egsr 2018

29th EG Symposium on Rendering.
Karlsruhe June, 27-29, 2018.

Stratified Sampling of Projected Spherical Caps

Carlos Ureña

Universidad de Granada

Iliyan Georgiev

Autodesk

Solid Angle

Index

- Motivation & problem statement
- Known: spherical cap sampling
- New: projected spherical cap sampling
    - Geometry
    - Area preserving maps
- Evaluation
    - Warping properties
    - Rendering

Motivation & problem statement

Spherical light sources: frequently used in scenes (isotropic emitted radiance)

Variance reduction via stratified sampling (Shirley et. al, 1991)

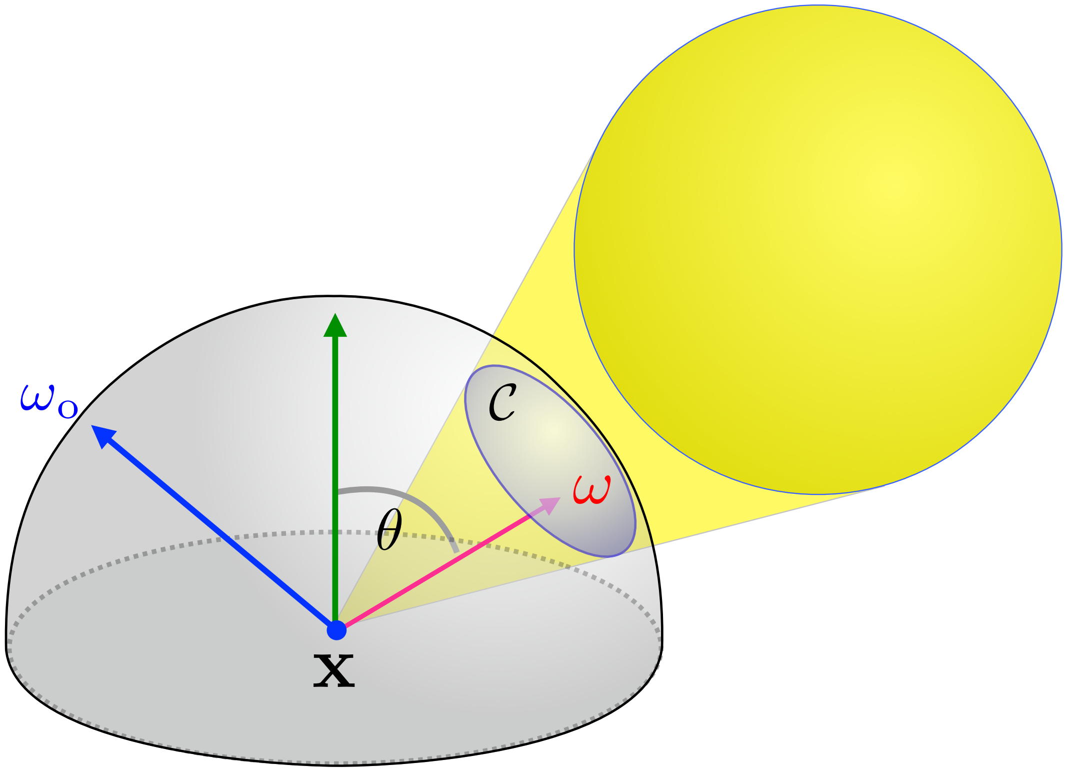
Related prior work

- Solid angle sphere sampling spheres (Shirley et al. 1994)
- Projected solid angle sampling fully visible spheres (Shirley et al. 1994)
- Solid angle sampling triangles (Arvo 95)
- Projected solid angle sampling planar polygons (Arvo 2001)
- Solid angle sampling rectangles (Ureña et al. 2013)
- Rejection based solid angle sampling disks and cylinders (Gamito 2016)
- Solid angle sampling disks (Guillén et al. 2017).

We address **projected solid angle sampling spheres**

Problem statement

- Compute outgoing radiance due to direct illumination from spherical light source
- Integral over a **spherical cap**



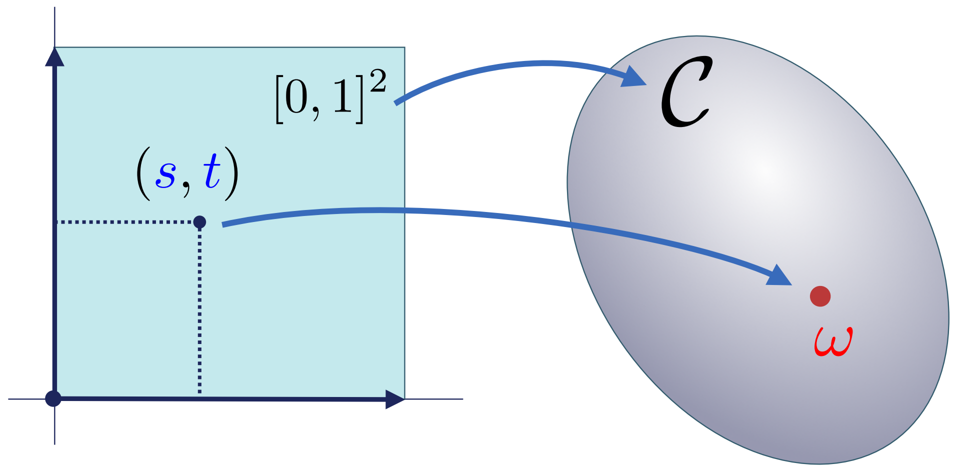
$$L_o(\mathbf{x}, \omega_o) = \int_{\mathcal{C}} L_i(\mathbf{x}, \omega) f_s(\mathbf{x}, \omega_o, \omega) |\cos\theta| \, d\sigma(\omega)$$

Known solution:
Stratified sampling of spherical caps

Solid angle → unit square

- Use a map ω

$$(s,t) \in [0,1]^2 \;\longrightarrow\; \omega(s,t) \in \mathcal{C}$$



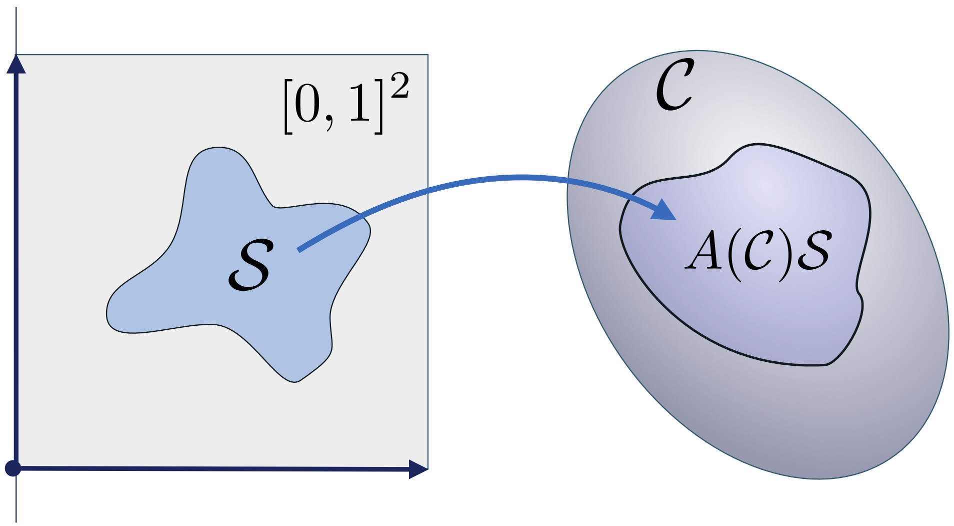
$$L_o(\mathbf{x},\omega_o) \;=\; \int_{[0,1]^2} L_i(\mathbf{x},\omega)\, f_s(\mathbf{x},\omega_o,\omega)\,|\cos\theta|\, J_\omega(s,t)\,\mathrm{d}s\mathrm{d}t$$

Area-preserving mapping

- Jacobian can introduce variance
- Avoid variance by making Jacobian constant

$$J_{\omega}(s,t) \;=\; A(\mathcal{C})$$

- Makes map **area preserving**

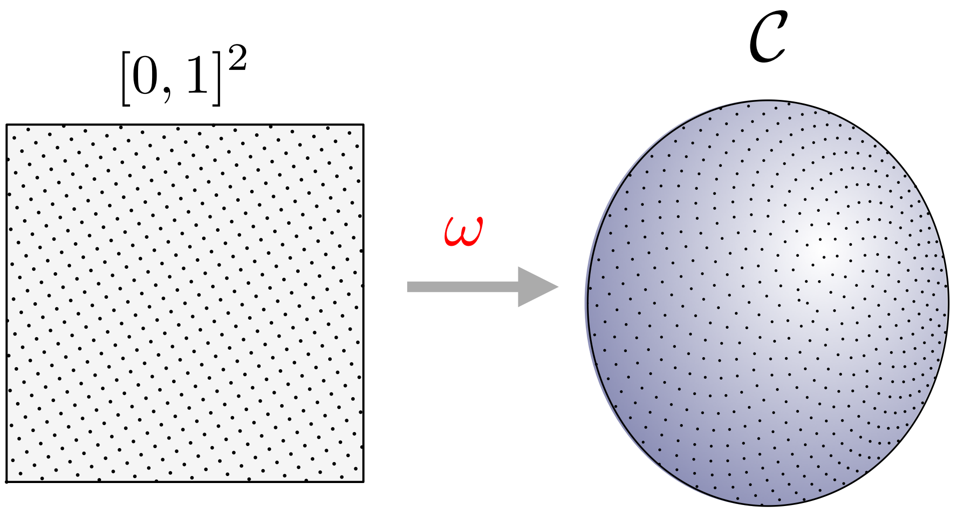


$$L_{\mathrm{o}}(\mathbf{x},\omega_{\mathrm{o}}) \;=\; A(\mathcal{C}) \int_{[0,1]^2} L_{\mathrm{i}}(\mathbf{x},\omega)\, f_{\mathrm{s}}(\mathbf{x},\omega_{\mathrm{o}},\omega)\, |\cos\theta|\, \mathrm{d}s\mathrm{d}t$$

Monte Carlo estimation

- Use a stratified unit-square sample set

$$\{ (s_0, t_0),\ (s_1, t_1),\ \ldots,\ (s_{N-1}, t_{N-1}) \}$$



$$L_{\mathrm{o}}(\mathbf{x}, \omega_{\mathrm{o}}) \approx \frac{A(\mathcal{C})}{N} \sum_{i=0}^{N-1} L_{\mathrm{i}}(\mathbf{x}, \omega_i)\, f_{\mathrm{s}}(\mathbf{x}, \omega_{\mathrm{o}}, \omega_i)\, |\cos\theta_i| \qquad \text{where: } \omega_i = \omega(s_i, t_i)$$

Drawbacks

- Every sample weighted by cosine term
- Zero-weight samples when spherical cap is partially below horizon

Our maps

- Factor in cosine term
- No samples below horizon

Stratified sampling of projected spherical caps

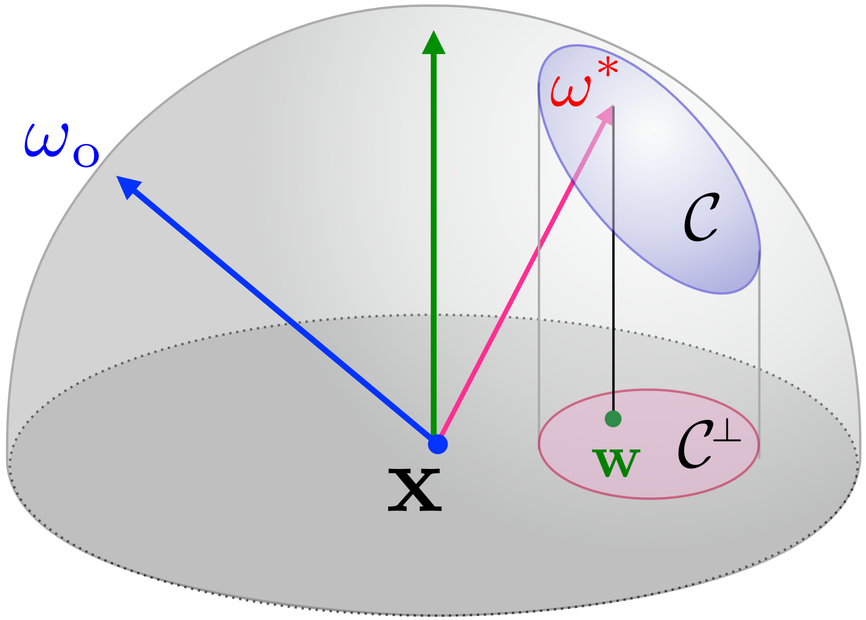
Projected solid angle integral

- Use a map ω^* from the **projected spherical cap** $\mathcal{C}^\perp$

$$\mathbf{w} \in \mathcal{C}^\perp \longrightarrow \omega^*(\mathbf{w}) \in \mathcal{C}$$

- Variance reduction due to

$$J_{\omega^*}(\mathbf{w}) = \frac{1}{|\cos\theta|}$$



$$L_o(\mathbf{x}, \omega_o) = \int_{\mathcal{C}^\perp} L_i(\mathbf{x}, \omega^*)\, f_s(\mathbf{x}, \omega_o, \omega^*)\, \mathrm{d}A(\mathbf{w})$$

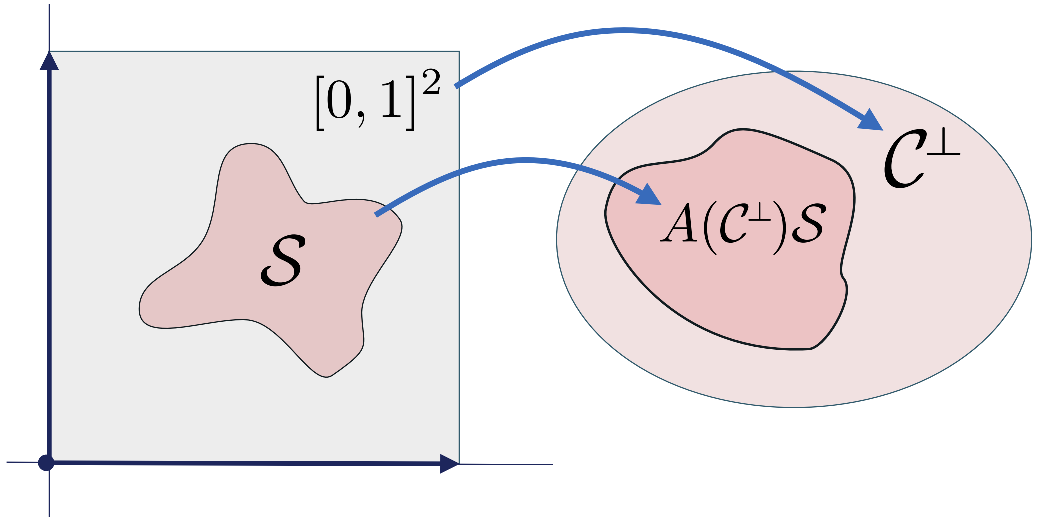
Projected solid angle → unit square

- Use a map $\mathbf{w}$

$$(s,t) \in [0,1]^2 \longrightarrow \mathbf{w}(s,t) \in \mathcal{C}$$

- Again, constant Jacobian:

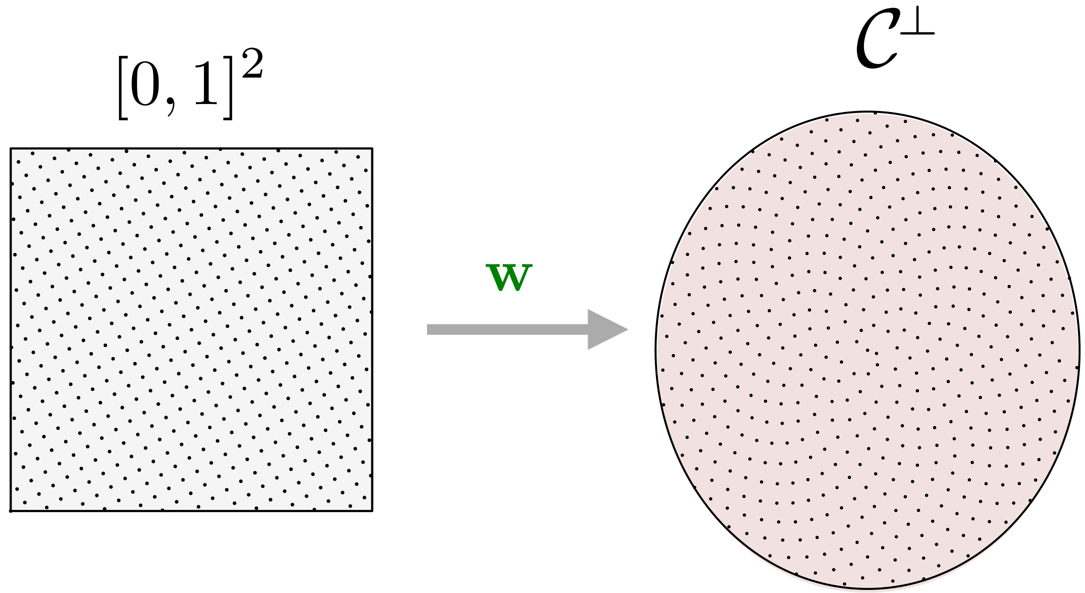
$$J_{\mathbf{w}}(s,t) = A(\mathcal{C}^{\perp})$$



$$L_{\mathrm{o}}(\mathbf{x},\omega_{\mathrm{o}}) = A(\mathcal{C}^{\perp}) \int_{[0,1]^2} L_{\mathrm{i}}(\mathbf{x},\omega^*)\, f_{\mathrm{s}}(\mathbf{x},\omega_{\mathrm{o}},\omega^*)\, \mathrm{d}s\mathrm{d}t$$

Stratified projected solid angle sampling

- No cosine term in estimator

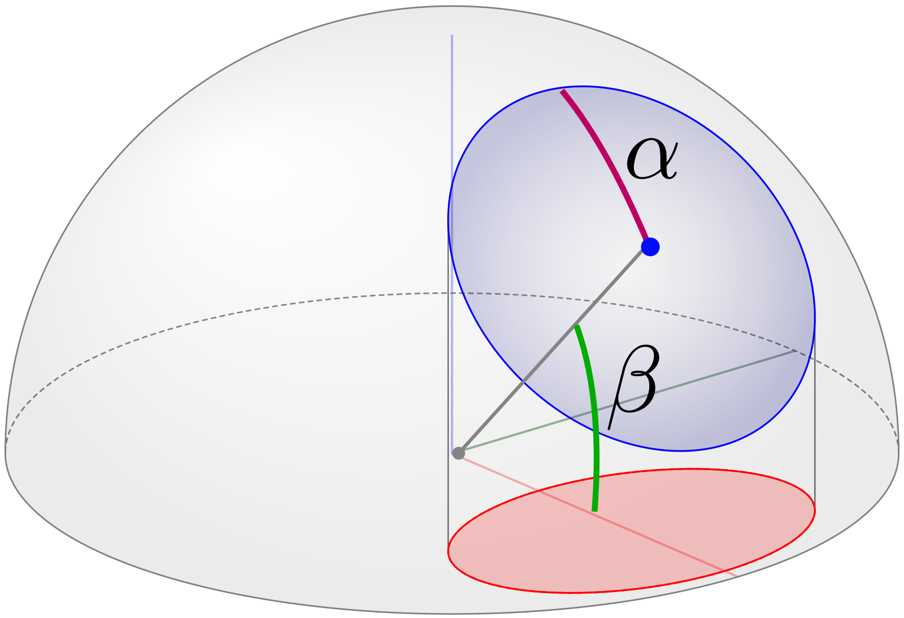


$$L_{\mathrm{o}}(\mathbf{x},\omega_{\mathrm{o}}) \approx \frac{A(\mathcal{C}^{\perp})}{N}\sum_{i=0}^{N-1} L_{\mathrm{i}}(\mathbf{x},\omega_i^*)\, f_{\mathrm{s}}(\mathbf{x},\omega_{\mathrm{o}},\omega_i^*) \qquad \text{where:} \quad \omega_i^* = \omega^*(\mathbf{w}(s_i,t_i))$$

Projected spherical cap geometry

Cap parameterization

Two angles: **aperture** α and **elevation** β

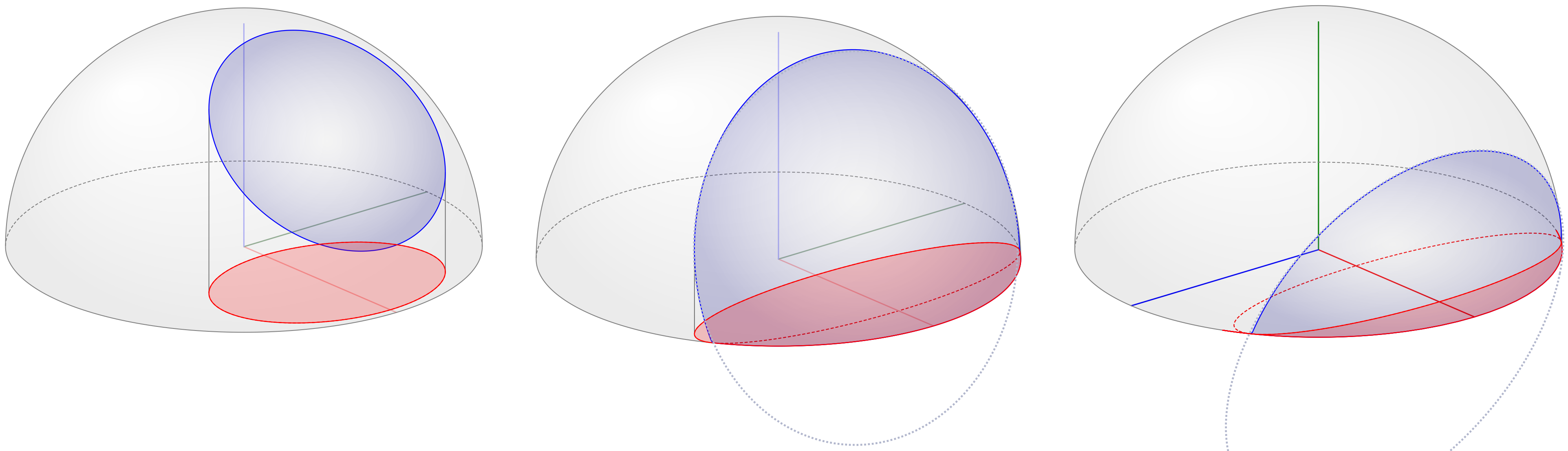


Three cases

$0 < \alpha \leq \beta$

$0 \leq \beta < \alpha$

$-\alpha < \beta < 0$



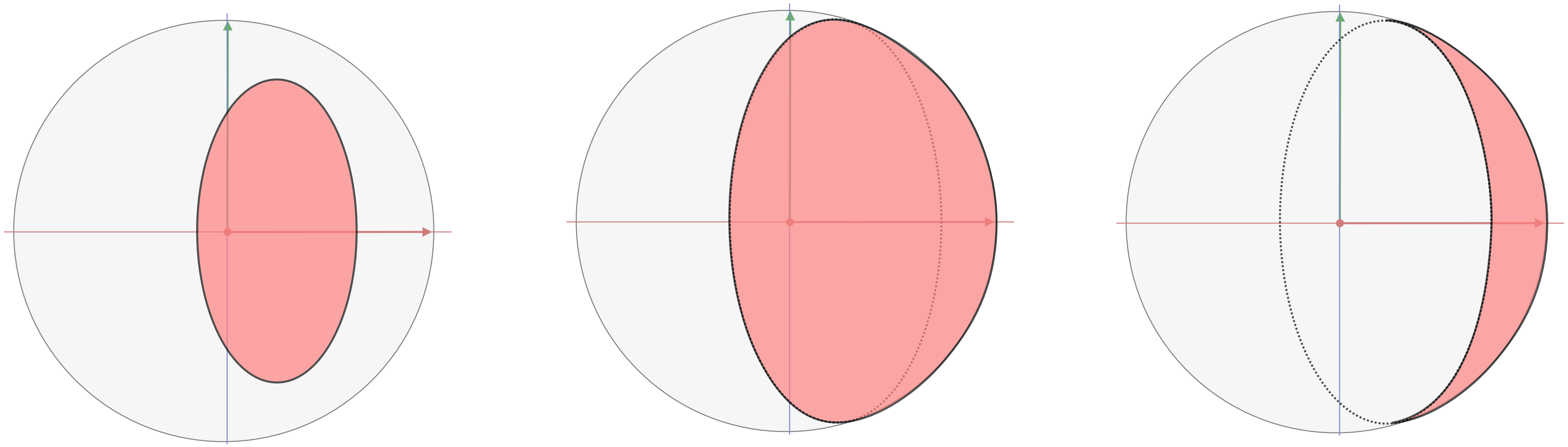
Cap fully visible
- center above horizon
- projection is ellipse

Cap mostly visible
- center above horizon
- projection is ellipse + lune

Cap mostly invisible
- center below horizon
- projection is lune

Three cases

(Cap bounding circumference projection is always ellipse)



Cap fully visible
- projection is ellipse
- ellipse included in unit disk

Cap mostly visible
- projection is ellipse + lune
- ellipse is tangent to unit disk

Cap mostly invisible
- projection is lune.
- ellipse is tangent to unit disk

Map design & evaluation

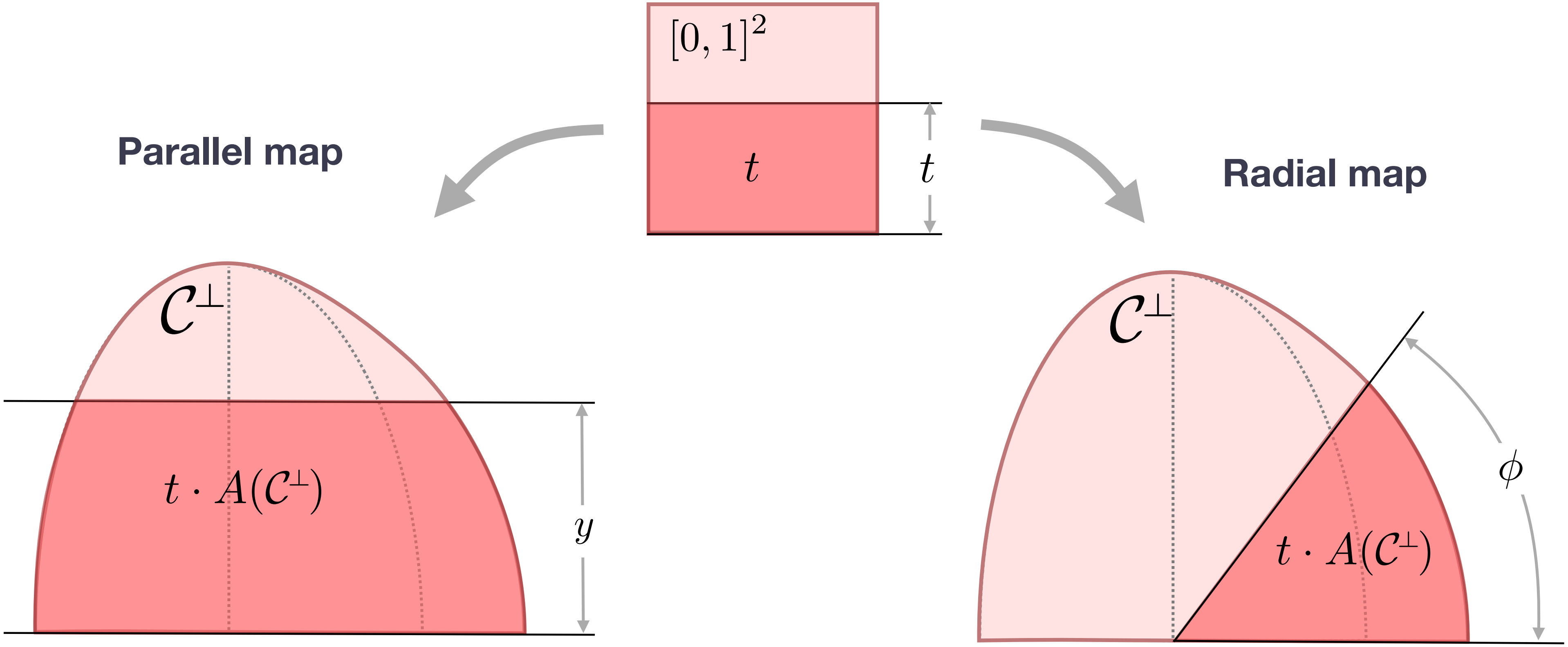
Desired properties

- **Area preserving**
- **Continuous** under continuous variations of parameters s and t
- **Continuous** under continuous variations of **aperture** and **elevation** angles

Two maps

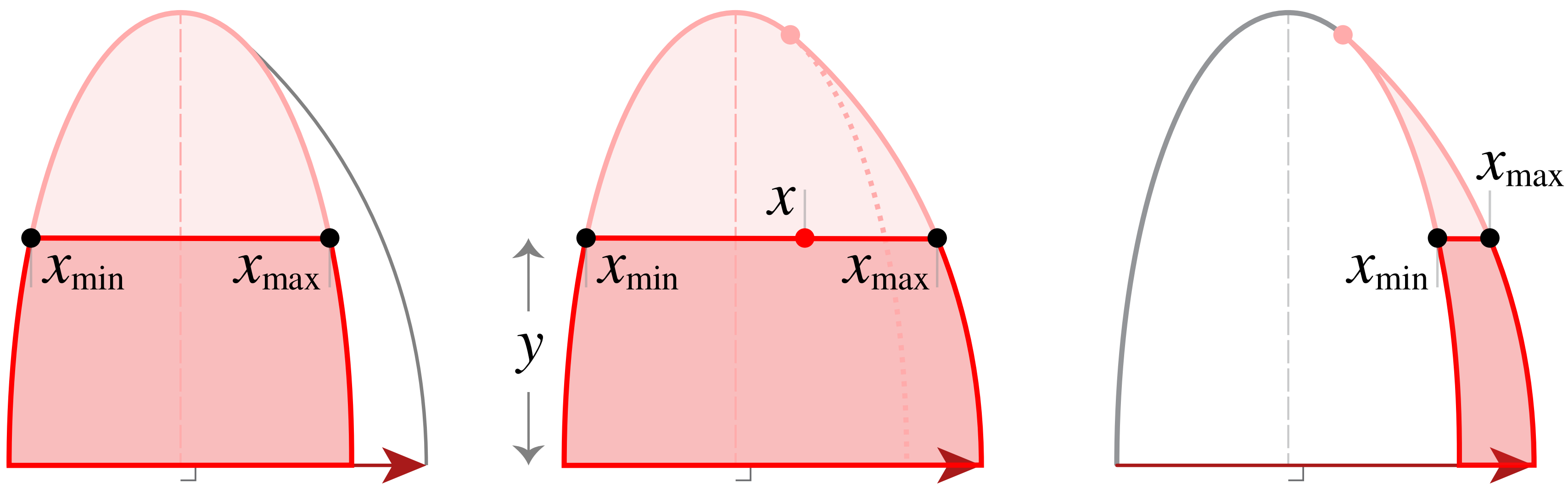
- **Parallel**: t-isocurves are lines parallel to X-axis
- **Radial**: t-isocurves are radii through ellipse center

For given (s,t) determine height/angle as functions of t, so that **area is preserved**



Parallel map

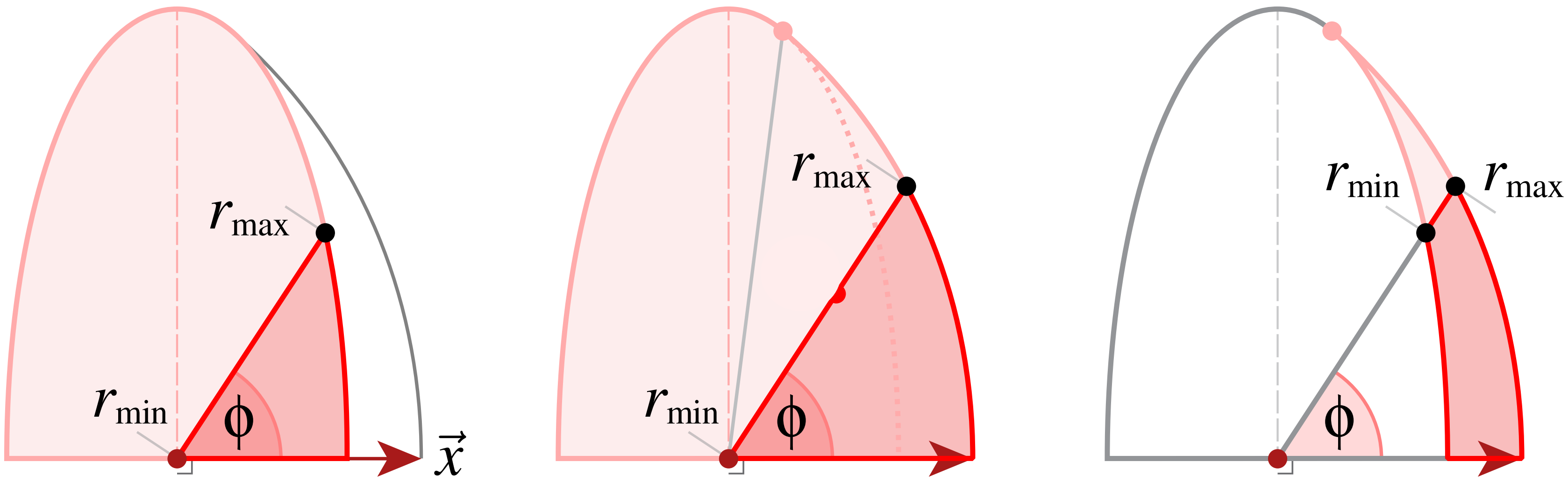
Partial area between X-axis and line expressed analytically



$$A_{\mathrm{p}}(y) = \int_0^y \left[\, x_{\mathrm{max}}(y') - x_{\mathrm{min}}(y') \,\right] \mathrm{d}y'$$

Radial map

Partial area between radius and X-axis expressed analytically



$$A_r(\phi) = \frac{1}{2}\int_0^{\phi}\left[r^2_{max}(\phi') - r^2_{min}(\phi')\right] d\phi'$$

Sampling: find line/radius

- Need to invert partial-area function: **no simple analytical expression**
- Resort to **numerical inversion**

Parallel map

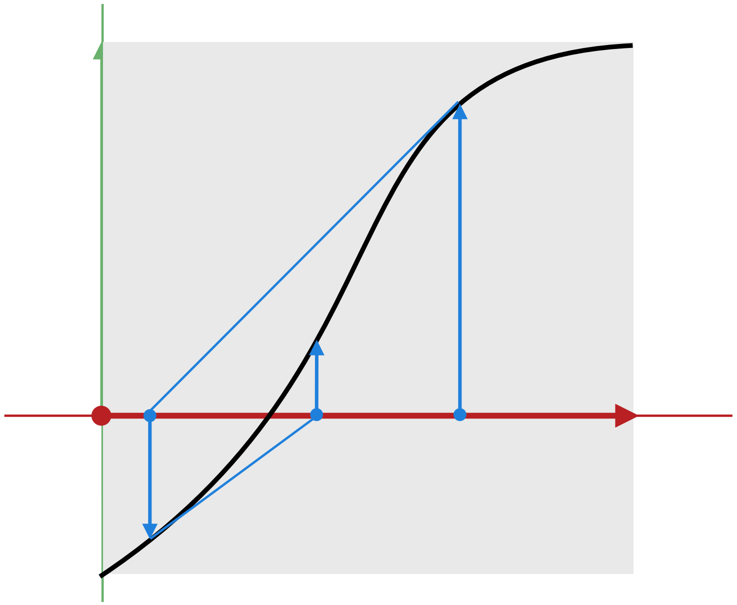
$$t = \frac{A_{\mathrm{p}}(y)}{A(\mathcal{C}^{\perp})} \implies y = A_{\mathrm{p}}^{-1}\left(t\,A(\mathcal{C}^{\perp})\right) \implies f_t(y) = \frac{A_{\mathrm{p}}(y)}{A(\mathcal{C}^{\perp})} - t = 0$$

Radial map

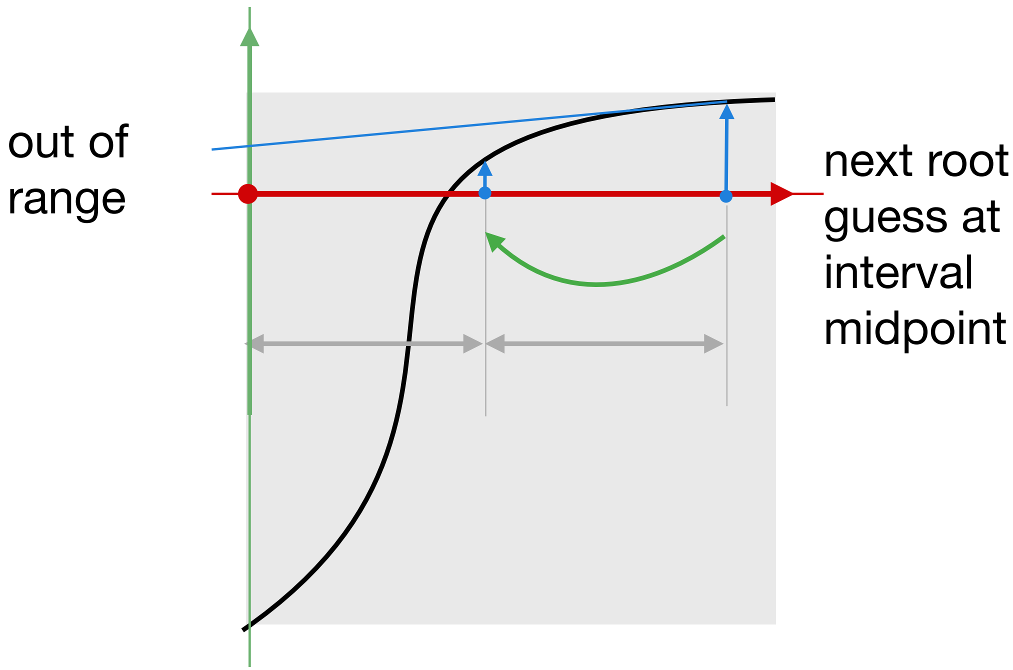
$$t = \frac{A_{\mathrm{r}}(\phi)}{A(\mathcal{C}^{\perp})} \implies \phi = A_{\mathrm{r}}^{-1}\left(t\,A(\mathcal{C}^{\perp})\right) \implies g_t(\phi) = \frac{A_{\mathrm{r}}(\phi)}{A(\mathcal{C}^{\perp})} - t = 0$$

Newton-Raphson root finding

- Susceptible to **out-of-range root guesses**
- Solved via **binary search**



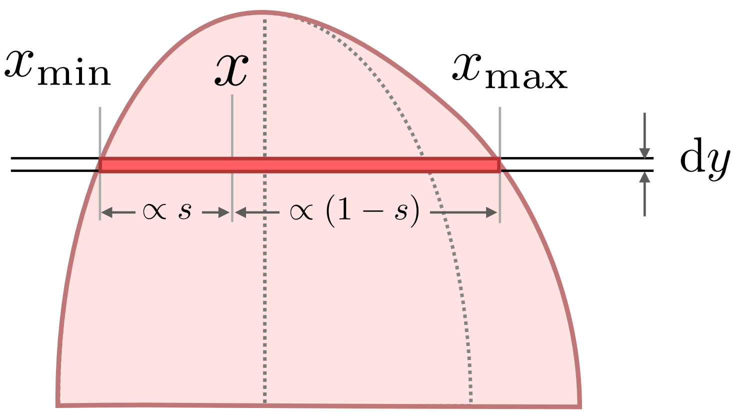
Easy case: **just Newton**



Hard case: **binary split**

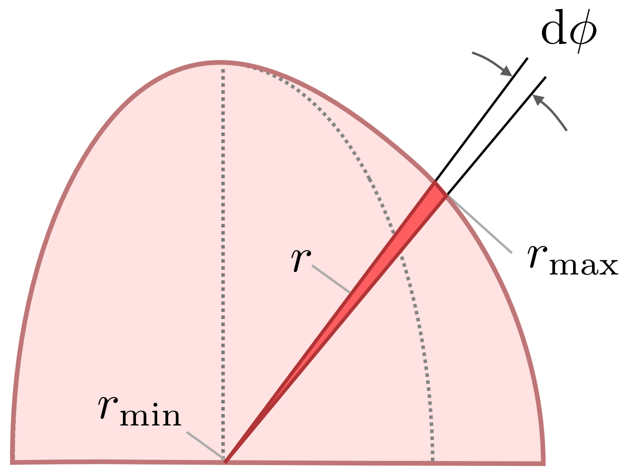
Sampling: find point along line/radius (by using s-coordinate)

- **Parallel map:** constant density along line (differential rectangle)



$$x = (1-s)x_{\min} + s x_{\max}$$

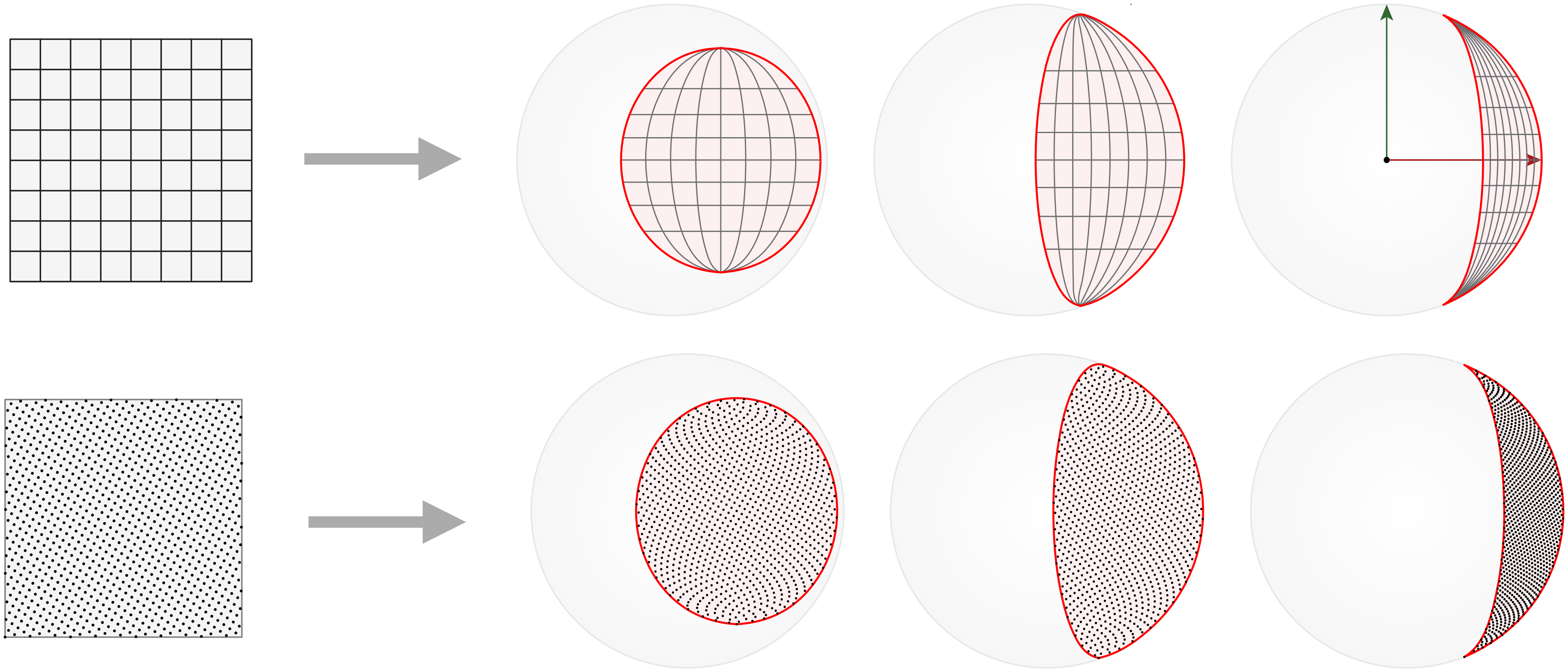
- **Radial map:** linear density along radius segment (differential trapezoid)



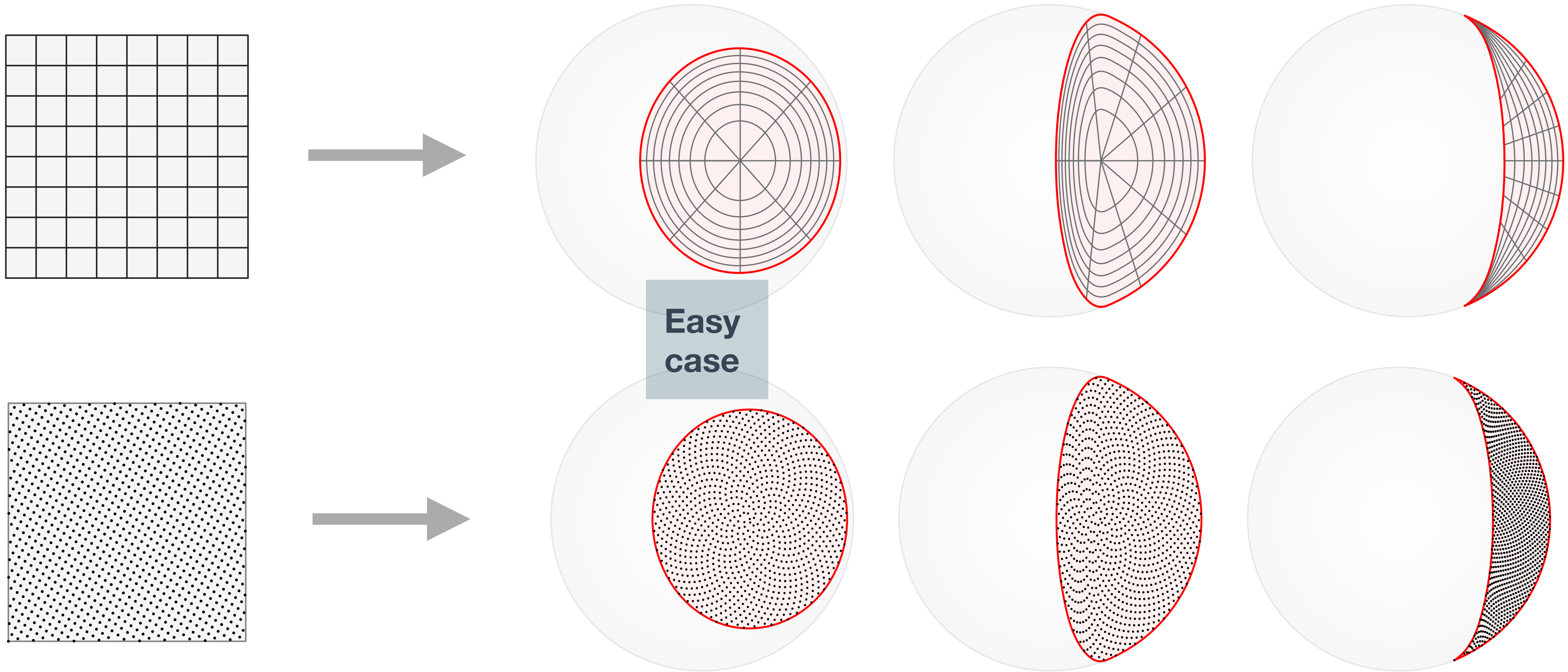
$$r = \sqrt{(1-s)r_{\min}^2 + s r_{\max}^2}$$

Warping evaluation

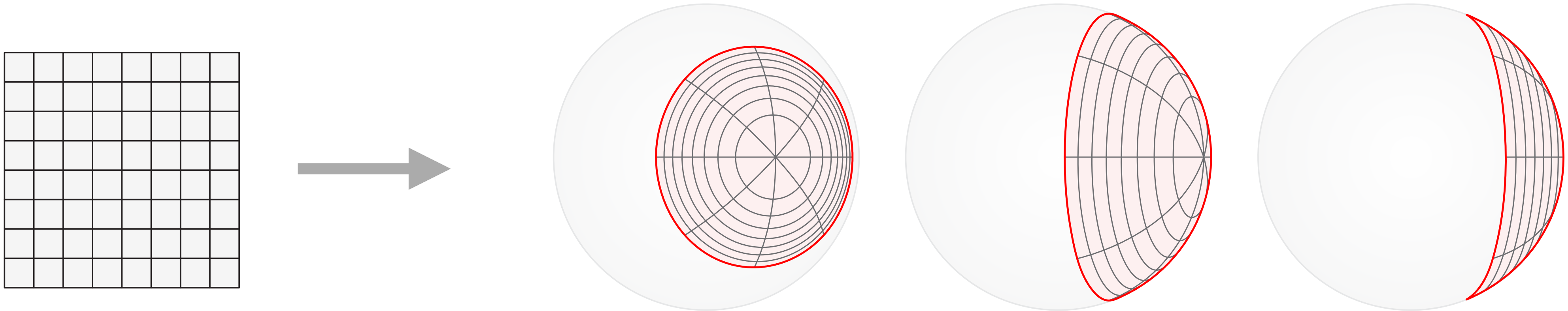
Parallel map

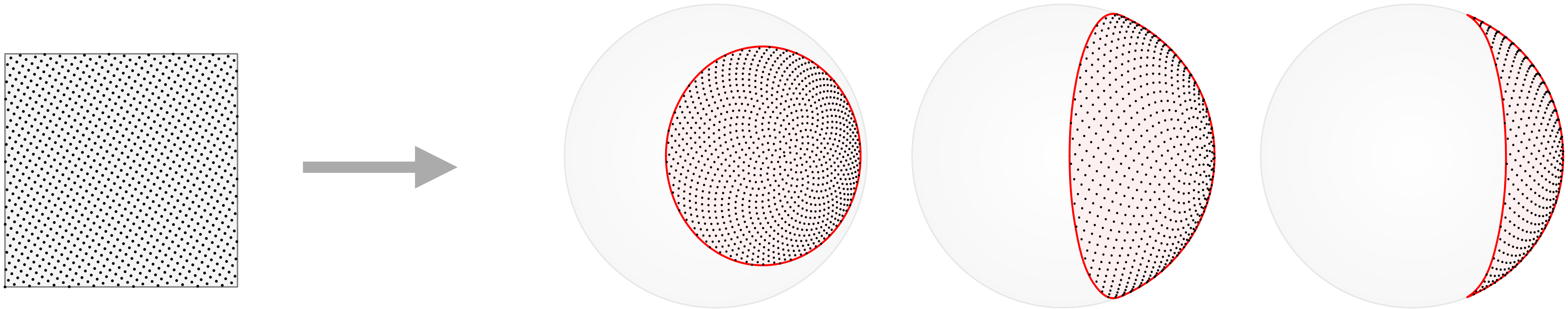


Radial map



Solid-angle map

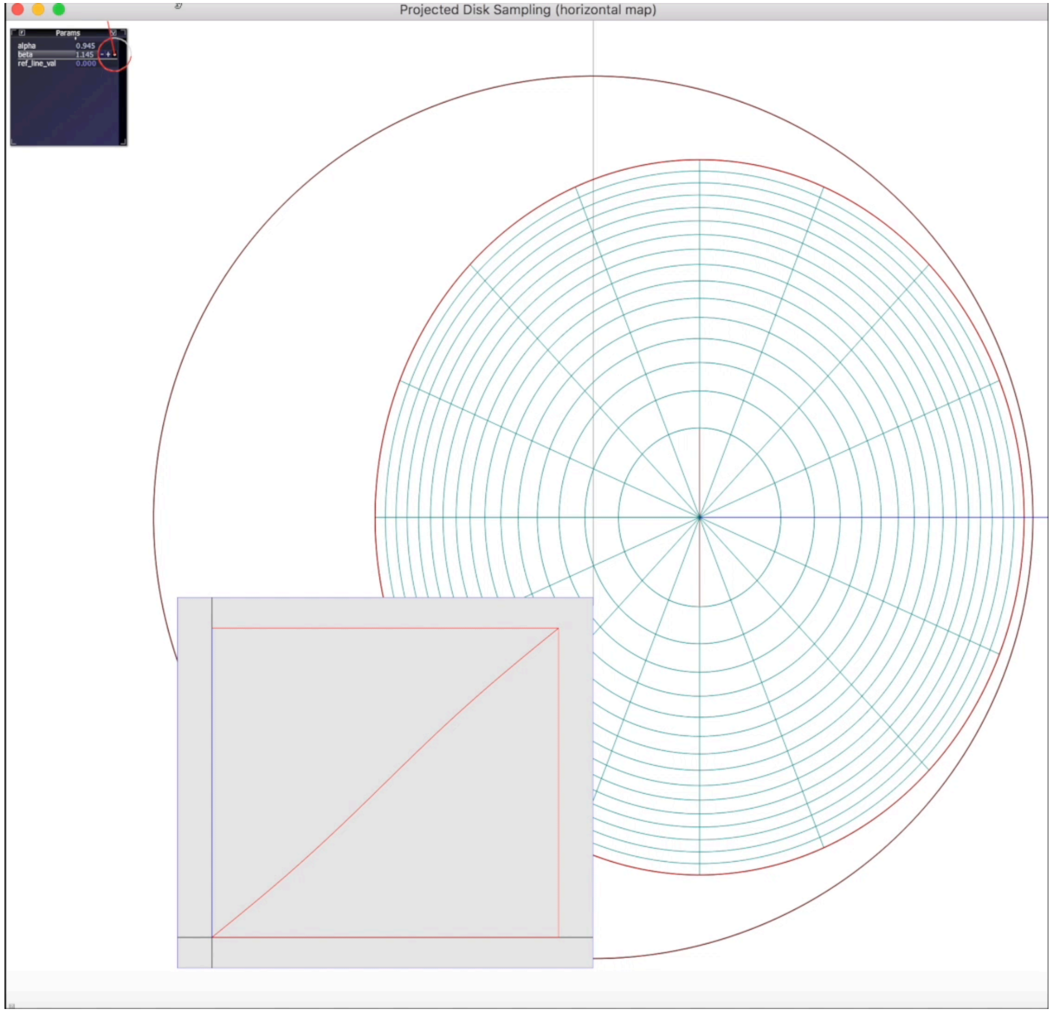




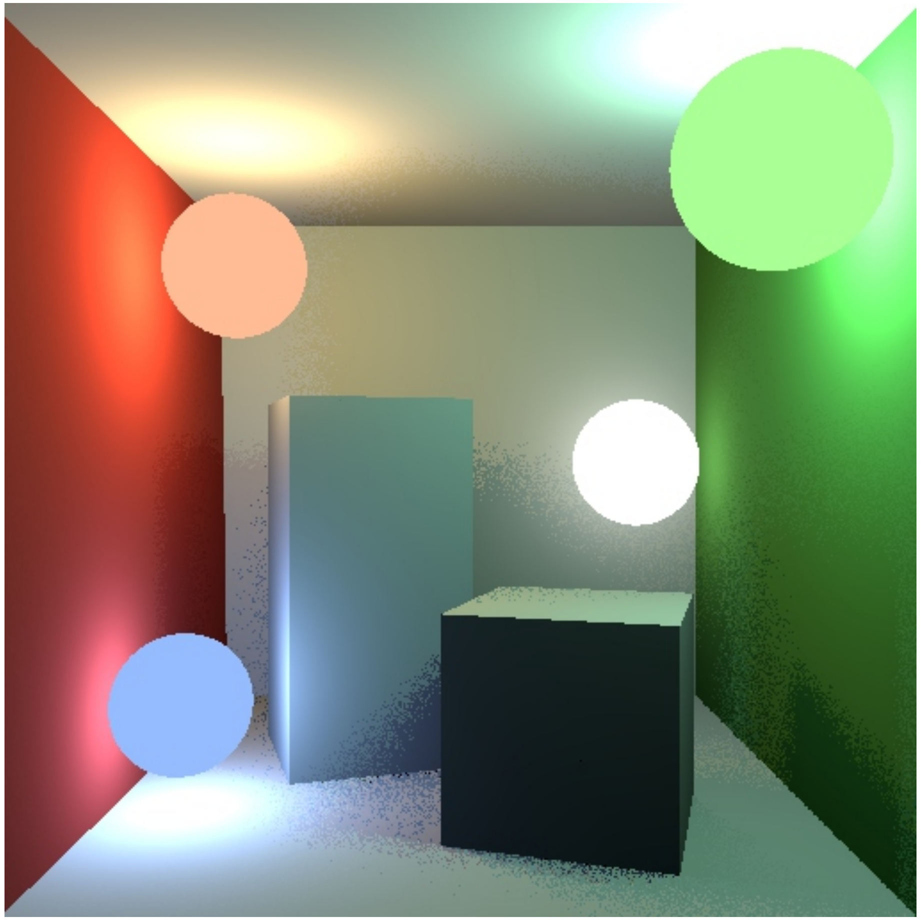
Radial map continuity

continuously changing elevation

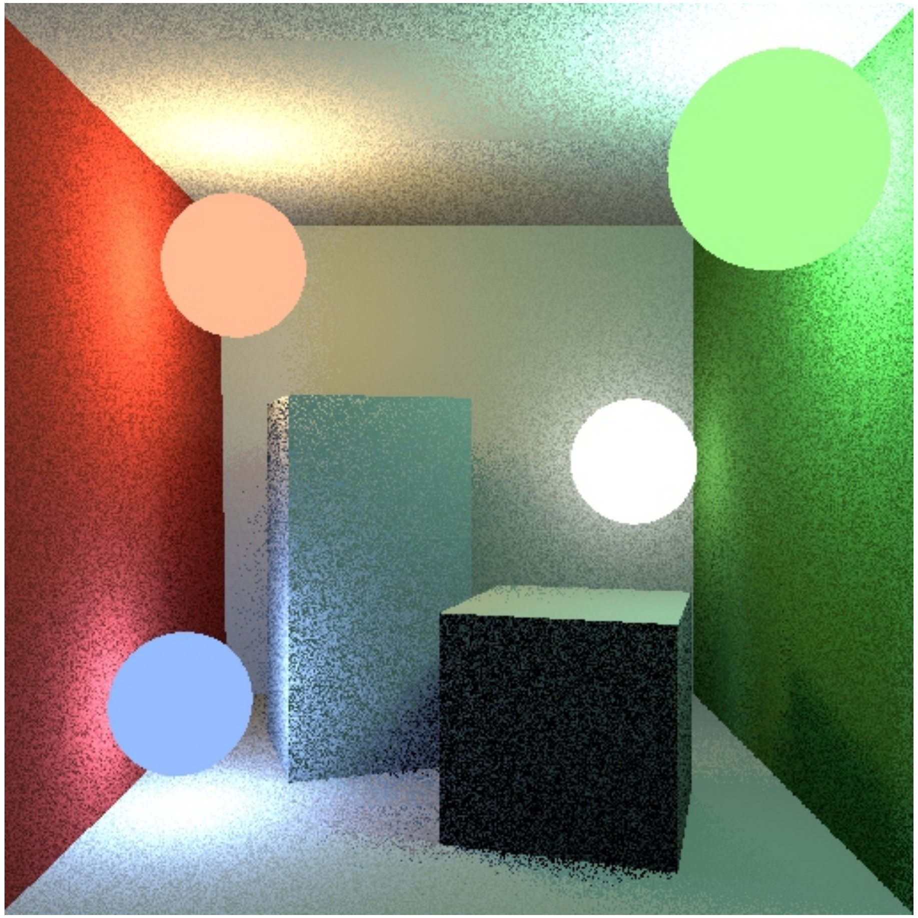
(click for video)



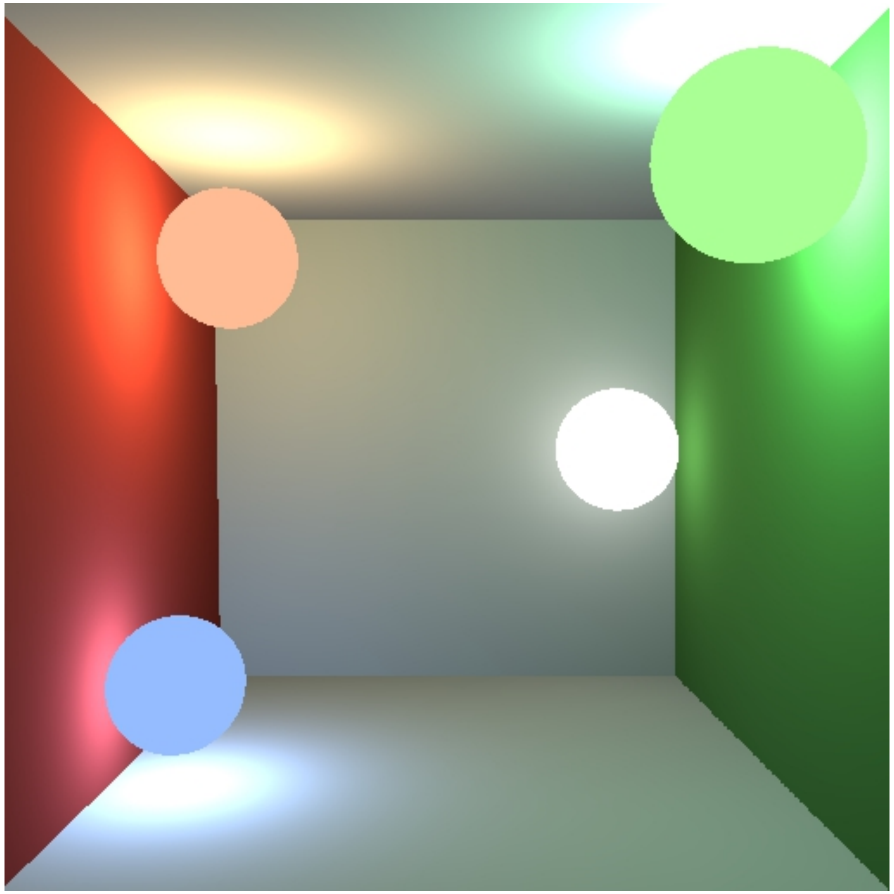
Rendering evaluation



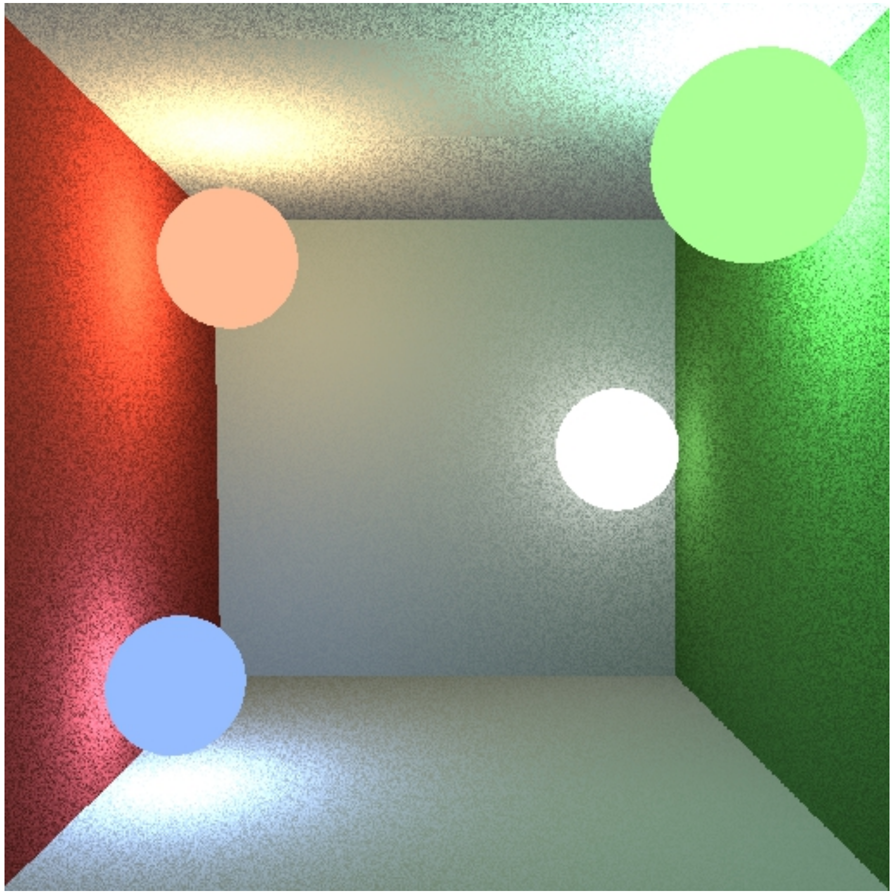
Projected solid angle sampling



Traditional solid angle sampling

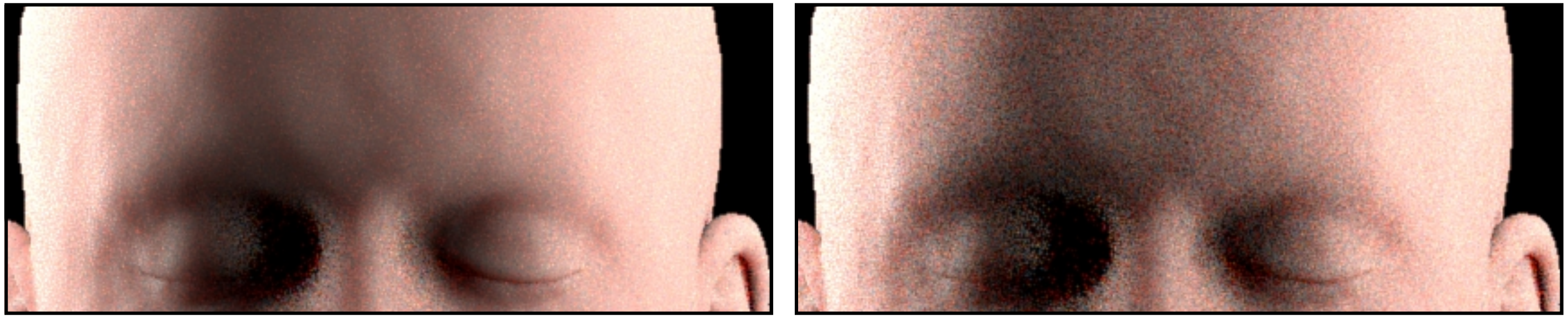


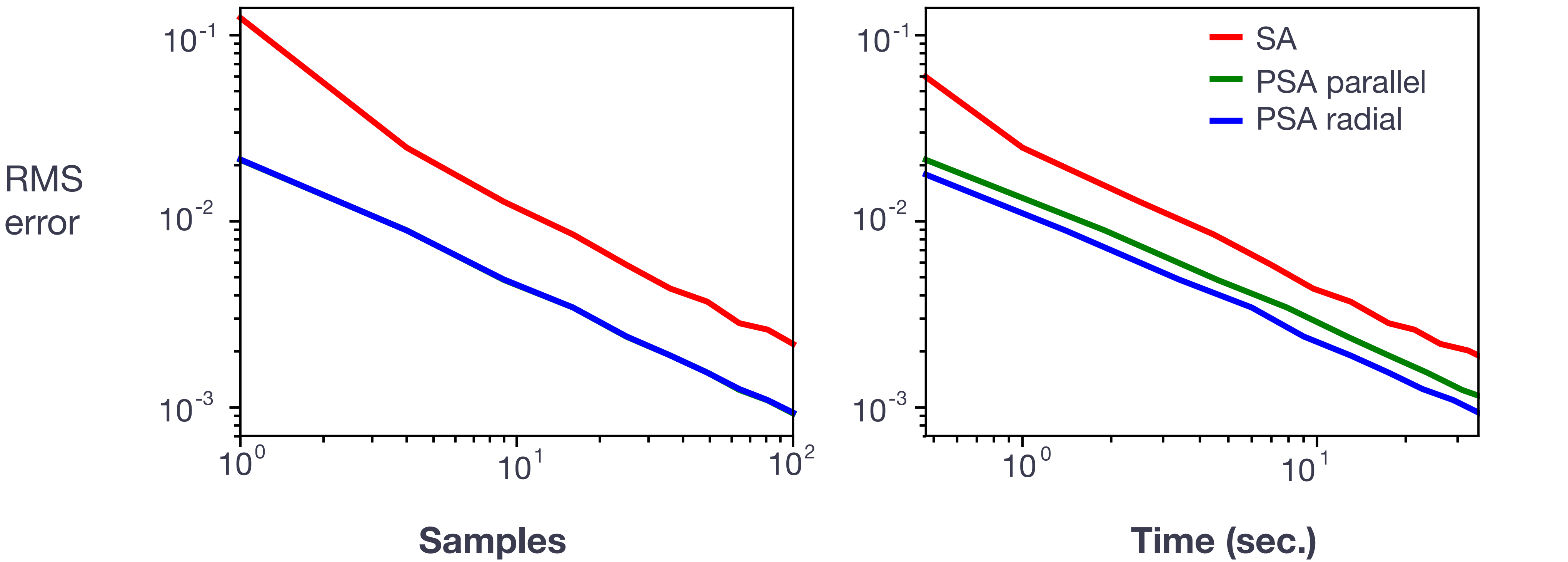
Projected solid angle sampling



Traditional solid angle sampling

Projected solid angle Traditional solid angle




RMS error
10^-1
10^-2
10^-3
10^0
10^1
10^2
Samples
SA
PSA parallel
PSA radial
Time (sec.)

Conclusions

Pros

- Simple, efficient, continuous map
- Reduces noise

Cons (extremely thin lune)

- Slow root-finding convergence
- Not robust in single precision

Future work

- Enhance robustness, convergence rates
- Broader: other shapes, take into account BRDF

Map evaluation code available at:

`github.com/carlos-urena/psc-sampler`

That’s all, thanks.