1T−7

Manipulation of Diffuse Light in Single Input Image by Modeling Diffuse Light

Wu, Jung-Hsuan Suguru Saito
Graduate School of Information Science & Engineering, Tokyo Institute of Technology

1 Abstract

In this paper, we present a simple method to model the diffuse reflection in an image which allows the user to turn on and off the lights, without knowing the scene geometry nor the camera parameters. The user first roughly marks the area lit by a light source in the input image and then our method will approximate the diffuse lighting by creating a virtual light source and fitting its lighting effect to the input image. With the obtained model, the user can modify the lighting in the input image very intuitively.

2 Related Work

As a typical problem in computer graphic field, there are a large amount of research about rendering and estimating the lighting in a scene or in an image. Yu et al.[3] estimate the specular reflectance and diffuse albedo of the surfaces in a scene by several input images. Bai et al.[1] presented a method of seeking the inverse lighting process from forward lighting process to eliminate the whole lighting from a single image. However, both research require the knowledge of geometry of the whole scene and all of the parameters of camera, which are difficult to be computed even with assistance from user.

3 Algorithm

We assume that the diffuse lighting which is to be processed in the input image is produced by a point light source, and is a circle on a plane of isotropic material, which means that the intensity of reflection light is independent of viewing direction.

Our method starts by requiring the user to mark five points on the boundary of the lit area: the center, the top, the left, the right, and the bottom of the area, as shown in Figure 1(b). Section 3.1 describes how intensity and position of light source will be iteratively optimized, and then we introduce an idea to estimate the color of reflection by material segmentation in Section 3.2. The method of light intensity modification will be mentioned in Section 3.3.SSSSSSSS

3.1 Diffuse Light Approximation

Eqn. 1 is the diffuse lighting model used in our method.

$$L(p) = M(P)L_{src}\cos(P) + M(P)L_{min}$$
$$\text{where } \cos(P) = \frac{\Delta P \cdot N}{|\Delta P|} \text{ and } \Delta P = P_{src} - P. \tag{1}$$

$L(p)$ is the color of pixel p in image. P, $M(P)$, and N are the corresponding position of pixel p in virtual 3D space, the color of material on P, and the normal vector on P, respectively. L_{src} and P_{src} are the color and the position of the light source. L_{min} is the ambient light and is assumed to be a constant gray light everywhere.

Initial Guess

Proper estimation of P_{src} is needed to optimize the approximation, however, during optimization, the intensity of L_{src} is also necessary, so we have to deal with both of them together. For simplisity, we let $I(p)$, I_{min}, and I_{src} represent the intensity of $L(p)$, L_{min}, and L_{src}, and let A represent the area marked by user.

The initial guess I^0_{src} is obtained by subtracting I_{min} from the intensity of brightest pixel in A. The roughly estimation of I_{min} is computed by averaging the intensity of the darkest 1/10 pixels in A. 0 can be the initial guess of x and z coordinate of P^0_{src} but not for the y coordinate unless the light source is set on the ground. The initial guess of y coordinate is inferred as

$$P^0_{src}y = \frac{1}{n}\sum_{p\in A}\sqrt{\frac{\cos^2(P)\Delta Px^2 + \cos^2(P)\Delta Pz^2}{1-\cos^2(P)}}, \tag{2}$$

where n is the number of pixels in A. Notice that $\cos(P)$ in Eqn. 1 is just a common definition and the value is still unknown. Fortunately, we can calculate it by

$$\cos(P) = \frac{I(p) - I_{min}}{I^0_{src}}. \tag{3}$$

Optimization

We first define the error caused by one pixel as

$$E(P) = (I(p) - I_{min}) - \frac{\Delta Py}{|\Delta P|}I_{src}. \tag{4}$$

The error can be minimized by updating P_{src} and I_{src} as

$$P^{k+1}_{src}y = P^k_{src}y + \frac{1}{n}\sum_{P\in A}\left(E(P)I^k_{src}\frac{\Delta P^k x^2 + \Delta P^k z^2}{|\Delta P^k|^3}\right), \tag{5}$$

$$P^{k+1}_{src}x = P^k_{src}x - \frac{1}{n}\sum_{P\in A}\left(E(P)I^k_{src}\frac{\Delta P^k x \Delta P^k y}{|\Delta P|^3}\right), \tag{6}$$

$$I^{k+1}_{src} = \frac{\Sigma_{P\in A}\left(\frac{\Delta P^k y}{|\Delta P^k|}(I(p) - I_{min})\right)}{\Sigma_{P\in A}\left(\frac{\Delta P^k y}{|\Delta P^k|}\right)^2}, \tag{7}$$

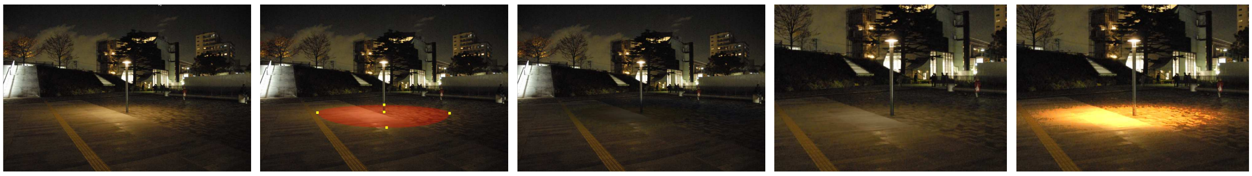
The z coordinate of P^k_{src} can be updated by Eqn. 6 with $\Delta P^k x$ replaced by $\Delta P^k z$.

3.2 Material Segmentation

Since we know the position of light source, it is possible to esitimate the reflection color in pixel p by

$$M(P)L_{src} = \frac{L(p) - M(P)L_{min}}{cos(P)}. \tag{8}$$

The $M(P)L_{min}$ term can be determined by similar way how a rough I_{min} is computed if all pixels have similar reflection property. Nevertheless, as shown in Figure 2, in most cases the materials of pixels are so different that it is difficult to find a $M(P)L_{min}$ for all pixels. Therefore, it is necessary to distinguish the materials and their reflection property.



(a) Input image (b) User input (c) Result with the lamp turned off (d) Result with the lamp intensity decreased to 80% (e) Result with the lamp intensity increased to 120%

Figure 1: The input and result images.

Figure 2: Comparison between the result of turning off the lamp after material segmentation process(left) and the result computed with an unique reflection color(right).

We refer to the work of Wang et al.[2] to build superpixels. Their work uses only the pixel's color when separating superpixels, however, the color is largely effected by lighting. To avoid being effected, we use both color and reflection coefficient, where reflection coefficient is $L(p)/(L'_{src}\cos(P))$ and L'_{src} is a rough estimation of L_{src} by averaging Eqn. 8 over all pixels in A. Then apply kNN Clustering to superpixels to segment the different materials. The result of segmentation is shown in Figure 3.

For each segment i, all pixels in it should have a unique material M^i. We sort and separate the superpixels into two groups A^i_{max} and A^i_{min} according to $\cos(P^j)$, where P^j is the centroid of superpixel j. Then we can compute $M^i L^i_{src}$ for segment i by

$$M^i L^i_{src} = \frac{L^i(p)_{max} - L^i(p)_{min}}{\cos^i(P)_{max} - \cos^i(P)_{min}}, \quad (9)$$

where $L^i(p)_{max}$ and $L^i(P)_{min}$ are the average color, and $\cos^i(p)_{max}$ and $\cos^i(P)_{min}$ are the average value of $\cos(P^j)$ of superpixels in A^i_{max} and A^i_{min}. Then $M^i L^i_{min}$ can be estimated as

$$M^i L^i_{min} = \frac{1}{n^i} \sum_{j \in A^i} L(p) - M^i L^i_{src} \cos(P^j), \quad (10)$$

where A^i is the set of superpixels in segment i and n^i is the size of the set.

Finally, the terms $M(P)L_{src}$ and $M(P)L_{min}$ of pixel p can be computed by weighted average over all estimated $M^i L^i_{src}$ and $M^i L^i_{min}$.

$$M(P)L_{src} = \frac{1}{\mathrm{prob}_t} \sum_i \mathrm{prob}^i(L(p)) M^i L^i_{src}$$

$$M(P)L_{min} = \frac{1}{\mathrm{prob}_t} \sum_i \mathrm{prob}^i(L(p)) M^i L^i_{min}, \quad (11)$$

where $prob^i(L(p))$ is computed by first using 3-dimensional gaussian model to approximate the color distribution of pixels in segment i and then calculating the possibility of pixel p belonging to segment i, which stands for similarity between pixel

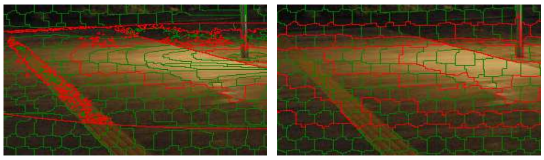


Figure 3: The result of segmentation using reflection coefficient(left) and without using reflection coefficient(right). The red line is the boundary of segmented materials and the green line is the boundary of superpixels.

p and segment i, and prob_t is the sum of all $prob^i(L(p))$ for normalization.

3.3 Light Intensity Modification

Now all necessary information is known, the lighting in input image can be modified by decreasing or increasing the $M(P)L_{src}$ term in Eqn. 1.

Notice that if we set the value of $M(P)L_{src}$ term to 0, then we actually "turned off" the light, as shown in Figure 1(c).

4 Result

Figure 1 shows the result of lighting after intensity modification by our algorithm. The processor of the machine we used for experiments is AMD Phenom II X4 940, and the memory is 4GB. The size of input image is 800x536 and the total computation time for the estimation is about 10 seconds and the relighting process costs less than 0.1 second.

5 Conclusion

We proposed a simple method to model the diffuse reflection in an image without any requirements of pre-knowledge or assumptions on the scene and camera and a novel idea to utilize the reflection coefficient for segmentation to prevent from being effected by lighting. Furthermore, we provided a friendly user interface to manipulate the lighting intuitively.

However, there are still many issues to be solved such as estimating the lighting on a complex geometry, and challenging more different types of light sources.

References

[1] Jiamin Bai, Manmohan Chandraker, Tian-Tsong Ng, and Ravi Ramamoorthi. A dual theory of inverse and forward light transport. In *Proceedings of ECCV'10: Part II, pp. 294-307.* Springer-Verlag, 2010.

[2] Jie Wang and Xiaoqiang Wang. Vcells: Simple and efficient superpixels using edge-weighted centroidal voronoi tessellations. *IEEE Trans. PAMI, 34(6): pp. 1241-1247*, June 2012.

[3] Yizhou Yu, Paul Debevec, Jitendra Malik, and Tim Hawkins. Inverse global illumination: recovering reflectance models of real scenes from photographs. In *Proceedings of SIGGRAPH '99, pp. 215-224*, 1999.