

# 人工光を考慮した夜景画像のインタラクティブな生成手法

## Interactive Nighttime Image Synthesis With Artificial Light Sources

山口 祐一郎†, パウロ シルバ†, 陳 炳宇§, 坂東 洋介†‡, 西田 友是†  
Yuichiro YAMAGUCHI†, Paulo SILVA†, Bing-Yu CHEN§, Yosuke BANDO†‡, and  
Tomoyuki NISHITA†

† 東京大学 † The University of Tokyo  
‡ 東芝 ‡ Toshiba Corporation  
§ 国立台湾大学 § National Taiwan University

E-mail: †{yauchi,paulo,ybando,nis}@is.s.u-tokyo.ac.jp, §robin@ntu.edu.tw

## 1 Introduction

Methods to convert daytime images into nighttime images are desirable for a number of reasons. For example, access restrictions, schedule, camera lacking nighttime shooting capabilities. Hence, some methods are developed for automatic manipulation of contrast in order to try to generate natural results. However, these methods only darken the image as a whole, and thus some effects such as artificial lights (street lamps, headlights, etc.) are not visible. Therefore, in this paper we propose a method that adds artificial lights (i.e., human-created landscapes) with a minimal amount of user interaction. In the first step, we remove shadow areas from the input image. Next we leverage previous research in order to globally darken the image using a model of the human eye. Then, we extract window areas from buildings in the image using a stroke-based user interface to help in the segmentation. We blur these regions and change the colors in them to mimic the interior lights that one would expect to see in a usual nighttime photograph. Finally, in our system the user can add other artificial light sources, which are computed using a physically based calculation.

## 2 Related Work

Relevant work based on visual adaptation and tone reproduction was presented by Ferwarda *et al.* [1], which proposed a model that simulates the human visual adaptation and aims at obtaining a realistic image synthesis. Pattanaik *et al.* [2], proposed a multi-scale model, which also reproduces the human visual adaptation and spatial vision. Larson *et al.* [3], proposed a system which uses a histogram adjustment technique and considers human contrast sensitivity, glare, spatial acuity, and color sensitivity. Durand and Dorsey [4], proposed a model that simulates visual adaptation for a three dimensional input. Photographic tone reproduction for digital images by Reinhard *et al.* [5], presents an approach based on tone reproduction for two dimensional images. Tompson *et al.* [6], introduced two important effects for nighttime image synthesis: visible noise and the loss of acuity. Another approach for synthesizing a natural looking night scene was proposed by Chen *et al.* [7], which introduced a predictable scene algorithm. Haro *et al.* [8], introduced a method that simulates not only color transition but also contrast modification, loss of acuity, etc. Although this method considers several aspects of the human visual sys-

tem in the process of image synthesis. However, none of these methods considers artificial light effects.

High luminosity in dark scenes originates glare effects in human eyes and photographs. Work that models these effects was presented by Rokita [9]. This author proposed a model for high intensity light, which is based on the human eye structure and human physiology. Spencer *et al.* [10], proposed a simple and easy-to-calculate glare model. Since this method is used in our system, we describe it in more detail in Sec. 3.4. Kakimoto *et al.* [11], introduced a model based on wave optics. This model considers Fraunhofer diffraction in wave optics, which makes it flexibility to design various types of glare effects.

However, to the best of our knowledge no previous research on nighttime image synthesis considers artificial lights or the glare effects resulting from those lights.

## 3 Nighttime Image Synthesis

### 3.1 Overview

Our method can be divided into four phases to create a night scene. In the first phase, the system erases the shadows in the input image. Without this process, synthesized images would have shadows, which should not be visible at night. In the second phase, the user indicates the value of desired luminance for the scene. Then, the system synthesizes a new image from the shadow-removed image, such that the average luminance of all pixels becomes the user-defined value. In the third phase, our system segments the windows in the original input image through a few user strokes. In the final phase, the system adds artificial lights in the identified regions. Next, we describe each of these steps in detail.

### 3.2 Shadow Removal

In a daytime image, there may be some shadowed regions due to the sunlight. If we synthesize a nighttime image using previous methods, the color difference between shadow and non-shadow areas on the walls or ground may remain to some degree. That is because the previous methods ignore the shadow regions and change the colors of each pixel independently for the whole image. However, where there was shadow in the photo taken during daytime, should be equal to the color of the rest of the surface. In our method, we avoid these unnatural shadows by removing them from the input image using a minimal

amount of user interaction. Of course, if there are no shadows in the input image, the user can skip this phase.

In this step, the user just needs to make a pair of strokes. One in the shadow and the other in the sunlit region as shown in Fig. 2 (a). Then, our method segments this region using the Graph Cut algorithm [12]. Next, our method calculates the Gaussian Mixture Model (GMM) that represents the color distribution of the shadow region segmented in the previous step. The Gaussian Mixture Model expresses the distribution of the data by the summation of several Gaussian distributions. We then estimate the average  $RGB$  color of the main Gaussian distribution, which has the maximum weight in all distributions. We call this  $\overline{RGB}_s$ . Finally, we sample pixels from the user-specified stroke in the sunlit region. Here, we call the average color of these sample pixels  $\overline{RGB}_n$ . We then roughly estimate the shadow effects as the difference of these two vectors  $\overline{RGB}_s$  and  $\overline{RGB}_n$ . After this sampling, the system changes the color of each pixel in the segmented shadow region using the following equation:

$$RGB_n = RGB_s + \overline{RGB}_n - \overline{RGB}_s, \quad (1)$$

where,  $RGB_n$  represents the computed color that is used in the shadow area, and  $RGB_s$  represents the original pixel color.

### 3.3 Interactive Image Darkening

The previous method presented in [8] takes several seconds to synthesize a nighttime image. Therefore, it cannot be regarded as an interactive process. In this step, we use a simplified version of [8] in order to decrease its processing time aiming at interactive rates. The original method recursively repeats a relatively expensive partial differential equation computation. We replace the recursive PDE solving step in [8] with a Gaussian filter. Though Gaussian blurring does not model the physical and physiological phenomena perfectly, the difference of these two methods is negligible for dark images. The parameter of the Gaussian filter  $\sigma$  is determined from the following equation according to the average luminance  $L_a\sigma = 8 - 8L_a/255$ , where we estimated the max value of luminance as 255. By these simplifications, our method can synthesize nighttime images in real-time.

In Fig. 1 (a), we show the data of a transition of luminance from a video taken by a web camera. The horizontal axis is time of the day, and the vertical axis is the average luminance of the captured image (from 0 to 255). Using this data, user can roughly indicate the average luminance for the desired time, which the user wishes to synthesize. The user can also input a desired time, and the system computes the corresponding luminance using this data.

The average luminance of the input image is an approximation of the actual luminance of the real scene, because we do not consider the differences originated by camera settings such as diaphragm, exposure time, etc. However, we assume that these differences are small such that the resulting image still looks natural.

### 3.4 Artificial Light Addition

To calculate the light effects, we divide them into surface light sources (windows, neon signs, etc.) and point light sources (traffic lights, headlights, etc.).

In the case of surface lights, the glare effect may surround the surface. On the other hand, point lights produce a circular glare effect around them.

In the case of surface light sources (mostly window lights), our method first segments these regions. In our work, we use the Graph Cut algorithm [12] with a stroke-based user interface for the segmentation process. We start by presenting some functions designed to model the glare effect used in this step. These glare effects result from the scattering and diffraction of the light in the cornea, lens, and retina of the human eye. Here we experimented several point spread functions in order to produce a natural and physically-based glare effect.

The glare effect is regarded as the point spread function, with a parameter  $\theta$  representing the degree of spreading. In the previous research by Spencer *et al.* [10], the glare function is separately represented by the following point spread functions  $P_p(\theta)$  (photopic or day vision),  $P_s(\theta)$  (scotopic or night vision), and  $P_m(\theta)$  (mesopic or mixed day and night). The corresponding luminance thresholds are  $< 0.01cd/m^2$  for night vision,  $0.01 < 3cd/m^2$  for mixed vision, and  $\geq 3cd/m^2$  for day vision [13]. For brevity's sake, we refer the reader to [10] for these equations definition. These functions are defined in Eq. 2.

$$\begin{aligned} P_p(\theta) &= 0.384f_0(\theta) + 0.478f_1(\theta) + 0.138f_2(\theta), \\ P_s(\theta) &= 0.282f_0(\theta) + 0.478f_1(\theta) \\ &\quad + 0.207f_2(\theta) + 0.033f_3(\theta, \lambda), \\ P_m(\theta) &= 0.368f_0(\theta) + 0.478f_1(\theta) \\ &\quad + 0.138f_2(\theta) + 0.016f_3(\theta, \lambda), \end{aligned} \quad (2)$$

where the functions  $f_0(\theta)$ ,  $f_1(\theta)$ ,  $f_2(\theta)$ , and  $f_3(\theta, \lambda)$  are empirically established as follows:

$$\begin{aligned} f_0(\theta) &= 2.61 \times 10^6 e^{-\left(\frac{\theta}{0.002}\right)^2}, \\ f_1(\theta) &= \frac{20.91}{(\theta + 0.02)^3}, \\ f_2(\theta) &= \frac{72.37}{(\theta + 0.02)^2}, \\ f_3(\theta, \lambda) &= 436.9 \frac{568}{\lambda} e^{-(\theta - 3\frac{\lambda}{568})^2}. \end{aligned} \quad (3)$$

In previous research, the glare effect is represented by the weighted summation of these four functions, and these weights change together with the average luminance. To produce the glare samples, we calculate these functions with for three discrete wavelengths. Red at  $700.0nm$ , green at  $546.1nm$ , and blue at  $435.8nm$ . Therefore, we can change the color of the glare effect by modifying the distribution of these three colors at the light sources.

Again, we make use of the Graph Cut algorithm. The user indicates through some strokes which areas correspond to foreground or background areas. After segmentation, our method pastes the light effect samples on each

pixel as follows:

$$C'_s = C_s + \sum_{T \in Window} P_*(\theta_{st}), \quad (4)$$

where  $C_s$  and  $C'_s$  represent the original and changed *RGB* color of the pixel  $S$ ,  $T$  represents the pixel inside the window area,  $P_*$  is the chosen point spread function (as defined in [10]) according to average illumination, and  $\theta_{st}$  represents the degree of spreading between pixel  $S$  and  $T$  at light source. For example, if the average illumination can be regarded as scotopic the system chooses function  $P_s$  to be function  $P_*$ . Because  $\theta_{st}$  is small in these point spread functions, we assume that it is linearly proportional to the Euclidean distance between these pixels.

In the case of point light sources, we apply the light effect sample onto the user selected pixel directly. Then, our system generates the point light effects radially by subdividing the circular halo effect as described in [10].

$$C'_s = C_s + P_*(\theta_{st}). \quad (5)$$

In these two cases, our method automatically chooses the light effect sample from three types. As mentioned above, the characteristics of the human eye change accordingly to the environmental luminance. So in our method, in nighttime image synthesis (as described in Sec. 3.3), the user-requested average luminance is used to choose a glare effect which is most adequate for the image luminance, from the three previously mentioned thresholds. This can be done using normalized value  $l \in [0, 1]$  which maps to the possible day to night luminance values. By default we use white color for the glare effect, but the user may also specify a different color.

While the light of windows and street lamps can be regarded as irradiating directly into our eyes, light reflection is largely affected by the color and shape of the reflecting object. Therefore, we simulate indirect light by the following equation:

$$(XY'Z) = (X, (1+k)Y, Z), \quad (6)$$

where  $X, Y, Z$  are color values of the original pixel in the  $XYZ$  color space,  $X', Y', Z'$  are modified values, and  $k$  is the illumination constant, which means the illumination level. In the *RGB* color space, this equation can be expressed as follows:

$$\begin{pmatrix} R' \\ G' \\ B' \end{pmatrix} = \begin{pmatrix} R - 1.53715k \\ G + 1.875991k \\ B - 0.204043k \end{pmatrix}, \quad (7)$$

where  $R, G, B$  are also the color values of the original pixel, and  $R', G', B'$  are the modified values. By increasing the  $Y$  value in the  $XYZ$  color space, which is the luminance of the pixel, this equations models the color reflection from the object. However, we do not extract the spatial information of the light source and reflecting object, which is left for future research. Therefore, it is necessary to modify the shape and angle of this effect manually to make it adequate to the surface.

Finally, we apply the method by [14] for the light interruption. This method can segment the foreground object

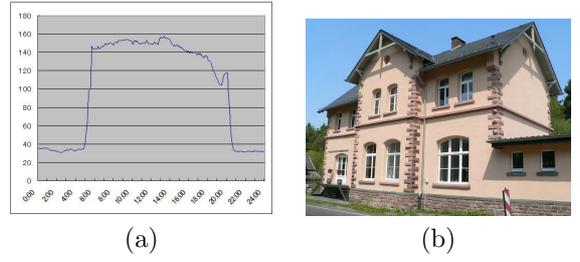


Figure 1: (a) Example data of the average luminance in a city environment. The horizontal axis represents time, and the vertical axis represents the average luminance (0 to 255). (b) Input house.

and compose it over other objects automatically simply by dragging. Both direct light and reflected light may be interrupted by other objects. Particularly, the reflected light may be divided at the boundary of the reflecting surface. By applying this method, we can drag these light effects behind the occluding objects without manual segmentation.

## 4 Results

The input image is shown in Fig. 1 (b). In the shadow segmentation, the user-given strokes are illustrated in Fig. 2 (a) where red indicates the sunlit surface, and blue strokes indicate the surface in the shadow. Fig. 2 (b) shows illustrates the segmentation results.

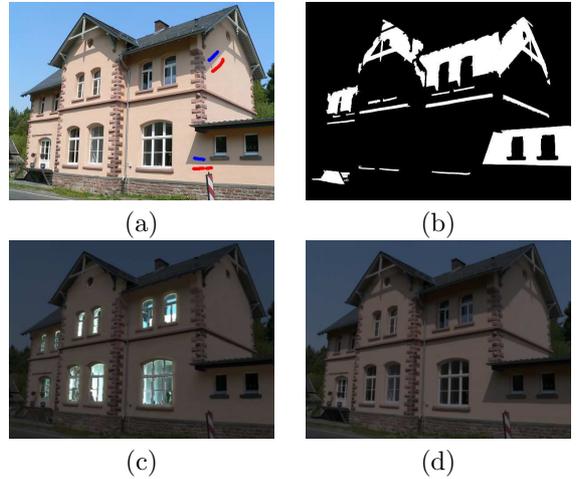


Figure 2: (a) Segmentation user input example. (b) Shadow segmentation result. (c) Our result when using Eq. 6. (d) Result from [8].

We compare our result with the result of the method [8] in Fig. 2 (c) and Fig. 2 (d). Because that method does not take into consideration the shadows due to the sunlight, the shadows are still visible in the synthesized image. Additionally, as seen on the figure, this method also does not introduce any artificial light. Fig. 3 shows another example generated with our method. In this example, a normal glare effect is shown around the window lights and, a radial glare effect is shown on the car headlights. These two

types of the glare can be applied independently. When user point regions which he or she wants to illuminate on night synthesized image, system calculates average luminance of that image, estimates glare size, and applies them onto the image. In addition to this, if the light source indicated by the user can be regarded as the point light source because of its size, the system replaces the lenticular halo effect with a more powerful radial halo effect.

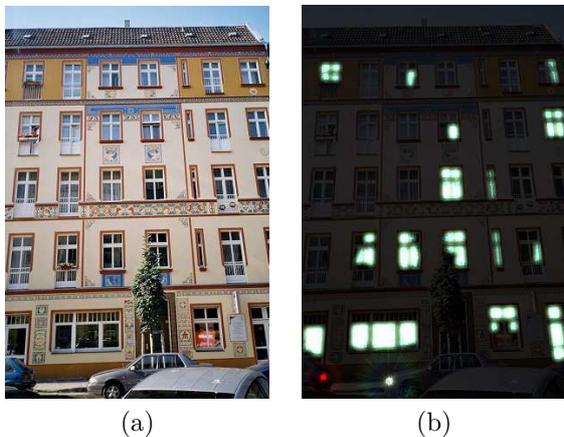


Figure 3: (a) Apartment building and cars. (b) Synthesized image when using Eq. 4. Notice the halo effect around the headlights and rear lights.

Our method can also make stationary video from one input image and some user input. The segmentation step is applied only once and then, the system swipes the parameter range from original luminance to the target luminance.

## 5 Conclusion and Future Work

We presented an interactive nighttime image synthesis method that takes into account the characteristics of the human visual system. Our method makes it possible to add artificial light sources, with minimal user interaction. Such effects were not considered in previous research. Our method only requires the user to indicate which areas are lit, all other parameters are automatically computed. Thanks to this estimation, the user interaction is minimized when compared with using standard off-the-shelf image editing software, making our method particularly attractive to novice users. In addition to this, our method introduces a step to remove sunlight shadows from the input image.

However, there are some limitations in our approach. First, we do not have any spatial information and, therefore there are some regions which might be incorrectly illuminated. Nevertheless, we believe that even in those situations our method is still valuable for novice users.

As future work, we consider mainly two objectives. One is to overcome current limitations in order to achieve more natural results. We believe it may be possible to estimate three dimensional spatial information from surface characteristics such as color, shape, etc.

Also, currently our method needs user interaction in the segmentation of window and shadowed areas. Therefore, our objective is to reduce the user interaction even further, and possibly to achieve full automatism.

## References

- [1] J. A. Ferwerda, S. N. Pattanaik, P. Shirley, and D. P. Greenberg, "A model of visual adaptation for realistic image synthesis," in *ACM SIGGRAPH 1996 Conference Proceedings*, pp. 249–258, 1996.
- [2] S. N. Pattanaik, J. A. Ferwerda, M. D. Fairchild, and D. P. Greenberg, "A multiscale model of adaptation and spatial vision for realistic image display," in *ACM SIGGRAPH 1998 Conference Proceedings*, pp. 287–298, 1998.
- [3] G. W. Larson, H. Rushmeier, and C. Piatko, "A visibility matching tone reproduction operator for high dynamic range scenes," *IEEE Transactions on Visualization and Computer Graphics*, vol. 3, no. 4, pp. 291–306, 1997.
- [4] F. Durand and J. Dorsey, "Interactive tone mapping," in *Proceedings of the 11th Eurographics Workshop on Rendering Techniques*, pp. 219–230, 2000.
- [5] E. Reinhard, M. Stark, P. Shirley, and J. Ferwerda, "Photographic tone reproduction for digital images," *ACM Transactions on Graphics*, vol. 21, no. 3, pp. 267–276, 2002.
- [6] W. B. Thompson, P. Shirley, and J. A. Ferwerda, "A spatial post-processing algorithm for images of night scenes," *Journal of Graphics Tools*, vol. 7, no. 1, pp. 1–12, 2002.
- [7] C.-C. Liu and K.-W. Chuang, "An outdoor time scenes simulation scheme based on support vector regression with radial basis function on dct domain," *Image Vision Comput.*, vol. 27, pp. 1626–1636, September 2009.
- [8] G. Haro, M. Bertalmío, and V. Caselles, "Visual acuity in day for night," *International Journal of Computer Vision*, vol. 69, no. 1, pp. 109–117, 2006.
- [9] P. Rokita, "A model for rendering high intensity lights," *Computers & Graphics*, vol. 17, no. 4, pp. 431–437, 1993.
- [10] G. Spencer, P. Shirley, K. Zimmerman, and D. P. Greenberg, "Physically-based glare effects for digital images," in *ACM SIGGRAPH 1995 Conference Proceedings*, pp. 325–334, 1995.
- [11] M. Kakimoto, K. Matsuoka, T. Nishita, T. Naemura, and H. Harashima, "Glare generation based on wave optics," *Computer Graphics Forum*, vol. 24, no. 2, pp. 185–193, 2005.
- [12] Y. Y. Boykov and M. P. Jolly, "Interactive graph cuts for optimal boundary & region segmentation of objects in ND images," in *Proceedings of the*

*8th IEEE International Conference on Computer Vision*, vol. 1, pp. 105–112, 2002.

- [13] M. S. Rea, *The Illumination Engineering Society Lighting Handbook*. Illumination Engineering Society, 8th ed., 1993.
- [14] Y. Yamaguchi, T. Saito, Y. Bando, B.-Y. Chen, and T. Nishita, “Interactive image composition through draggable objects,” in *ACM SIGGRAPH Asia 2009 Sketches*, pp. 32:1–32:1, 2009.