Feature-Based Interpolation of Intensities for Interactive Editing of Environmental Light Effects

Tada, Munehiro
Graduate School of Information Science and Technology, Hokkaido University

https://hdl.handle.net/2324/1430833

出版情報：MI lecture note series. 50, pp.116-121, 2013-10-21. 九州大学マス・フォア・インダストリ研究所
バージョン：
権利関係：

Feature-based Interpolation of Intensities for Interactive Editing of Environmental Light Effects

Munehiro TADA

Graduate School of Information Science and Technology, Hokkaido University

(joint work with Yoshinori DOBASHI, Tsuyoshi YAMAMOTO)

In computer graphics, shading techniques play an important role to display virtual objects. Environmental map is a popular technique to create realistic results with low computational cost. However, it is not guaranteed to generate user-desired shading effects. Therefore, many methods have been proposed for editing shading effects obtained by environmental map. However, these methods are limited to the editing of global shading effects. In this paper, we propose a system that allows the user to specify desired-intensities on arbitrary positions. Our system enables the user to design both local and global shading effects intuitively.

1. INTRODUCTION

Recently, computer graphics techniques have been used in many applications such as movies, games and commercial films. One of the most important techniques for displaying realistic virtual objects is the ability to achieve appropriate shading. To create photorealistic results, many physically-correct methods have been proposed[1][2]. However, these physically-correct methods require high computational cost. Therefore, it is a time consuming process to obtain user-desired shading results by using these methods. To address this problem, image-based lighting(IBL) techniques have been proposed to create realistic results with reasonable computational cost[3]. These techniques employ environmental maps as virtual light sorces[4]. Environmental maps are used to model real-world lighting environment and reproduce complicated shading effects caused by the natural illumination. One problem in these techniques is that the results are often different from the user-desired shading effects. Therefore, many methods have been proposed to address this problem[5][6]. These methods allow the user to edit environmental maps intuitively to create user-desired shading effeces. However, the results are limited to global shading effects obtained by environmental lights, since these methods are based essentially on editing environmental maps. Therefore, it is difficult to edit shading effects locally. In this paper, we propose an editing system that allows the user to design both local and global shading effects intuitively.

Our system is based on feature-based interpolation of intensities and enables the user to edit shading effects obtained by environmental maps. In our system, the user directly specifies desired intensities on arbitrary positions of surfaces by using control points. Then, the shading effects are generated by feature-based interpolation of the user-specified intensities. We use features that are often used in the shading calculations, such as positions, normals, and visibilities, in order to produce plausible shading effects. By taking into account of positions as features for interpolation, the system achieves to edit shading effects locally. The interpolation function for the intensities is represented by a linear combination of radial basis functions (RBFs). The user can interactively add, move, or delete the control points, and the resulting shading effects

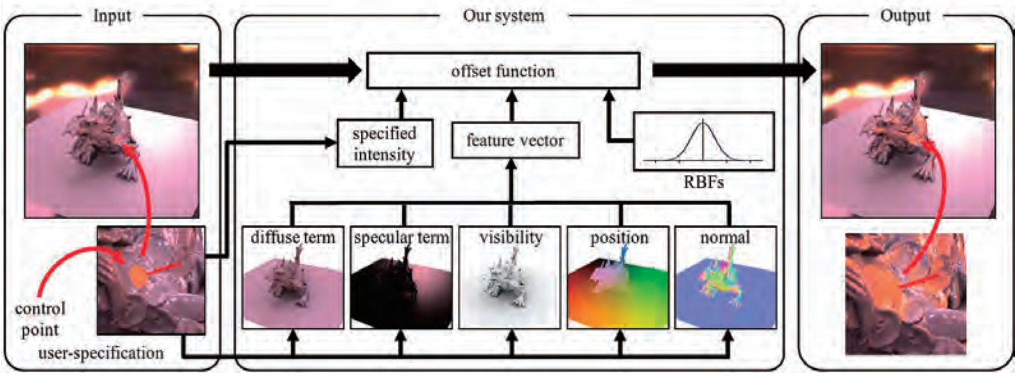

Figure 1. Overview of proposed method

are displayed in real time. Furthermore, our system allows the user to edit diffuse, and specular terms separately.

2. Previous Work

As we mentiond before, it is difficult to generate desired shading effects by using physically-correct shading models. Therefore, various methods have been proposed to edit shading effects even though the results are not physically-correct. Ritschel et al. proposed an editing method for specular reflection [7]. However, this method is limited to ideal specular reflections. Todo et al. proposed an editing method for cartoon shading effects [8]. However, this method is not suitable for photorealistic effects. Obert et al. proposed an editing method for shading effects by editing visibility functions [9]. Ritschel et al. proposed a method for deforming on-surface signals [10]. However, these methods aim to deform shadow and shading, and then it is difficult to specify user-desired intensities on arbitrary positions by using these methods. Methods for solving an inverse lighting problem for automatic placement of light sources has been proposed [11][12]. However, these methods cannot reflect the user's intention on the resulting image.

In this paper, we propose an interactive editing system for shading effects obtained by environmental map. Okabe et al. proposed a method for calculating an environmental light by solving inverse problem that can produce specified shading effects [5]. However, it is not guaranteed that the specified shading effects are always achieved. Therefore, Pellacini proposed a method for modifying exiting environmental map [6]. However, in these methods, it is difficult to edit shading effects locally. Our system enables the user to edit shading effects locally.

3. Proposed Method

In our method, the intensity at point x on the object surfaces viewed from ω_o is computed by using the following equation.

$$I\left(x,\omega_o\right)=k_d\left(x\right)I_d\left(x\right)+k_s\left(x\right)I_s\left(x,\omega_o\right), \tag{1}$$

where I_d and I_s represent the intensities due to diffuse reflection and specular reflection, k_d and k_s are reflection coefficients for each reflection term respectively. Fig. 1 shows an overview of our method. The user selects one of the two terms in Eq. 1 and places multiple control points on the surfaces of objects to specify the desired intensities at the positions of the control points. In Fig. 1, a single control point is placed as indicated by the orange sphere. The color of the sphere corresponds to the user-specified intensity. The red line indicates the normal direction at the control point. Our system then computes intensities on the surfaces of objects by using offset functions. Offset functions are determined by user-specified intensities and features that are often used in the shading calculation such as normal, position, and visibility. There are two offset functions corresponding to the two terms in Eq. 1. I_d and I_s are represented by:

$$I_d\left(x\right) = \hat{I}_d\left(x\right) + h_d\left(x\right) \tag{2}$$

$$I_s\left(x, \omega_o\right) = \hat{I}_s\left(x, \omega_o\right) + h_s\left(x, \omega_o\right), \tag{3}$$

where h_d and h_s are the offset functions. The offset functions are represented by using RBFs. $\hat{I}_d$ and $\hat{I}_s$ are intensities before editing, and represented by:

$$\hat{I}_d\left(x\right) = \int_\Omega L\left(\omega_i\right) V\left(x, \omega_i\right) \left(n_x, \omega_i\right) d\omega_i \tag{4}$$

$$\hat{I}_s\left(x, \omega_o\right) = \int_\Omega L\left(\omega_i\right) V\left(x, \omega_i\right) \left(\left(n_x, \omega_o\right)\right)^s d\omega_i, \tag{5}$$

where Ω is a hemispherical domain, V is a visibility, ω_i is an incident direction, n_x is a normal at point x, and s is a coefficient to determine the shininess of specular reflections. In our system, visibility is precomputed at each position. The user can interactively add, move, or delete the control points. During the user's operation, the offset functions are updated in real time and the resulting shading effects are displayed.

4. OFFSET FUNCTIONS

4.1. **Constructing Offset Functions.** The offset functions, h_d and h_s are constructed by using the user-specified intensities. We will explain the construction of the offset function for the diffuse term, h_d, only. h_s is constructed in the same way. Let us assume that the number of control points is N, the user-specified intensity for each control point is $I_d^\star\left(x_i\right)$ $(i = 1, 2, \cdots, N)$, and the position of the i-th control point is x_i. The shading interpolator h_d is represented by:

$$h_d\left(x\right) = \sum_{i=1}^{N} a_i \phi\left(\left\|\mathbf{f}_d\left(x_i\right) - \mathbf{f}_d\left(x\right)\right\|\right), \tag{6}$$

where $\mathbf{f}_d\left(x_i\right)$ is a feature vector at x_i (see Section 4.2). a_i is a coefficient for each interpolation kernel ϕ, which is expressed by RBF. ϕ is given by:

$$\phi\left(r\right) = \exp\left(-r^2\right). \tag{7}$$

a_i is computed by using the least square method such that the following function is minimized:

$$\arg\min_{a_i} \sum_{j=1}^{N} \left(I_d^\star\left(x_j\right) - \left(\hat{I}_d\left(x_j\right) + h_d\left(x_j\right)\right)\right)^2, \tag{8}$$

where $\hat{I}_d\left(x_j\right)$ is diffuse component before editing. Our system computes a_i in real-time.

4.2. **Feature Vectors.** This section describes the features for the diffuse and specular terms. We employed each terms before editing, $\hat{I}_d(x)$ and $\hat{I}_s(x, \omega_o)$, as features to edit each term. In addition to $\hat{I}_d(x)$ and $\hat{I}_s(x, \omega_o)$, the coordinate p_x, the normal n_x, and visibility vector $\mathbf{V}(x) = (V(x, \omega_1), V(x, \omega_2), \cdots, V(x, \omega_n))$ of the calculation point x are also included in the feature vector. Use of the visibility vector $\mathbf{V}$ enable the user to edit soft shadows. Use of the coordinate p_x allows the user to edit the shading effects locally. Use of the normal n_x makes it possible to control the shading effects according to the direction of surface. Consequently, the feature vectors, $\mathbf{f}_d(x)$ and $\mathbf{f}_s(x)$, are defined by:

$$\mathbf{f}_d(x) = \left(\frac{\hat{I}_d(x)}{\sigma_d}, \frac{\mathbf{V}(x)}{\sigma_v}, \frac{p_x}{\sigma_p}, \frac{n_x}{\sigma_n} \right) \tag{9}$$

$$\mathbf{f}_s(x, \omega_o) = \left(\frac{\hat{I}_s(x, \omega_o)}{\sigma_s}, \frac{\mathbf{V}(x)}{\sigma_v}, \frac{p_x}{\sigma_p}, \frac{n_x}{\sigma_n} \right), \tag{10}$$

where σ_d, σ_s, σ_p and σ_n are coefficients to control the effect of each component and are specified by the user.

5. RESULTS

This section shows results produced by using our method. We used a desktop PC with an Intel(R) Core(TM) i7-2600k 3.40GHz (CPU) and a NVIDIA GeForce GTX 560 (GPU).

Fig. 2 shows examples of the edited shading effects on a sappho (left column), a figurine (middle) and an armadillo. In Fig. 2, images in upper row show shading results before editing, and images in lower row show edited results. Fig. 2(d) is an example of glossy specular effects like as metalic object. Fig. 2(e) is an example of the effects of the skylight in sunset. Fig. 2(f) is an example of the effects of soft shadows. Our system is useful for inserting synthetic 3D objects into real photographs, as shown in Fig. 3. The upper row in Fig. 3 shows rendered image by the Phong reflection model. By editing the shading of the synthetic objects (an asian doragon, a bunny and a space ship), the objects are naturally composited onto the real photographs as shown in lower images.

In general, the artificial shading effects shown in this section are often created by placing virtual light sources, but adjusting the parameters for such light sources is difficult and time-consuming. In contrast, our system allows the user to easily create the desired shading effects by directly specifying the desired color and intensity at the desired positions.

6. CONCLUSION AND FUTURE WORK

In this paper, we have proposed an interactive editing system for shading effects by feature-based interpolation. In our method, offset functions represented by RBFs are introduced. Our system allows a user to edit the shading effects for the diffuse, and the specular terms separately. By using our method, the user can intuitively design the desired shading effects.

One of our future work is taking into account of interreflection effects. We are also interested in extending our system to other advanced shading models.

ACKNOWLEDGEMENTS

This research is supported in part by Core Research for Evolutional Science and Technology (CREST) Program “Mathematics for Computer Graphics” of Japan Science and Technology Agency (JST).

REFERENCES

[1] E. Veach, and L. J. Guibas, "Metropolis Light Transport," In SIGGRAPH 97, pp.65-76, 1997.

[2] H. W. Jensen, "Global Illumination using Photon Maps, " Proceedings of the Seventh Eurographics Workshop on Rendering, pp.21-30, 1996.

[3] P. Debevec, "Rendering Synthetic Objects into Real Scenes," In SIGGRAPH 98, pp.189-198, 1998.

[4] J. F. Blinn, and M. E. Newell, "Texture and reflection in computer generated images," Communications of the ACM, vol.19, no.10, pp.542-547, 1976.

[5] M. Okabe, Y. Matshshita, L. Shen, and T. Igarashi, "Illumination Brush: Interactive Design of All-frequency Lighting," Proceedings of Pacific Graphics 2007, pp.171-180, 2007.

[6] F. Pellacini, "envyLight: An Interface for Editing Natural Illumination," ACM Transactions on Graphics (In Proceedings of SIGGRAPH 2010), vol.29, no.4, 2010.

[7] T. Ritschel, M. Okabe, T. Thormahlen, and H.P. Seidel, "Interactive Reflection Editing," ACM Transactions on Graphics (In Proceedings of SIGGRAPH Asia 2009), vol.28, no.5, 2009.

[8] H. Todo, K. Anjyo, W. Baxter, and T. Igarashi, "Locally Controllable Stylized Shading," ACM Transactions on Graphics (In Proceedings of SIGGRAPH 2007), vol.26, no.3, 2007.

[9] J. Obert, F. Pellacini, S. Pattanaik, "Visibility Editing For All-Frequency Shadow Design," Computer Graphics Forum (In Proceedings of Eurographics Symposium on Rendering 2010), vol.29, no.4, pp.1441-1449, 2010.

[10] T. Ritschel, T. Thormahlen, C. Dachsbacher, J. Kautz, and H.P. Seidel, "Interactive On-Surface Signal Deformation," ACM Transactions on Graphics (In Proceedings of SIGGRAPH 2010), vol.29, no.4, 2010.

[11] R. Shacked, and D. Lischinski, "Automatic Lighting Design using a Perceptual Quality Metric," Computer Graphics Forum (In Proceedings of Eurographics 2001), vol.20, no.3, pp.215-227, 2001.

[12] S. Gumhold, "Maximum Entropy Light Source Placement," In Proceedings of Visualization 2002, pp. 275-282, 2002.

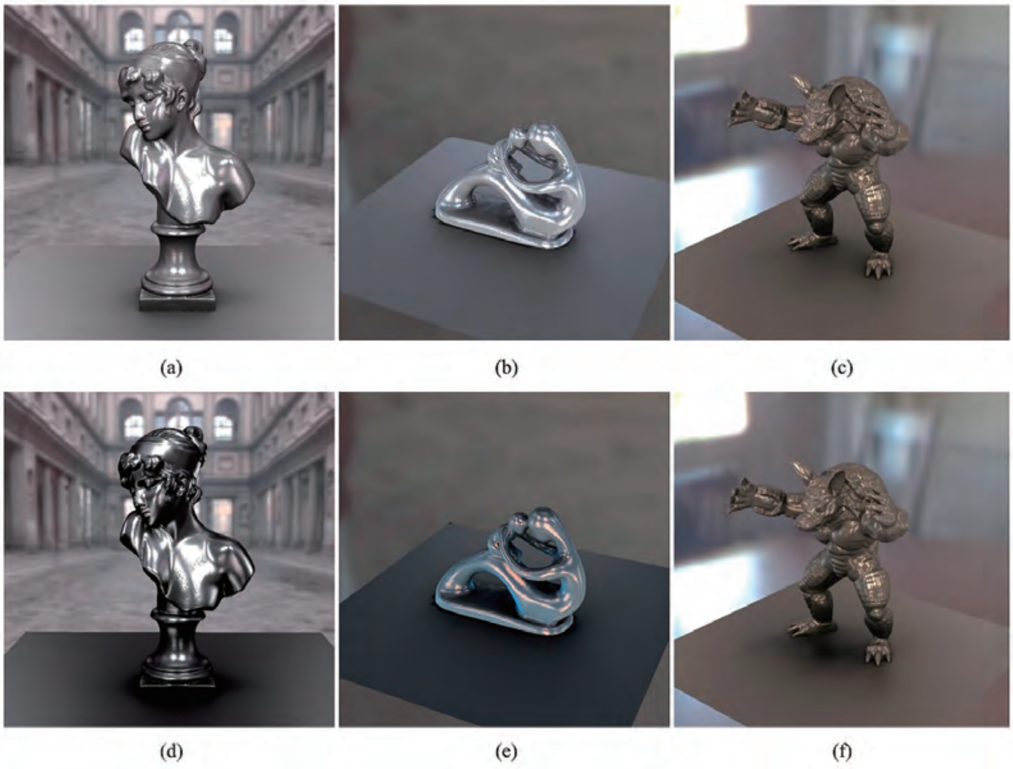

FIGURE 2. Examples of edited shading effects

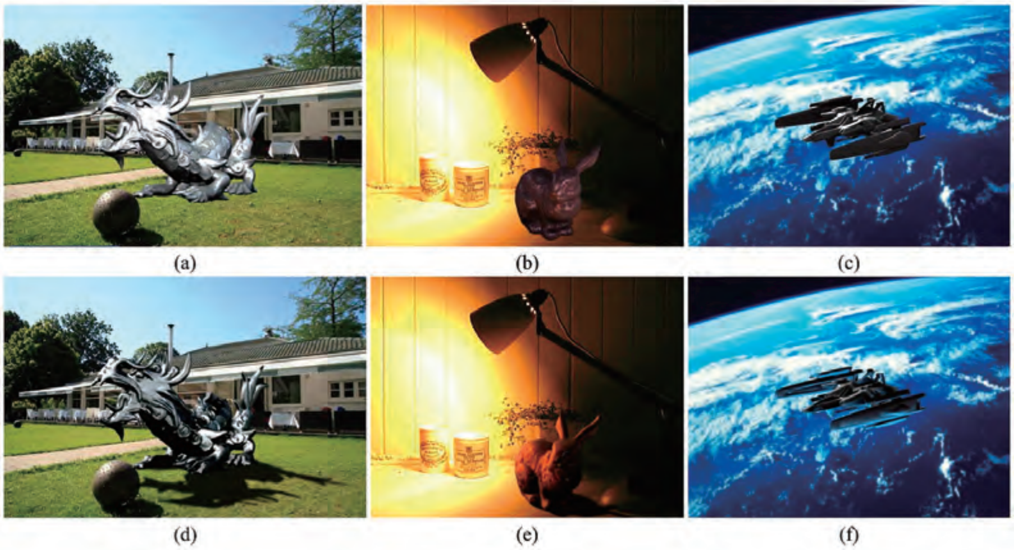

FIGURE 3. Composition of synthetic objects into real photographs