEE Transactions on Visualization and Computer Graphics*, vol. 5, no. 1, pp. 1-12, March 1999.

[10] Y. Sato, M. D. Wheeler and K, Ikeuchi, ”Object shape and reflectance modeling from observation”, *Proc. of SIGGRAPH 97*, pp. 379-387, August, 1997.

[11] S. A. Shafer and T. Kanade, ”Using Shadows in Finding Surface Orientations”, *Computer Vison, Graphics, and Image Processing*, 22(1), pp. 145-176, 1983.

[12] Y. Yu, P. Debevec, J. Malik and T. Hawkins, ”Inverse global illumination: Recovering reflectance models of real scenes from photographs”, *Proc. of SIGGRAPH 99*, pages 215–224, August 1999.