Illumination Estimation from Shadow and Incomplete Object Shape Captured by an RGB-D Camera

Takuya Ikeda[1], Yuji Oyamada[2], Maki Sugimoto[3], and Hideo Saito[4]
Graduate School of Science and Technology, Keio University, Japan
{[1]*ikeda,* [2]*charmie,* [4]*saito*}*@hvrl.ics.keio.ac.jp,* [3]*sugimoto@ics.keio.ac.jp*

Abstract

Optical consistency between the real world and the virtual objects is one of the important issues in Augmented Reality (AR). This paper proposes a method to estimate illuminations from an object shadow and incomplete object shape information captured by an RGB-D camera. The environmental illumination can be estimated without any prior knowledge of the object shape. The radiance of each light source is computed by solving linear equations derived from color and depth images. Since a priori knowledge of object shape is not necessary, we can flexibly use/combine any objects. Thus, we can use/combine multiple objects in the target scene to increase the accuracy and flexibility. Experimental results show the characteristics of our proposed method.

1. Introduction

AR adds virtual information to the real world so that human can perceive more information from their experience. To realize realistic AR, especially, any virtual objects need to be consistent with the real world. One of the important consistencies is optical one such as shading on the virtual objects and shadow cast by them. For this purpose, two key components are necessary; realistic rendering technique and accurate environmental information.

To calculate the light condition of the target scene, researchers have proposed various methods. To realize the optical consistency in AR, one of the well-used solutions is to estimate the environmental light condition from camera feedback.

There exist various methods estimating light distribution.The methods are categorized into two types. The first method directly acquires the light distribution of the target scene from the camera observation. The point of this method is how to observe wider angle view from the camera(s).one of the aims one intuitive method is to attach a fish eye lens to the camera [2]. Another solution puts a spherical mirror in the scene and estimates the light distribution from the reflection on the mirror [1, 4]. The Second indirectly estimates the light distribution. This method is based on a physics phenomenon that shadow is generated by objects obstructing light sources. The idea of the indirect method is to estimate light distribution from shadow appearing on the image(s), so called illumination from shadow. Modeling shadow as a product of scene geometry, reflectance property, and light distribution inverse operation can recover the light distribution from the shadow. Sato et al. assumed that there is a known shape object in the scene so that light distribution can be computed with simple least square methods [5].

Merit of the former method is simplicity. Since camera directly observes light distribution, we can easily recover it. However, we should prepare special equipments such as fish eye lens and spherical mirrors. On the other hand, illumination from shadow methods does not require them. However, their assumption about a priori knowledge of object shape limits the usage environment.

The purpose of this paper is to reduce the limitation of existing illumination from shadow methods. Specifically, our proposed method aims to estimate the light distribution without a priori knowledge of object shape. For this purpose, we introduce a depth camera and use the incomplete object shape information obtained by it instead of using the complete shape information. Even though the single fixed depth camera provides incomplete shape information, such information can be useful to estimate the light distribution. Since our method does not require a priori knowledge of object shape, we can flexibly use/combine any objects. This characteristic allows us to use multiple objects in the target scene so that we can expect that the objects contribute to increase the accuracy of light distribution estimation.

2. Illumination Estimation using an RGB-D Camera

This section describes our proposed method. We use calibrated color camera and depth camera that every pixel has color information (RGB) and depth information (D). First of all, we briefly derive the basic idea of calculating illumination from shadow, the reader can refer to [5] for complete equations. Our method takes three steps: Scene segmentation (Sec.2.2), scene geometry acquisition (Sec.2.3), and solution of linear equation (Sec.2.4). In this paper, we assume that an object is placed on a flat Lambertian plane.

2.1. Illumination from Shadow

Considering the light source in all directions, we model that light sources are distributed on a hemisphere with an infinite distance radius. We follow a mathematical form presented by Sato et al. [5]. Due to the lack of space, let us define discrete equation. Pixel value $i(\boldsymbol{u})$ observed at a point $\boldsymbol{x}$ on a flat plane is written as

$$i(\boldsymbol{u}) = \sum_{n=1}^{N} V_n(\boldsymbol{x}) R_d L_n \cos\theta_n, \qquad (1)$$

where $\boldsymbol{u}$ represents pixel position, $V_n(\boldsymbol{x})$ is a visibility term of n-th light source w.r.tlet@tokeneonedota point $\boldsymbol{x}$. The function $V_n(\boldsymbol{x})$ is 1 if the n-th light source is visible, otherwise is 0. A constant R_d is a diffuse reflection parameter of the plane. Since Lambertian surface uniformly reflects the light to all directions, R_d is set to a constant value. L_n represents the radiance of n-th sampling light source. Its direction is defined by θ_i and ϕ_i which are the elevation and azimuth angles w.r.tlet@tokeneonedotthe point $\boldsymbol{x}$ respectively. By increasing θ_i and ϕ_i by a constant degree, it is possible to be set the sampling light source on the hemisphere. Light intensity $\cos\theta_n$ is computed from the angle of the light source and the point $\boldsymbol{x}$.

Equation (1) tells us that a pixel in the shadow area gives one linear equation. If we know the scene geometry, *i.e*let@tokeneonedot, 3D shape of an object in the target scene, we can estimate N light source distribution by least square methods from $M >= N$ pixels deriving M linear equations [5].

2.2. Segmentation

To obtain the visibility term $V_n(\boldsymbol{x})$ and light intensity $\cos\theta_n$, we need to know the location of shadow and object area separately. Our proposed method estimates these two by background subtraction using both color images and depth images.

We use four input images and their pixel values are $c_{\text{obj}}(\boldsymbol{u})$, $c_{\text{bg}}(\boldsymbol{u})$, $d_{\text{obj}}(\boldsymbol{u})$, and $d_{\text{bg}}(\boldsymbol{u})$, where the variable c and d denote color image and depth image and the subscripts bg and obj denote background image and object image respectively. Each difference image is defined as follows

$$c_{\text{diff}}(\boldsymbol{u}) = \begin{cases} 1 & \text{if } |c_{\text{obj}}(\boldsymbol{u}) - c_{\text{bg}}(\boldsymbol{u})| > \tau_{\text{c}} \\ 0 & \text{otherwise} \end{cases} \qquad (2)$$

$$d_{\text{diff}}(\boldsymbol{u}) = \begin{cases} 1 & \text{if } |d_{\text{obj}}(\boldsymbol{u}) - d_{\text{bg}}(\boldsymbol{u})| > \tau_{\text{d}} \\ 0 & \text{otherwise} \end{cases} \qquad (3)$$

where τ represents the threshold for detection, the subscript diff denotes difference image.

When we put an object in the scene, the depth image gets difference only on object area while the color image does on both object and shadow area. Using these differences, we segment the shadow area and object area separately. First, we detect object area. Since object area corresponds to $d_{\text{diff}}(\boldsymbol{u}) = 1$, we set object area $a_{\text{obj}}(\boldsymbol{u}) = d_{\text{diff}}(\boldsymbol{u})$. Next, we detect shadow area. Utilizing the difference between $c_{\text{diff}}(\boldsymbol{u})$ and $d_{\text{diff}}(\boldsymbol{u})$ mentioned above, we extract the shadow area by taking exclusive OR of the difference images as $a_{\text{shadow}}(\boldsymbol{u}) = c_{\text{diff}}(\boldsymbol{u}) \bigoplus d_{\text{diff}}(\boldsymbol{u})$.

2.3. Computation $V_n(\boldsymbol{x})$

Next, we compute the visibility function $V_n(\boldsymbol{x})$ from the depth image $d_{\text{obj}}(\boldsymbol{u})$ and the segmented areas $a_{\text{obj}}(\boldsymbol{u})$ and $a_{\text{shadow}}(\boldsymbol{u})$. As mentioned above, the depth camera provides incomplete shape information of the object because the backside of it w.r.tlet@tokeneonedotthe depth camera is invisible. Thus, there are three types of possibilities for the visibility function: visible and n-th light reaches the point $\boldsymbol{x}$ resulting $V_n(\boldsymbol{x}) = 1$; visible and object blocks n-th light to the point $\boldsymbol{x}$ resulting $V_n(\boldsymbol{x}) = 1$; and invisible. There are several choices how to deal with the invisible case. One omits the corresponding $V_n(\boldsymbol{x})$ from linear equations while another somehow sets the term 1 or 0. To increase the number of linear equations, we do not omit the corresponding $V_n(\boldsymbol{x})$. If we set the $V_n(\boldsymbol{x})$ to 1, this means that the backside of the object has a flat shape like a cliff, but this assumption is too restrictive. Thus, we set $V_n(\boldsymbol{x})$ corresponding to the invisible area to 0.

2.4. Solution of Linear Equations

Now, we have $V_n(\boldsymbol{x})$, $\cos\theta_n$, and the constant value R_d computed from the color background image c_{bg}.

When the shadow area has M pixels, we can derive M linear equations each of which corresponding to each pixel on its. Since any light intensity should have a positive value, we estimate the light distribution by non-negative least-square algorithm [3].

3. Experiments and Results

This section validates our proposed method by using real images of indoor environments. We estimated the light distribution using our proposed method and Sato et al.'s [5] method. After that, we superimposed the shadow of the virtual object in real scene by using estimated light distribution. In this experiment, Microsoft Kinect is used as an RGB-D camera. We used eight various objects (shown in Fig. 1). They include both simple shapes (e.g., **Box**, **Hemisphere**) and complex ones (e.g., **Duck1**, **Doll1**). As mentioned above, our method can flexibly allow multiple objects. We combined the linear equations of the multiple objects, except for **Box**, will be called **Combined**.

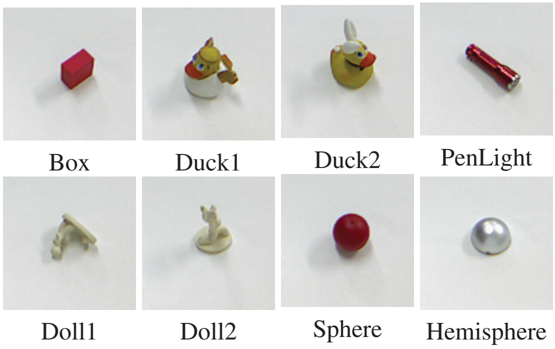
Bottom row of the right column in Fig. 2 shows the scenes captured by a fish-eye lens camera. The positions of the real light sources (fluorescent lamps) are marked with blue circles and the position of the RGB-D camera is marked with a red circle.

For the discretizing the light distribution, we had one for every 5 degrees in $0 \leq \theta \leq 70$, assuming the light sources are not located around the ground plane and for each 10 degrees in $0 \leq \phi < 360$. Thus, the number of the light sources N is 505 degrees. Our method is implemented as follow, OS:Windows 7, CPU:Intel Core i7-2600 3.40GHz, RAM:3.49GB, implement environment:Microsoft Visual C++ 2010.

3.1. Light Source Estimation

Fig. 2 shows the estimated results. Due to the lack of space, we show some of the results. First row shows object images; second row is the estimated light distributions. First three columns show the results of three of eight objects: **Box**, **Duck2**, and **Hemisphere**. Fourth column is **Combined**. Fifth column shows the Sato et al.'s method. Since it requires a priori knowledge of object shape, we used same **Box** for estimation.

The objects' shadows show that it seems to have two bright light sources. As mentioned above, our method uses incomplete shape information, the result of Sato et al.'s method with the complete shape information can be regarded as an ideal result. Thus, it cannot overcome their method in estimation accuracy. In this experiment, we regarded the result of Sato et al.'s method as Ground Truth and evaluated the errors between our method and theirs one.


Box Duck1 Duck2 PenLight

Doll1 Doll2 Sphere Hemisphere

Figure 1. Eight Objects using out experiments.

In second row of Fig. 2, even the results are different according to the objects, all the results have a bright light source at the bottom center and darker light sources are distributed. Furthermore, compared with **Real scene**, they tend to have errors of light sources around the RGB-D camera. It is due to the lack of information of linear equations corresponding to an invisible shadow area which is mentioned in [6].

We evaluated the light distribution of our method and the Ground Truth. Table. 1 shows the RMSE and standard deviation (Std. dev.) between our proposed and Sato et al.'s methods. Each estimated sampling light takes a value from 0 to 1. As mentioned above, these methods have theoretical mis-estimation around the RGB-D camera. In this evaluation, we ignored 13 light sources around the camera to reduce the effect of the mis-estimation. From Table. 1, **Hemisphere** and **PenLight** are worse than the others. Compared with other object images in Fig. 1, especially, **Hemisphere** has only one visible shadow from center to bottom left direction and the shadows cast smaller area. Other object images seem to have two shadows from center to bottom left and right direction. It causes that the linear equations can't be included the information of true shadow from center to right direction cast by present light sources. On the other hand, **Combined** is best of all results. Because each object shadow has different shape of its shadow, linear equations can be included many shadow information by using multiple objects.

3.2. Shadow Rendering

For quantitative evaluation, we performed the shadow rendering results by using each estimated light distribution as shown in the first row of Fig. 3. As same as the estimated light distribution, we regarded Sato et al.'s result as Ground Truth. The second row of Fig. 3 shows error map comparing with Ground Truth. Comparing with the real scene, Sato et al.'s method well-rendered two large shadows. On the other hand, our proposed method seems to have lack of them. We computed the evaluation value of the rendering shadows. Table. 2 shows the RMSE, Std. dev., and Maximum (Max) value of the absolute difference between our proposed and Sato et al.'s methods.

These results concern how clearly shadows are observed on the image. Looking at object images, the shadows appear differently. **Box** case has two large shadows, thus, both methods can estimate the corresponding light sources. Since the difference between **Box** and Sato et al.'s method is only the incompleteness of shape information, we can conclude that **Box**'s error is caused by shape incompleteness. **Duck2** case has also two shadows but they are relatively less clear than **Box**. These unclearer shadows and object's complex shape cause the result worse than **Box**. Surprisingly, **Hemisphere** shows worst result even though it's one of the simple objects. Comparing with the other two cases, as mentioned before, **Hemisphere** has only one visible shadow. This condition can be the error cause. **Combined** result is almost as same as **Duck2** and worse than **Box**. Comparing with **Duck2**, its maximum error is reduced and mean and standard deviation is slightly reduced. One plausible cause is that some error/noise makes the light distribution result worse.

4. Discussions

These experiments show some characteristics of our proposed method. Comparing with Sato et al.'s result, illumination estimation with incomplete shape information provides slightly worse results. This is the limitation of our proposed method. Basically, a simpler shape object such as **Box** provides better result than a complex shape object. There are two possible causes of this observation. One is the invisible area. Since **Box** has less invisible volume than other cases, the number of mis-estimation of $V_n(\boldsymbol{x})$ is less than the others. Another cause is shape complexity. With a complex shape objects, depth camera provides worse measurement errors. This makes $V_n(\boldsymbol{x})$ worse even $V_n(\boldsymbol{x})$ corresponding to visible area. Comparing **Box** and **Hemisphere**, it is clear that our method requires as much linear equations as possible even with simple shape objects. If one object is used in our method, the ideal object is like **Box**, because its shape is simple and its shadows can be clearly observed.

Combining linear equations from multiple objects has both pros and cons. The important contribution of our proposed method is that **Combined** provided better result than each component such as **Duck2** and **Hemisphere** as we expected. However, **Box**'s result is better than **Combined**, which means that combining the linear equations also accumulates mis-estimation of $V_n(\boldsymbol{x})$. Since we rely on simple non-negative least squares, our method is sensitive to noises or large errors. One potential idea to conquer this difficulty is to apply more sophisticated optimization such as sparsity constraint or object shape constraint. Since the object shape indirectly relates to $V_n(\boldsymbol{x})$, the developing optimization can be a good future direction.

In our method, the estimation process does not run in real time. (For example, computation time of **Box** is about 40 min.) Our method takes a lot of time in the calculation of $V_n(\boldsymbol{x})$. Also, the shadow rendering process takes about 50 min. However, we consider that its process can be real time by using computer graphic technics which are used in [4].

5. Conclusions

This paper proposes a method to estimate light distribution from shadow and incomplete object shape using an RGB-D camera. In our method, to estimate the light distribution, visibility term $V_n(\boldsymbol{x})$ is computed from incomplete object shape. The experimental results show that our proposed method can estimate the light distribution even with incomplete object shape information. In our experiments, we confirmed the limitation of our proposed method, derived some characteristics of it, and found potential future direction.

References

[1] P. Debevec. Rendering synthetic objects into real scenes: bridging traditional and image-based graphics with global illumination and high dynamic range photography. *In Proc.IEEE SIGGRAPH*, pages 189–198, 1998.

[2] M. Knecht, C. Traxler, O. Mattausch, W. Purgathofer, and M. Wimmer. Differential instant radiosity for mixed reality. *In Proc.IEEE ISMAR 2010*, pages 99–107, 2010.

[3] C. L. Lawson and R. J. Hanson. *Solving Least-Squares Problems*, chapter 23, page 161. Prentice-Hall, 1974.

[4] D. Nowrouzezahrai, S. Geiger, K. Mitchell, R. Sumner, W. Jarosz, and M. Gross. Light factorization for mixed-frequency shadows in augmented reality. *In Proc.IEEE ISMAR 2011*, pages 173–179, 2011.

[5] 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. *In Proc.IEEE CVPR*, pages 875–882, 1999. 86.

[6] I. Sato, Y. Sato, and K. Ikeuchi. Stability issues in recovering illumination distribution from brightness in shadows. *In Proc.IEEE CVPR,*, pages 400–407, 2001.

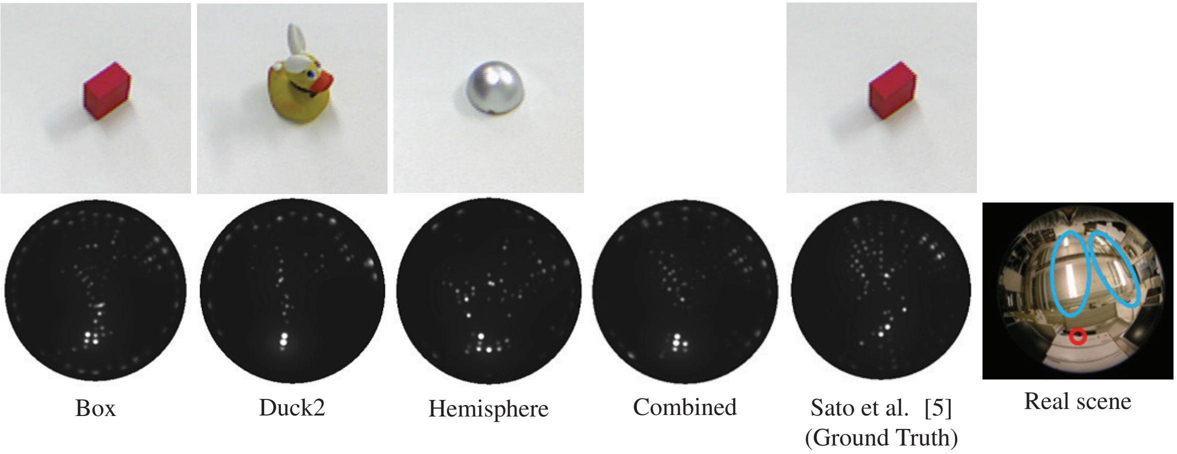


Figure 2. Light estimation results:The top row shows object images using our experiments, the bottom row shows estimated light distributions. the bottom row of right column shows the scene captured by a fish-eye lens camera.

Table 1. Evaluation values of estimated light distributions in Fig. 2.

Object name	Box	Duck1	Duck2	PenLight	Doll1	Doll2	Sphere	Hemisphere	Combined
RMSE ($\times 10^{-3}$)	6.9	6.6	6.9	8.3	7.3	7.0	7.6	8.2	**6.4**
Std.dev. ($\times 10^{-3}$)	7.6	7.1	7.5	9.1	8.0	7.0	7.6	8.2	**7.0**

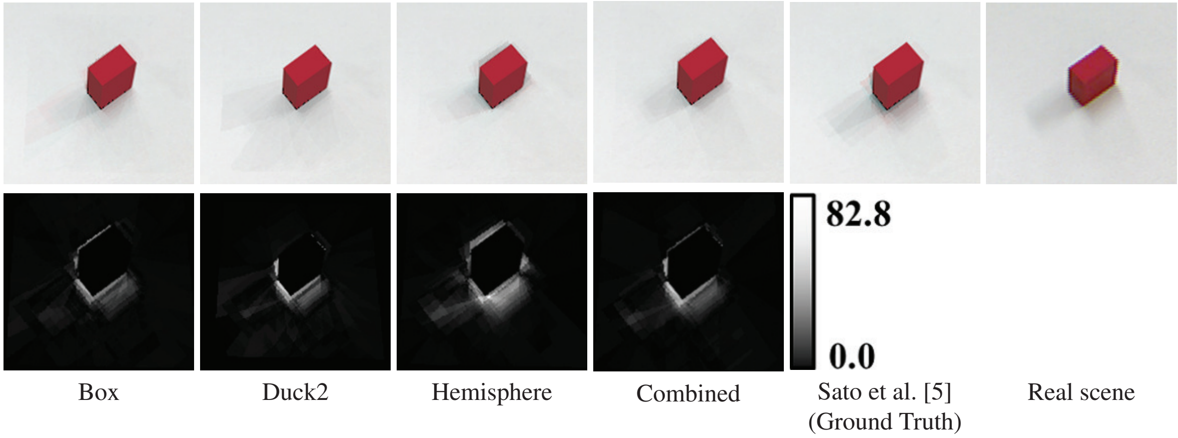


Figure 3. Shadow rendering results: the top row shows shadow rendering results using estimated light distributions, the bottom row shows error maps with Sato et al. [5].

Table 2. Evaluation values of shadow rendering results shown in Fig. 3.

Object name	Box	Duck1	Duck2	PenLight	Doll1	Doll2	Sphere	Hemisphere	Combined
RMSE [pixel value]	**6.1**	9.1	7.3	9.1	9.1	7.8	7.6	9.8	7.2
Std.dev. [pixel value]	**5.3**	8.1	6.4	7.7	8.3	6.3	6.8	8.9	6.4
Max [pixel value]	**58.9**	128.7	70.6	90.0	140.7	68.6	66.5	82.8	66.0