

Frequency Analysis and Dual Hierarchy for Efficient Rendering of Subsurface Scattering

David Milaenen
Weta Digital
Université de Montréal

Laurent Belcour
Unity Labs
Université de Montréal

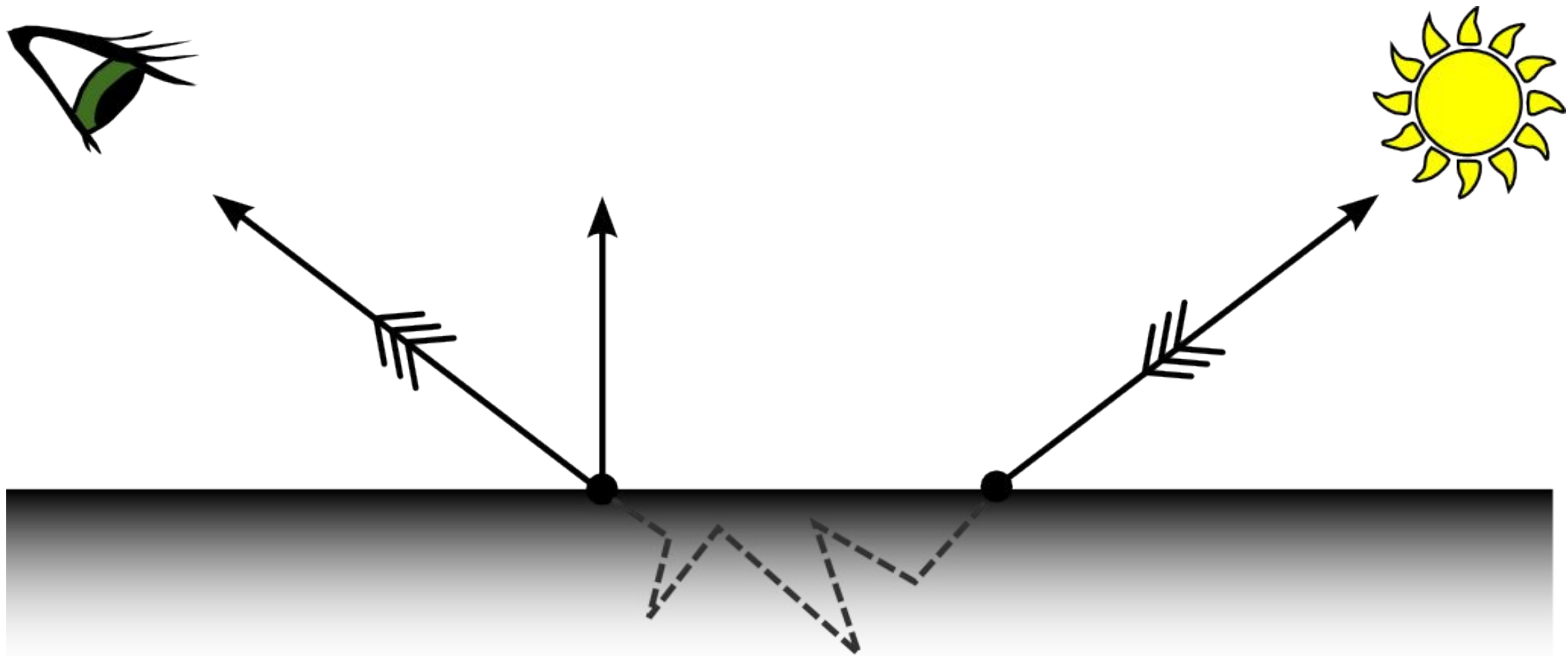
Jean-Philippe Guertin
Université de Montréal

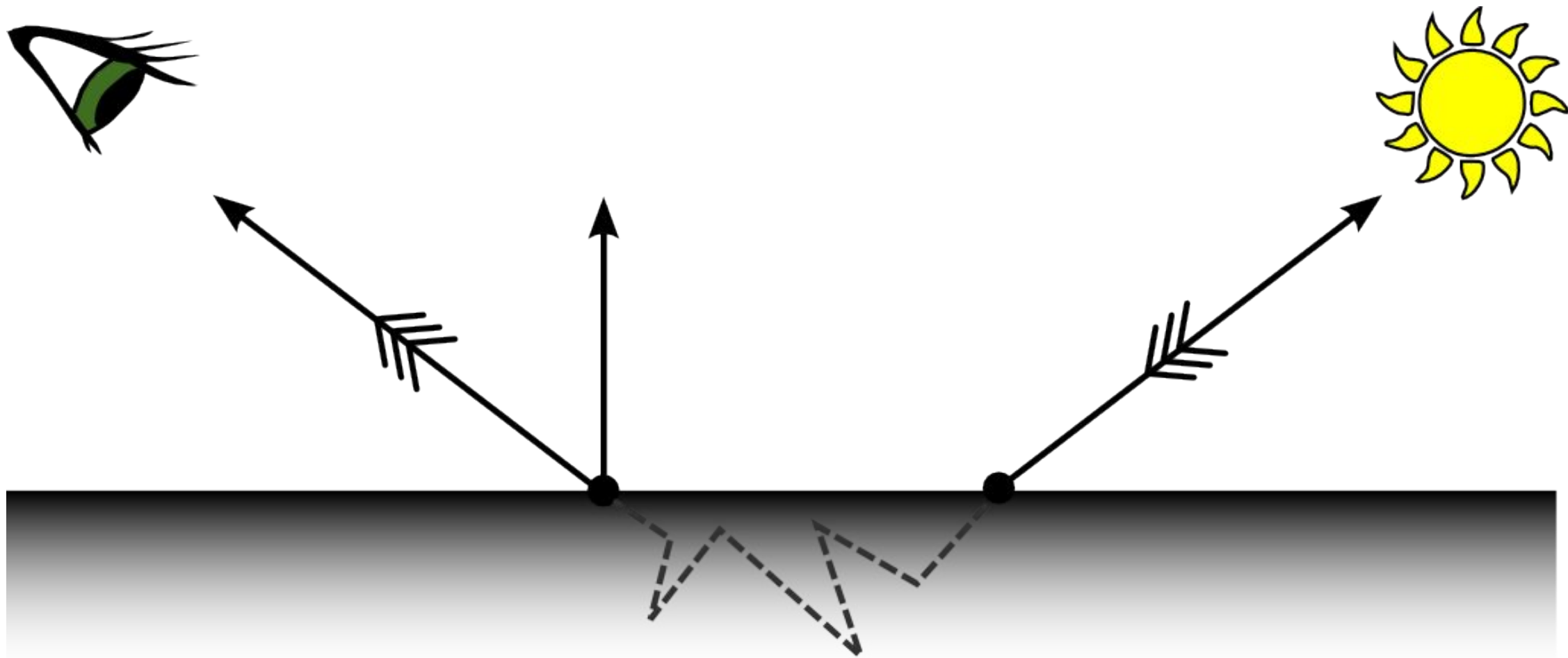
Toshiya Hachisuka
University of Tokyo

Derek Nowrouzezahrai
McGill University
Université de Montréal

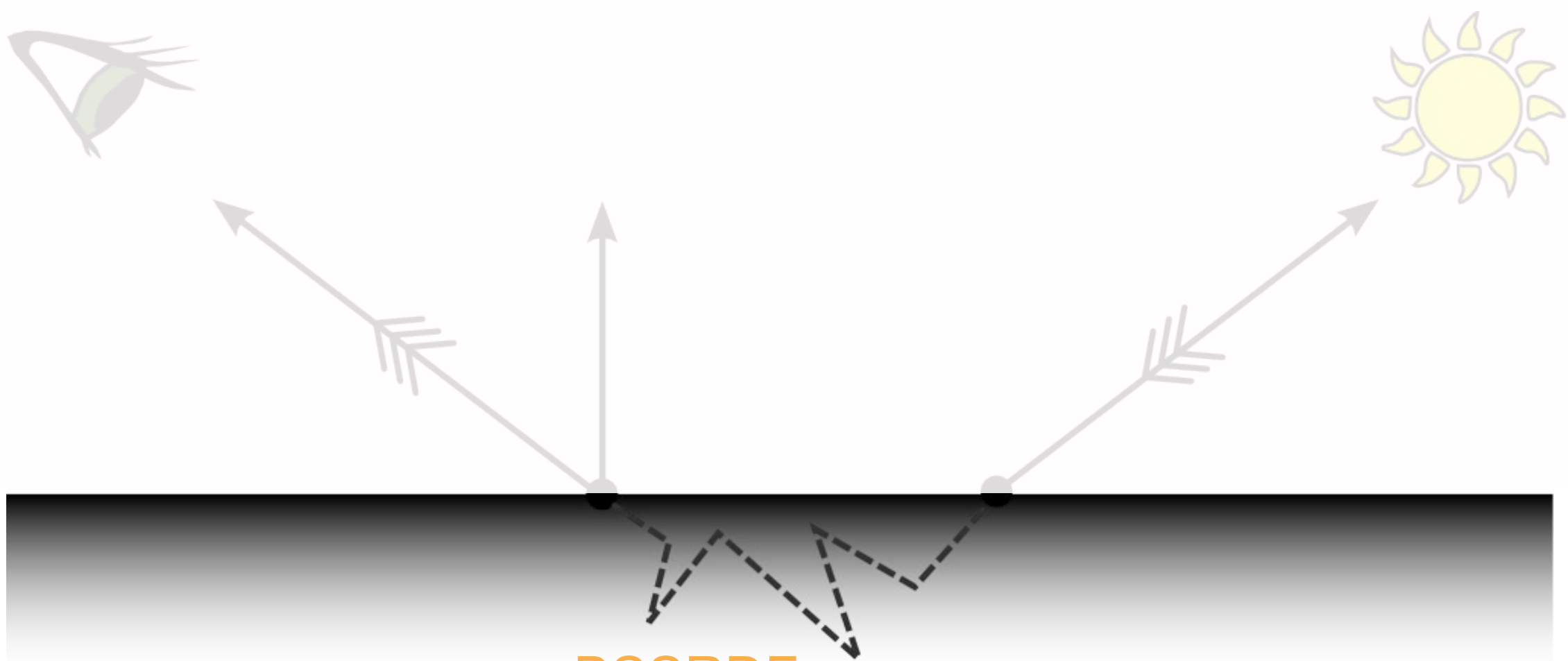








$$L_o(x_o, \omega_o) = \int_A \int_{2\pi} S(x_i, \omega_i, x_o, \omega_o) L_i(x_i, \omega_i) (n \cdot \omega_i) d\omega_i dA(x_i)$$



BSSRDF

$$L_o(x_o, \omega_o) = \int_A \int_{2\pi} \boxed{S(x_i, \omega_i, x_o, \omega_o)} L_i(x_i, \omega_i) (n \cdot \omega_i) d\omega_i dA(x_i)$$



Classical Dipole [Jensen et al. 2001]



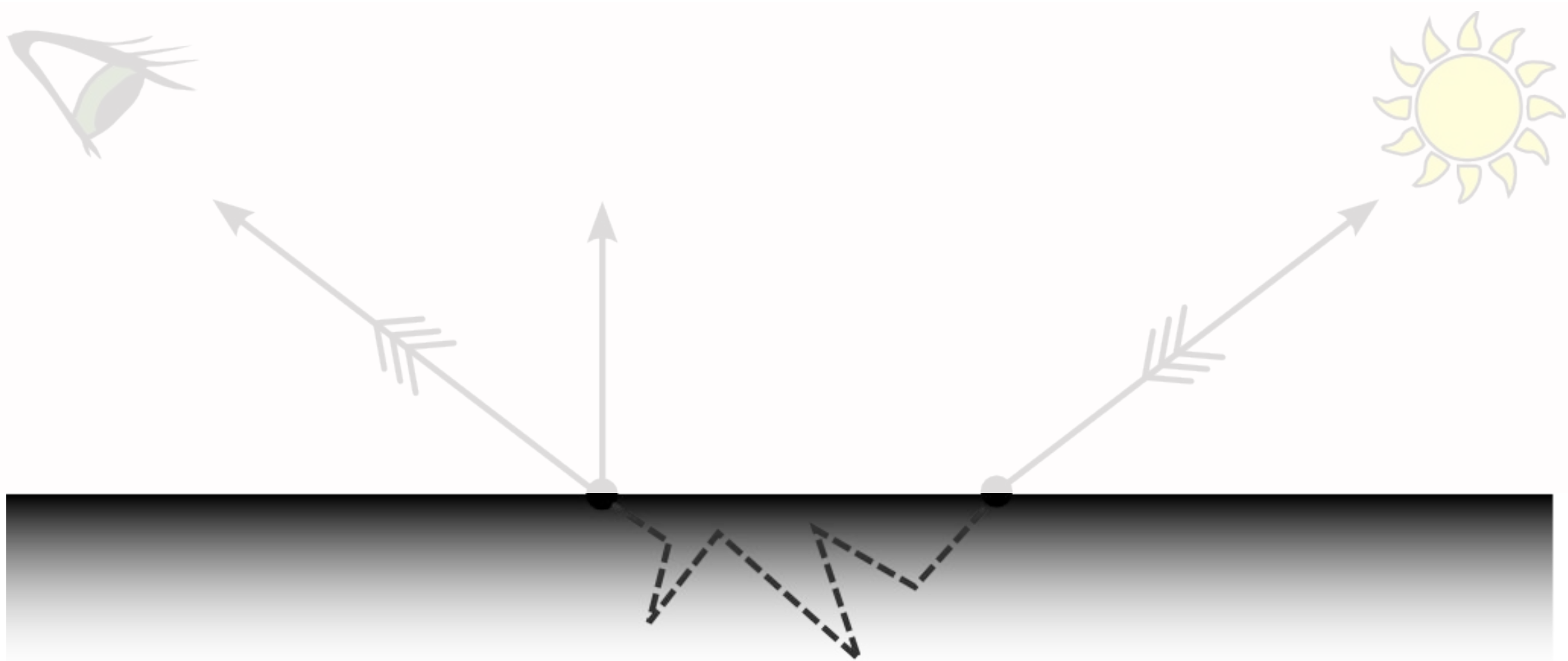
Directional Dipole [Frisvad et al. 2014]



Beam Diffusion [Habel et al. 2013]

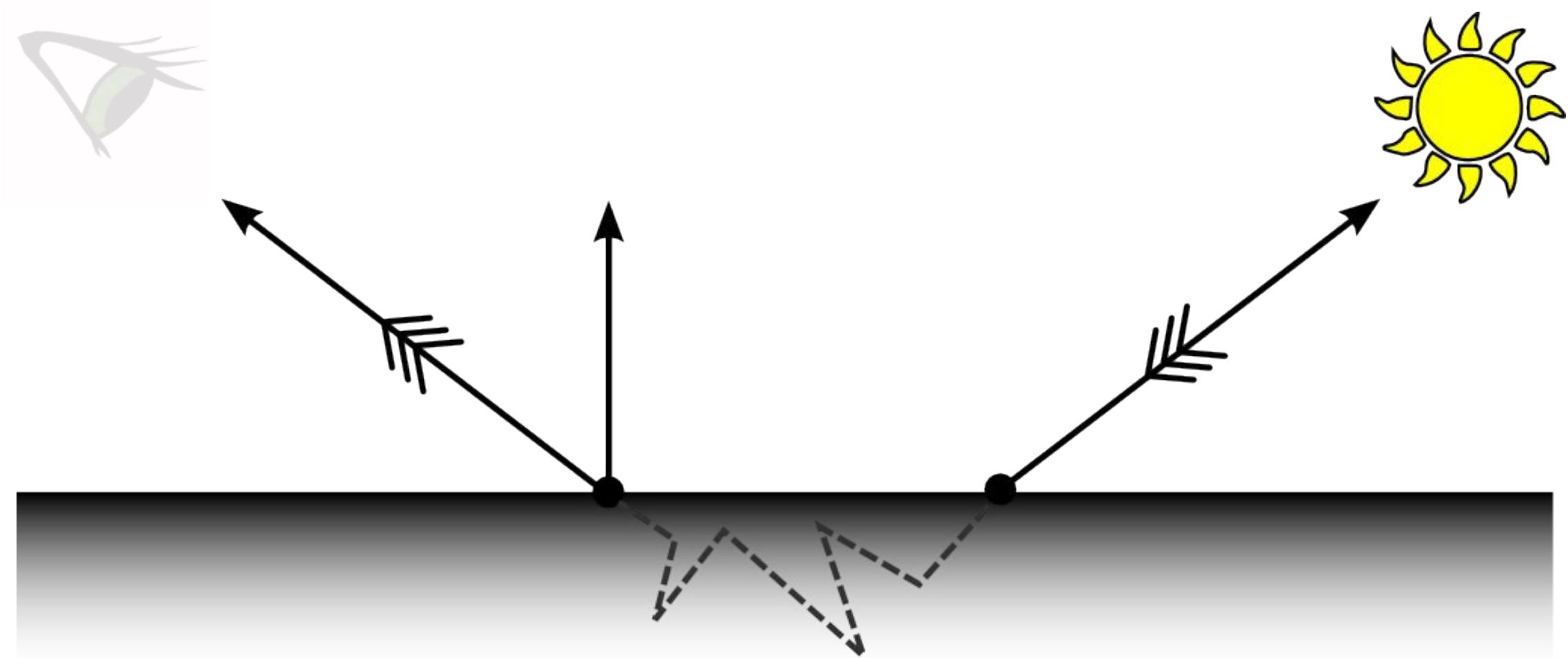


Quantized-Diffusion [d'Eon and Irving 2011]

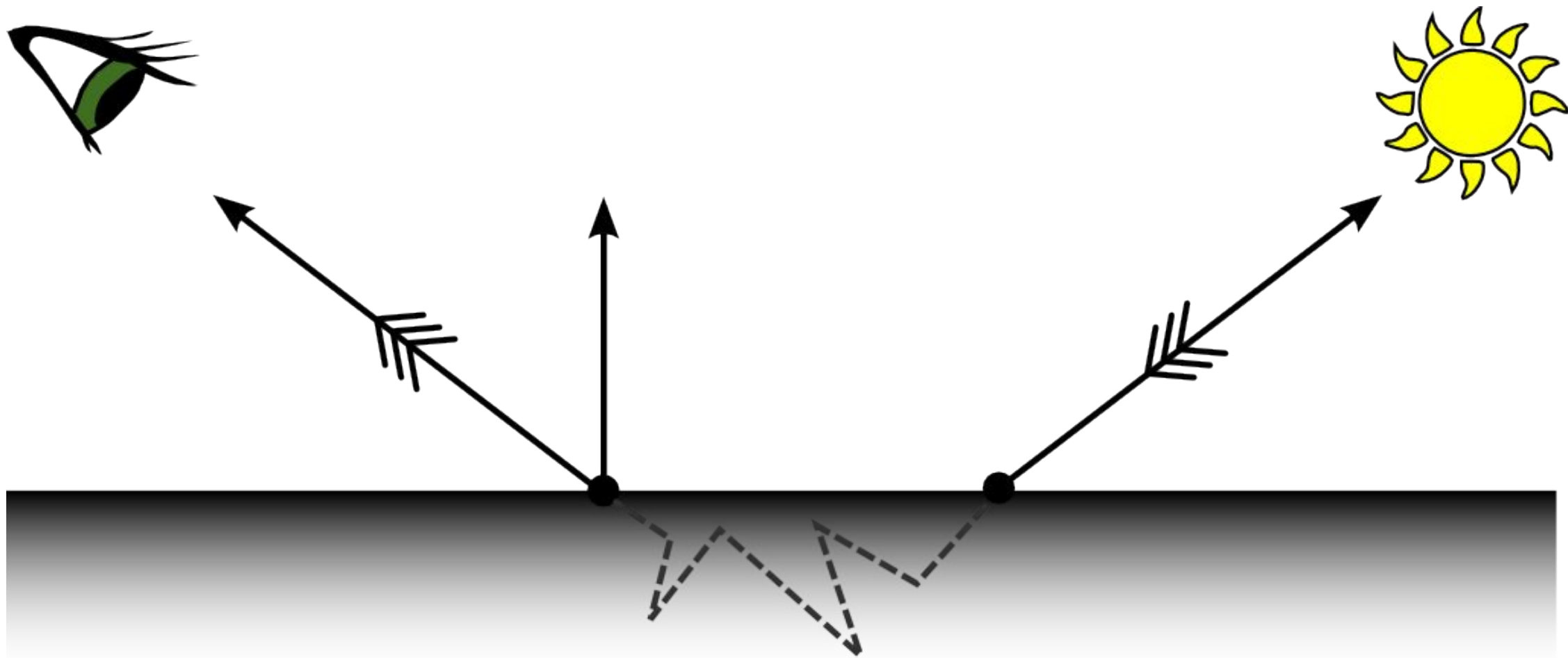


$$L_o(x_o, \omega_o) = \int_A \int_{2\pi} S(x_i, \omega_i, x_o, \omega_o) L_i(x_i, \omega_i) (n \cdot \omega_i) d\omega_i dA(x_i)$$

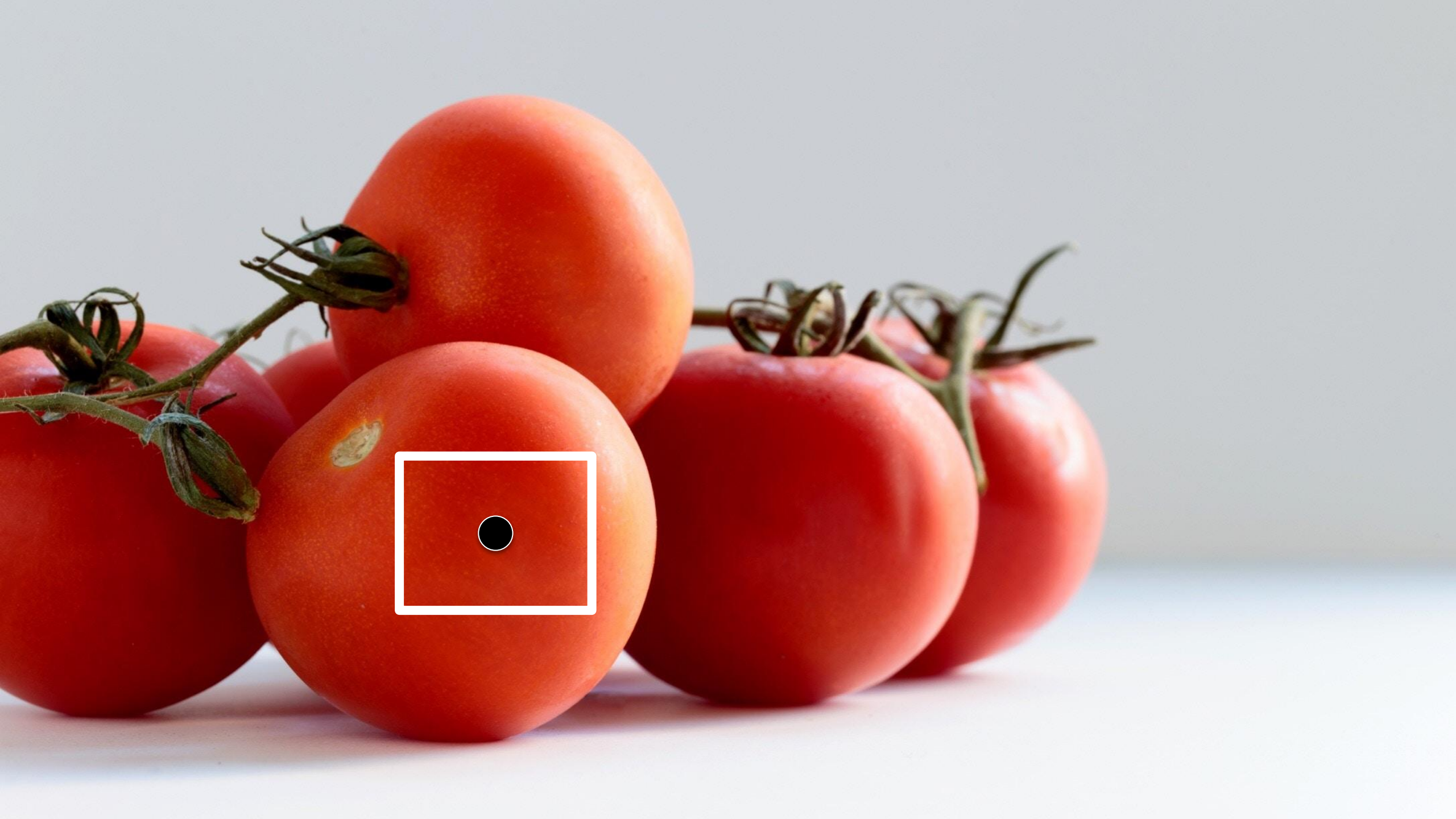


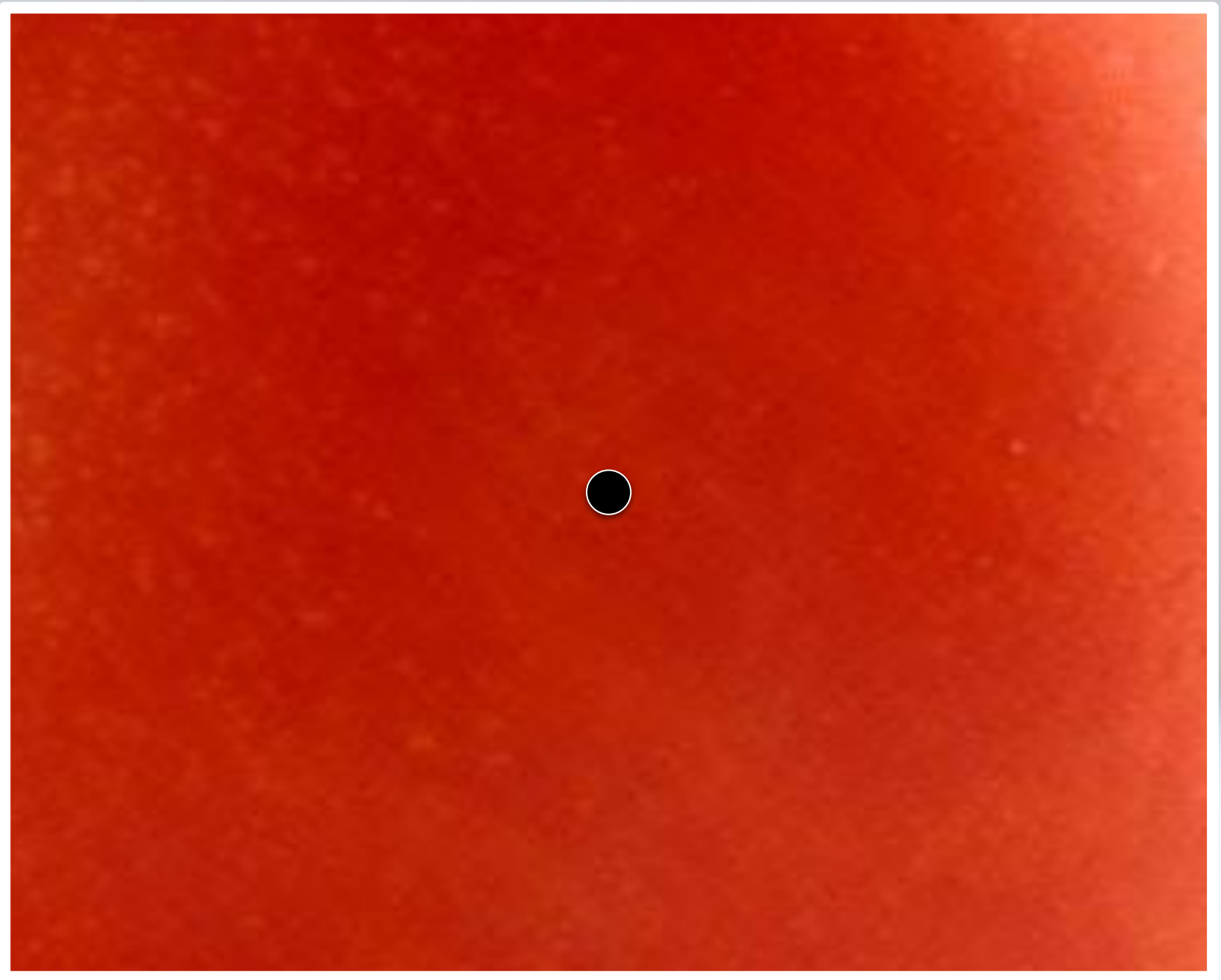


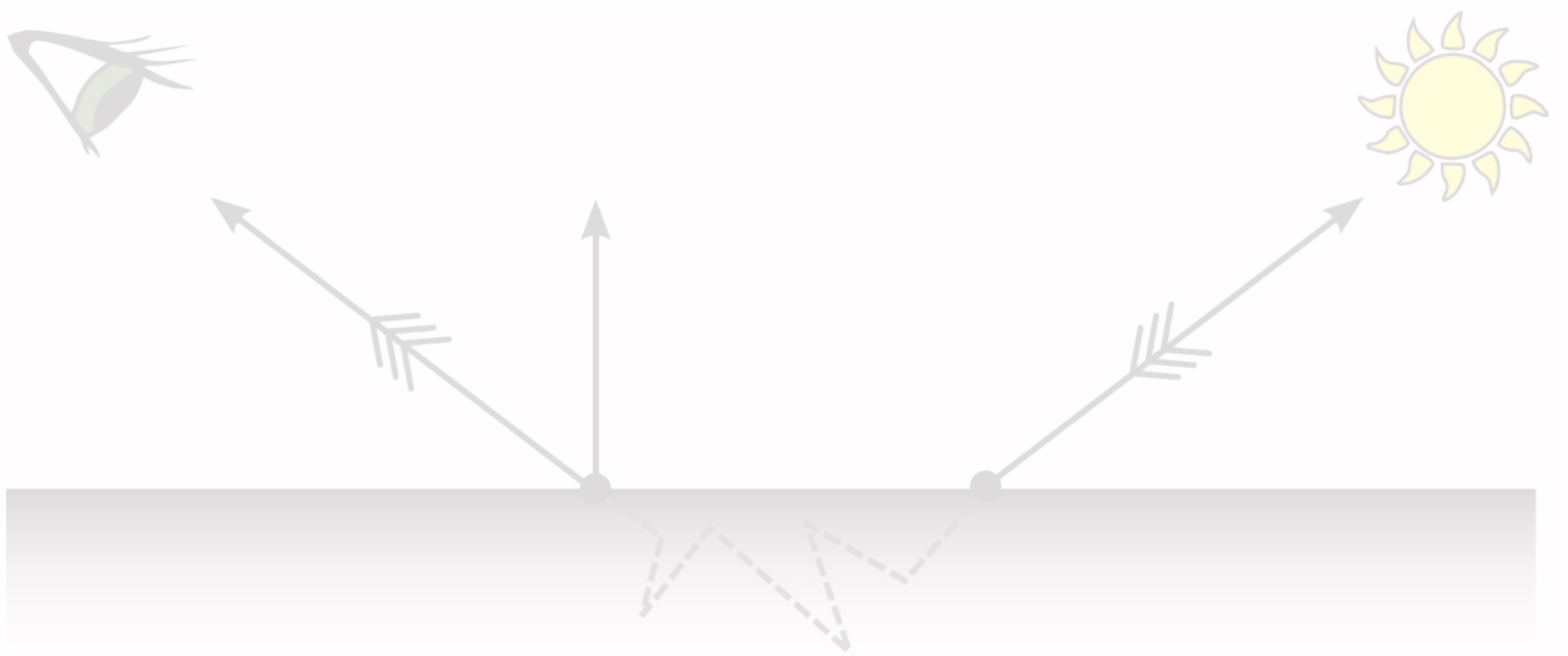
$$L_o(x_o, \omega_o) = \int_A \int_{2\pi} S(x_i, \omega_i, x_o, \omega_o) L_i(x_i, \omega_i) (n \cdot \omega_i) d\omega_i dA(x_i)$$



$$L_o(x_o, \omega_o) = \int_A \int_{2\pi} S(x_i, \omega_i, x_o, \omega_o) L_i(x_i, \omega_i) (n \cdot \omega_i) d\omega_i dA(x_i)$$

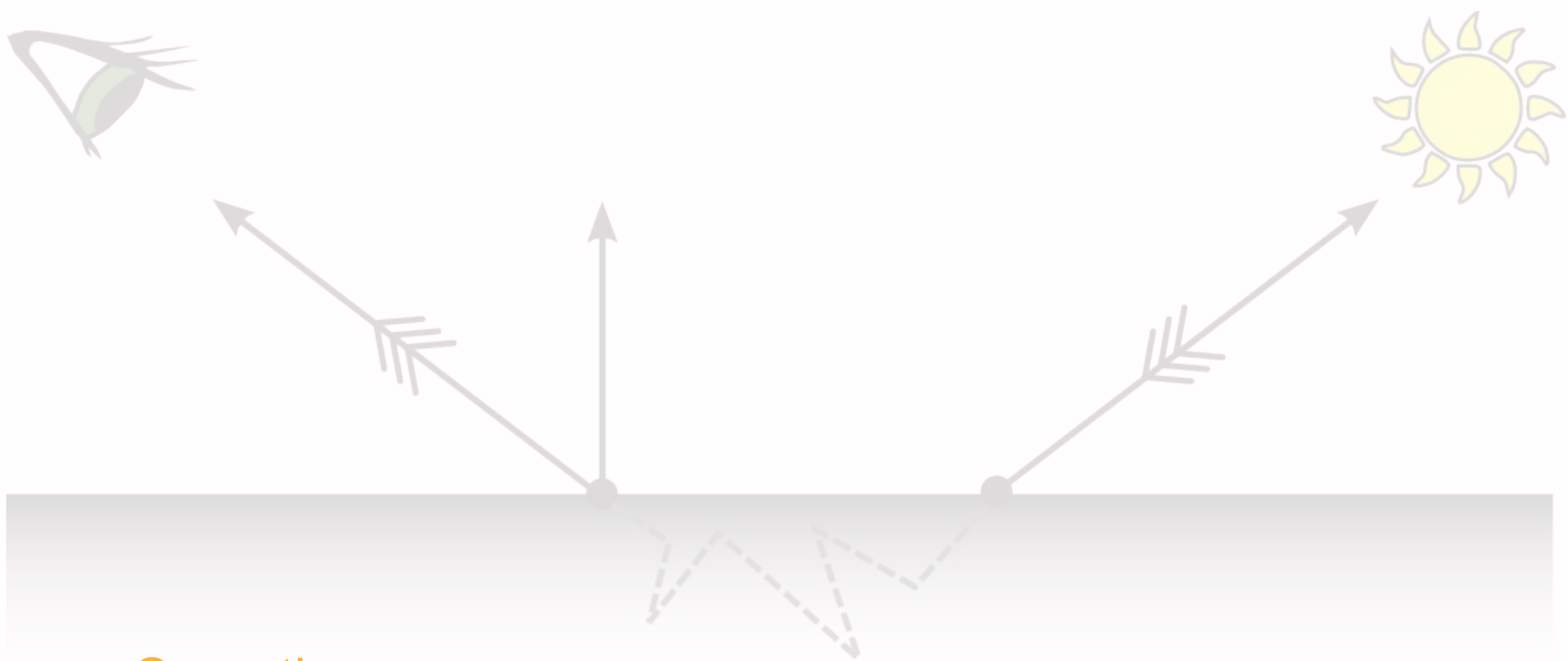






$$L_o(x_o, \omega_o) = \int_A \int_{2\pi} S(x_i, \omega_i, x_o, \omega_o) L_i(x_i, \omega_i) (n \cdot \omega_i) d\omega_i dA(x_i)$$





Smooth

$$L_o(x_o, \omega_o) = \int_A \int_{2\pi} S(x_i, \omega_i, x_o, \omega_o) L_i(x_i, \omega_i) (n \cdot \omega_i) d\omega_i dA(x_i)$$

➤ Observations

- Costly double integral per pixel
 - Directional integral (just like BSDFs)
 - Spatial integral (specific to BSSRDFs)
- Result is often smooth
 - Individually solving each pixel is **wasteful**

➤ Observations

- Costly double integral per pixel
 - Directional integral (just like BSDFs)
 - Spatial integral (specific to BSSRDFs)
- Result is often smooth
 - Individually solving each pixel is **wasteful**

$$L_o(x_o, \omega_o) = \int_A \int_{2\pi} S(x_i, \omega_i, x_o, \omega_o) L_i(x_i, \omega_i) (n \cdot \omega_i) d\omega_i dA(x_i)$$

$$L_o(x_o, \omega_o) = \int_A \int_{2\pi} I_{x_o, \omega_o}(x_i, \omega_i) d\omega_i dA(x_i)$$

$$L_o(x_o, \omega_o) = \int_A \int_{2\pi} I_{x_o, \omega_o}(x_i, \omega_i) d\omega_i dA(x_i)$$

Illumination function

$$L_o(x_o, \omega_o) \approx \frac{1}{N} \sum_{k=1}^N I_{x_o, \omega_o}(x_{i,k}, \omega_{i,k})$$

Illumination sample

$$L_o(x_o, \omega_o) \approx \frac{1}{N} \sum_{k=1}^N I_{x_o, \omega_o}(x_{i,k}, \omega_{i,k})$$

Illumination sample

Cost: $\mathcal{O}(N)$

➤ Clustered integration

- **Hierarchical** [Jensen and Buhler 2002]
 - **Vectors** [Frisvad et al. 2014]
 - **Radiance bin** [d'Eon and Irving 2011]
- **Multidimensional Lightcuts** [Walter et al. 2006]
- **Illumination cuts** [Bus et al. 2015]
- **Double hierarchy**
 - **Micropolygon grid** [d'Eon and Irving 2011, PRMan]
 - **GI VPL clustering** [Jarabo et al. 2015]

➤ Clustered integration

$$L_o(x_o, \omega_o) \approx \frac{1}{N} \sum_{k=1}^N I_{x_o, \omega_o}(x_{i,k}, \omega_{i,k})$$

➤ Clustered integration

$$L_o(x_o, \omega_o) \approx \frac{1}{M} \sum_{k=1}^M C_{x_o, \omega_o}(x_{i,k}, \omega_{i,k})$$

Clustered sample

➤ Clustered integration

$$L_o(x_o, \omega_o) \approx \frac{1}{M} \sum_{k=1}^M C_{x_o, \omega_o}(x_{i,k}, \omega_{i,k})$$

Clustered sample

Cost: $\mathcal{O}(N) \rightarrow \mathcal{O}(M)$

➤ Observations

- Costly double integral per pixel
 - Directional integral (just like BSDFs)
 - Spatial integral (specific to BSSRDFs)
- Result is often smooth
 - Individually solving each pixel is **wasteful**

Smooth

$$\boxed{L_o(x_o, \omega_o)} \approx \frac{1}{M} \sum_{k=1}^M C_{x_o, \omega_o}(x_{i,k}, \omega_{i,k})$$



➤ Frequency analyses of light transport

● Theory [Durand et al. 2005]

- Acquired Materials [Bagher et al. 2012]
- Defocus, Motion Blur [Belcour et al. 2013]
- Participating media [Belcour et al. 2014]
- Animation cache [Dubouchet et al. 2017]
- Global Illumination [Belcour et al. 2017]



A Frequency Analysis of Light Transport: from Theory to Implementation. [Belcour 2017].

➤ Frequency analyses of light transport

- Theory [Durand et al. 2005]

- Acquired Materials

- [Bagher et al. 2012]

- **Can we use such analyses for clustering**

- [Belcour et al. 2013]

- **both pixels and illumination samples?**

- [Belcour et al. 2014]

- Animation cache

- [Dubouchet et al. 2017]

- Global Illumination

- [Belcour et al. 2017]



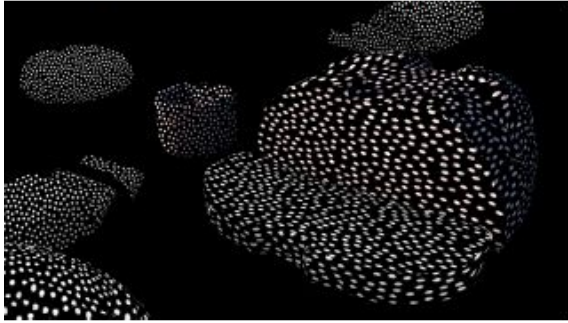
A Frequency Analysis of Light Transport: from Theory to Implementation. [Belcour 2017].

➤ Contributions

- Clustered integration for translucent objects
 - **Both** pixels and illumination samples
 - Using a **dual-tree** structure
 - Based on novel **frequency analysis**
 - Supports **any** BSSRDF models

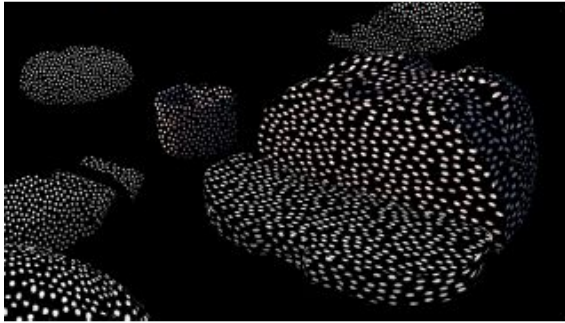
➤ Overview

I. Illumination sampling

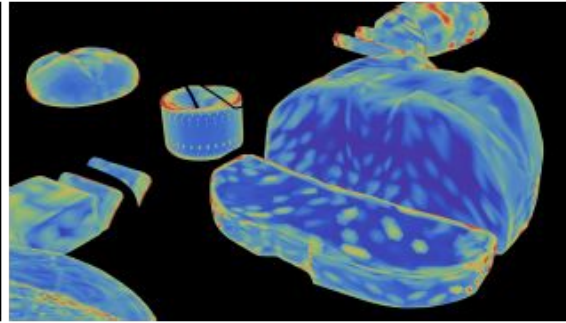


➤ Overview

I. Illumination sampling

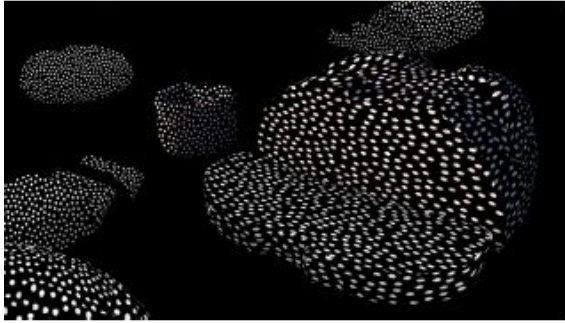


II. Predicted sampling rate

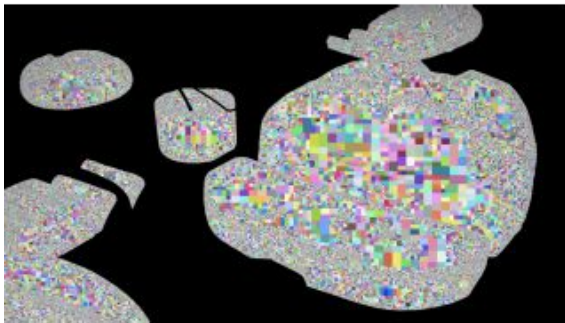
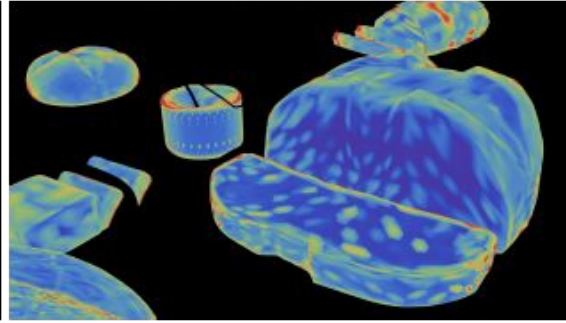


➤ Overview

I. Illumination sampling



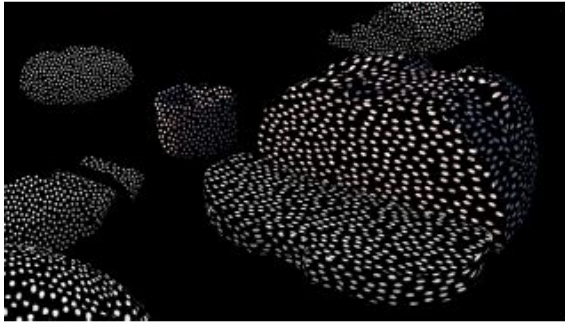
II. Predicted sampling rate



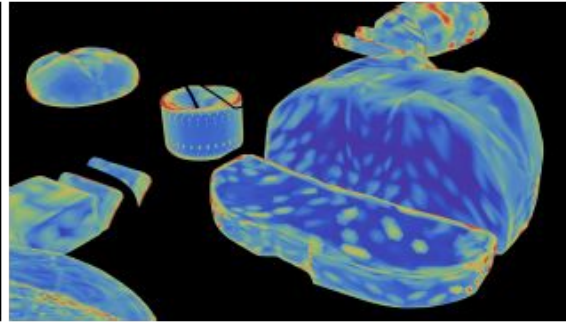
III. Clustered pixels

➤ Overview

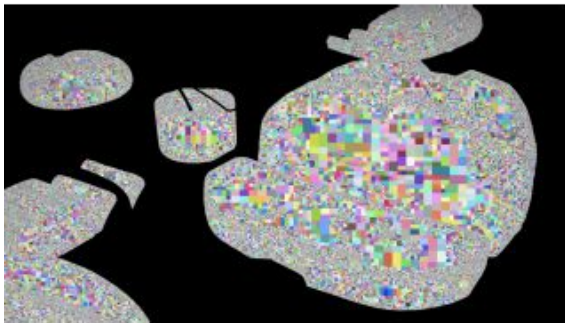
I. Illumination sampling



II. Predicted sampling rate



III. Clustered pixels



IV. BSSRDF contribution

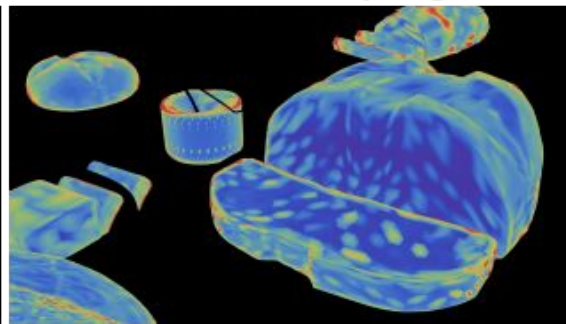


➤ Overview

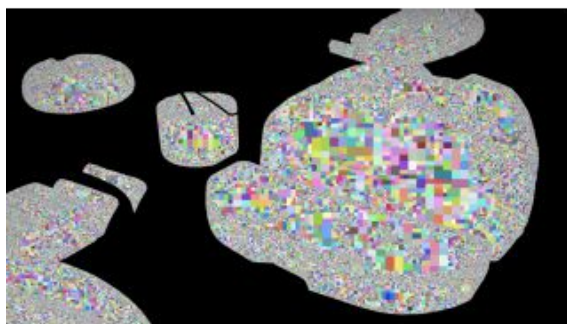
I. Illumination sampling



II. Predicted sampling rate



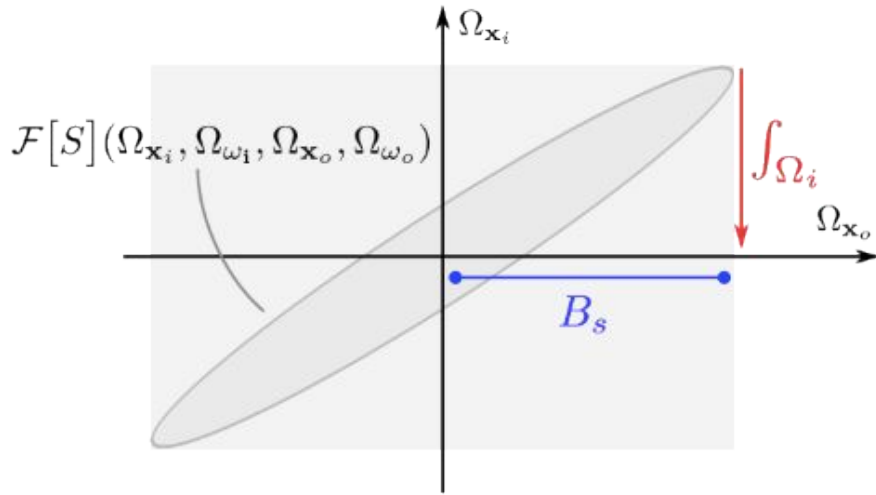
III. Clustered pixels



IV. BSSRDF contribution

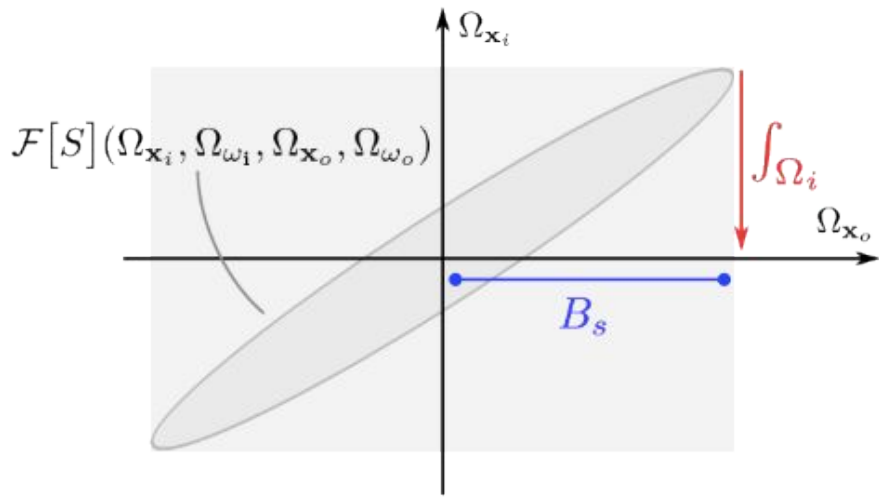


➤ Outgoing radiance bandwidth

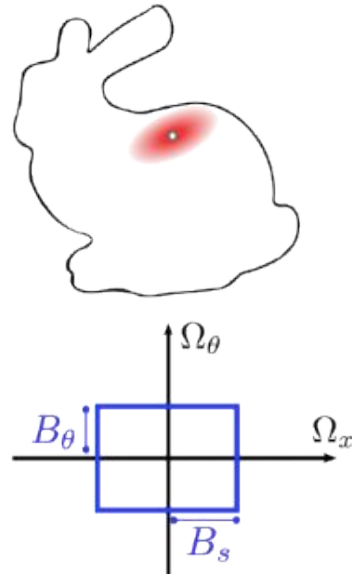


a) Outgoing radiance bandwidth computation

➤ Outgoing radiance bandwidth

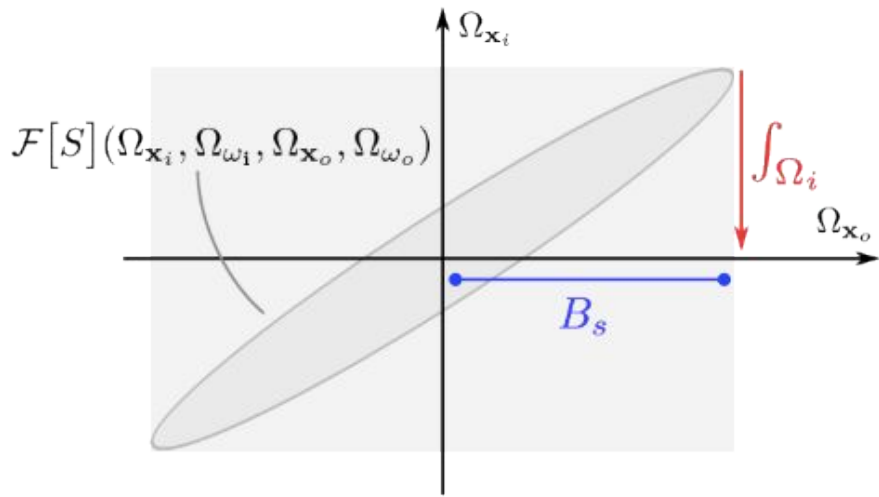


a) Outgoing radiance bandwidth computation

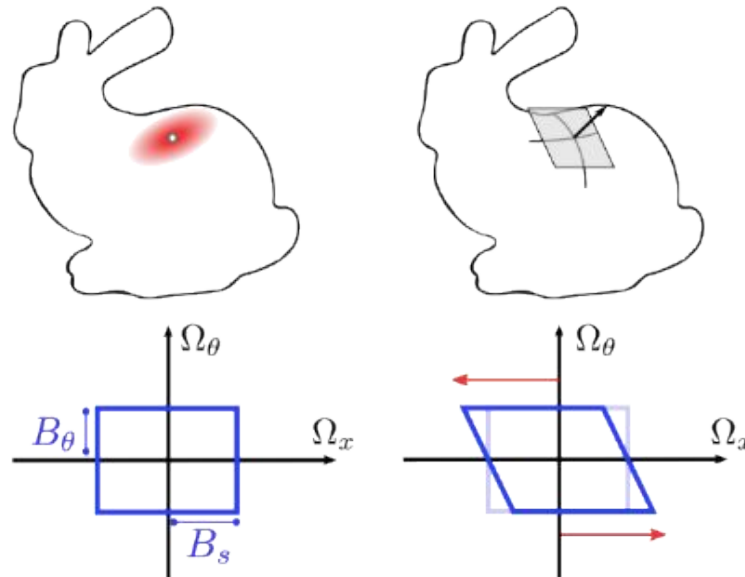


b) BSSRDF bandwidth

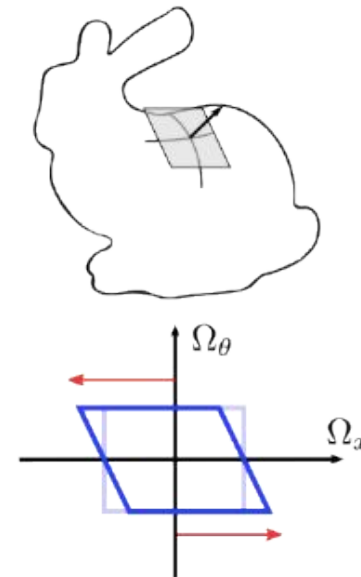
➤ Outgoing radiance bandwidth



a) Outgoing radiance bandwidth computation

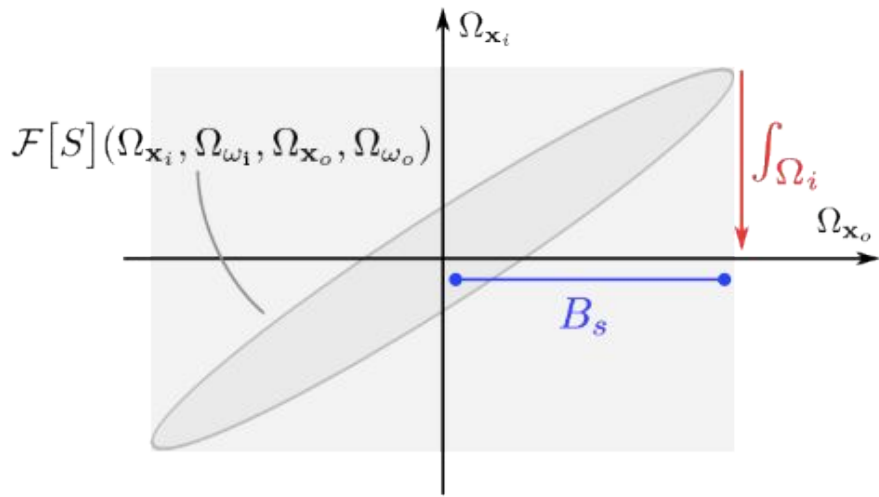


b) BSSRDF bandwidth

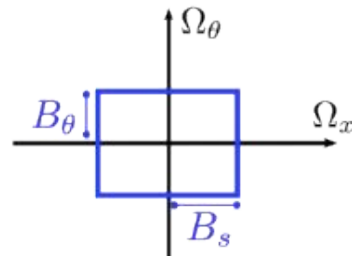


c) Curvature shear

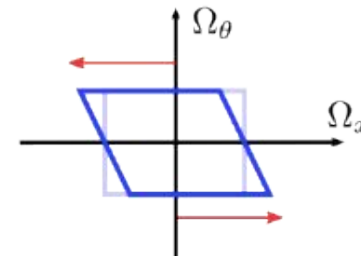
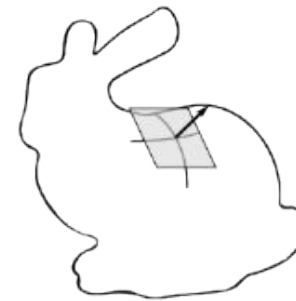
➤ Outgoing radiance bandwidth



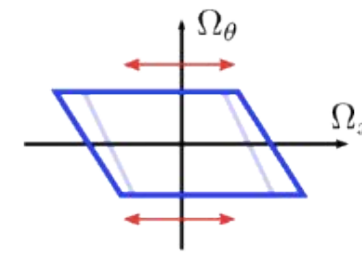
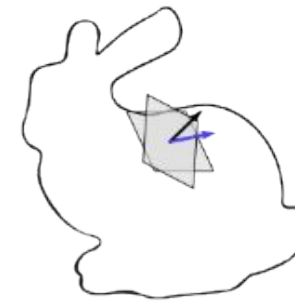
a) Outgoing radiance bandwidth computation



b) BSSRDF bandwidth

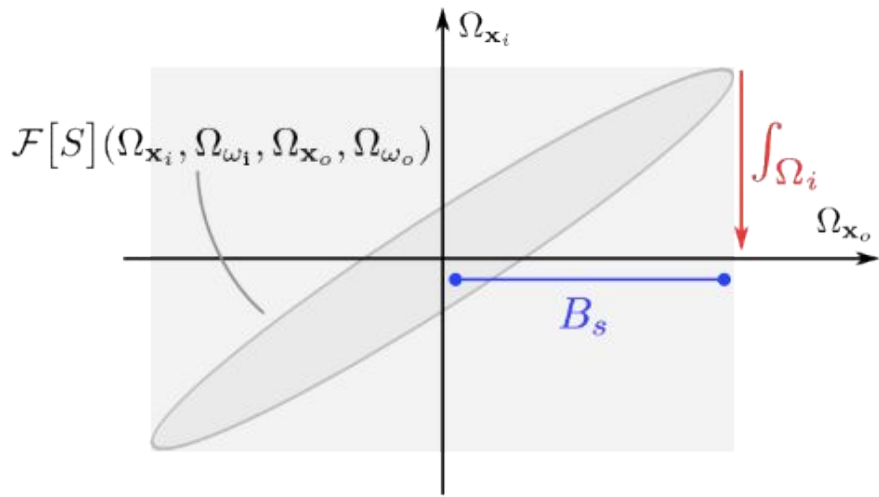


c) Curvature shear

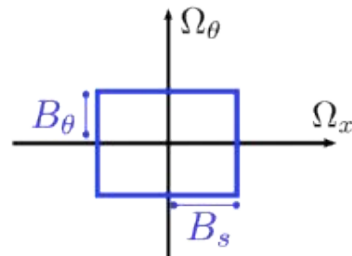


d) Cosine scale

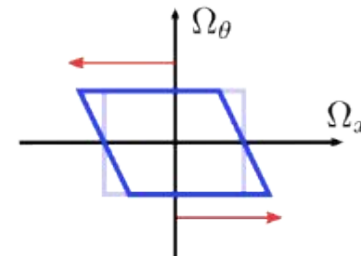
➤ Outgoing radiance bandwidth



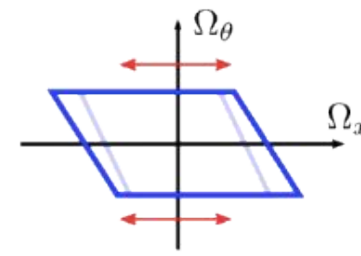
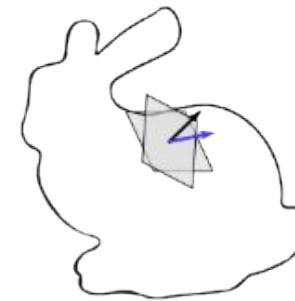
a) Outgoing radiance bandwidth computation



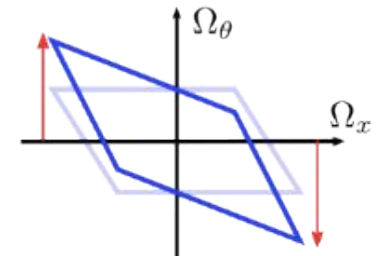
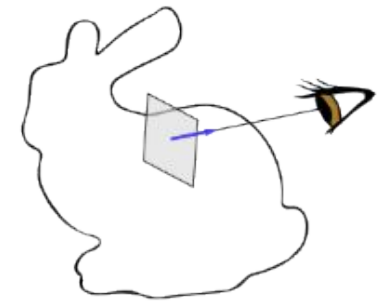
b) BSSRDF bandwidth



c) Curvature shear



d) Cosine scale

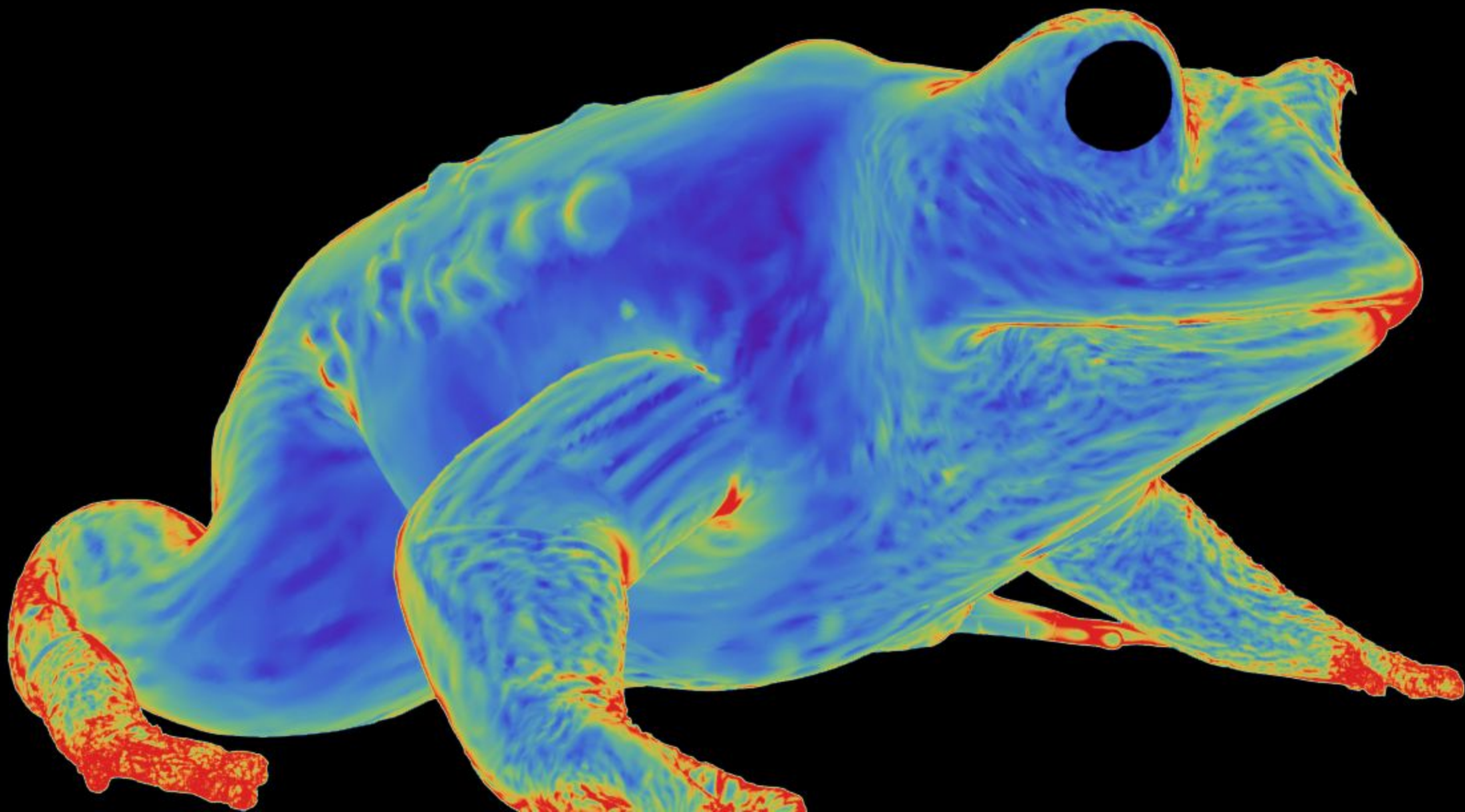


e) Transport shear

Image



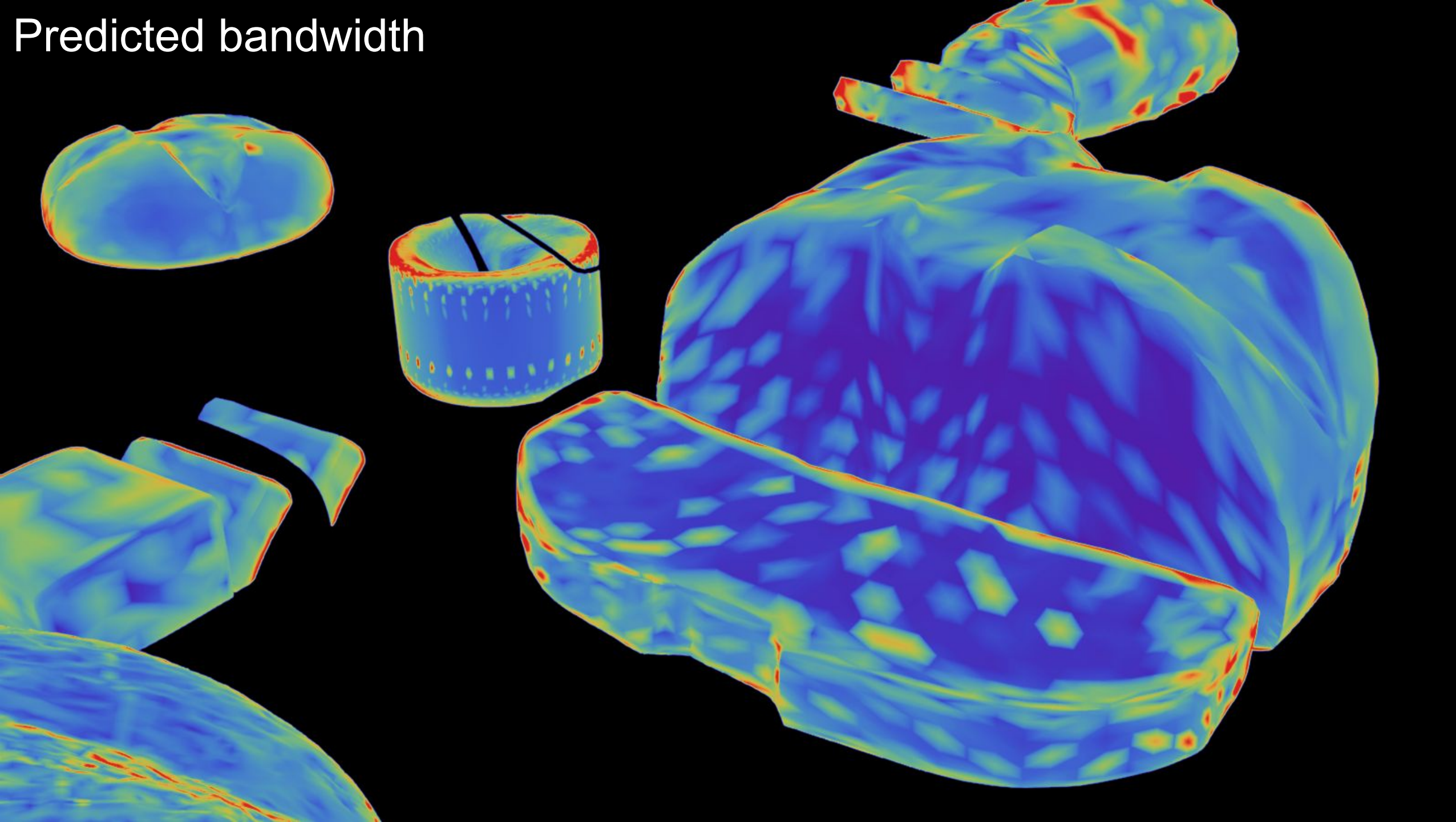
Predicted bandwidth



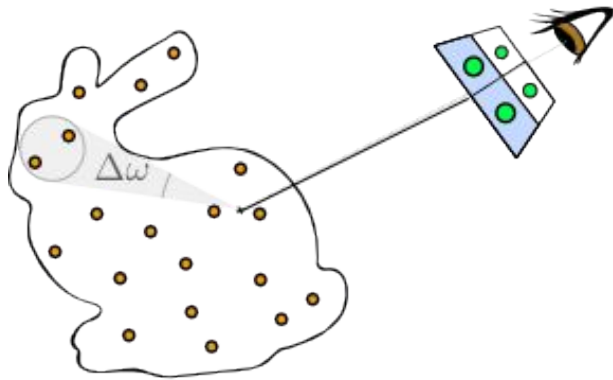


Image

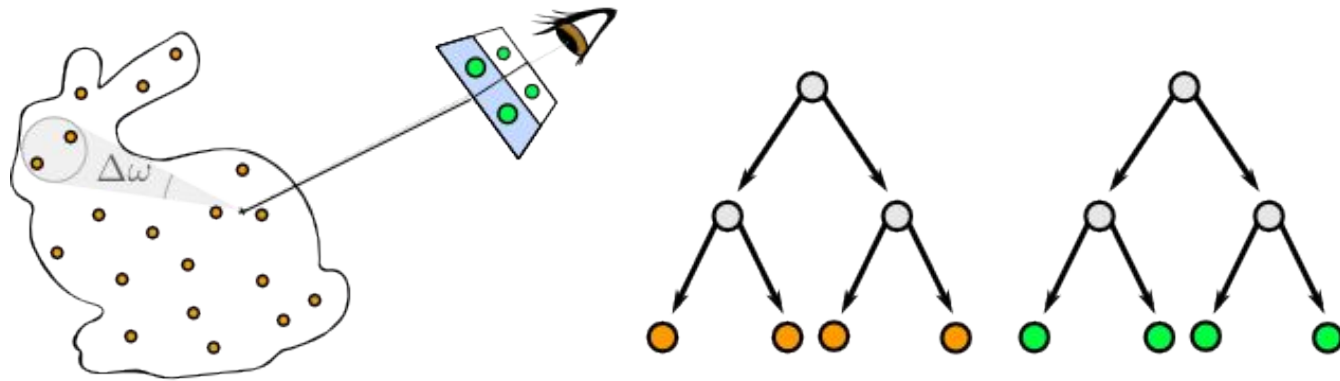
Predicted bandwidth



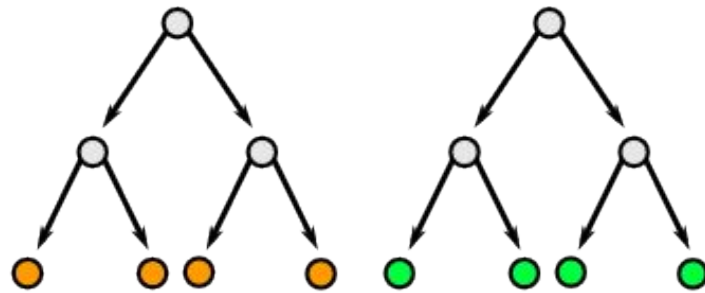
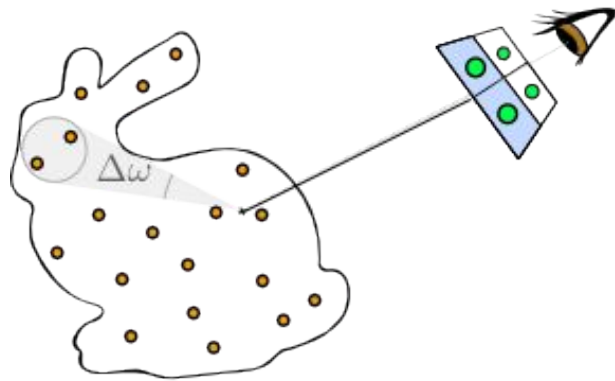
➤ Dual-tree traversal



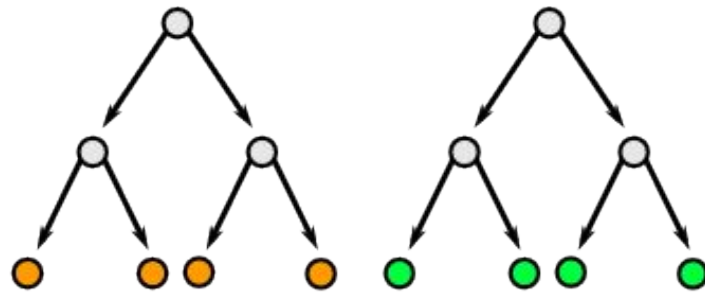
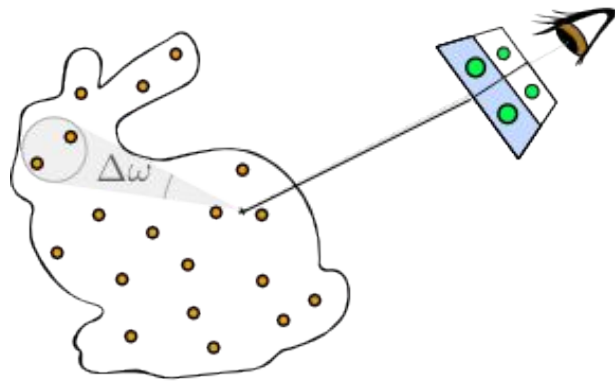
➤ Dual-tree traversal



➤ Dual-tree traversal



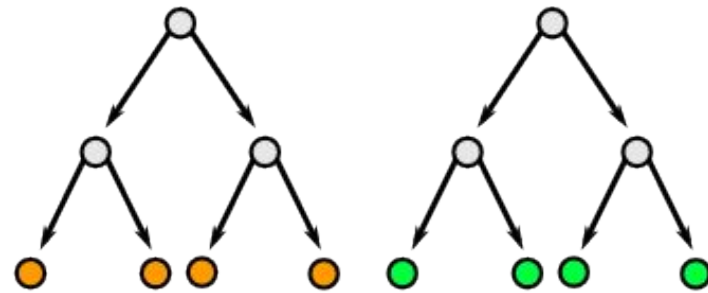
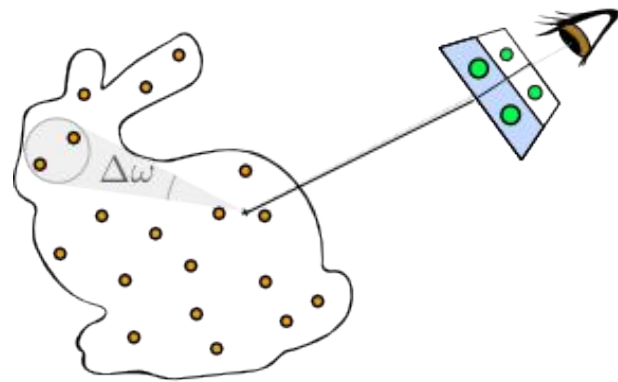
➤ Dual-tree traversal



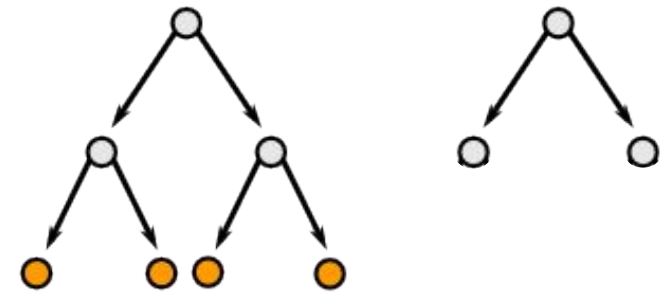
Simultaneous refinement



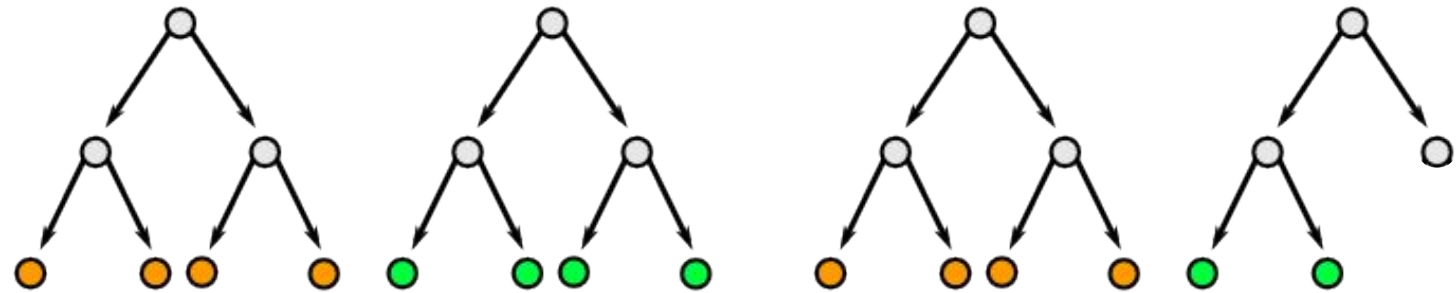
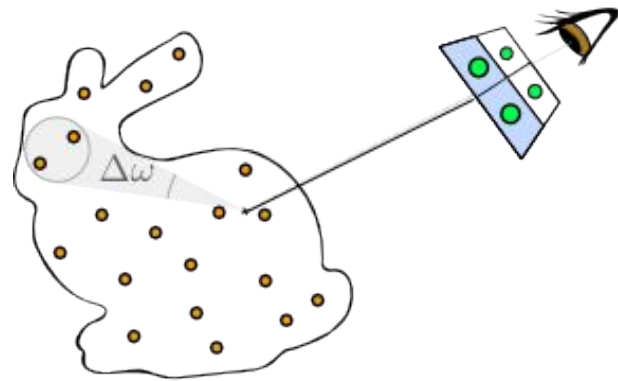
➤ Dual-tree traversal



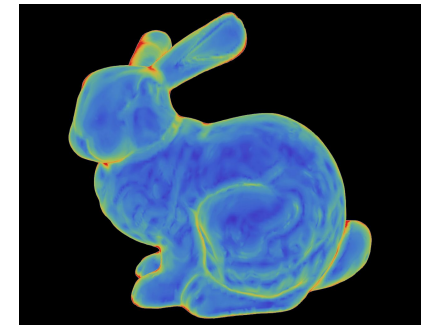
Simultaneous refinement



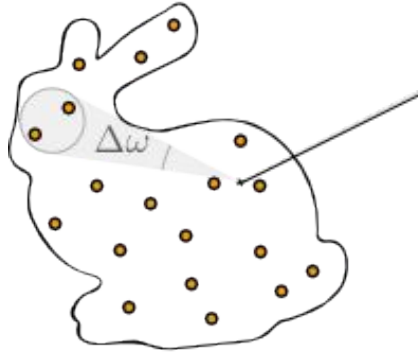
➤ Dual-tree traversal



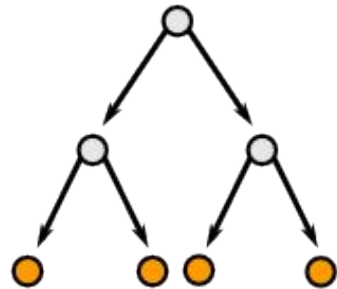
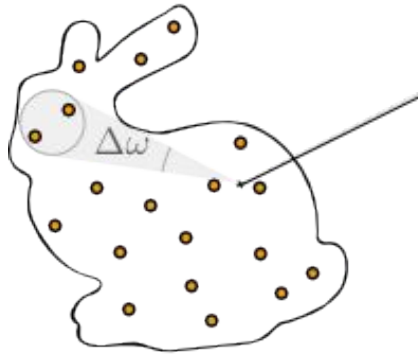
Simultaneous refinement



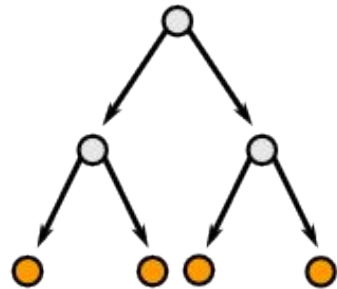
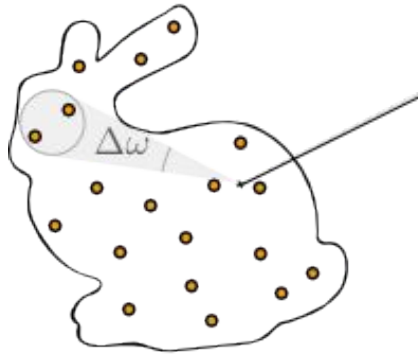
➤ Single-tree traversal (previous work)



➤ Single-tree traversal (previous work)

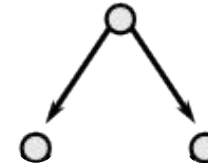
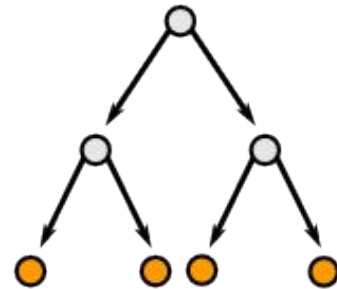
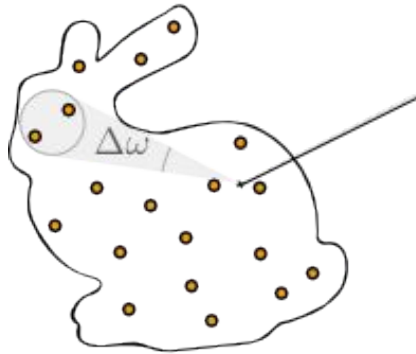


➤ Single-tree traversal (previous work)



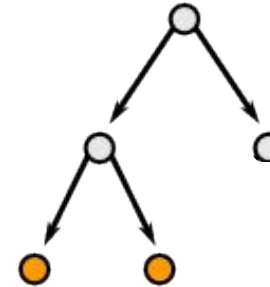
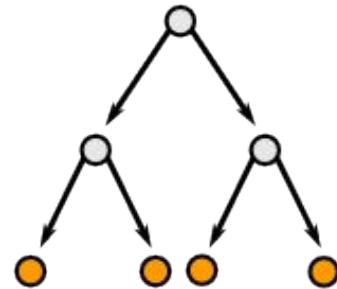
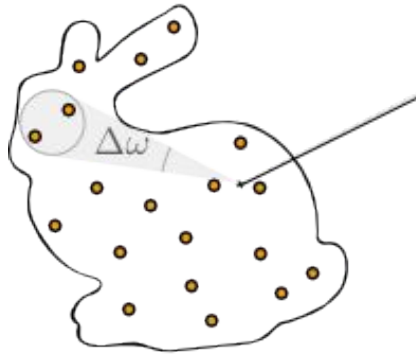
➤ Single-tree traversal (previous work)

Independent refinement per pixel

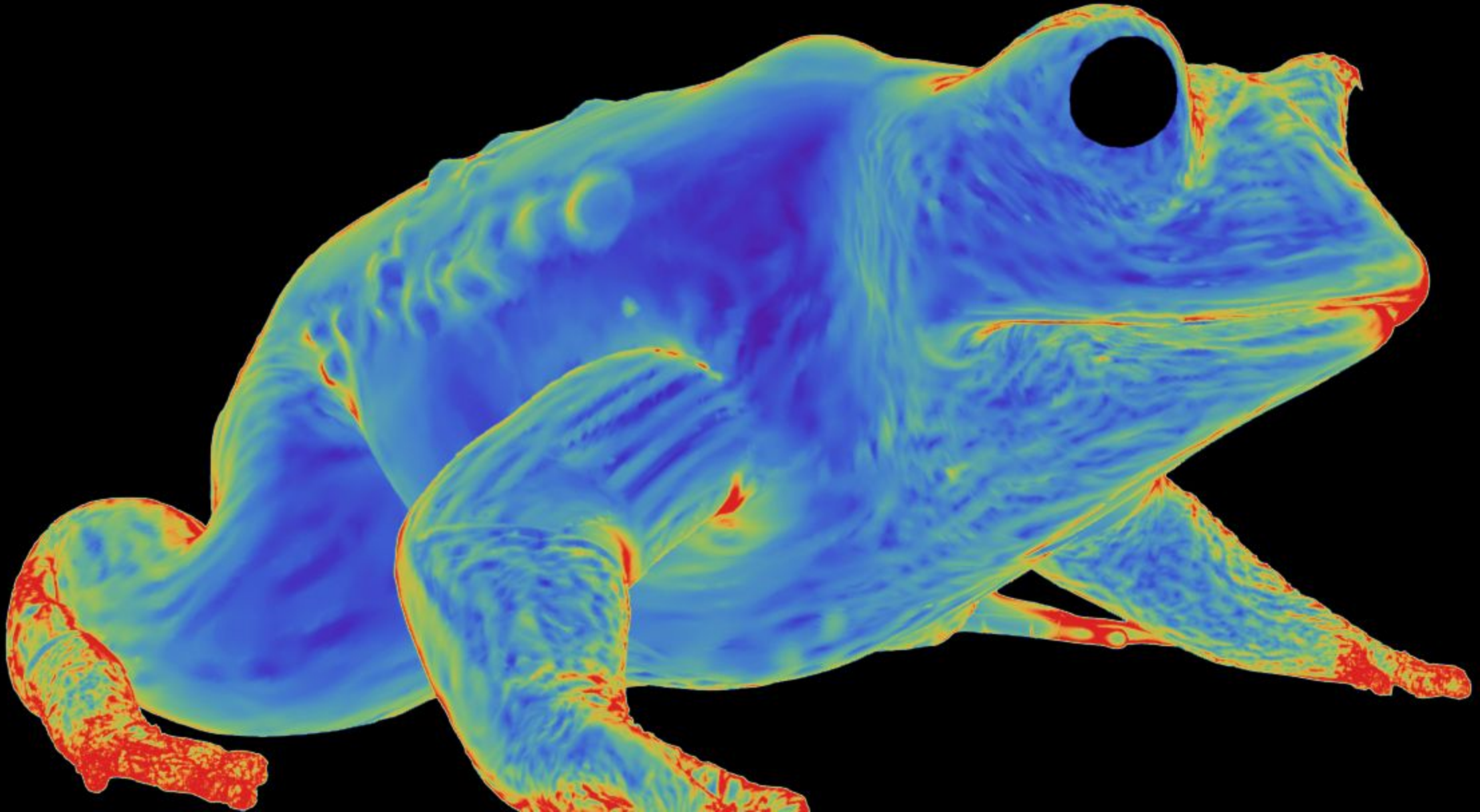


➤ Single-tree traversal (previous work)

Independent refinement per pixel



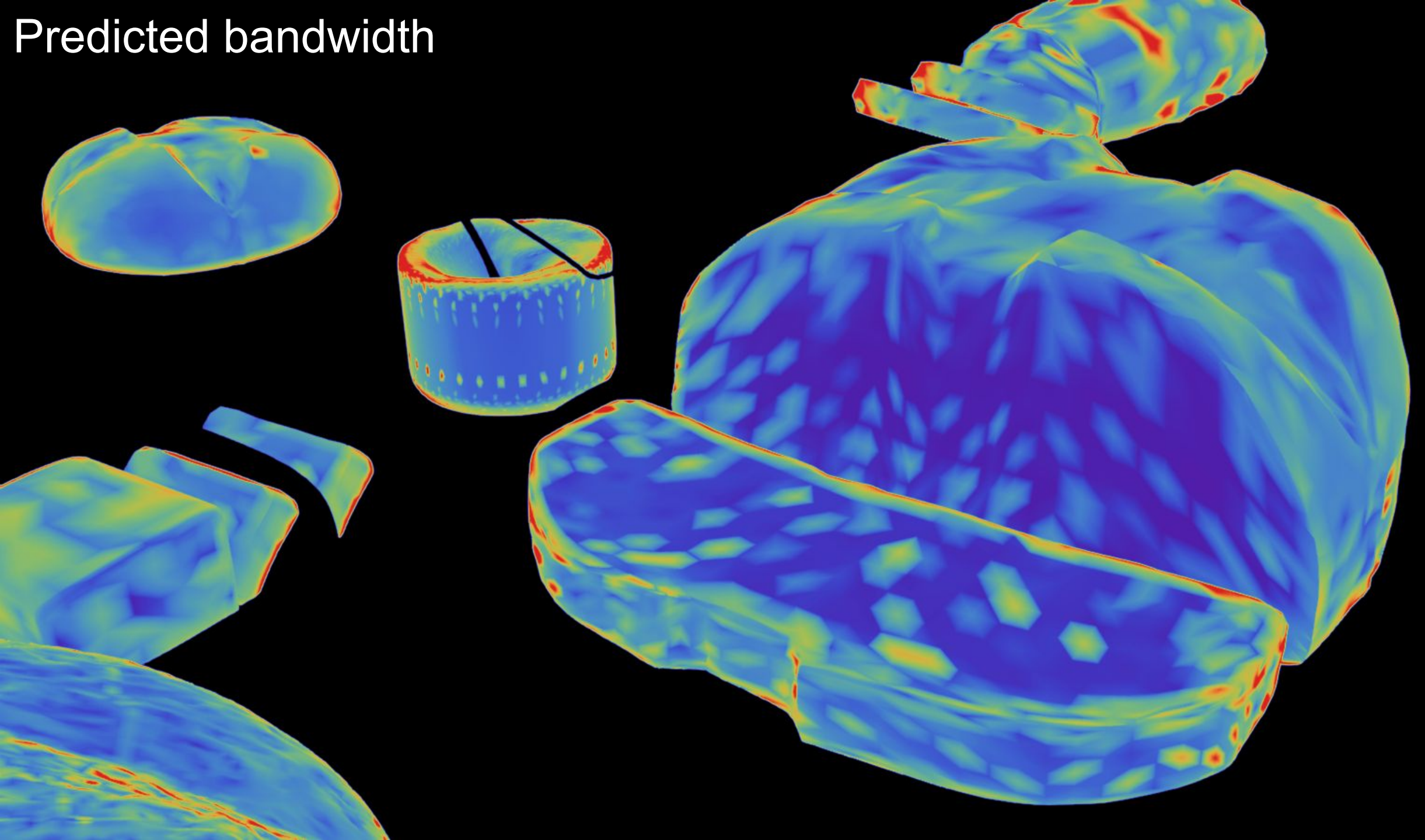
Predicted bandwidth



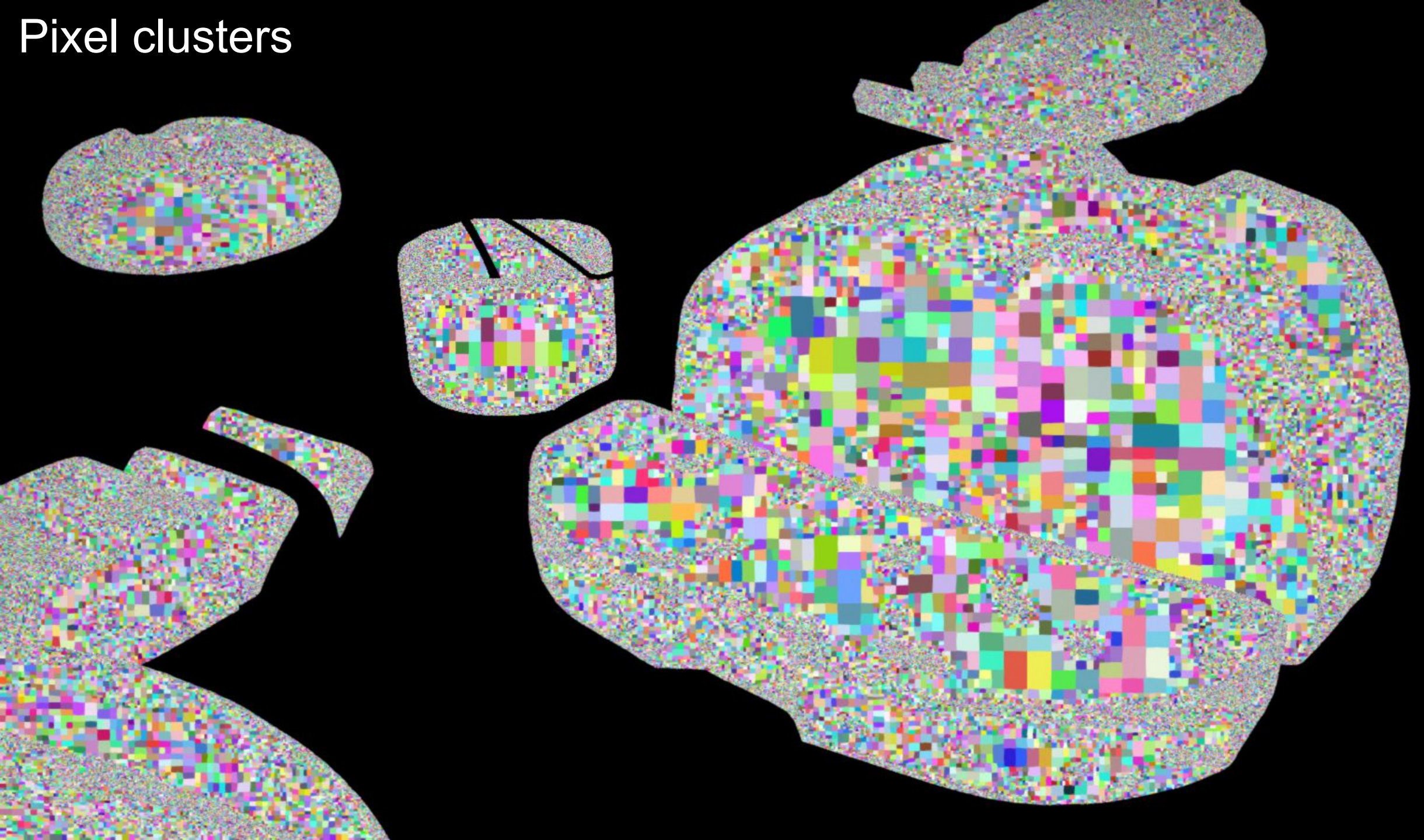
Pixel clusters



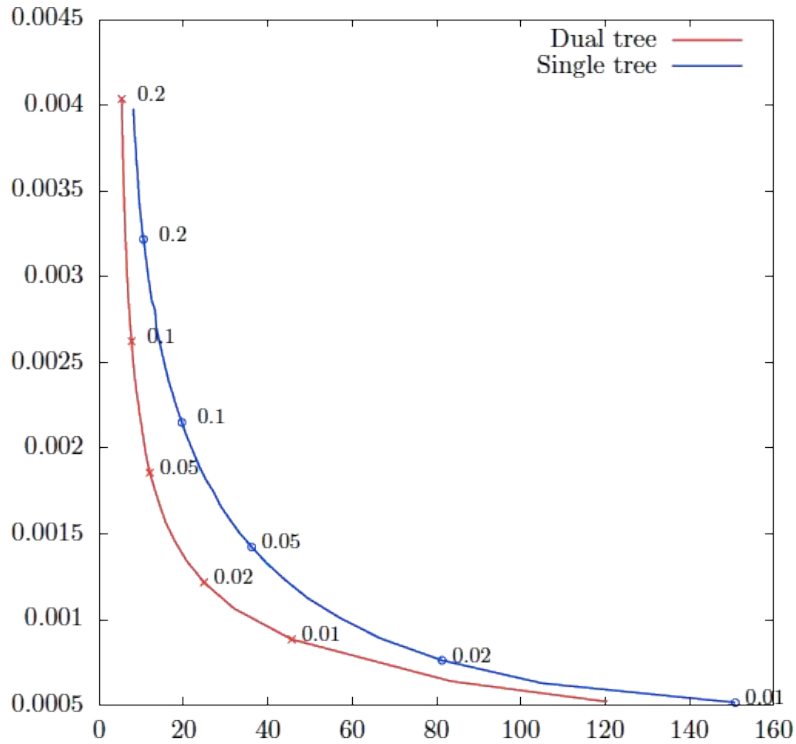
Predicted bandwidth



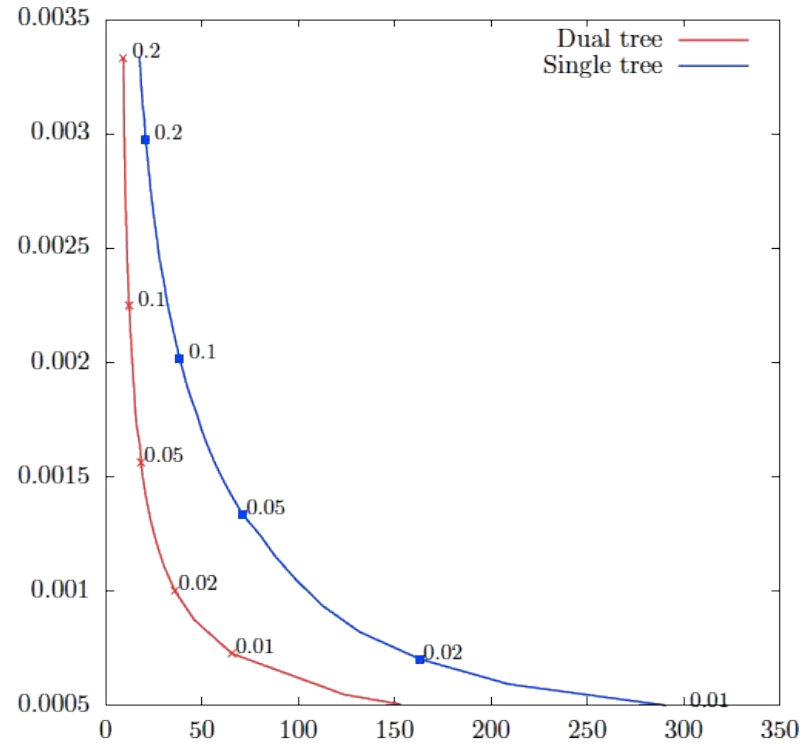
Pixel clusters



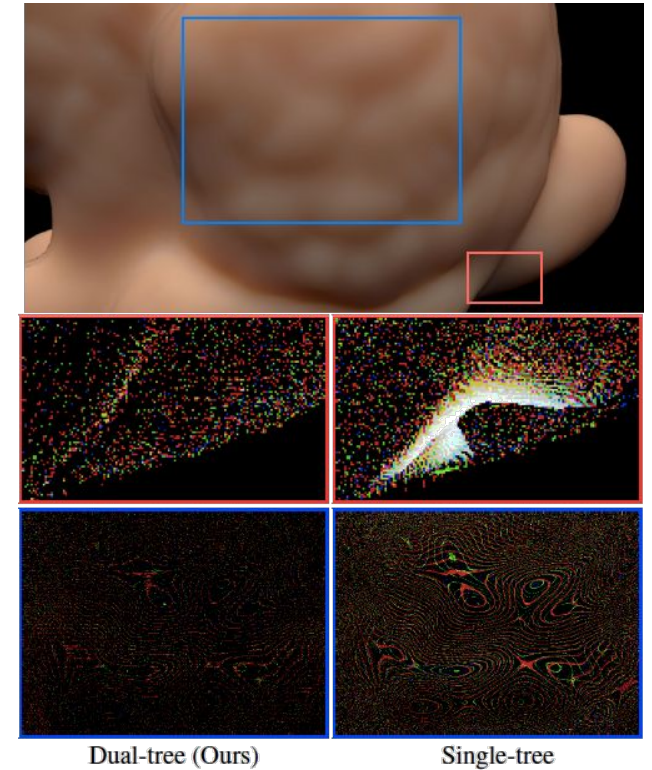
Results



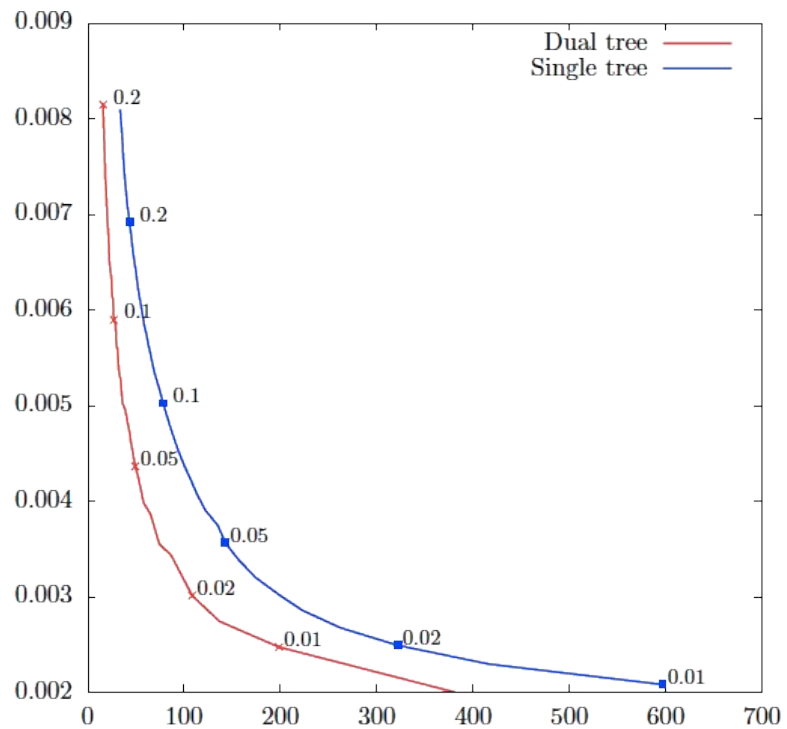
(a) BUNNY



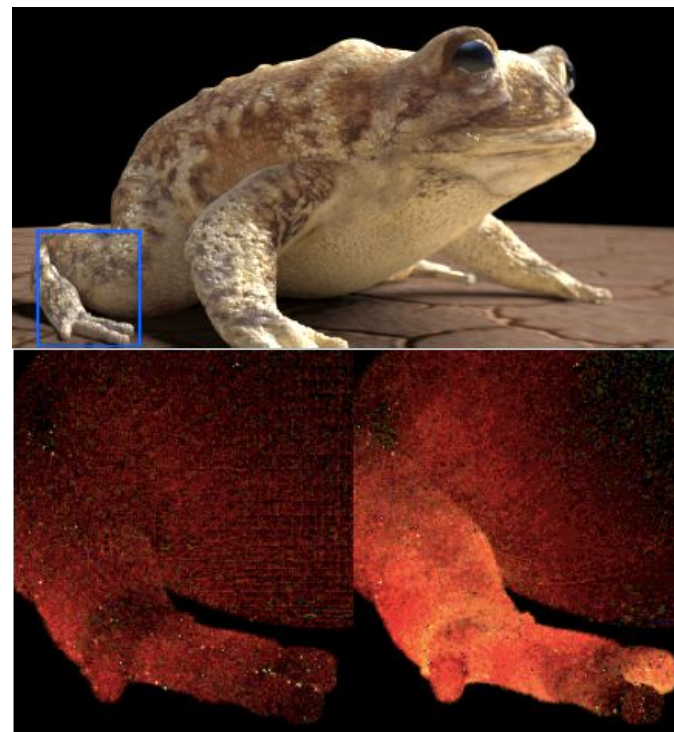
(b) BUNNY (CLOSE-UP)



Results



(c) TOAD



Dual-tree (Ours)

Single-tree

➤ Limitations

- Subpixel variation will not be detected faithfully
 - Aliasing can cause wrong clustering
 - Need more conservative bandwidth estimation
- Manual clustering for separated objects
 - One tree per “separated” translucent object is needed
 - Same for any existing approaches

➤ Conclusion

- Translucent objects often exhibit smoothness across pixels due to their blurring nature
- Smoothness is exploited via clustering
 - Dual-tree structure for pixels and illumination samples
 - Frequency analysis conservatively predicts cluster sizes