# Virtual Spherical Gaussian Lights for Real-time Glossy Indirect Illumination

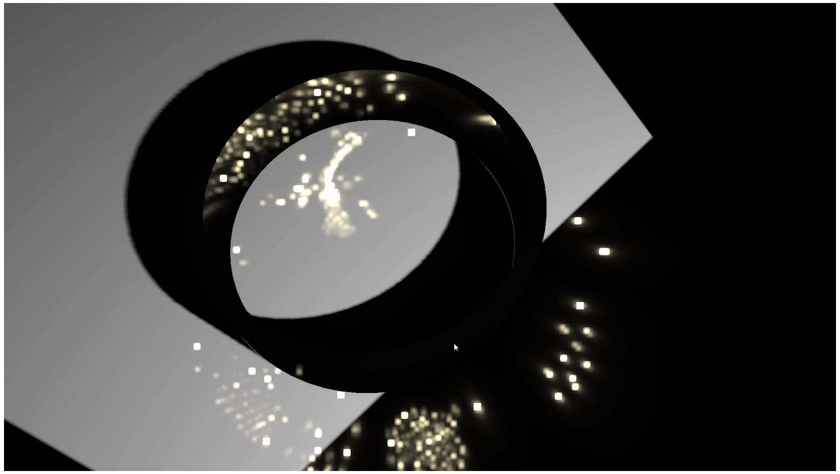
Yusuke Tokuyoshi

Square Enix Co., Ltd.

PACIFIC GRAPHICS 2015

# Limitations of Virtual Point Lights (VPLs) [Keller 97]

- Spiky artifacts
  - VPL singularities
- Flickering
  - Discontinuous geometry
  - Textures
  - Normal maps
  - Glossy BRDFs



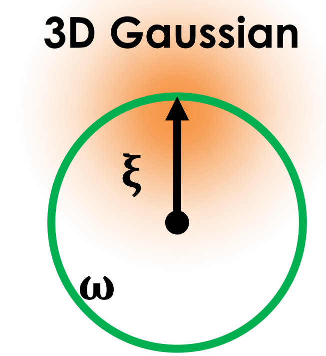
# Spherical Gaussians (SGs)

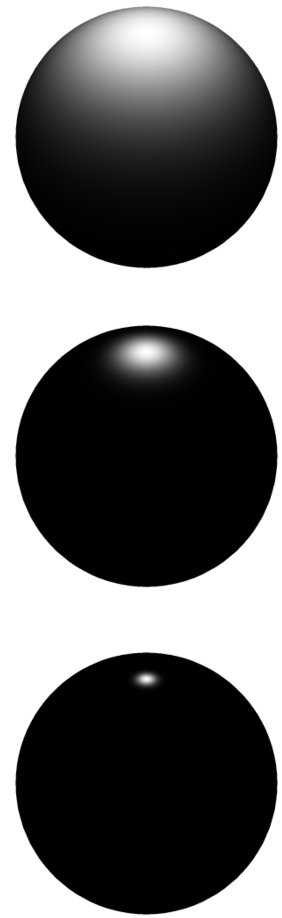
$$G(\boldsymbol{\omega}, \boldsymbol{\xi}, \lambda) = g\left( \|\boldsymbol{\omega} - \boldsymbol{\xi}\|, \frac{1}{\lambda} \right)$$

Gaussian function

lobe axis (mean)

lobe sharpness (inverse of variance)



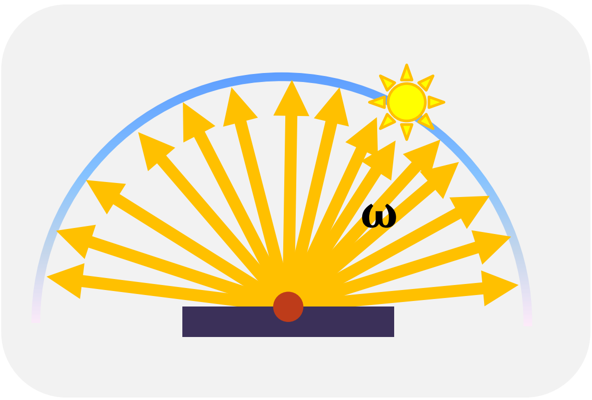



- Gaussian function on a unit sphere
- Can represent all-frequency signals
  - All-frequency lighting

# Rendering Using Spherical Gaussians

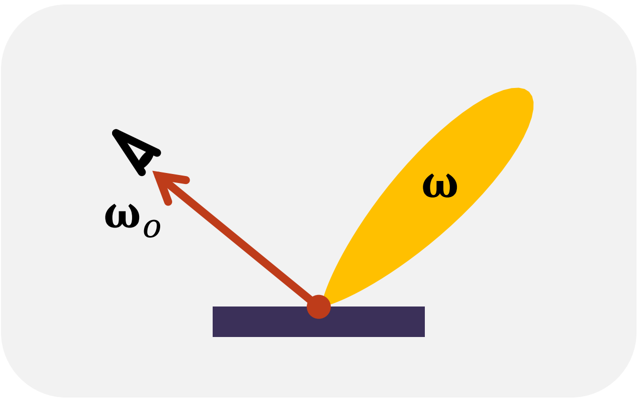
- Closed-form solutions for SGs
  - **Product**, **integral**, and **product integral**
  - Used for approximating the rendering equation

$$L(\mathbf{x}_p, \boldsymbol{\omega}_o) = \int_{S^2}$$




**Incoming radiance**

$\times$




**Reflection lobes**

$\mathrm{d}\boldsymbol{\omega}$

# Rendering Using Spherical Gaussians

- Closed-form solutions for SGs
  - **Product**, **integral**, and **product integral**
  - Used for approximating the rendering equation

$$L(\mathbf{x}_p, \boldsymbol{\omega}_o) = \int_{\mathbb{S}^2} \underbrace{\sum_i^N c_i G_i(\boldsymbol{\omega})}_{\textbf{Incoming radiance}} \times \underbrace{\sum_j^M c_j G_j(\boldsymbol{\omega})}_{\textbf{Reflection lobes}} \, \mathrm{d}\boldsymbol{\omega}$$

# Rendering Using Spherical Gaussians

- Closed-form solutions for SGs
  - **Product**, **integral**, and **product integral**
  - Used for approximating the rendering equation

$$L(\mathbf{x}_p, \boldsymbol{\omega}_o) = \int_{S^2} \underbrace{\sum_{i}^{N} c_i G_i(\boldsymbol{\omega})}_{\textbf{Incoming radiance}} \times \underbrace{\sum_{j}^{M} c_j G_j(\boldsymbol{\omega})}_{\textbf{Reflection lobes}} \, \mathrm{d}\boldsymbol{\omega}$$

**Incoming radiance**

**Reflection lobes**

[Wang et al. 09]
[Xu et al. 13]

# Rendering Using Spherical Gaussians

- Closed-form solutions for SGs
  - **Product**, **integral**, and **product integral**
  - Used for approximating the rendering equation

$$L(\mathbf{x}_p, \boldsymbol{\omega}_o) = \int_{S^2} \sum_{i}^{N} c_i G_i(\boldsymbol{\omega}) \times \sum_{j}^{M} c_j G_j(\boldsymbol{\omega}) \, \mathrm{d}\boldsymbol{\omega}$$

**Incoming radiance** **Reflection lobes**

[Wang et al. 09]
[Xu et al. 13]

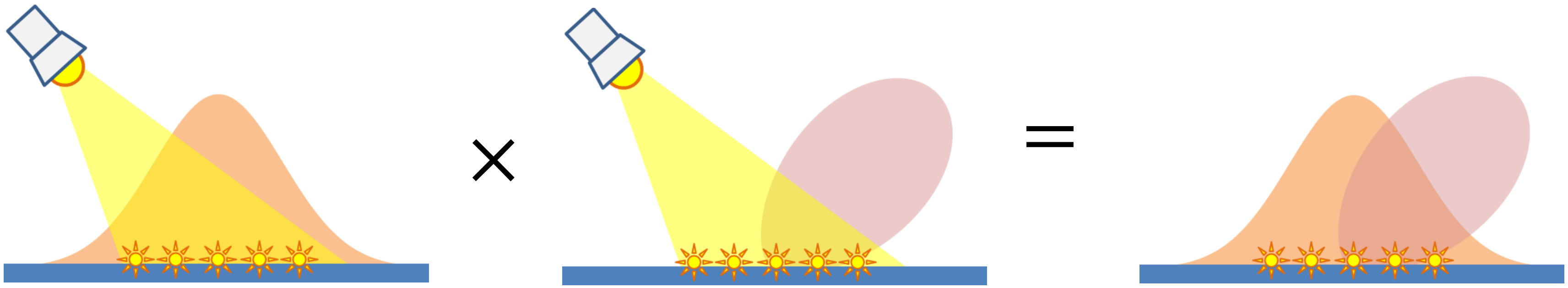
# SG Approximation of Incoming Radiance

- **[Xu et al. 14]**
  - High quality
  - Near-interactive frame rate
  - Restricted to distant light sources

- **Proposed method**
  - Lower quality than [Xu et al. 14]
  - Real-time frame rate (20-30 ms)
  - Directional/point/spot light sources
  - Simpler

# Virtual Spherical Gaussian Lights (VSGLs)

- Rough approximation of a set of VPLs



**Distribution of VPL positions (isotropic Gaussian)**

mean & variance

**Total radiant intensity (spherical Gaussians)**

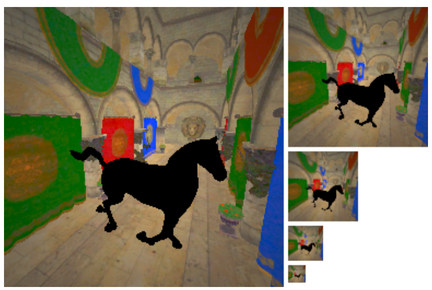
Toksvig filter [Toksvig 05]

**Emitted radiance**

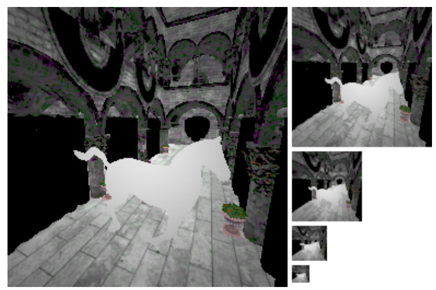
Generated by an averaging operation (mipmap)

# VSGL Generation Using Mipmaps

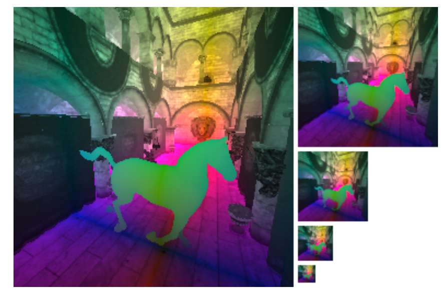
- Mipmapping of reflective shadow maps
- Filtered importance sampling [Krivanek and Colbert 08]



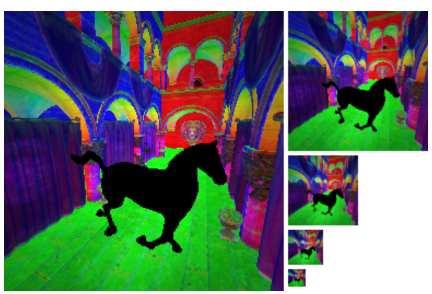
Diffuse weight



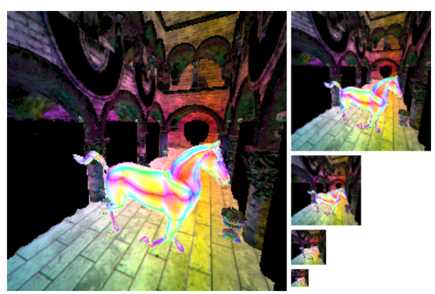
Specular weight



Weighted position



Weighted avg. direction of diffuse reflection

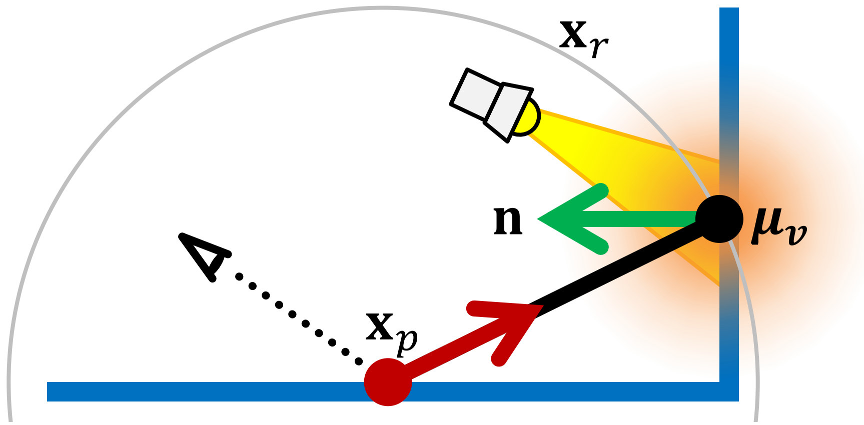


Weighted avg. direction of specular reflection

Weighted squared norm of the position

# Incoming Radiance from a VSGL

- Incoming radiance = emitted radiance of a VSGL




$$L_{in}(\mathbf{x}_p, \boldsymbol{\omega}) \approx \frac{I_v(-\boldsymbol{\omega})}{2\pi\sigma_v^2 |\boldsymbol{\omega} \cdot \mathbf{n}|} g(\|\mathbf{x} - \boldsymbol{\mu}_v\|, \sigma_v^2)$$

Unknown reflector position $\mathbf{x}$
Unknown reflector normal $\mathbf{n}$

$$\approx \frac{I_v(-\boldsymbol{\omega})\cancel{|\boldsymbol{\omega} \cdot \mathbf{n}|}}{2\pi\sigma_v^2 \cancel{|\boldsymbol{\omega} \cdot \mathbf{n}|}} \underbrace{g(\|\mathbf{x}_r - \boldsymbol{\mu}_v\|, \sigma_v^2)}_{\text{Gaussian on a sphere}} = \frac{I_v(-\boldsymbol{\omega})}{2\pi\sigma_v^2} \underbrace{G\left(\boldsymbol{\omega}, \frac{\boldsymbol{\mu}_v - \mathbf{x}_p}{\|\boldsymbol{\mu}_v - \mathbf{x}_p\|}, \frac{\|\boldsymbol{\mu}_v - \mathbf{x}_p\|^2}{\sigma_v^2}\right)}_{\text{Spherical Gaussian}}$$

# Incoming Radiance from a VSGL

$$L_{in}(\mathbf{x}_p, \boldsymbol{\omega}) \approx \frac{I_v(-\boldsymbol{\omega})}{2\pi\sigma_v^2} G\left(\boldsymbol{\omega}, \frac{\boldsymbol{\mu}_v - \mathbf{x}_p}{\left\|\boldsymbol{\mu}_v - \mathbf{x}_p\right\|}, \frac{\left\|\boldsymbol{\mu}_v - \mathbf{x}_p\right\|^2}{\sigma_v^2}\right)$$

$$\approx \frac{c_v}{2\pi\sigma_v^2} \underline{G(\boldsymbol{\omega}, -\boldsymbol{\xi}_v, \lambda_v)} G\left(\boldsymbol{\omega}, \frac{\boldsymbol{\mu}_v - \mathbf{x}_p}{\left\|\boldsymbol{\mu}_v - \mathbf{x}_p\right\|}, \frac{\left\|\boldsymbol{\mu}_v - \mathbf{x}_p\right\|^2}{\sigma_v^2}\right) = c_{in} G(\boldsymbol{\omega}, \boldsymbol{\xi}_{in}, \lambda_{in})$$

$$L(\mathbf{x}_p, \boldsymbol{\omega}_o) = \int_{S^2} \sum_i^N c_i G_i(\boldsymbol{\omega}) \times \sum_j^M c_j G_j(\boldsymbol{\omega}) \, \mathrm{d}\boldsymbol{\omega}$$

**Incoming radiance** **Reflection lobes**

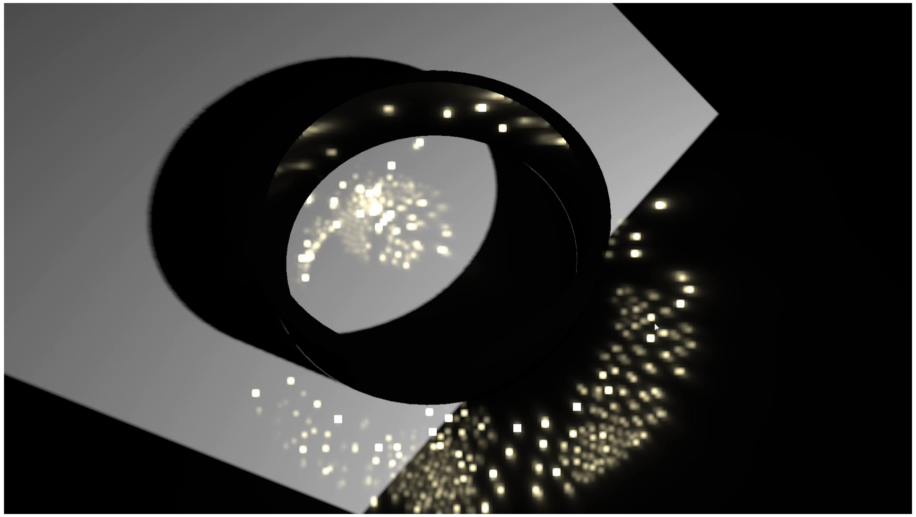
# Results

1024 VPLs
(23 ms)

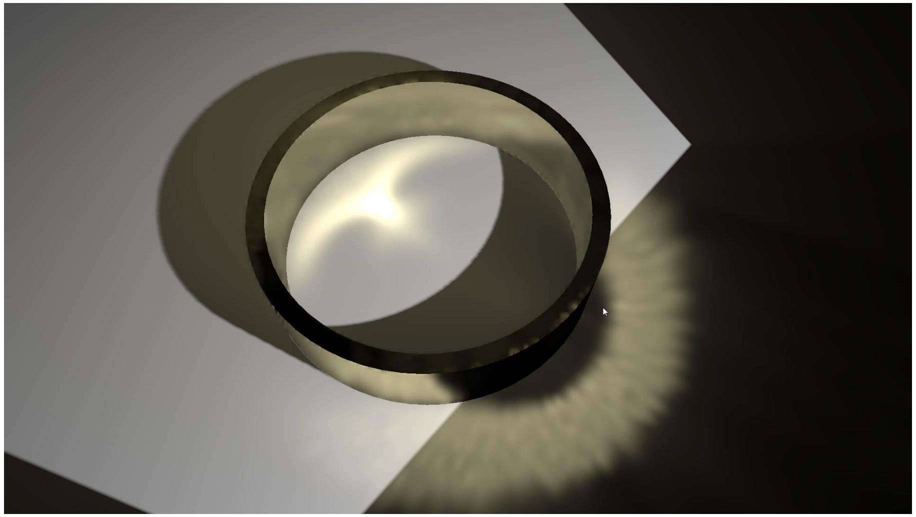
1024 VSGLs
(26 ms)

Resolution: 1920×1088, GPU: Radeon R9 290X

# Results



1024 VPLs
(21 ms)



1024 VSGLs
(23 ms)

Resolution: 1920×1088, GPU: Radeon R9 290X

# Limitations

- Light leaks
  - Rough approximation for real-time rendering
- Flickering
  - Can be reduced but cannot be avoided completely
  - Our implementation of VSGL sampling is not sophisticated enough yet

# ~~Future Work~~ Work in Progress (1)

- Temporal coherence
  - Investigate sampling methods for VSGLs

Improved VSGL sampling

# ~~Future Work~~ Work in Progress (2)

- Specialization for more time-sensitive applications (e.g., video games)


2 VSGLs (0.7 ms)

(to appear in SIGGRAPH ASIA 2015 Posters)

# Conclusion

- VSGL = rough approximation of a set of VPLs
  - VPL positions: Gaussian distribution
  - Radiant intensity: Spherical Gaussians
- Dynamic glossy interreflections and caustics
- Simple solution for real-time rendering
  - Over 30 FPS

Thank you for your attention

PACIFIC GRAPHICS 2015