JOURNAL OF INFORMATION SCIENCE AND ENGINEERING 33, 1163-1176 (2017)
DOI: 10.6688/JISE.2017.33.5.4

Lighting and Foggy Effect Transfer

TSUNG-SHIAN HUANG, FU-HSI HUANG, WEN-CHIEH LIN+
AND JUNG-HONG CHUANG
Department of Computer Science
National Chiao Tung University
Hsinchu, 300 Taiwan
E-mail: {csie9624; oriyayo}@gmail.com; {wclin; jhchuang}@cs.nctu.edu.tw

Participating media, *e.g.*, fog and smoke, play an important role in computer animation and games. They influence the atmosphere of scenes by inducing some special lighting effects, such as the distant blur, glows, and god rays. To design the lighting effects in a scene with fog or smoke, users usually need to tune many parameters to achieve their goal. In this paper, we propose a method that aims to transfer the desired lighting and homogeneous foggy effects in target images, including glow, god rays, and fog attributes, to a 3D scene by solving an inverse problem. The experimental results show that our approach is effective, intuitive, and user-friendly.

Keywords: computer graphics, rendering, foggy effect, lighting design, inverse design

1. INTRODUCTION

Participating media exist in many natural phenomena, *e.g.*, fog, mist, and haze. Scattering in participating media produces interesting lighting effects, such as the glow lighting up the foggy street or god rays shining through trees in a forest. These effects can greatly improve the atmosphere and story-telling for computer animation and video games. To design a scene with these effects in participating media using commercial software, users usually need to tune many parameters about lights and the participating media by trial and error. It is often a complicated process to adjust the parameters and obtain desired results.

We propose an approach that is able to transfer the desired foggy and lighting effects in a target image to a 3D scene by solving an inverse problem, aiming to find a set of parameters with which the scene would have similar foggy and lighting effects as the target image. On the target image and the rendered scene image, users draw some control patterns to guide the sampling. The inverse problem then tries to minimize the error between the target image and rendered scene image on samples drawn from the control patterns. As a result, a set of optimal parameters about lighting and fog is obtained.

The contributions of this paper can be summarized as follows. We propose an approach that can transfer light glows, god rays, and the homogeneous foggy effects in a target image to a 3D scene by solving an inverse problem involving the lighting and fog parameters. Our approach can help both novices and artists. For novices, our system does not require the understanding of physical parameters, such as extinction coefficient and albedo. For artists, they can rely on the optimization system to achieve the desired effects automatically instead of tediously tuning parameters manually. Also, they can specify the

Received March 25, 2016; revised September 19, 2016; accepted December 10, 2016.
Communicated by Yung-Yu Chuang.
+ Corresponding author.

desired effects by providing exemplary photos. Therefore, the required user intervention is intuitive and minimized.

2. RELATED WORK

Lighting Design Schoeneman *et al.* [15] first described a method to set lighting configurations with known light positions by solving an inverse problem. The user paints on a 3D scene, and the intensities of lights can be derived automatically. Marks *et al.* [9] proposed the Design Galleries system that provides an intuitive user interface. The user chooses desired galleries rendered by configurations suggested by the system as targets, and then the optimal result similar to those targets is computed.

The method proposed by Okabe *et al.* [12] generates the environment light automatically by solving an inverse shading problem to meet the painted illumination effects specified by the user. It provides an interactive lighting design system due to the use of precomputed radiance transfer. Although it can handle environment lighting design well, point lights are not supported. In the method presented by Pellacini *et al.* [13], the user paints color, light shape, shadows, highlights and reflections on the 2D image, and the optimal parameters of lights for meeting the desired illumination effects can be found. However, the required painting effort is still inconvenient to the user who is unfamiliar with lighting design.

Lin *et al.* [8] proposed a guiding system to specify lighting and shadows. In the first step, the positions and number of lights are determined based on shadow strokes drawn by the user. Some initial lighting parameters are precomputed, and an appropriate lighting configuration is obtained using the least-squares method. Then they use Nelder-Mead simplex to find better light positions with a light tree. Other details including color, intensity, and shadows can be adjusted based on the result of the first step. In addition to directly paint desired illumination, the user can assign input by modifying the rendering result. This makes the system more stable and user-friendly.

Participating Media Design Zhou *et al.* [18] proposed an interactive design system for smoke or fog by incorporating their analytic method for handling isotropic, single-scattering media illuminated by point lights. The user can model the distribution of media by directly painting strokes or placing the particle emitters in the 3D scene to simulate the foggy effect.

Dobashi *et al.* [3] proposed a method to synthesize clouds by analyzing the density distribution of clouds from an input photograph. They further proposed another method that finds parameters for rendering clouds using a reference photograph [2]. Giving cloud density data and a photograph of real clouds, they apply the genetic algorithm to compute the rendering parameters by minimizing the difference between the color histogram of the synthesized image and the photograph. The final image rendered by the optimal parameters is visually similar to the input photograph. Their problem is related to ours, but it aims for rendering clouds. The rendering of cloud scenes and fog scenes are quite different though. The background of clouds is usually only the sky, which can be regarded as existing in an infinite distance; however, the background of fog is diversified and complicated, *e.g.*, trees, roads, buildings, or terrain.

Nowrouzezahrai *et al*. [11] presented a method for generating artistic volumetric effect by decoupling the modeling and shading. Artists first sketch the shape for volumetric effects and specify the desired color of smoke, fire, or beam. Their system then solves the parameters. Klehm *et al.* [6] proposed a method for stylizing volumetric data. The user needs to paint on the images from different views of the volumetric data. Emission, albedo, and extinction coefficients of the volumetric data are derived to make the rendered image close to those images drawn by the user. They reformulate the rendering equation such that the emission, albedo, and extinction coefficients can be solved from a linear system. However, painting on multiple views requires the user to be familiar with image editing software. Since the fitting targets are created by user rather than transferred from photo, the result sometimes becomes too artificial and unnatural.

Fog Removal In image processing, several approaches [4, 5, 16] were proposed to remove or enhance the fog in a single image. These approaches focused on editing the fog effects in 2D images, while our work aims to transfer the fog in a target image into a 3D scene.

3. SYSTEM OVERVIEW

We aim to simplify the process of designing lighting and foggy effects in 3D scenes. In the traditional design methods, parameters of lighting and participating media are adjusted manually on a trial-and-error basis, which could be very tedious and time-consuming for artists. It may be even more difficult for novices as they may not have any prior knowledge or clues on tuning the parameters to achieve the desired effects. In the backward or inverse methods, the desired effects or some references are provided, and the solution is found through an optimization process which minimizes the difference between the desired effects and the synthesized result.

In this paper, we propose a backward approach which allows users to utilize some lighting and foggy effects in a target image as the reference. The input of our system is a 3D scene with one or several target images. Some control patterns for sampling are placed by users on the target image I_t and the rendered image I_s with the default lighting and fog parameters X_0. Then the optimization compares the samples on both sides, and the rendering parameters X_i are updated automatically until the appropriate parameters X^* are found. The 3D scene rendered with the optimal lighting and fog parameters X^* is the final output of our system. Fig. 1 shows the flowchart of our approach consisting of the following components.

User Input To obtain the desired effects, the user can input one or few real-world photos that contain the desired lighting and foggy effects as the target images. The desired effects do not always suffuse the entire image. For instance, light glows often just appear in a local region of the image. Instead of sampling the whole image, we provide pattern tools for users to sample and map the desired effects in the target image to the 3D scene.

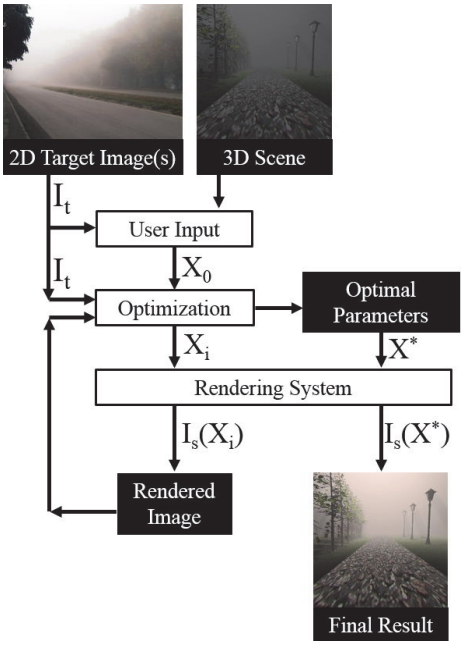


Fig. 1. Flowchart of our method.

Optimization Our goal is to transfer the desired effects in the target image to the 3D scene. Thus the problem is to find a set of parameters X^* with which the rendered image is as similar as possible to the target image on the selected effects. We formulate this as an optimization problem and solve it using Nelder-Mead simplex algorithm [10].

Rendering System We use ray marching to render scenes with the participating media. As the optimization is an iterative process and the 3D scene is rendered for each iteration, the high rendering cost is a major bottleneck of the optimization process. To ensure interactive performance, we use GPU to accelerate rendering.

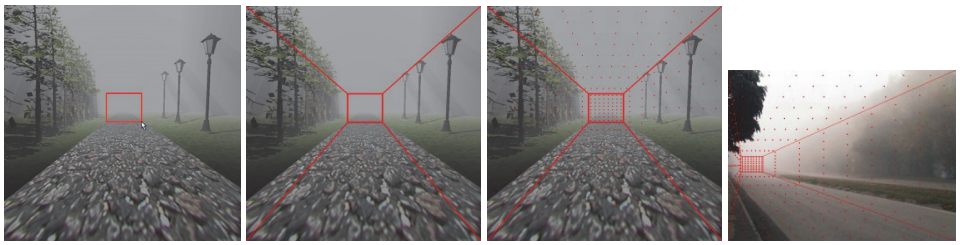
4. OUR METHOD

4.1 User Input

Our system provides two types of patterns for users to control how the desired effects are sampled. These control patterns are placed in pair on both of the target image and the rendered image of the input scene.

In general, the shape of control patterns should match the variation of fog in the target image. The vanishing pattern is designed to capture the variation of fog opacity associated with the viewing distance. Our original idea is to use a single line as a pattern to sample the image. However, the line sampling is too coarse, and the estimated fog variation from the target image can be greatly influenced by the background color. Therefore, for better stability, the number of samples needs to be increased. Instead of

placing multiple lines from near to far, the vanishing pattern is used to represent the desired depth cue and provide a structure for generating samples in correspondence on the rendered scene image and target image. Figs. 2 and 6 shows the vanishing pattern and its usage. The user drags rectangles in the rendered scene and the target image. The corners of the dragged rectangle are connected to the corners of the image, partitioning the image into five pieces. Each pair of corresponding pieces on the rendered scene image and target image are then sampled, producing samples in one-to-one correspondence. A better approach is to extract the depth structure of the target image, but the vanishing pattern seems to work well in our experiments.



(a) (b) (c) (d)

Fig. 2. The vanishing pattern (a) A user drags a rectangle to indicate the vanishing point in the rendered scene; (b) The created vanishing pattern; (c) The system generates the samples corresponding to the pattern; (d) The vanishing pattern on target image.

The glow pattern is designed to capture the glow effect, in which the light intensity decays from the center to the outside. The descriptive parameters of the glow effect include glows range, color, and the intensity gradient. Thus, we use multiple concentric circles to sample the glow from inside to outside. Fig. 3 depicts how the glow patterns are placed in the scene and target image. In Fig. 3 (a), a start point as the center of a circle and an end point representing the expected scope of the glow effect are placed first. Then sample points which distributed in a way of concentric circles are generated (Fig. 3 (b)). Similar glow patterns are placed on the target image (Fig. 3 (c)).

(a) (b) (c)

Fig. 3. The glow pattern; (a) User drags and places the glow pattern; (b) and (c) are examples of a pair of patterns in the form of concentric circles for the scene image and target image, respectively.

4.2 Optimization

A set of optimized lighting and fog parameters is computed in the optimization process. Lighting parameters include the *L*a*b* color of each light source with known position. Fog parameters include extinction σ_t and albedo α for global homogeneous media. We assume that the color of fog is gray, so the albedo is a scaling value rather than a RGB color vector. If the number of lights is L_{num}, then the total dimension of X is (L_{num}×3+2).

Problem Formulation We compute the optimal solution X^* by minimizing the objective function O defined as follows:

$$\arg\min_X O(I_t, I_s(X)), \tag{1}$$

where X is a vector consisting of light and fog parameters; I_t is the target image(s) for the desired effects and $I_s(X)$ is the image of the scene rendered using parameter X. The objective function O includes E_v for the vanishing pattern and E_g for the glow pattern defined as follows:

$$O(I_t, I_s(X)) = \beta_v E_v(I_t, I_s(X)) + \beta_g E_g(I_t, I_s(X)), \tag{2}$$

where β_v, and β_g are scalar weights. In our experiments, we set $\beta_v = 1$, $\beta_g = 1$ for all test scenes. Larger β_v causes the optimization to focus more on global brightness and the environmental atmosphere. Larger β_g can preserve the contrast of glow effect better.

The vanishing pattern transfers the overall atmosphere. Thus, we define E_v as

$$E_v(I_t, I_s(X)) = \frac{1}{N_v}\sum_{i=1}^{N_v}[\| L(I_t^i) - L(I_s^i) \|^2 + \| Lab(A_t^i) - Lab(A_s^i) \|^2]. \tag{3}$$

N_v is the number of sample points. *Lab*() is the function that transfers the RGB color to the CIELAB color. We use *L*a*b** color space as it matches human perception well. *L*() it the function to extract the brightness value in the *L*a*b** color space. I_t^i and I_s^i is the color of the *i*th sample point in a target image and a rendered scene, respectively. A_t^i and A_s^i are airlight color of the *i*th sample point in a target image and rendered scene, respectively. I_t^i, I_s^i, A_t^i and A_s^i are all in the RGB color space. The airlight of the target image is obtained by the image dehazing method [5], while the airlight of rendered image can be computed by rendering system.

The difference between $L(I_t^i)$ and $L(I_s^i)$ can be utilized to control the overall brightness to avoid the scene getting too dark or too bright. The difference between $Lab(A_t^i)$ and $Lab(A_s^i)$ captures the foggy atmosphere which varies from far to near. We do not directly minimize the difference between $Lab(I_t^i)$ and $Lab(I_s^i)$ because I_t^i and I_s^i are influenced by the color of the background. We only want the foggy atmosphere, so we minimize the difference of the airlight. If the users want to transfer the fog opacity without the color of the atmosphere, our system provides them an option to ignore the difference of hue. In this situation, $Lab(A_t^i)$ and $Lab(A_s^i)$ are replaced with $L(A_t^i)$ and $L(A_s^i)$.

The glow effect mainly depends on the color and intensity of light. We define E_g as the appearance difference between samples in the rendered scene and the target image:

$$E_g(I_t, I_s(X)) = \frac{1}{N_g}\sum_{i=1}^{N_g} \| Lab(I_t^i) - Lab(I_s^i) \|^2 \quad (4)$$

where N_g is the number of sample points.

Compute Optimal Configuration We solve the light and fog parameters simultaneously. Since the objective function is nonlinear, we use the Nelder-Mead simplex algorithm [10] to minimize Eq. (1). For minimizing the function with d dimension input, Nelder-Mead simplex algorithm requires (d+1) configurations to form a simplex in high dimension space. One of the initial configurations is X_0, where the lights and the fog parameters are given by users. The other d configurations can be created by slightly disturbing an element of X_0. Then the optimization repeatedly replaces the worst configuration with a better one until the simplex is small enough. The final configuration is the optimal parameters X^*.

4.3 Rendering System

We render the participating media using the ray marching method. As scene rendering is required at each iteration of the optimization process, the rendering speed should be fast enough. Thus, we consider only single scattering and use the GPU to accelerate the ray marching process. Note that our framework does not restrict to any rendering technique. The design of our approach is just a trade-off between speed and quality. The considered light transport is shown as Fig. 4.

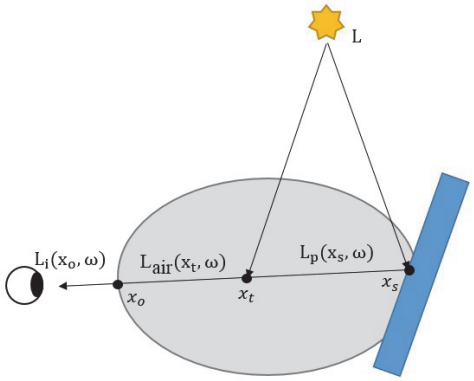


Fig. 4. Light transport goes into the viewer's eyes along the direction ω. It includes the radiance scattered by air and the radiance reflected by the surfaces of objects.

In Fig. 4, the light $L_i(x_o, \omega)$ passing through x_o along the direction ω can be represented as a sum of two terms:

$$L_i(x_o, \omega) = T_r(x_o, x_s)L_p(x_s, \omega) + \int_0^{\|x_o - x_s\|} T_r(x_o, x_o - u\omega)L_{air}(x_o - u\omega, \omega)du. \quad (5)$$

The first term represents how the light $L_p(x_s, \omega)$, which is emitted from a light source and reflects off an object at x_s, is attenuated by participating media between x_s and x_o. The second term represents how the light $L_{air}(x_t, \omega)$, which comes from the light source

and scatters at x_t, is attenuated by participating media between x_t and x_o. When light penetrates the media from x_s and x_o, its intensity is reduced due to the absorption and out-scattering. The transmittance $T_r(x_o, x_s)$ is the ratio of the remaining light leaving the media between x_o and x_s:

$$T_r(x_o, x_s) = e^{-\int_0^{\|x_o - x_s\|} \sigma_t(x_o - u\omega)du} \tag{6}$$

where $\sigma_t(x)$ is the extinction coefficient composed of absorption coefficient $\sigma_a(x)$ and scattering coefficient $\sigma_s(x)$. The radiance $L_{air}(x_t, \omega)$ at point x_t in direction ω comes from the scattering:

$$L_{air}(x_t, \omega) = \sigma_s(x_t) \int_\Omega f(x_t, \omega, \omega_\ell) L_i(x_t, \omega_\ell) d\omega_\ell \tag{7}$$

where $L_{air}(x_t, \omega_\ell)$ is incoming radiance at xt in direction ω_ℓ. The outgoing radiance $L_i(x_t, \omega_t)$ in direction ω is determined by the phase function $f(x_t, \omega, \omega_t)$, which defines how much light in direction ω_ℓ will scatter to the direction ω after penetrating x_t, and $\sigma_s(x_t)$ is the scattering coefficient for controlling the strength of scattering effect at x_t. $\sigma_s(x_t)$ can be computed by $\sigma_t(x_t) \times \alpha$, where α is the albedo of homogeneous fog in the scene. In our implementation, we assume that the phase function is isotropic and the fog parameters are homogeneous.

5. EXPERIMENTAL RESULTS

We first show that our system is robust and insensitive to the initial parameters. In Fig. 5, four different sets of initial parameters are used for the optimization. Then the optimization tries to fit the artificial target image generated by the render system with given parameters. Although the optimized parameters are different for the same target image since there are multiple solutions for achieving the result, the images of fitting results are very similar to the target image. The PSNR between fitting results and the target image are all greater than 60 dB, which implies that human eyes can barely tell the difference perceptually.

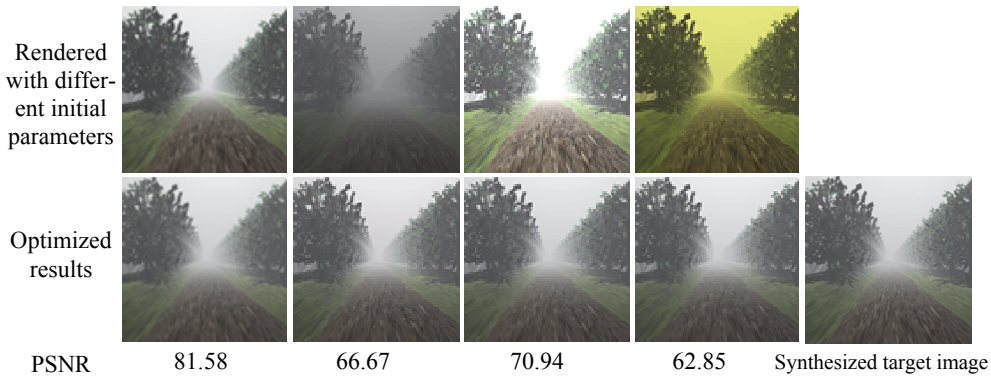
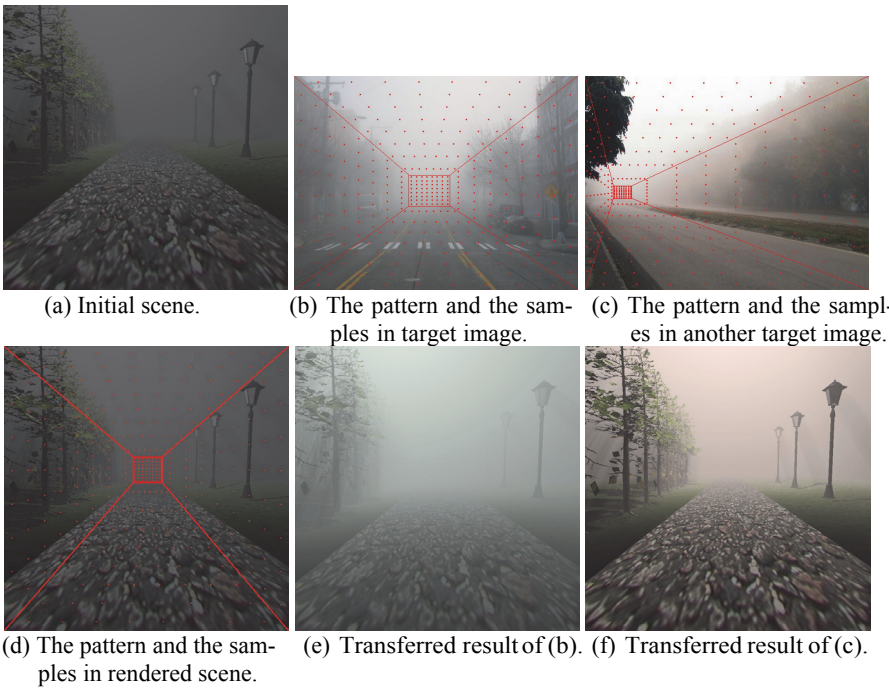


Fig. 5. Optimization with different sets of initial parameters.

Next, we demonstrate our system by testing with different scenes and target photos. Fig. 6 shows an example that transfers the foggy effect using the vanishing pattern. Fig. 6 (a) is rendered with the initial parameters for a 3D scene. Figs. 6 (d) and (b) show the patterns drawn by the user in the rendered scene and target image, respectively. Fig. 6 (e) is the optimized result generated by our system. Comparing with the initial parameter, the entire scene becomes brighter and the region near the vanishing point becomes more foggy. Fig. 6 (c) is another target image. Given Figs. 6 (d) and (c), our system generates the optimal result shown in Fig. 6 (f). The atmosphere of Fig. 6 (f) is very similar to that of the Fig. 6 (c).



(a) Initial scene. (b) The pattern and the samples in target image. (c) The pattern and the samples in another target image.

(d) The pattern and the samples in rendered scene. (e) Transferred result of (b). (f) Transferred result of (c).

Fig. 6. Transfer the foggy effect by vanishing pattern.

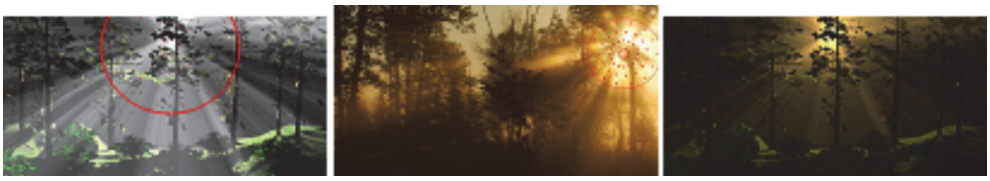
(a) Initial scene. (b) First target image. (c) Second target image.

Fig. 7. Multiple Target Images; (a) The initial parameter of rendered scene; (b) The first target image contains one sample pattern which is mapping to the red pattern in (a); (c) The second target image contains two sample patterns which are mapping to the green and blue patterns in (a).

(d) Transferred result (e) Another viewpoint of (d). (f) Other viewpoint of (d).

Fig. 7. Multiple Target Images; (d) The transferred result of (a); (e)-(f) Another viewpoints of (d).

Our system can also support multiple target images. In Fig. 7, we aim to transfer the glow with different colors in target images to street lamps in a 3D scene. (Fig. 7 (a) shows the initial rendered scene. There are two target images in this example: Figs. 7 (b) and (c).) Note that the glow patterns in Figs. 7 (b) and (c) are mapped to those in Fig. 7 (a) with the same corresponding color. Fig. 7 (d) demonstrates the transferred result. The red glow pattern in the 3D scene is larger than the other two patterns. Thus, the extent of glow effect for left light is also larger than that of the other two. Figs. 7 (e) and (f) show the results viewed at different positions. We can see that the glow and foggy effects are still visually pleasant and consistent for the view change. Since the vanishing pattern is not used in this example, this result only represents the transfer of glow effects, and hence the ground is a little over bright. The usage of glow pattern can also be utilized for other similar effects. Fig. 8 is an example that utilizes the glow pattern for the transfer of god rays effect.



(a) The pattern and the samples in rendered scene. (b) The pattern and the samples in target image. (c) Optimized result.

Fig. 8. Transfer the god rays effect by glow pattern.

(a) Initial scene. (b) Target image. (c) Optimized result.

Fig. 9. The vanishing and glow patterns can be used at the same time; (a) The initial configuration and the samples in the rendered scene; (b) The patterns and samples in the target image. The corresponding patterns in the scene and target images are shown in the same color; (c) The optimized result.

The vanishing and glow patterns are generally used at the same time for the scenes with lights. Fig. 9 is a bridge scene with street lights. As the vanishing pattern tends to spread the samples globally, we used it to transfer the overall atmosphere. On the other hand, the glow pattern tends to place the samples locally, it is suitable to anchor the light source. Both patterns are utilized and the glow and foggy effects in the target image are transferred to the 3D scene.

Computational performance Our system is implemented on a PC equipped with Intel Xeon E3-1230 and NVIDIA GTX 670. The major computational cost is the rendering, whose performance depends on the total number of lights and metaballs and the complexity of the input scene. Each ray in ray marching is divided into 160 sample points, and the screen size is 768×768. Optimizing the example in Figs. 6-9 takes 5.68, 15.1, 3.48, and 7.9 seconds, respectively.

6. DISCUSSION

Although our system works well in all the tested examples, there are still some issues that need to be addressed. For the objective function, the E_v term in Equation 3 is mainly designed for handling the global atmosphere. Other error metrics can also be considered. In fact, we tried some other error metrics, such as transmittance and saturation, but they do not work well. The transmittance error failed due to two reasons. First, the extracted transmittance by image de-hazing [5] is mainly affected by depth, but the depth distribution in the target image may be quite different to that in the 3D scene. Besides, the transmittance can only preserve the order of the depth reliably, but the entire appearance of transferred fog may be too dense or too thin. We think a depth mapping between the scene and the target image may be needed if transmittance is adopted as the error metric. As for the saturation, it mainly depends on the final rendered appearance which is easily influenced by material color of the scene. Moreover, it suffers from low-saturation target images. The worst case is a gray scale target image.

For the optimization, the convergence and the initial parameter sensitivity are also important to our system. In fact, there are multiple parameters which can achieve visually similar results. This characteristic is also explored in Zhao *et al.*'s study [17]. The objective function definitely has multiple peaks and valleys. Since the Nelder-Mead simplex algorithm tends to converge to a local minimum, we also try the simulated annealing to explore the search space more completely. Its result is similar to that of Nelder-Mead, but the computation time of simulated annealing method is much longer than that of Nelder-Mead simplex algorithm. Note that the light source should be parameterized in $L^*a^*b^*$ color space rather than RGB color since the error metric is based on $L^*a^*b^*$ color space; Otherwise, it creates unnecessary local minimums.

Because our system uses the image de-hazing to extract the airlight of the target image, the quality of extracted airlight also affects the quality of the optimized result. In some cases, the hue of extracted airlight is wrong, and the optimization fails to converge to a desired result. If users can provide some hints to the image dehazing process, this problem can be ameliorated.

7. CONCLUSION AND FUTURE WORK

We propose an approach for solving the inverse problem involving both lighting and fog parameters in this paper. Our approach can handle the glow effect of light and foggy effects from homogeneous fog. The optimal parameters of lights and fog are obtained simultaneously by the Nelder-Mead simplex algorithm. Based on the proposed approach, we further develop a user-friendly system that allows users to intuitively transfer the lighting and foggy effects in real-world photos to 3D scenes. Compared to the traditional lighting design approach, which usually requires intensive user interventions on a trial and error basis to achieve desired effects, our approach provides a much easier way to design the lighting and foggy effects in 3D scenes.

In the future, we want to improve our rendering system so that more effects can be rendered, *e.g.*, multiple scattering and environment lighting. We would also like to add other functions for transferring inhomogeneous media with details. We believe our approach may stimulate many related research and potential applications in the lighting design field. In particular, many lighting effects indeed involve the interaction between light objects, and environment. Thus, it is reasonable to solve the lighting design problem by taking into account these three factors together. From this viewpoint, we consider the main contribution of this work is suggesting a novel direction for solving the lighting design problem.

REFERENCES

1. E. Cerezo, F. P´erez, X. Pueyo, F. J. Seron, and F. X. Sillion, "A survey on participating media rendering techniques." *The Visual Computer*, Vol. 21, 2005, pp. 303-328.
2. Y. Dobashi, W. Iwasaki, A. Ono, T. Yamamoto, Y. Yue, and T. Nishita, "An inverse problem approach for automatically adjusting the parameters for rendering clouds using photographs," *ACM Transactions on Graphics*, Vol. 31, 2012, Article No. 145.
3. Y. Dobashi, Y. Shinzo, and T. Yamamoto, "Modeling of clouds from a single photograph," in *Computer Graphics Forum*, Vol. 29, 2010, pp. 2083-209.
4. R. Fattal, "Single image dehazing," *ACM Transactions on Graphics*, Vol. 27, 2008, Article No. 72.
5. K. He, J. Sun, and X. Tang, "Single image haze removal using dark channel prior," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 33, 2011, pp. 2341-2353.
6. O. Klehm, I. Ihrke, H.-P. Seidel, and E. Eisemann, "Volume stylizer: tomography-based volume painting," in *Proceedings of ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, 2013, pp. 161-168.
7. J. Krüger and R. Westermann. "Acceleration techniques for gpu-based volume rendering," in *Proceedings of the 14th IEEE Visualization*, 2003, p. 38.
8. W.-C. Lin, T.-S. Huang, T.-C. Ho, Y.-T. Chen, and J.-H. Chuang, "Interactive lighting design with hierarchical light representation," *Computer Graphics Forum*, Vol. 32, 2013, pp. 133-142.

9. J. Marks, B. Andalman, P. A. Beardsley, W. Freeman, S. Gibson, J. Hodgins, T. Kang, B. Mirtich, H. Pfister, W. Ruml, *et al.*, "Design galleries: A general approach to setting parameters for computer graphics and animation," in *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques*, 1997, pp. 389-400.
10. J. A. Nelder and R. Mead, "A simplex method for function minimization," *The Computer Journal*, Vol. 7, 1965, pp. 308-313.
11. D. Nowrouzezahrai, J. Johnson, A. Selle, D. Lacewell, M. Kaschalk, and W. Jarosz, "A programmable system for artistic volumetric lighting," *ACM Transactions on Graphics*, Vol. 30, 2011, Article No. 29.
12. M. Okabe, Y. Matsushita, L. Shen, and T. Igarashi, "Illumination brush: Interactive design of all-frequency lighting," in *Proceedings of the 15th IEEE Pacific Conference on Computer Graphics and Applications*, 2007, pp. 171-180.
13. F. Pellacini, F. Battaglia, R. K. Morley, and A. Finkelstein, "Lighting with paint," *ACM Transactions on Graphics*, Vol. 26, 2007, Article No. 9.
14. E. Reinhard, M. Ashikhmin, B. Gooch, and P. Shirley, "Color transfer between images," *IEEE Computer Graphics and Applications*, Vol. 21, 2001, pp. 34-41.
15. C. Schoeneman, J. Dorsey, B. Smits, J. Arvo, and D. Greenberg, "Painting with light," in *Proceedings of the 20th Annual Conference on Computer Graphics and Interactive Techniques*, 1993, pp. 143-146.
16. R. T. Tan, "Visibility in bad weather from a single image," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2008, pp. 1-8.
17. S. Zhao, R. Ramamoorthi, and K. Bala, "High-order similarity relations in radiative transfer," *ACM Transactions on Graphics*, Vol. 33, 2014, Article No. 104.
18. K. Zhou, Q. Hou, M. Gong, J. Snyder, B. Guo, and H.-Y. Shum, "Fogshop: Real-time design and rendering of inhomogeneous, single-scattering media," in *Proceedings of the 15th Pacific Conference on Computer Graphics and Applications*, 2007, pp. 116-125.

Tsung-Shian Huang (黃聰賢) received his BS degree from the Department of Computer Science, National Chiao Tung University, Taiwan in 2007. Currently, he is a Ph.D. student of National Chiao Tung University. He is interested in computer graphics and software engineering.

Fu-Hsi Huang (黃富熙) received her BS and MS degrees from National Chiao Tung University, Taiwan, in 2011 and 2014. Currently, she is an Engineer of MediaTek.

Wen-Chieh Lin (林文杰) is an Associate Professor at the Department of Computer Science, National Chiao Tung University, Taiwan. He received the BS and MS degrees in Control Engineering from National Chiao Tung University, Taiwan, in 1994 and 1996, respectively, and the Ph.D. degree in Robotics from the School of Computer Science at Carnegie Mellon University, Pittsburgh, in 2005. His research interests span several areas of computer graphics, human computer interaction, and visualization.

Jung-Hong Chuang (莊榮宏) is a Professor of Computer Science and Information Engineering Department at National Chiao Tung University, Taiwan. His research interests include 3D computer graphics, virtual reality, geometric modeling, and visualization. Chuang received his BS degree in applied mathematics from National Chiao Tung University, Taiwan, in 1978, and MS and Ph.D. degrees in Computer Science from Purdue University in 1987 and 1990, respectively.