

Glare Generation Based on Wave Optics

Masanori Kakimoto
The University of Tokyo
and
SGI Japan, Ltd.
kaki@sgi.co.jp

Kaoru Matsuoka
The University of Tokyo
matsuoka@hc.t.u-tokyo.ac.jp

Tomoyuki Nishita
The University of Tokyo
nis@is.s.u-tokyo.ac.jp

Takeshi Naemura
The University of Tokyo
naemura@hc.t.u-tokyo.ac.jp

Hiroshi Harashima
The University of Tokyo
hiroshi@hc.t.u-tokyo.ac.jp

Abstract

This paper proposes a novel and general method of glare generation based on wave optics. A glare image is regarded as a result of Fraunhofer diffraction, which is equivalent to a 2D Fourier transform of the image of given apertures or obstacles. In conventional methods, the shapes of glare images are categorized according to their source apertures, such as pupils and eyelashes and their basic shapes (e.g. halos, coronas or radial streaks) are manually generated as templates, mainly based on statistical observation. Realistic variations of these basic shapes often depend on the use of random numbers. Our proposed method computes glare images fully automatically from aperture images and can be applied universally to all kinds of apertures, including camera diaphragms. It can handle dynamic changes in the position of the aperture relative to the light source, which enables subtle movement or rotation of glare streaks. Spectra can also be simulated in the glare, since the intensity of diffraction depends on the wavelength of light. The resulting glare image is superimposed onto a given computer-generated image containing high intensity light sources or reflections, aligning the center of the glare image to the high intensity areas. Our method is implemented as a multi-pass rendering software. By pre-computing the dynamic glare image set and putting it into texture memory, the software runs at an interactive rate.

1. Introduction

Rendering of high intensity light is a difficult problem because of physical limitations in the brightness of

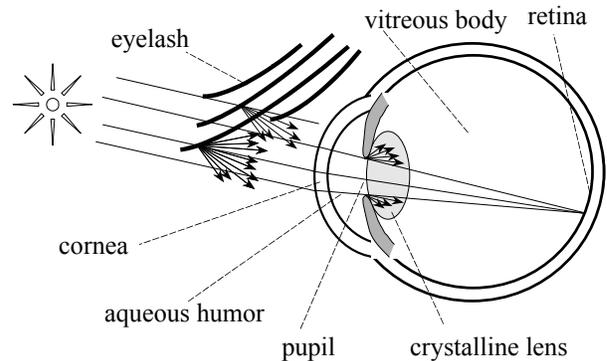


Figure 1. Light rays passing through the eye structure.

computer displays. A software-based solution to this issue is to use the psychological aspects of human vision. Several researchers have worked on the problems of glare because it is a phenomenon that is always associated with high intensity lights.

Glare is a phenomenon whereby bright light sources or reflections cause a spreading of light, mainly in human eyes. It is perceived as a blurry circle or a set of radial streaks around the light source. Recently, glare generation has become a common rendering technique to enhance reality in computer-generated imagery.

It is widely known that glare is caused by diffraction and scattering at obstacles close-to or inside the eye (Figure 1). Rays from a light source are first diffracted by the eyelashes (Some people make their eyelashes curl upward or have such eyelashes by nature and in these cases the eyelashes have little affect.) and sometimes by the edge of the eyelids [1]. This produces long radial

streaks of glare. After entering through the cornea, the light rays are diffracted or scattered by the edge of pupil, which causes a blurry corona to appear around the light sources. Even suspended matter in the vitreous body is said to cause diffraction. Other than diffraction and scattering, aberration of the lens and anomalies of refraction are also sources of glare, but their impact is relatively limited. Figure 2 (in a color page) shows an experiment of glare using a piece of false eyelashes.

Computer graphics techniques to artificially generate and render glare on digital images have been pursued in the past and they are now widely used as glare filters. The previous work is mentioned in Section 2.

Glare rendering would be more important than ever in accordance with the spread of HDR (High Dynamic Range) imaging technology and its contents. Scenes in nature, which are captured as HDR images by the human eye, often cause the glare that we experience in daily life. On the other hand, current electronic display devices cannot produce glare because of their limited dynamic range. Thus, artificial glare effects could improve the appearance of HDR images mapped to those Low Dynamic Range displays.

Main Contributions: The method we propose in this paper is not merely a special effect for realism, but is a wave-optics based simulation of the physical phenomena caused by various obstacles between strong light sources and the retina. Changes in the condition of the individual obstacles or the light sources directly affect the resulting image. This means that the proposed method could potentially be utilized in more practical applications than special effects.

The application of the glare-generation method described in this paper has three aspects; 1) as a glare rendering technique, 2) as a vision system simulation (a.k.a. vision-realistic rendering) and 3) as a GPU accelerated technique.

Advantages: The proposed method is more physically based than previous methods because it uses a formula that represents the intensity of diffracted light, which is well established in the field of wave optics, or Fourier optics. In existing methods, on the other hand, glare images are usually designed manually, based on statistical observation.

Secondly, the algorithm is simple because the basic formula can be implemented as an FFT process. It is also robust for any type of input.

Since the input data for the glare generation process consists of image data of the obstacles that cause diffraction, a wide range of applications might be possible

by simply replacing or combining the input image data; e.g. eyelashes, eyelids or even camera diaphragms.

Another advantage would be that the algorithm may be easy to accelerate using modern programmable GPUs because the most time-consuming process is the execution for the Fourier transform, which now has been implemented on a GPU [2].

Organization: In Section 2 we give a brief survey of related work on glare rendering techniques and vision system simulation. In Section 3 we introduce the theory of Fraunhofer diffraction, then describe how glare images are computed using the theory. We also show details of the implementation and our results, including several examples of glare images for a couple of different types of obstacles, mainly focusing on eyelashes. Section 4 explains the whole image generation process, which is a multi-pass rendering technique into which the algorithm shown in the previous section is to be integrated. In Section 5 we conclude the paper by giving a summary and outlining future work.

2. Related work

In this section, we give a brief description of prior work on glare rendering and vision system simulation.

2.1. Glare rendering techniques

Computer-generated glare was first introduced by Shinya et al. [3]. They simulated a cross-screen filter attached to a camera. Their algorithm produces glare consisting of a pair of horizontal and vertical streaks crossing at high intensity pixels rendered with a ray tracer. The glare image is generated by post-processing, and is actually an image convolution of the ray-traced image.

The glare generation technique using a convolution filter was expanded and applied to a night driving simulator by Nakamae et al. [4]. They made an observation of eyelashes and prepared a realistic filter by considering the distribution of angles for each eyelash. They also added a spectrum effect on the glare image. Spencer et al. [5] proposed a general glare filter synthesizer. This enabled designers to compose flexible glare shapes, such as coronas for diffraction at the edge of the pupil and streaks for diffraction at eyelashes, as well as blooms caused by scattering of light at various places of the eye.

Rokita [6] proposed a practical approach for glare generation. He made more detailed observations of the

structure of the human eye to create a glare image and attached it as a billboard overlaid on the pixel corresponding to a strong light source.

The combination of a glare filter and an HDR image is an effective method for rendering realistic scenes with high intensity pixels, as mentioned in Section 1. Debevec et al. [7] applied glare filters to an HDR image of the interior of a cathedral.

Recent innovations in programmable GPUs have enabled real-time rendering with glare filters. Mitchell [8] demonstrated an interactive version of an existing batch-rendered motion picture containing some glare effects with two-to-three orders of magnitude of acceleration.

The glare filter techniques established by the above researchers have been widely applied to image rendering as a special effect in the game [9] and film-making industries.

However, all of the previous works use more or less human-designed shapes or fixed shapes for the glare image. As far as we know, there has been little research that has taken the individual shape of ‘diffractors’ (i.e., structural elements of the eye or various types of diaphragms) directly into account. Our proposed method directly handles glare caused by diffraction using a wave optics theory. The only computer graphics research work that explicitly coped with wave optics is Moravec [10]. Our approach to glare generation shares the basic idea with his work.

While the proposed method focuses on diffraction, it is less applicable to scattering, which is also a major source of glare. For practical applications, hybrid implementation with a popular glare-filtering technique such as Spencer et. al. [5] may be needed.

2.2. Vision system simulation

Simulations of the optical aspects of the structure of the eye have been pursued. Depth-of-field was an early research topic when considering optical vision for computer graphics. Potmesil and Chakravarty [11] first introduced a depth of field simulation using a convolution filter for blurred images. Cook et al. [12] invented distributed ray tracing and solved an occlusion problem in the depth-of-field effect. Haeberli and Akeley [13] proposed a hardware-accelerated depth-of-field effect using an accumulation buffer.

Research has been carried out that focused more on the characteristics of individual eyes and which was oriented towards ophthalmologic applications. Santamaria et al. [14] measured patient’s eye as a lens system with optical equipment, acquiring a point-spread function for use in visual acuity simulation. Mostafawy et al. [15] employed distributed ray tracing and a multi-lens camera model [16] in order to evaluate refractive errors in human vision.

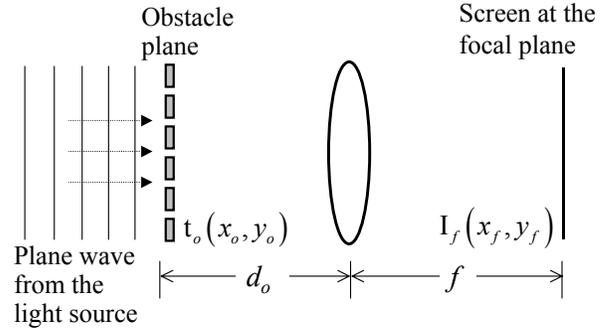


Figure 3. A simplified lens system configuration.

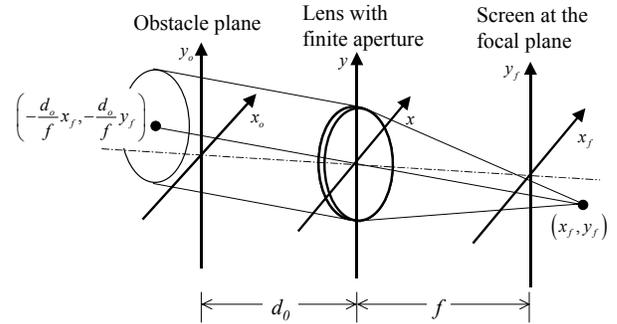


Figure 4. A lens system model considering a finite aperture on the lens.

They also simulated vision before and after photorefractive keratectomy (PRK) surgery.

Barsky et al. [17] proposed the concept of Vision-Realistic Rendering. For example, they used a Shack-Hartmann wavefront aberrometry device to measure accurate aberrations of the human eye, and determined the wavefront of the lens system for a point-light source. A set of depth point-spread functions was computed from the wavefront and applied to an image of each depth for creation of a final vision-realistic image.

Although several optical aspects of the eye are considered in the vision-system simulation approach, nobody attempted to take diffraction effects into account in any of this work.

3. Glare filter based on wave optics

The structure of the human eye as a lens system is fairly complicated, as shown in Figure 1. The cornea is the first lens that light rays pass through, and it has the greatest refracting power. The aqueous humor and the vitreous body have the same refractive index as water and

do not have a positive refracting effect in this environment. The crystalline lens has variable refracting power and is used for focusing. The three major diffraction-causing obstacles, the eyelashes, the eyelids and the pupil edge, are located at different positions relative to the lenses.

3.1. A simplified lens system model for diffraction

First we use a simplified model to approximate diffraction. Figure 3 illustrates the optical model for an eye as a lens system. Here, d_o is the distance between the given obstacle plane and the lens. Obstacles, e.g. a set of eyelashes, are simply placed on the obstacle plane. The distance f is the distance between the lens and the screen or the retina. It is assumed that the screen is on the focal plane of the lens, which means f is the focal distance.

According to the basic theory of wave optics [18][19], the distribution of the complex amplitude $U_f(x_f, y_f)$ on the screen becomes

$$U_f(x_f, y_f) = \frac{A}{j\lambda f} \exp \left[j \frac{\pi}{\lambda} \left(1 - \frac{d_o}{f} \right) (x_f^2 + y_f^2) \right] \times \iint_{-\infty}^{\infty} t_o(x_o, y_o) \exp \left[-j \frac{2\pi}{\lambda f} (x_o x_f + y_o y_f) \right] dx_o dy_o, \quad (1)$$

where A and λ are the amplitude and the wavelength of the incoming plane wave, respectively. The 2D function $t_o(x_o, y_o)$ is the amplitude transmittance of the obstacle. It can be represented as a 2D grayscale image, each of whose pixels is the opacity of the corresponding position of the obstacle.

This formula is known to represent Fraunhofer diffraction. Note that $U_f(x_f, y_f)$ is equivalent to the Fourier transform of $t_o(x_o, y_o)$ except for the phase factor preceding the integrals. When $d_o = f$, Equation (1) becomes an exact Fourier transform. Also note that the equation is an approximation that cannot be applied when $d_o \gg f$.

Since it is the intensity distribution across the screen that is of real interest in computer graphics applications, the absolute value of the amplitude distribution is squared and the phase distribution has no effect. Thus the intensity across the screen on the focal point becomes

$$I_f(x_f, y_f) = |U_f(x_f, y_f)|^2 = \frac{A^2}{\lambda^2 f^2} \left| \iint_{-\infty}^{\infty} t_o(x_o, y_o) \exp \left[-j \frac{2\pi}{\lambda f} (x_o x_f + y_o y_f) \right] dx_o dy_o \right|^2 = \frac{A^2}{\lambda^2 f^2} \left| \mathbf{F} \left[t_o(x_o, y_o), \lambda f \right] \right|^2, \quad (2)$$

where \mathbf{F} is an operator denoting Fourier transform of an input function (in this case $t_o(x_o, y_o)$) with a parameter value (in this case λf).

3.2. A model taking lens aperture into account

In the previous subsection, we assumed that the extent of the lens aperture is infinity, or that the lens size is sufficiently large compared to the obstacle. To include the effect of the finite aperture, we employ a geometrical approximation. Figure 4 shows the configuration of the lens system with a finite aperture.

The finite extent of the lens aperture can be accounted for by projecting the aperture back onto the obstacle plane. The projection is centered on a line joining the coordinates (x_f, y_f) with the center of the lens. The value of I_f at (x_f, y_f) can be found from the Fourier transform of that portion of the obstacle masked by a pupil function P , centered at $[-(d_o/f)x_f, -(d_o/f)y_f]$. Thus

$$I_f(x_f, y_f) = \frac{A^2}{\lambda^2 f^2} \left| \mathbf{F} \left[t_o(x_o, y_o) P \left(x_o + \frac{d_o}{f} x_f, y_o + \frac{d_o}{f} y_f \right), \lambda f \right] \right|^2, \quad (3)$$

where the pupil function $P(x, y)$ is defined by

$$P(x, y) = \begin{cases} 1 & \text{inside the lens aperture} \\ 0 & \text{otherwise.} \end{cases} \quad (4)$$

Note that the input function for the Fourier transform varies depending on the position of interest on the output screen (x_f, y_f) . This forces us to compute the Fourier transforms from the obstacle images slightly differently from each other for each pixel across the screen.

In order to simulate the diffraction that takes place in actual human eyes, we regard $t_o(x_o, y_o)$ as an image representing the silhouette of the eyelashes and the eyelids. It is natural to let image $P(x, y)$ represent the silhouette of the pupil. For cameras, the pupil function $P(x, y)$ is the shape of the diaphragms and $t_o(x_o, y_o)$ is the pattern of an attached filter (if any).

Since the computation of Equation (3) is rather complicated, we assume another approximation with a fixed mask image, still keeping some of the advantages of the pupil function.

$$I_f(x_f, y_f) = \frac{A^2}{\lambda^2 f^2} \left| \mathbf{F} \left[t_o(x_o, y_o) P(x_o, y_o), \lambda f \right] \right|^2. \quad (5)$$

Thus, the problem resolves itself into a single Fourier transform process for an image $t_o(x_o, y_o)$ masked by $P(x_o, y_o)$.

The spectrum effect caused by the wavelength factor in Equation (5) is approximated by simply using different wavelength for each pixel component (R:680nm, G:530nm, B:460nm) and composing the three results. This assumption implies that the light source contains three, equal-strength, pure coherent spectrums of those wavelengths.

3.3. Non-linear characteristics between real world luminance and display luminance

Prior to implementing a diffraction simulator using the theories described in the previous subsections, we need to consider the non-linear characteristics of the relationship between the luminance intensity $I_f(x_f, y_f)$ and the pixel intensity, or frame buffer value.

Tone mapping techniques for high contrast images or high dynamic range images have received much attention [20]-[25]. Among the techniques, we employed the scene layering and illumination compression method by Tumblin et. al. [23] since, in our proposed method, glare image is treated as a separate layer from the scene rendering result (see Section 4 for details).

The tone mapping function for the illumination compression forms a ‘‘sigmoid’’ curve which looks like an ‘‘S’’ curve when plotted on log-log axes for the input real world luminance $I_f(x_f, y_f)$ and the output display luminance $I_p(x_f, y_f)$.

$$I_p(x_f, y_f) = \text{sig}\{I_f(x_f, y_f)\}. \quad (6)$$

where $\text{sig}()$ is a sigmoid function,

$$\text{sig}(x) = \frac{x^g + 1}{x^g + k} \cdot D. \quad (7)$$

The parameter D defines the maximum limit of the output, k^2 defines the contrast between the minimum and the maximum value of the output, and the exponent g , together with k , defines the maximum slope of the function. The shape of the function can be adjusted by varying the three parameters.

The raw FFT result $I_f(x_f, y_f)$ is typically such an image that a very bright and small spot appears at the

center and the diffracted light appears around the center with brightness smaller than that of the spot by orders of magnitude. The central bright spot is treated as being on a separate image layer and is excluded from the tone mapping input image $I_f(x_f, y_f)$ by clipping and expanding the input value. This enables us to reserve better dynamic range for the glare image.

3.4. Implementation Results

The results of the implementation of Fourier transform of Equations (5) and the tone mapping function of Equation (6) for several practical input images are presented in Figure 5. An FFT program followed by a sigmoid function is used to generate the image $I_p(x_f, y_f)$. The image size of $t_o(x_o, y_o)$, $I_f(x_f, y_f)$ and $I_p(x_f, y_f)$ is 512×512 . The dynamic range of the pixels of $t_o(x_o, y_o)$ and $I_p(x_f, y_f)$ is $[0, 255]$. All the calculations in between are done with double precision floating point. The tone mapping parameters are $D = 640$, $k = 20$ and $g = 0.465$.

Figure 6 shows results for an HDR sample image. The glare image is overlaid for each high intensity pixel as a semi-transparent billboard. The spectral effect is obtained by applying the algorithm for each of RGB components with different λ values in Equation (5).

4. Integration of glare into a real-time image generation process

In this section, we describe how the glare filter generated in the process explained above can be used in common 3D rendering frameworks, such as OpenGL and DirectX.

4.1. Overview of the whole rendering process

As with most glare filtering techniques, our proposed method uses the glare generation results at the post-processing stage in the main rendering loop.

Figure 7 illustrates the flow of the rendering process for each frame. The main rendering loop consists of 3D scene rendering and of post-processing for glare. There are two 3D scene-rendering routines of similar type: one for regular, low dynamic range rendering and another for the rendering of high intensity pixels that correspond to light sources or to the highlights of specular reflective surfaces. The resulting image of the latter is read from the frame memory back to the main memory and used to find out where and how strong the glare image should be when it is attached over the former regular rendering result.

A glare image is stored as a texture image and mapped onto an alpha-channel-enabled translucent billboard,

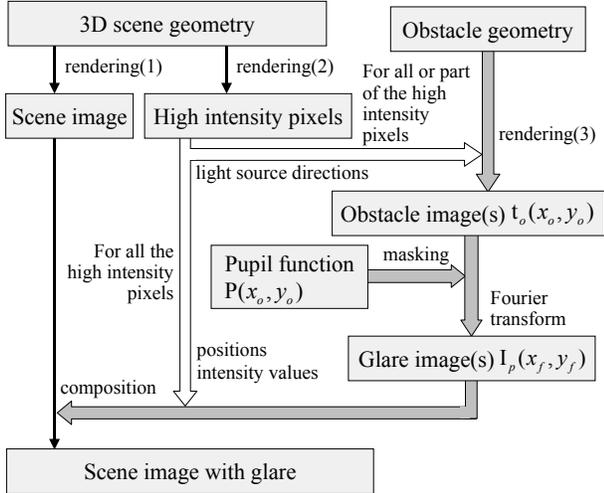


Figure 7. Flow diagram of the multi-pass rendering for each frame. The thin arrows represent per-frame operations, the gray block arrows represent per-pixel operations for each high intensity pixel and the white block arrows represent per-pixel basis parameter transfers. Of these six operations, rendering(3), masking and the Fourier transform can be run in the pre-computation stage and the glare images can be stored as intermediate data. Obstacle geometry and rendering(3) can be optional if the user provides an obstacle image.

appropriately placed at each high-intensity pixel. A single, fixed glare image can be reused throughout the whole sequence of the scene. Even so, the users can still evoke some shape variations of glare images caused by the shapes of high-intensity pixel clusters. In addition, the glare image intensity is multiplied by the values of the high intensity pixels.

4.2. Dynamic glare image generation

The main advantage of the proposed method is that the glare image changes as the input condition changes.

Glare image can be generated on the fly, which is helpful when the input image $t_o(x_o, y_o)$ or $P(x_o, y_o)$ changes dynamically frame by frame. For example, when the viewer turns the head horizontally, the relative movement of the position of an existing high-intensity light source requires scrolling of the eyelash silhouette $t_o(x_o, y_o)$. By

varying the input image, the users can enjoy subtle changes or rotation of the glare shape as the scene is panned.

However, an FFT calculation needs to be completed for every frame, which might cause a performance penalty.

4.3. Glare generation as a pre-computation stage

In case the computational power is insufficient for dynamic glare generation, one can pre-compute and store all the glare images for possible high intensity pixels, assuming in this case that the only changeable condition is the direction of the high intensity pixels. The shapes of the eyelashes or other diffraction-causing obstacles need to be static.

The pre-computation process forms a loop for each of the pixels over the final output image. Since each pixel is a potential high intensity pixel, we can regard it as corresponding to a potential light source direction. 2D or 3D aperture obstacles (may often be a set of eyelashes) are rendered using parallel projection (since plane wave is assumed for light sources) along this direction and the result becomes an image $t_o(x_o, y_o)$. The glare generation routine then produces a glare image $I_p(x_f, y_f)$ for that pixel. This is carried out for all or some parts of the pixels across the output viewport, and as many glare images as that number of pixels are stored in the memory.

In the main rendering loop, an appropriate stored glare image is simply selected according to the location of the high intensity pixel. Thus, an interactive change of shape in accordance with the glare location is realized.

This is rather a ‘brute-force’ method, and its full implementation is not practical since huge amounts of memory are required. For example, we will require 64G pixels for a 1000×1000 output viewport and a 256×256 glare image. However, if we get an array of pixels along the same vertical line to share a 256×256 16bit glare image, the amount of memory required will be 128MB, which is possible for today’s typical systems. We used this ‘x position dependent’ pre-computed glare image set for implementation. Figure 8 demonstrates how such dynamic glare images are created. See 4.5 for details.

4.4. Rendering of reflective objects

Reflection is an important source of high intensity pixels, and thus is a source of glare. For example, automobile bodies (when placed in sunshine) strongly

reflect sunlight and therefore cause glare. Kakimoto et al. [26] proposed a technique of locating high intensity pixels due to reflection using HDR environment maps, while acquiring surface reflection attributes for those pixels. We combined their technique with our glare generator.

4.5. Implementation results of dynamic glare

We have implemented our proposed dynamic glare generation method using the pre-computation and applied it to reflective objects. The eyelash and pupil images in Figure 5 (b) (left and middle) are used as an obstacle input. The eyelash image is actually wider than it is shown in the figure and is scrolled according to horizontal component of light direction. 600 scrolled images of 512×512 pixel are prepared and stored on the texture memory to be used for mapping on the billboards.

It took 0.21sec for an FFT of a 512×512 image using Pentium 4 1.4GHz. Including file I/O, the pre-computation process took 22 minutes to load 600 obstacle images, run FFTs, and save 600 glare images.

During the interactive drawing process, it took 15 msec (60fps) to over 1000 msec (1fps) per frame to render a scene, depending upon how many high intensity pixels are in the view. Figure 9 shows snapshots from interactive demo scenes. The graphics hardware used is a Silicon Graphics InfiniteReality4 with 1GB texture memory. The billboard size and the alpha (translucent) values for the billboards were adaptively defined for each scene. Since many translucent billboards may often overlap, the frame memory needs to have sufficient bits per pixel, preferably 12bits for each of RGB and alpha components, in order to avoid overflow of significant digits.

5. Conclusion and future work

In this paper, we have proposed a physically based, fully automatic glare generation method using the theory of Fraunhofer diffraction in wave optics. Our method utilizes a set of obstacles that cause diffraction as another input in addition to the main 3D input scene. These are in the form of 3D geometry, a hand-drawn picture or a snapshot picture. A glare image is computed as an intermediate result using a Fourier transform and this is finally added to each high intensity pixel of the 3D scene rendering result as a billboard texture.

An issue of the 2D approximation of 3D obstacles is left and needs to be further investigated. In Fraunhofer diffraction, depth of obstacle affects only the phase factor of the diffracted light wave (see Equation (1)) and has no influence on the intensity. Thus, in this paper, the amount of change of the depth or distance between the obstacle and the lens is regarded to having little affect. However,

Fraunhofer approximation does not handle 3D or multiply layered obstacles. More theoretical analysis in wave optics may be required to handle the condition.

A major benefit of the proposed method is flexibility. Users can design various types of realistic glare by providing obstacle models or pictures, which is much more intuitive and easier than directly-painting glare images. The ability to handle dynamic glare is also an advantage of this method over conventional glare filter techniques.

Interactive-rate performance can only be achieved under limited conditions: with a static glare image and with fewer high-intensity pixels. Dynamic glare generation on the fly would be possible by executing FFT on a GPU. Currently, the rendering speed drops when lots of high intensity pixels are detected, because the pixel-fill operations for the glare billboards increase in proportion to the high intensity pixels. Since a high-intensity area is often detected as a cluster of continuous pixels, there might be some clever way of dramatically reducing the number of billboards.

Acknowledgements

We would like to thank Dr. Mikio Shinya of Toho University for inspiring us with discussions and for his advice on wave optics theory. Thanks also go to Norio Izumi and Seiji Matsuki of SGI Japan for their continuous support of our work. Yukiteru Mukai also helped us make submission demo video. Kei Iwasaki of the University of Tokyo provided us with the ocean geometry data that he generated.

References

- [1] M. G. J. Minnaert (Translated and revised by Seymour L.), *Light and Color in the Outdoors*. Springer-Verlag New York, 1993.
- [2] K. Moreland, E. Angel, "The FFT on a GPU", *Graphics Hardware 2003*, July 2003.
- [3] M. Shinya, T. Saito, T. Takahashi, "Rendering Techniques for Transparent Objects", In *Proc. Graphics Interface '89*, 1989, pp. 173-182.
- [4] E. Nakamae, K. Kaneda, T. Okamoto, T. Nishita, "A Lighting Model Aiming at Drive Simulators", In *Proc. SIGGRAPH '90*, August 1990, pp. 395-404.
- [5] G. Spencer, P. Shirley, K. Zimmerman, D. P. Greenberg, "Physically-Based Glare Effects for Digital Images", In *Proc. SIGGRAPH '95*, August 1995, pp. 325-334.
- [6] P. Rokita, "A Model for Rendering High Intensity Lights", *Computers & Graphics*, Vol. 17, No. 4, 1993, pp. 431-437.

- [7] P. E. Debevec, J. Malik, "Recovering High Dynamic Range Radiance Maps from Photographs", In *Proc. SIGGRAPH '97*, August 1997, pp. 369-378
- [8] J. L. Mitchell, "RADEON 9700 Shading", *State of the Art in Hardware Shading*, Course Note #17, SIGGRAPH '02, July 2002.
- [9] M. Kawase, M. Nagatani, "Real Time CG Rendering Techniques Used in DOUBLE-S.T.E.A.L.", CEDEC2002: CESA Game Developers Conference 2002, No. 1-3-A, Tokyo, September 2002. (In Japanese)
- [10] H. P. Moravec, "3D Graphics and the Wave Theory", In *Proc. SIGGRAPH '81*, 289-296, August 1981.
- [11] M. Potmesil, I. Chakravarty, "A Lens and Aperture Camera Model for Synthetic Image Generation", In *Proc. SIGGRAPH '81*, August 1981, pp. 297-305.
- [12] R. L. Cook, T. Porter, L. Carpenter, "Distributed Ray Tracing", In *Proc. SIGGRAPH '84*, 1984, pp. 137-146.
- [13] P. E. Haeberli, K. Akeley, "The Accumulation Buffer: Hardware Support for High-quality Rendering", In *Proc. SIGGRAPH '90*, August 1990, pp. 309-318.
- [14] J. Santamaria, P. Artal, J. Bescos, "Determination of the Point-Spread Function of Human Eyes Using a Hybrid Optical Digital Method", *Optical Society of America A*, Vol. 4, No. 6, 1987, pp. 1109-1114.
- [15] S. Mostafawy, O. Kermani, H. Lubatschowski, "Virtual Eye: Retinal Image Visualization of the Human Eye", *IEEE CG&A*, Vol. 17, No. 1, 1997, pp. 8-12.
- [16] C. Kolb, D. Mitchell, P. Hanrahan, "A Realistic Camera Model for Computer Graphics", In *Proc. SIGGRAPH '95*, August 1995, pp. 325-334.
- [17] B. A. Barsky, A. W. Bargteil, D. D. Garcia, S. A. Klein, "Introducing Vision-Realistic Rendering", 13th Eurographics Workshop on Rendering, Poster Papers, 2002.
- [18] J. W. Goodman. *Introduction to Fourier Optics*. McGraw-Hill Physical and Quantum Electronics Series, McGraw-Hill, 1968.
- [19] S. G. Lipson, H. Lipson. *Optical Physics, Second edition*. Cambridge University Press, 1981.
- [20] J. Tumblin, H. Rushmeier, "Tone Reproduction for Realistic Images", *IEEE CG&A*, Vol. 13, No. 6, November 1993, pp. 42-48.
- [21] C. Schlick, Quantization Techniques for Visualization of High Dynamic Range Pictures, In *Proc. 5th Eurographics Workshop on Rendering*, June 1994, pp. 7-18.
- [22] G. W. Larson, H. Rushmeier, C. Piatko, "A Visibility Matching Tone Reproduction Operator for High Dynamic Range Scenes", *IEEE Transactions on Visualization and Computer Graphics*, Vol. 3, No. 4, October 1997, pp. 291-306.
- [23] J. Tumblin, J. K. Hodgins, B. K. Guenter, "Two Methods for Display of High Contrast Images", *ACM Transactions of Graphics*, Vol. 18, No. 1, January 1999, pp. 56-94.
- [24] E. Reinhard, M. Stark, P. Shirley, J. Ferwerda, "Photographic Tone Reproduction for Digital Images", In *Proc. SIGGRAPH 2002*, 2002, pp. 267-276.
- [25] M. Ashikhmin, "A Tone Mapping Algorithm for High Contrast Images", In *Proc. 13th Eurographics Workshop on Rendering*, 2002, pp. 145-155.
- [26] K. Kakimoto, Y. Mukai, T. Haga, T. Nishita, T. Naemura, H. Harashima, "A Real-Time Glare Rendering Technique Taking into Account Surface Reflection Attributes", *Journal of IIEEJ*, Vol. 32, No. 4, July 2003, pp. 336-345. (In Japanese)

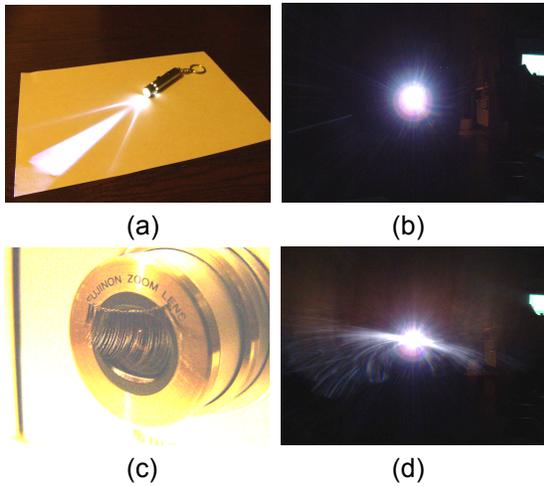


Figure 2. An experiment of glare generation. (a) An LED light used for the light source. (b) Snapshot of the light. (c) False eyelashes attached close to the camera lens. (d) Snapshot of the light with the false eyelashes.

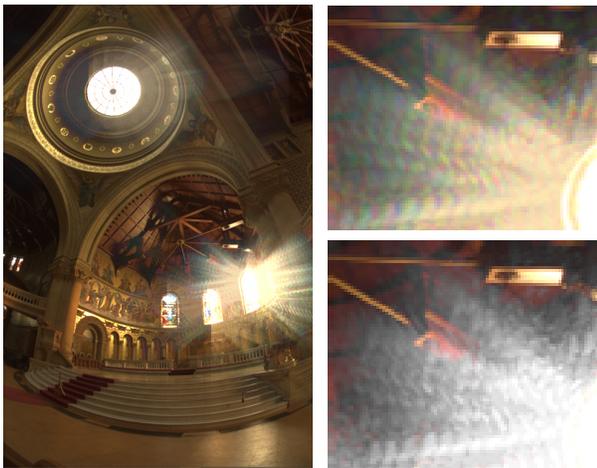


Figure 6. Left: Result of applying the glare output from an eyelash silhouette image to a Debevec's HDR sample image. Upper right: Close-up image of the left in the case where a single wavelength component is used, generating grayscale glare. Lower right: Close-up image of the left to highlight the spectral effect.

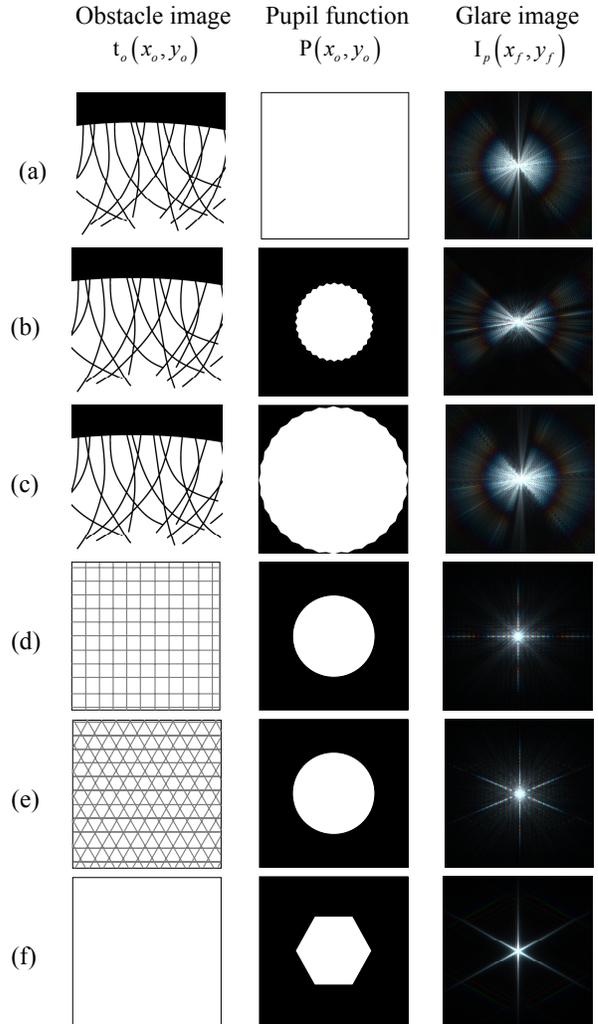


Figure 5. Results of Fraunhofer diffraction using Equations (5) and (6). (a). Eyelashes of both eyes and an upper eyelid taken into account as the obstacle image. (b). Small pupil masking the obstacle image, assuming human eye under daylight. (c). Expanded pupil assuming an eye during night. (d). A camera with a cross-screen filter and an ideal round diaphragm. (e). A camera with a trigonal cross-screen filter. (f). A camera with a hexagonal diaphragm.

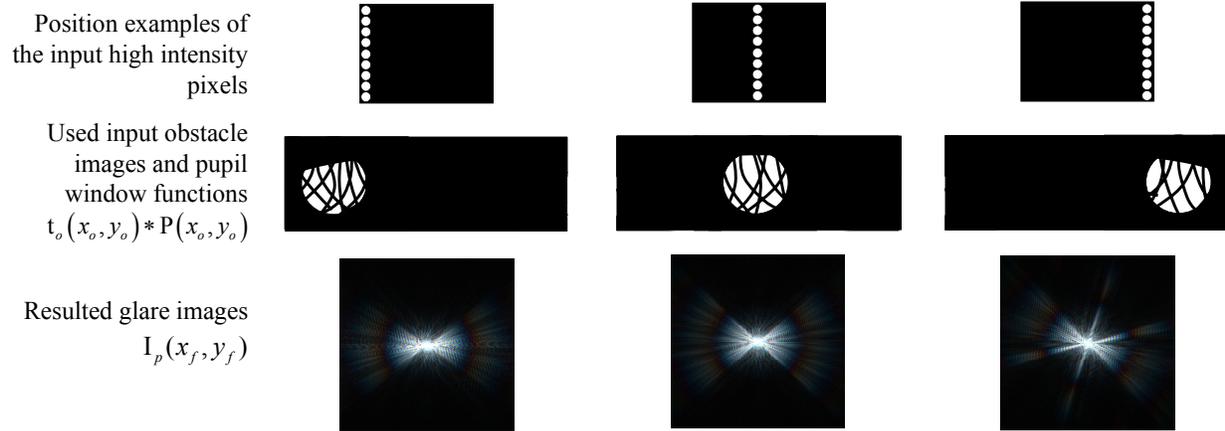


Figure 8. Dynamic glare image generation dependent on horizontal positions of the given high intensity pixel. This process can be carried out either on the fly or in the pre-computation stage. Examples for three horizontal positions of potential high intensity pixels are presented. In the final rendering stage, vertical movement of the high-intensity pixels has no effect on the change of glare because all pixels with the same x coordinates share the same window on $t_o(x_o, y_o)$.

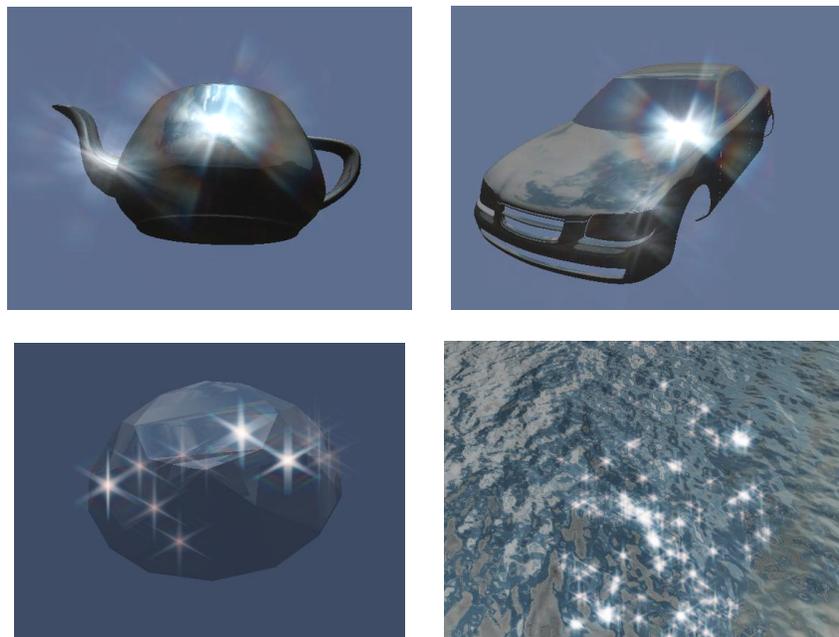


Figure 9. Snapshots from interactive reflection demos. Top left: The teapot model with a white light source through an eye. Top right: A car model with different shininess values on the windshield and the body. Bottom left: A diamond model under multiple incandescent lamps through a camera with a hexagonal diaphragm. Bottom right: An ocean model in the sun.