Glare Generation Based on Wave Optics

Masanori KAKIMOTO †‡  Kaoru MATSUOKA †
Tomoyuki NISHITA †  Takeshi NAEMURA †
Hiroshi HARASHIMA †

† The University of Tokyo  ‡ SGI Japan, Ltd.
Contents

• Introduction
• Previous Work
• Fraunhofer Diffraction
• Simulation of Glare and its Implementation
• Integration to Real-time Image Generation
• Conclusion and Future Work
• Introduction

• Previous Work

• Fraunhofer Diffraction

• Simulation of Glare and its Implementation

• Integration to Real-time Image Generation

• Conclusion and Future Work
What is glare?

- Diffraction or scattering of light ray in human eye caused by very bright light sources
- Blurred shape or streaks around the light source
- Undesired in real life
- Used as a visual effect in computer graphics
Where does glare occur?

- Scattering

- cornea
- crystalline lens
- retina
- sources of scattering (boundaries)
Where does glare occur?

- Scattering and Diffraction

- Sources of scattering (boundaries)
- Sources of diffraction

Diagram showing:
- Eyelashes
- Iris
- Cornea
- Crystalline lens
- Retina
Where does glare occur?

- Scattering and Diffraction

- eyelashes
- iris
- sources of diffraction
- cornea
- crystalline lens
- retina
- sources of scattering (boundaries)
An experiment of glare

- Are eyelashes prevailing sources of glare?

A pen-light used for the experiment

A direct snapshot of the pen-light
An experiment of glare

- Are eyelashes prevailing sources of glare?
  - Yes, they are!

A set of false eyelashes attached to the camera

A direct snapshot of the pen-light
An experiment of glare

• Are eyelashes prevailing sources of glare?

  - Yes, they are.

The same false eyelashes rotated 90 degree

A direct snapshot of the pen-light
An experiment of glare

- Are eyelashes prevailing sources of glare?
  - Yes, they are.

Another set of false eyelashes attached to the camera

A direct snapshot of the pen-light
Introduction

Previous Work

Fraunhofer Diffraction

Simulation of Glare and its Implementation

Integration to Real-time Image Generation

Conclusion and Future Work
Previous Work

• Shinya89
  – Simulated a cross filter of a camera
  – Crossing streaks of light ray is placed on a ray traced image

• Nakamae90
  – Glare design based on statistical distribution of eyelash direction, angles of streaks are generated using random numbers

Nakamae, Kaneda, Okamoto and Nishita, Siggraph ‘90
Previous Work

• Rokita93
  – Glare effect using billboards placed at light sources

• Spencer95
  – Thorough analysis of the human eye structure
  – Glare filter composition
  – Spectrum effect

Spencer, Shirley, Zimmerman and Greenberg, Siggraph ‘95
Previous Work

• Debevec97
  – Glare effect on HDR images

• Mitchell02
  – Glare filter using RADEON pixel shader

• Kawase02
  – Glare filter using NVIDIA pixel shader (XBOX)
## Previous Work

### Comparison

<table>
<thead>
<tr>
<th></th>
<th>glare shape</th>
<th>real-time (HW accelerated)</th>
<th>reflection</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nakamae90</td>
<td>manual design</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>Spencer95</td>
<td>manual composition</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>Mitchell02</td>
<td>manual design</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td><strong>Our method</strong></td>
<td>physically based generation</td>
<td>yes</td>
<td>yes</td>
</tr>
</tbody>
</table>
Our method

• Automatic generation of glare with the input of the sources of diffraction

• Glare computation based on diffraction theory in wave optics – more physically based

• Easy to get wide variety of glare by changing the input images
• Introduction
• Previous Work

• Fraunhofer diffraction
• Simulation of Glare and its Implementation
• Integration to Real-time Image Generation
• Conclusion and Future Work
What is Fraunhofer diffraction?

• A fundamental theory in wave optics
• An approximation of diffraction light distribution through lens onto a screen at the focal position
Equation of complex amplitude distribution

• Derived from Huygens’ principle and the wave equation

\[ t_o(x_o, y_o) \]

complex amplitude distribution

\[ U_f(x_f, y_f) = \frac{A}{j\lambda f} \exp \left[ j \frac{\pi}{\lambda} \left( 1 - \frac{d_o}{f} \right) (x_f^2 + y_f^2) \right] \]

light source amplitude

unit imaginary number

light wave length

\[ \times \int \int_{-\infty}^{\infty} t_o(x_o, y_o) \exp \left[ -j \frac{2\pi}{\lambda f} (x_o x_f + y_o y_f) \right] dx_o dy_o \]

[Goodman1968] *Introduction to Fourier Optics*
Factors of the amplitude distribution

- Represented using Fourier transform

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

- Square of absolute value of the amplitude

\[
I_f(x_f, y_f) = \left| U_f(x_f, y_f) \right|^2
\]

\[
= \frac{A^2}{\lambda^2 f^2} \left| \int \int_{-\infty}^{\infty} t_o(x_o, y_o) \exp \left[ -j \frac{2\pi}{\lambda f} \left(x_o x_f + y_o y_f\right) \right] dx_o dy_o \right|^2
\]

The \(e^{j\omega}\) factor (phase component) diminishes to 1.

\[
\omega = \frac{\pi}{\lambda} \left(1 - \frac{d_o}{f}\right)(x_f^2 + y_f^2)
\]
Luminance intensity distribution

- Square of absolute value of the amplitude

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

2D Fourier transform

Easy to implement using FFT
• Introduction
• Previous Work
• Fraunhofer Diffraction
• **Simulation of Glare and its Implementation**
• Integration to Real-time Image Generation
• Conclusion and Future Work
• The luminance intensity distribution \( I_f(x_f, y_f) \) is an HDR (High Dynamic Range) image

  - How to map the value to 8bit frame buffer value?

\[
t_o(x_o, y_o) \quad I_f(x_f, y_f) \quad ??? \quad I_p(x_p, y_p)
\]
Tone mapping function

- [Tumblin1999, ACM TOG] using sigmoid function

$$I_p(x_p, y_p) = \text{sig}\{I_f(x_p, y_p)\},$$

where

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

- \(g\) : defines slope at \(x = 1\)
- \(k\) : defines maximum contrast
- \(D\) : defines maximum luminance

Implementation
Tone mapping summary

• glare image =

\[
\text{toneMapping}\{\ \text{FFT(obstacle image)}\ \}
\]
Spectrum effect

• Composition of the results for multiple wave lengths

\[ I_f(x_f, y_f, \lambda) \]

\[ I_p(x_p, y_p) \]
Spectrum effect

- glare image = toneMapping

\[ \left\{ \sum_{\lambda} \text{FFT}(\lambda, \text{obstacle image}) \right\} \]
Aperture effect

- Employing a Pupil Function

\[
P \left( x_o + \frac{d_o}{f} x_f, y_o + \frac{d_o}{f} y_f \right)
\]

- Incoming light (plane wave)
- Obstacle
- Aperture
- Lens
- Screen (focal position)

\[
t_o(x_o, y_o) \quad I_f(x_f, y_f)
\]

\[
d_o \quad f
\]
Aperture effect

- An approximation

\[ t_o(x_o, y_o) P(x_o, y_o) \]

Implementation

Incoming light (plane wave) \( d_o \)

Screen (focal position) \( f \)
Implementation results

- Image generation using FFT, spectrum and tone mapping

\[ t_o(x_o, y_o) \quad P(x_o, y_o) \quad I_p(x_p, y_p) \]

- eyelashes and the upper eyelid
- iris shape
- Output glare image
Implementation results

- Image generation using FFT, spectrum and tone mapping

\[ t_o(x_o, y_o) \quad P(x_o, y_o) \quad I_p(x_p, y_p) \]

A cross filter of a camera

A diaphragm on the lens

Output glare image
Implementation results

- Image generation using FFT, spectrum and tone mapping

\[ t_o(x_o, y_o) \quad P(x_o, y_o) \quad I_p(x_p, y_p) \]

No filter

Trigonal diaphragms

Output glare image
Implementation results

• Overlaying glare images onto an HDR (High Dynamic Range) image

Regular range

High intensity range

Result
• Introduction
• Previous Work
• Fraunhofer diffraction
• Simulation of Glare and its Implementation
• Integration to Real-time Image Generation
• Conclusion and Future Work
Integration to real-time rendering

• Interactive operation of 3D models makes glare move along the reflective surfaces

A snapshot from a real-time interactive demo
Multi-pass rendering flow

---

3D model

Scene image

Scene image (high intensity pixels)

Obstacle image

Pupil funcs

FFT

Glare images

Billboards

Texture mapping

Intermediate data (texture memory)

Integration

---

Scene image with glare

Final result for a frame

---

Pre-computation

---

glReadPixels

Pixel position detection

Overlay
Multi-pass rendering flow

3D model

Environment map

Scene image

Image (high intensity pixels)

HeadPixels

Pixel position detection

Pupil funcs

Obstacle image

FFT

Glare images

Billboards

Texture mapping

Intermediate data (texture memory)

Scene image with glare

Final result for a frame

Integration

Pre-computation

Integration

glReadPixels

Environment map

Billboards

Texture mapping
Multi-pass rendering flow

Integration

Pre-computation

Scene image

3D model

Scene image (high intensity pixels)

Light map

Obstacle image

Pupil funcs

FFT

Glare images

Billboards

Texture mapping

Intermediate data (texture memory)

Scene image with glare

Final result for a frame
Multi-pass rendering flow

3D model

Scene image

Scene image (high intensity pixels)

glReadPixels

Pixel position detection

Pupil funcs

Obstacle image

FFT

Glare images

Billboards

Overlay

Scene image with glare

Final result for a frame

Integration

Pre-computation

Intermediate data (texture memory)
Multi-pass rendering flow

1. **3D model**
   - Scene image
     - glReadPixels
     - Pixel position detection
     - Scene image with glare

2. **Pre-computation**
   - Obstacle image
     - FFT
     - Glare images
     - Texture mapping
     - Intermediate data (texture memory)

3. **Integration**
   - Final result for a frame
Dynamic glare

- Moving glare changes shape

An example position of the high intensity pixels detected on the fly

Selected input obstacle image and pupil window function

\[ t_o(x_o, y_o) \ast P(x_o, y_o) \]

Output glare image

\[ I_p(x_f, y_f) \]
Dynamic glare

• Moving glare changes shape

An example position of the high intensity pixels detected on the fly

Selected input obstacle image and pupil window function

\[ t_o(x_o, y_o) \ast P(x_o, y_o) \]

Output glare image

\[ I_p(x_f, y_f) \]
Dynamic glare

- Moving glare changes shape

An example position of the high intensity pixels detected on the fly

Selected input obstacle image and pupil window function

\[ t_o(x_o, y_o) * P(x_o, y_o) \]

Output glare image

\[ I_p(x_f, y_f) \]
Integration to real-time rendering

- Demonstrations (recorded real-time)
  - SGI Onyx3400 InfiniteReality4
  - 1GB texture memory
Finally ...

- Introduction
- Previous Work
- Fraunhofer Diffraction
- Simulation of Glare and its Implementation
- Integration to Real-time Image Generation

- Conclusion and Future Work
Conclusion

• Established a glare simulation model based on Fraunhofer diffraction

• Implemented using FFT and a Tone mapping technique
  – Enabled users to design glare using obstacle and pupil image as inputs

• Integrated the result into real-time rendering loop
Future Work

• More accurate simulation model
  – Smoother spectrum effect (more sampling of $\lambda$)
  – 3D position of obstacles and aperture

• Optimization of performance
  – Especially when number of high intensity pixels is big

• Beyond a mere special effect
  – Application to ophthalmology or engineering
    Postoperative simulation
    Automobile headlamp evaluation
Q&A