Practical Image Based Lighting

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

Abstract

In this paper, we present a user-friendly and practical method for seamless integration of computer-generated images (CG) with real photographs and video. In general, such seamless integration is extremely difficult and requires recovery of real-world information to simulate the same environment for both CG and real objects. This real-world information includes camera positions and parameters, shapes, material properties, and motion of real objects. Among these, one of the most important is lighting.

Image-based lighting that is developed to recover illumination information of the real world from photographs has recently become popular in computer graphics. In this paper we present a practical image-based lighting method that is based on a simple and easily constructable 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

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 exists a number of large special effects companies specializing mainly in compositing[1]. For successful 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 and, **(3)** shapes, material properties, and motion of real objects. The special effects companies solve these problems by employing a large number of computer graphics experts and mathematicians.

The compositing (or integrating) of computer-generated images (CG) with real photographs can be an important part of the architectural design process. Such composited images can show how a new structure will change the existing landscape before actually building the structure. We have observed that some architects and architectural firms have already started to use composited images. The quality

[1] Examples of such special effects companies and samples of movies they have contributed are Industrial Light and Magic (ILM) (Jurassic Park, Terminator, Flubber, Jumanji, Pearl Harbor, Artificial Intelligence and Star Wars), Rythm and Hues (Babe, Men in Black, Cats and Dogs, Harry Potter and Lords of the Rings), Digital Domain (Titanic, What Dreams May Come and The Fifth Element), Sony Pictures Imageworks (Hollow Man, Stuart Little and Godzilla).

of such composited images is limited because most architectural firms cannot afford to employ large numbers of computer graphics experts and mathematicians. For architectural compositing there is a need for simple and effective methods of recovering the real-world information.

In this paper, we present a practical image-based lighting method to recover illumination information. Our method can be useful for the architectural practice because it does not require specific expertise in computer graphics or mathematics. The usability of our method was tested in a graduate level compositing course in which a majority of the students had an undergraduate background in architecture. Using our method, each student was able to seamlessly integrate CG characters into complicated real scenes. Figure 1 shows a frame from a student animation composited over real video. As seen in the example, the background photographs or videos can even include transparent objects. Based on our experience, we claim that even very small architectural firms can easily use our method.

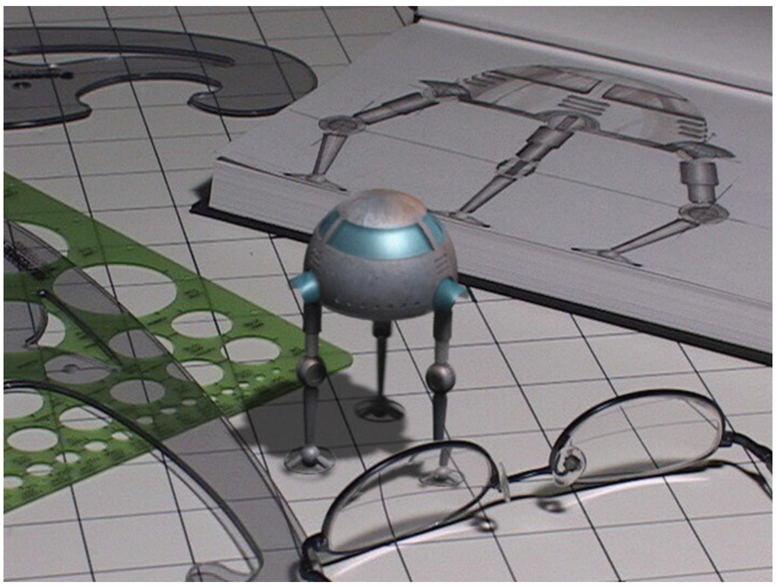


Figure 1. An example of student work: A frame from an animation that is composited over real video. In this image the robot is the only computer-generated object. Note that the refraction and shadow of the robot are also computed and composited by our student, Han Lei.

2 Introduction

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, image based rendering, and image based modeling. Among these techniques, we focus on image-based lighting that is used to recover illumination information from a photograph of the real world.
There are three major image-based lighting approaches**:** **(1)** *Fisheye lens*: Uses fisheye lens photographs that give a 180 degree field of view, which includes all lights in the scene (Sato et al. 1999c). **(2)** *Gazing ball*: Uses a photograph of a mirrored ball placed in a real scene for catching the reflected view on its surface (Debevec 1998, Yu et al. 1999) **(3)** *Shadow based*: Uses shadows in the photographs of the real world (Sato et al. 1999a, 1999b, 1999c).

The method we present here is a shadow-based approach. Shadow-based approaches have been proven to be reliable in recovering illumination information of the real world in conjunction with global illumination (Sato et al. 1999a, 1999b, 1999c). Shadow irradiance contains illuminant information of the real world, such as intensities of light sources. Orientations of illumination can also be predicted from shadow shapes (Shafer and Kanade 1983, Kender and Smith 1987).

Existing shadow-based methods, however, are not useful for practical applications. There are two major problems with these methods: **(1)** For any given photograph of the scene, basic shapes in the scene have to be constructed by the users. Material properties also need to be predicted. **(2)** The computation of illumination information is computed by solving a large matrix. The process may be difficult or computationally expensive.

We have developed a practical shadow-based method to reconstruct illumination information. The main advantages of our method over previous shadow-based approaches are:
(1) We use photographs of a simple pre-constructed device to recover lighting information. The device can be constructed easily by placing a cylindrical stick on the center of a white plate. The stick is used to cast shadows on the white surface. Because the shape of the device is already known, 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 are redundant and do not provide additional information. We have developed user-assisted software to choose the characteristic points in the photographs. Using these characteristic pixels, we can drastically reduce the size of the matrix to be inverted to compute illumination information.
(3) Parameters of the shaders can be adjusted to create the same surface colors of objects in the scene based on computed lighting information and the shader model used.

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 the practicality of our method.

3 Related Work

One of the successfully proven methods of estimating real-world illumination is to use an omni-directional image taken in a real scene (Sato et al. 1999c, Yu et al. 1999, Debevec 1998). Images of light in the real world are obtained in various ways, normally using a fisheye lens that has a 180-degree field of view (Sato et al. 1999c), or a mirrored ball which catches a surrounding environment onto its highly reflective surface (Debevec 1998). Once an omni-directional image is obtained through such methods, it is applied to a scene as an environment image surrounding synthetic objects. This method can successfully illustrate realistic lighting to achieve the seamless integration of synthetic objects and real photographs. However, its computation time becomes crucial because a number of high quality images are required. Depending on techniques of this kind, it is often essential for a user to have knowledge about the principles of illumination models in order to fully understand their numerical algorithms and modify them in practical ways for animation projects. As a result, the accessibility of such methods would be confined to certain users who would be able to handle such complex procedures.

Another well recognized approach is to use shadows in a real scene (Sato et al. 1999a, 1999b,1999c). The real-world illumination can be distilled from pixel values of shadows in conjunction with indirect illumination distributions caused by the surface reflectance of objects. Unlike the environment map method, which accesses lighting information directly from captured illumination, this method approaches the illuminant information indirectly through shadows. Shadows are also used for determining 3D shapes and orientations of objects that cast shadows onto a scene (Kender and Smith 1987, Shafer and Kanade 1983).

4 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 the environment except adjacent surfaces. These lights are often called fill lights in computer graphics and the 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:

Equation 1 $$\boldsymbol{A} \,.\, \boldsymbol{L} = \boldsymbol{C}$$

where $\boldsymbol{A}$ is a matrix whose coefficients are computed from the visibility of an incoming light source, $\boldsymbol{L}$ is a vector that includes the unknown values of light intensities for given set of light positions, and $\boldsymbol{C}$ is a vector that includes the color value of selected surface points (Sato et al. 1999c).

Solving equation 1 can be computationally difficult if the number of selected surface points is extremely large. For instance, the plate surface in 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.

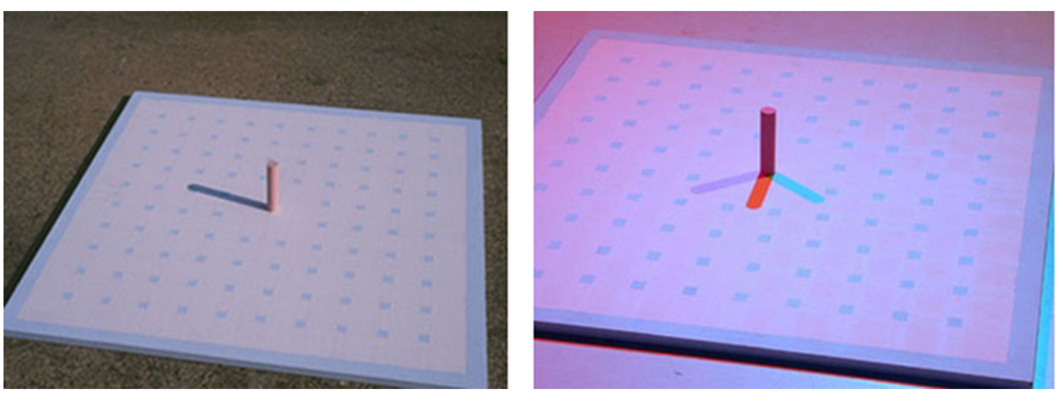


Figure 2. Two photographs of the device. The one on the left is shot under outdoor illumination. The one of the right is shot under studio lighting.

To simplify the problem, we only consider two types of regions on the plate: **(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 of these regions are almost constant in most cases. Thus, a few numbers of samples from each of the shadow regions and the rest of the plate are enough to recover the lighting information. By using samples from only these regions, the size of matrix $\boldsymbol{A}$ drastically reduces, and it becomes easy to solve equation 1. We further observe that key illumination can be recovered only from shadow colors cast by the stick, and direct illumination can be recovered from the color of the rest of the plate. 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. . (Note that, 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 used in collecting color samples.)

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.

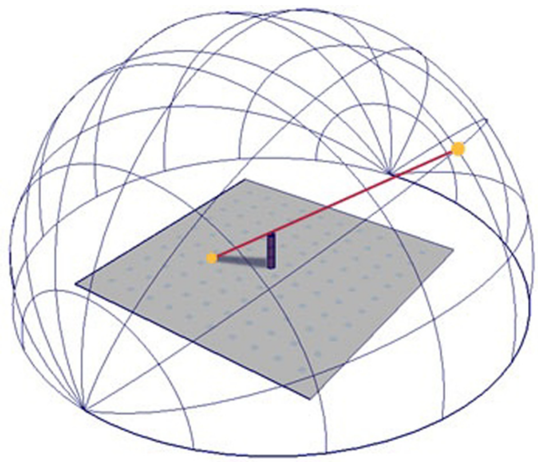


Figure 3. Computing the orientation of a key light from shadow positions.

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 the 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 both color values are close enough. In addition to the intensity adjustment, fill lights need to be improved further for better results. Because 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 bouncing off 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 the 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 under incandescent lighting. Therefore, if colors of synthetic objects need to be matched with real ones in a photograph, we first must find a common factor, that 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.

Equation 2 $$c = a \cdot u$$

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

5 Procedure

We have developed a C++ program that is called *LightRecon* to compute locations and intensities of light sources based on pixel values selected by users from a given photograph. We have an N x N size nonsingular matrix from the data, and find its solution through matrix inversion using an elementary operation.

We also use a commercial software program called 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 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 consists of the measurements from 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 provide additional information about the 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. The image 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 color of synthetic and real objects. Figure 4 shows the images that are needed for a field record.

2. **Camera parameter reconstruction**: We built a 3D replica of a real scene based on the survey data using a 3D application. A virtual model of the device is created as a part of the 3D replica; it helps us to find a camera orientation as well as illumination information of the scene. Note that the blue squares on the white surface of the device are designed to help identify the orientation of a camera; however, their color should not be involved in the illumination recovering process.

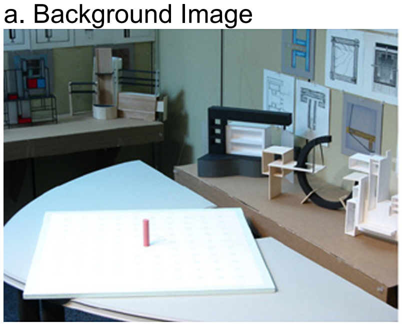
3. **Illumination reconstruction**: We run LightRecon to obtain lighting information of the scene, such as the intensities and locations of lights. The software allows us to select pixels of a photograph, and conducts matrix calculations from the selected pixel values. In order to run LightRecon, we need a scene data file from the 3D replica, which is a text file containing information about the matched camera and the device. As a final output, it creates a text file to provide the lighting information. Note that we provide a prototype of scene data, so that users can simply insert the required information into the file based on its instruction. There are two different versions of the software, one to accommodate Windows users and the other for Unix users.

4. **Illumination adjustment**: In order to improve initial intensities of the light sources that we get from LightRecon, we need to readjust them based on pixel values of both a rendered image and a background photograph. The overall procedure is as follows:
   4.1. Render the virtual model of the device with initial lighting intensities.
   4.2. Open both the rendered image and the background photograph in Adobe Photoshop or any similar 2D application, and then collect a pixel value from each image, especially pixels on the white surface of the device.
   4.3. Find a ratio of each color channel value of two selected pixels.
   4.4. Multiply the initial intensities by the ratios to get new intensities of the light sources.

4.5. Render the scene again with the updated illumination information. If necessary, conduct this process iteratively until the ratio values approach unity.

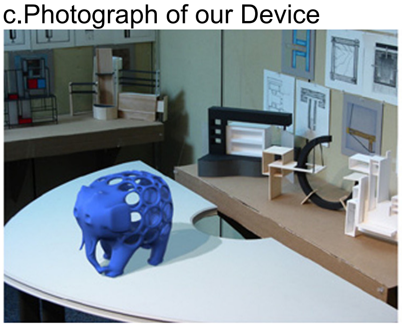
a. Background Image

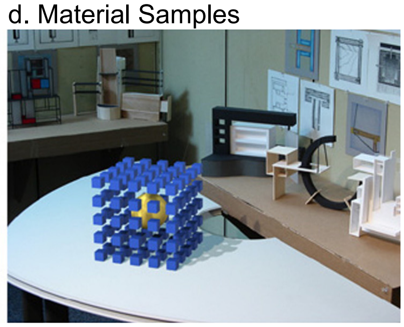
b. Photograph of Gazing Ball



c.Photograph of our Device

d. Material Samples





e/f. Two Examples of Compositing

Figure 4. The top four pictures show the images that are required for a field record. The two composited images illustrate that once material samples are collected, seamless integration is possible, even when extreme colors are chosen.

5. Material colors: Parameters of shader colors for synthetic objects are estimated from the photographs of material samples taken in a real scene. The procedure is the following:

5.1. Render a scene with synthetic objects using shader colors that are chosen from the photographs of material samples based on your observation.

5.2. Open the rendered image in Adobe Photoshop or any other 2D image processing program. Then collect color values of the synthetic objects in the image. 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 the lighting environment of the real scene, which affects the rendition of the material samples.

5.3. Estimate an unknown factor of the linear equation using 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.

5.3. Rerender the same scene with the updated shader colors. If necessary, conduct the process iteratively until you get the right colors.

Figures 5 and 6 show two examples of compositing with an outdoor scene. Notice that Figure 6 includes a glass building that reflects CG objects.

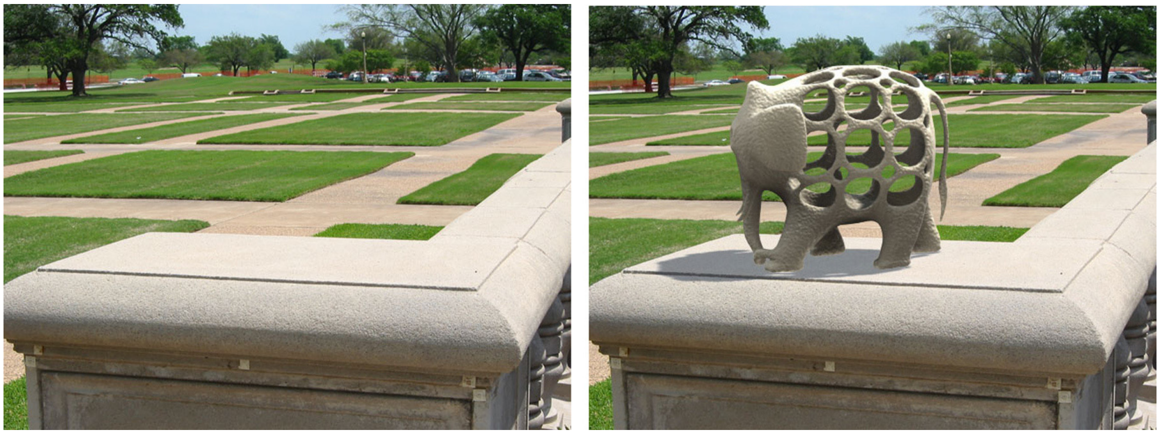


Figure 5. An example outdoor scene composited with a CG character.

a.Background Image b. Material Samples

c/d.Two Examples of Compositing

Figure 6. An example outdoor scene composited with CG characters.

6 Conclusion

Recovering illumination information with image-based lighting is essential for realistic rendering and compositing. Although many image-based lighting techniques have been introduced, most of these techniques require a substantial amount of computer graphics experience and mathematical knowledge. As a result, it is difficult for the people who have limited experience in computer graphics and mathematical knowledge, to use these techniques. In this work, we propose to develop a practical and user-friendly image-based lighting technique to overcome this limitation.

We developed a software that is compatible with popular 3D applications. Now, users have better control with flexible adjustments through the process of recovering illumination information. In addition, we believe our simple method helps artists to understand the overall concept of image based lighting techniques. Computation wise, we have a small-size nonsingular matrix based on a small number of characteristic pixels chosen by users from photographs. The process makes the computation of our method inexpensive enough for users to be able to apply the technique to their projects with less effort.

We tested our method in a graduate level compositing course where a majority of the students had an undergraduate background in architecture. Using our method, each student was able to seamlessly integrate CG characters into complicated real scenes.

References

P. E. Debevec. (1998). "Rendering Synthetic Objects into Real Scenes: Bridging Traditional and Image-based Graphics with Global Illumination and High Dynamic Range Photography", *Proceedings SIGGRAPH 98*, pp. 189-198, July 1998.

A. Fournier, A. Gunawan and C. Romanzin, (1993), " Common Illumination between Real and Computer Generated Scenes", *Proceedings Graphics Interface '93*, pp. 254-262, 1993.

B. K. P. Horn, (1975), "Obtaining shape from shading information", Chapter 4, *The Psychology of Computer Vision*, McGraw-Hill Book Co., New York, N. Y., 1975.

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

K. Ikeuchi and K. Sato, (1990), "Determining Reflectance using Range and Brightness Images*", Proceedings IEEE Intl. Conference on Computer Vision 90*, pp. 12-20, 1990.

J. R. Kender and E. M. Smith, (1987), "Shape from Darkness: Deriving Surface Information from Dynamic Shadows*", Proceedings IEEE Intl. Conference on Computer Vision 87*, pp. 539-546, 1987.

I. Sato, Y. Sato and K. Ikeuchi, (1999a), "Illumination Distribution from Shadows", *Proceedings IEEE Conference on Computer Vision and Pattern Recognition 99*, pp. 306-312, June 1999.

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

I. Sato, Y. Sato and K. Ikeuchi, (1999c), "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.

Y. Sato, M. D. Wheeler and K, Ikeuchi, (1997), "Object shape and reflectance modeling from observation", *Proceedings of SIGGRAPH 97*, pp. 379-387, August, 1997.

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

Y. Yu, P. Debevec, J. Malik and T. Hawkins, (1999), "Inverse global illumination: Recovering reflectance models of real scenes from photographs", *Proceedings of SIGGRAPH 99*, pages 215--224, August 1999.