

Distributed Ray Tracing

© 1996-2017 Josef Pelikán
CGG MFF UK Praha

pepca@cgg.mff.cuni.cz

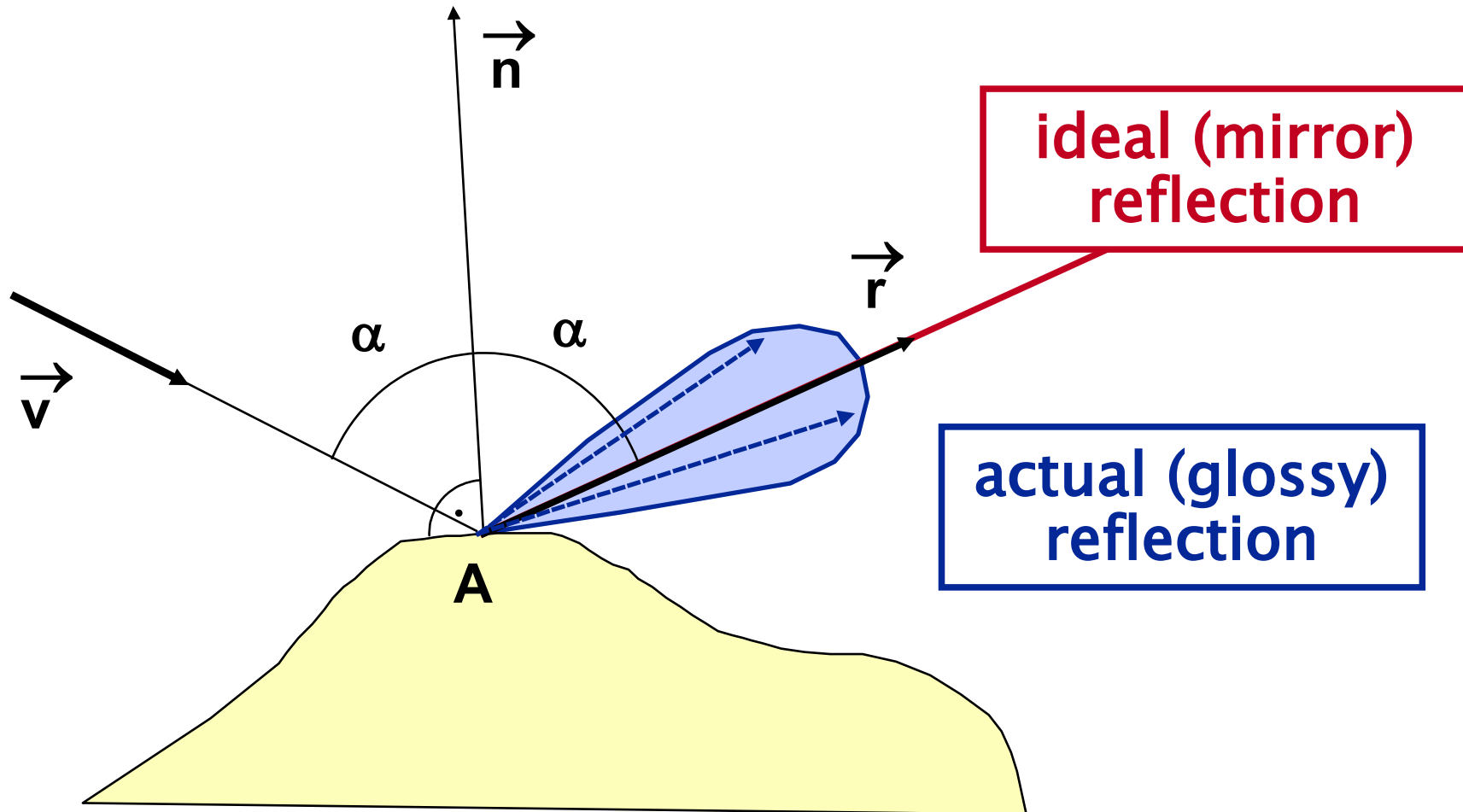
<http://cgg.mff.cuni.cz/~pepca/>



Distributed ray tracing

- ◆ **better image quality** (fidelity)
 - soft shadows, glossy reflections, soft refractions
 - motion blur
 - depth of field imitation
 - light dispersion (index of refraction depends on λ)
- ◆ introducing **new variables** to an image function
 - reflection or refraction angle, wavelength, light source point, lens entry point, time, ..

Glossy reflection





Glossy reflection computation

Sharp reflection:

$$I(\mathbf{V}) = I(\mathbf{R})$$

(one reflected ray)

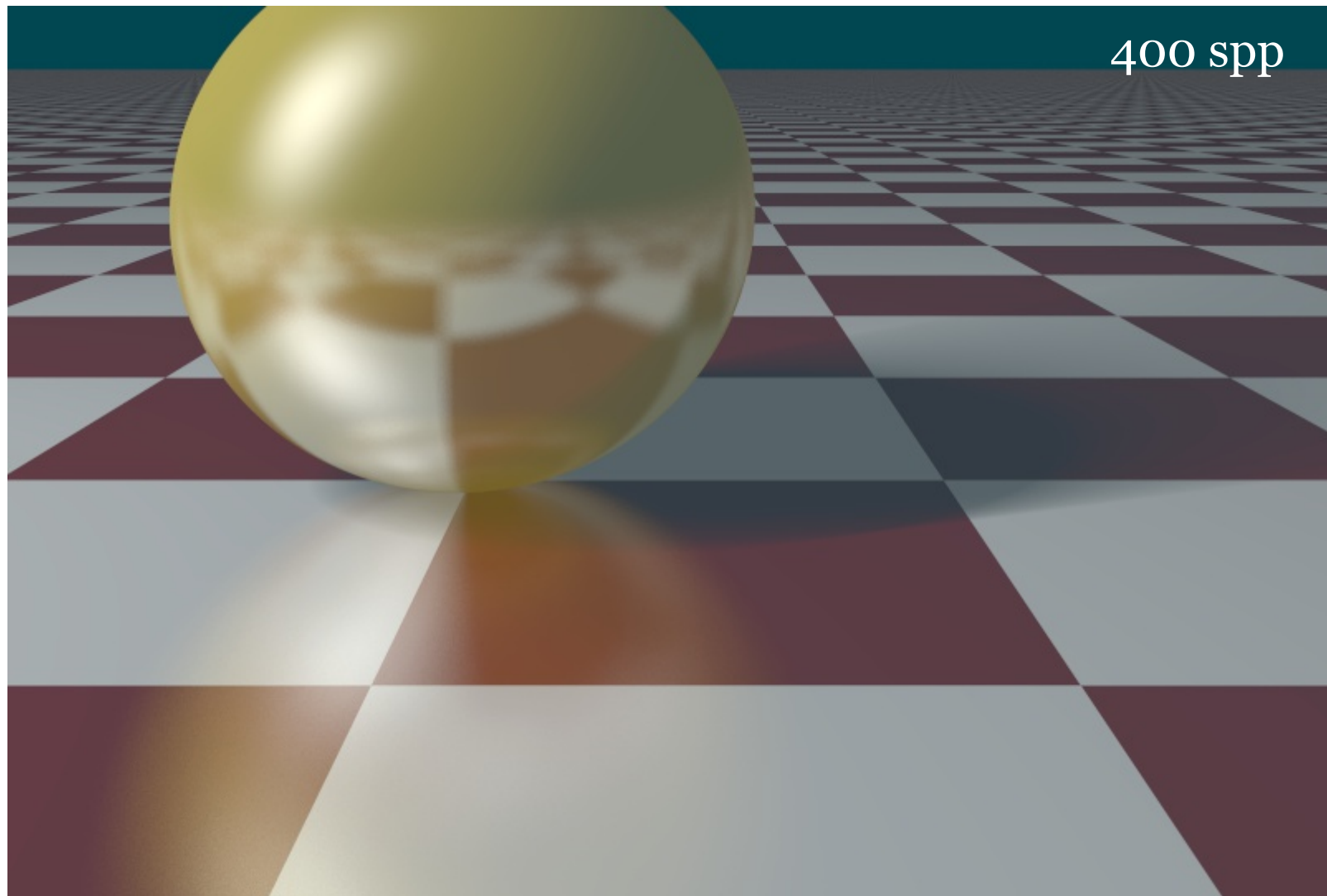
Glossy reflection:

$$I(\mathbf{V}) = \iint_{\text{sphere}} I(\mathbf{R}(\varphi, \theta)) \cdot \text{BRDF}(\alpha, \beta, \varphi, \theta) \, d\varphi \, d\theta$$

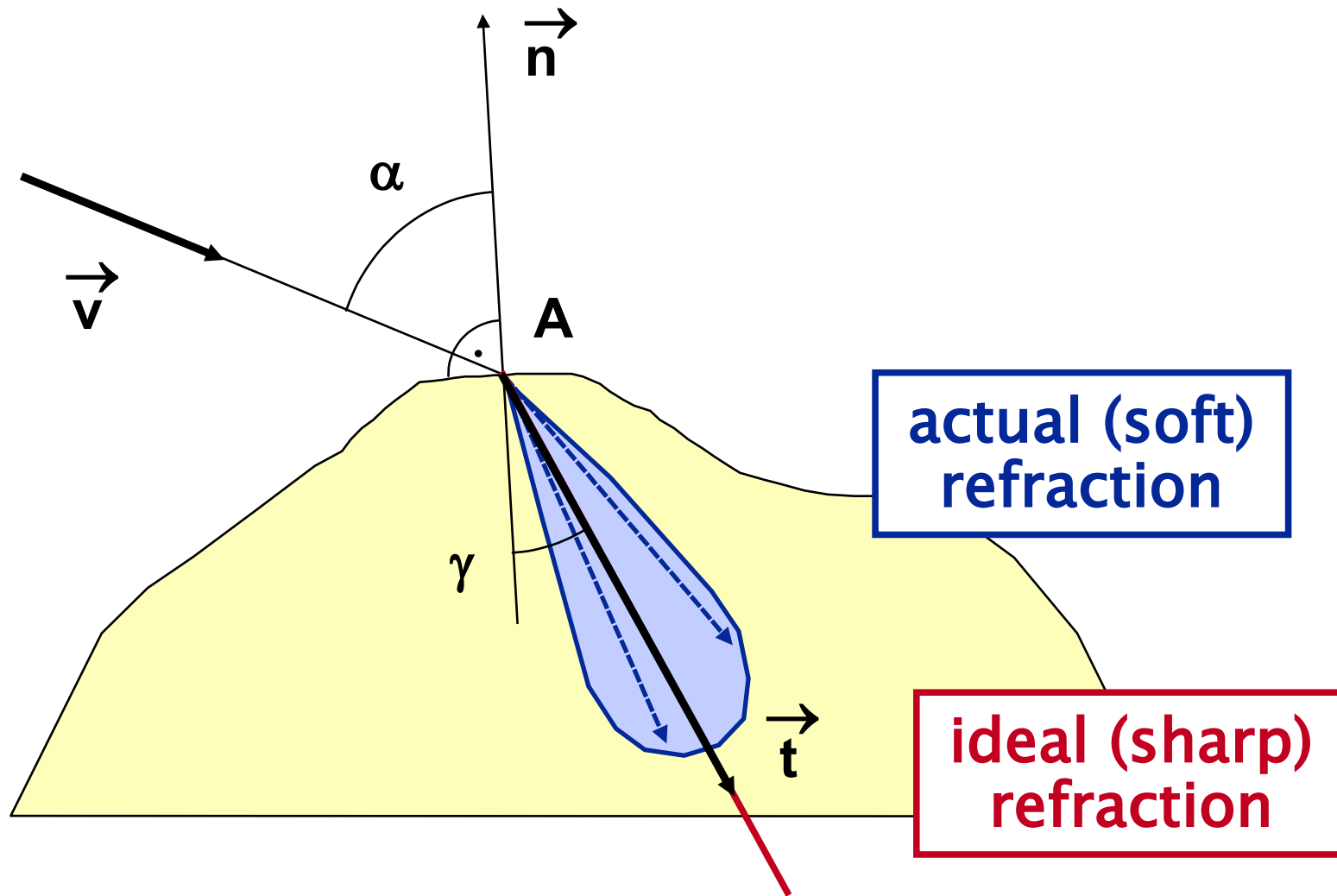
reflectance function

(weighted integral average .. over all reflection angles)

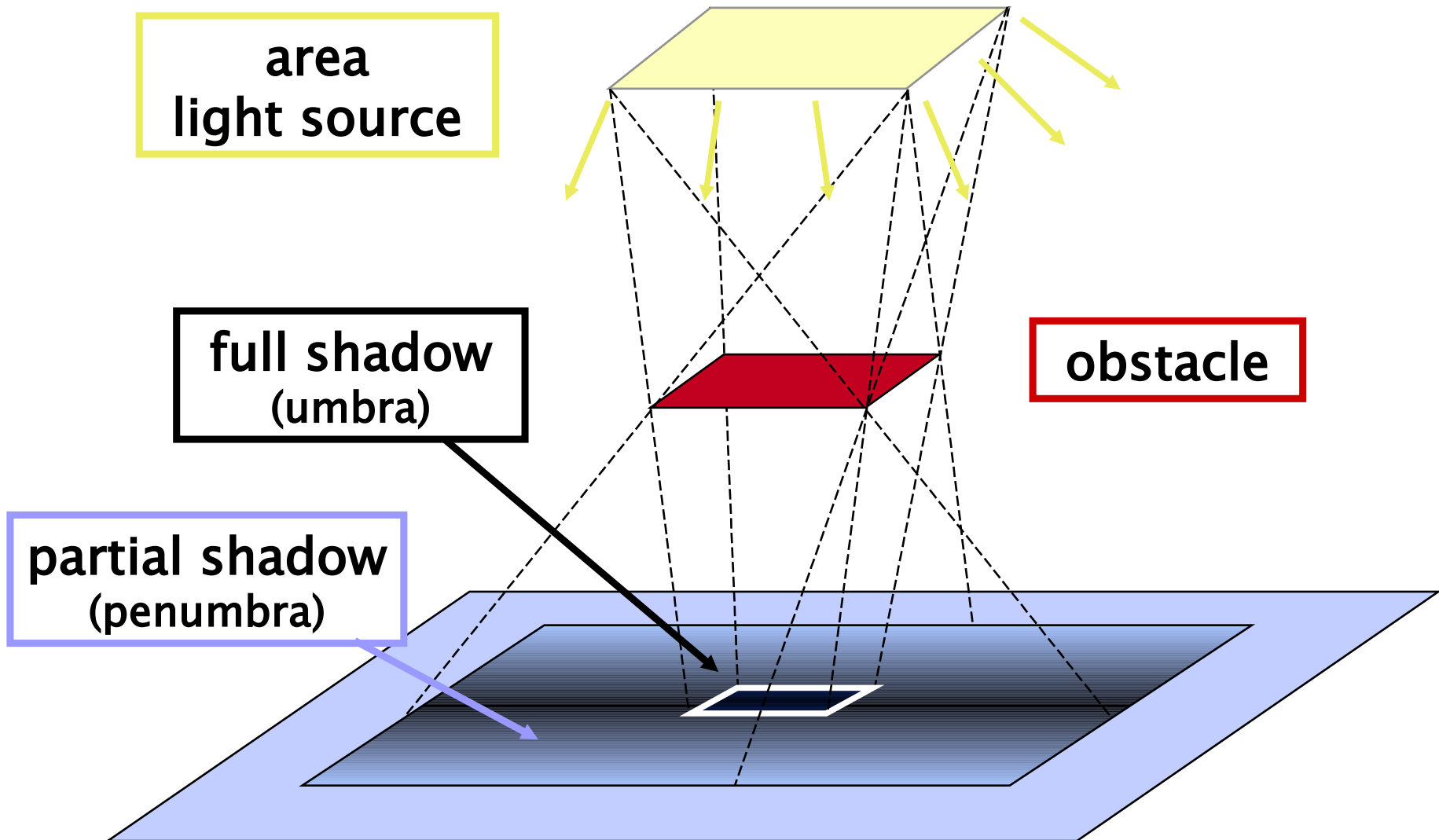
Glossy reflection - example



Soft refraction



Soft shadow





Soft shadow computation

Contribution of a point light source:

$$I(\mathbf{A}) = \begin{cases} I_L & \text{if source is visible from } \mathbf{A} \\ 0 & \text{else} \end{cases}$$

Contribution of an area light source:

$$I(\mathbf{A}) = I_L \cdot S[\%]$$

visible portion of a light source



Soft shadow computation

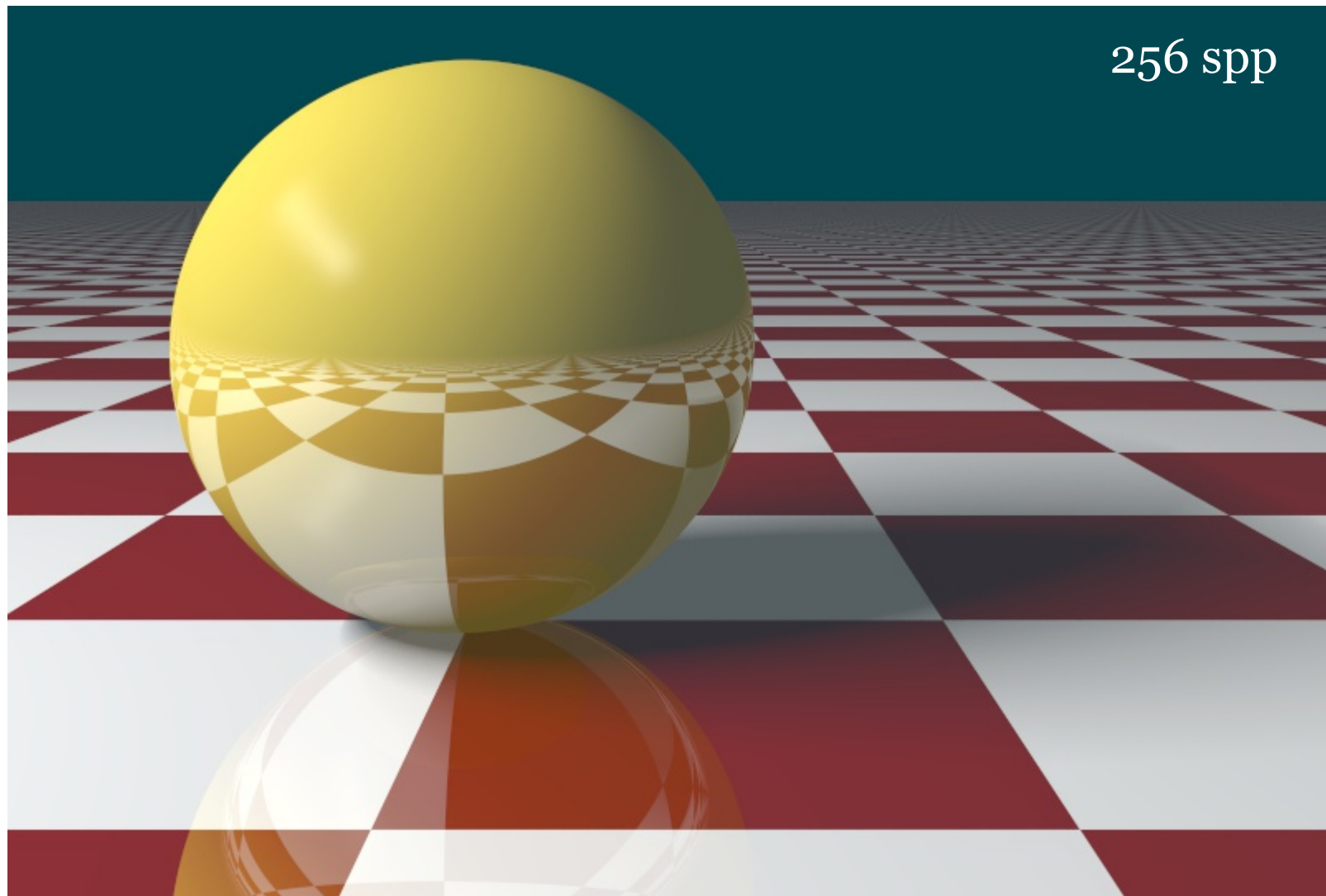
Contribution of an inhomogenous light source:

$$I(\mathbf{A}) = \iint_{\text{light source area}} I_L(\mathbf{u}, \mathbf{v}) \cdot \underline{\text{vis}(\mathbf{A}, \mathbf{u}, \mathbf{v})} \, du \, dv$$

visibility function

$$\underline{\text{vis}(\mathbf{A}, \mathbf{u}, \mathbf{v})} = \begin{cases} 1 & \text{if } \mathbf{S}(\mathbf{u}, \mathbf{v}) \text{ is visible from } \mathbf{A} \\ 0 & \text{else} \end{cases}$$

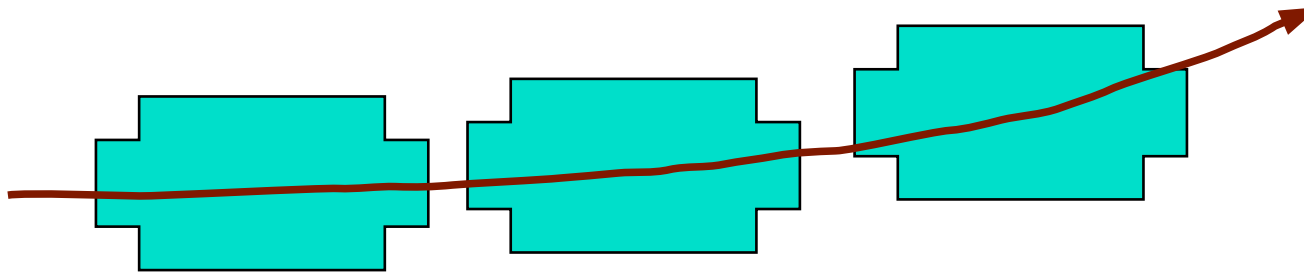
Soft shadow - example





Motion blur

path of the solid $s(t)$



rendered interval:
(shutter open time)

$[t_1, t_2]$

scene rendering in time t :

$f(t) = f(x, y, t)$



Motion blur

General motion blur:

$$\mathbf{f}_{\text{blurr}} = \int_{t_1}^{t_2} \mathbf{f}(t) dt$$



Motion blur

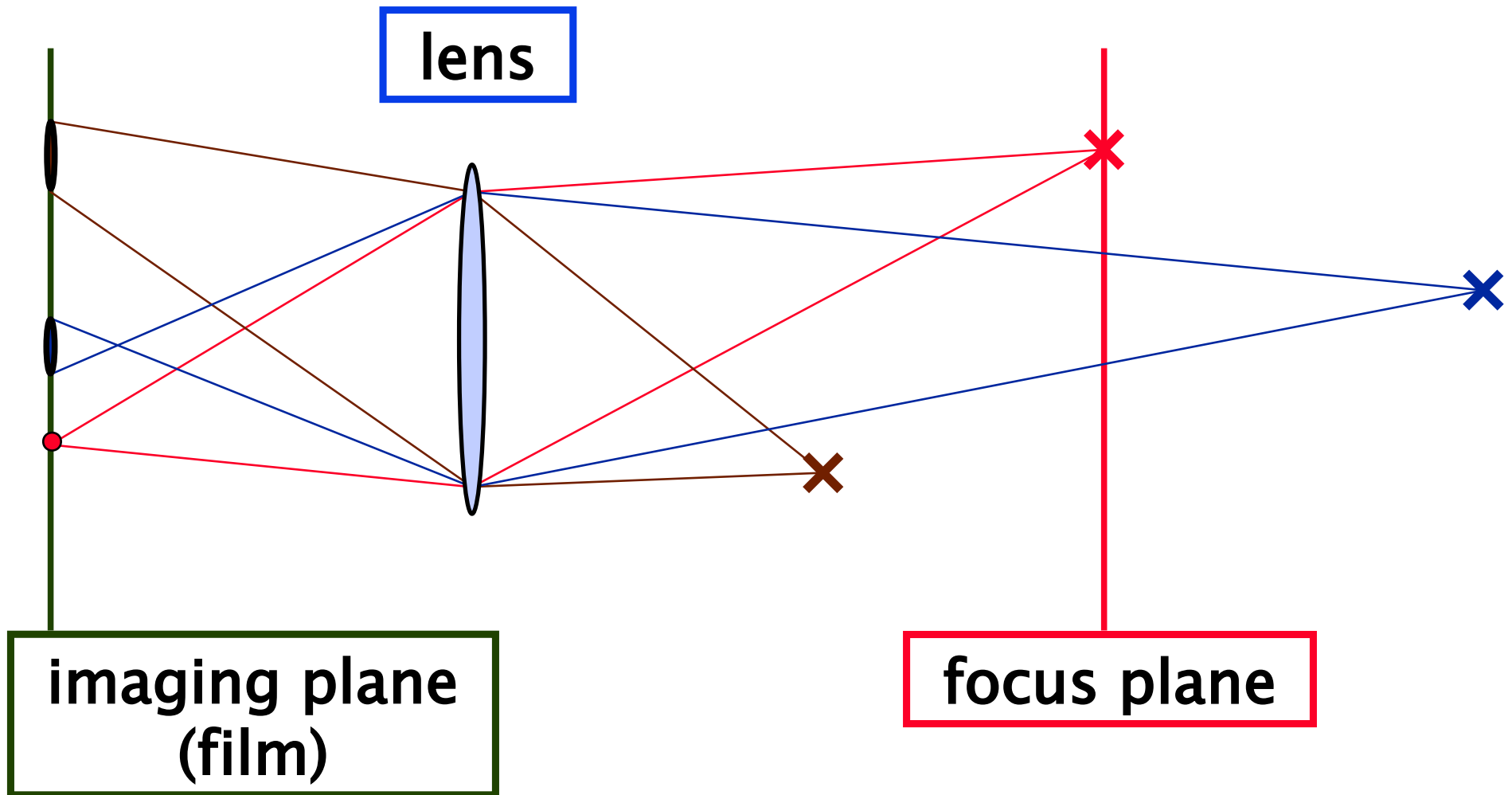
Scene with single moving object:

$$\mathbf{f}_{\text{blurr}} = \frac{\int_{t_1}^{t_2} \mathbf{f}(\mathbf{t}) \cdot |\mathbf{s}'(\mathbf{t})|^{-1} dt}{\int_{t_1}^{t_2} |\mathbf{s}'(\mathbf{t})|^{-1} dt}$$

$$|\mathbf{s}'(\mathbf{t})| \neq 0 \text{ on } \langle \mathbf{t}_1, \mathbf{t}_2 \rangle$$

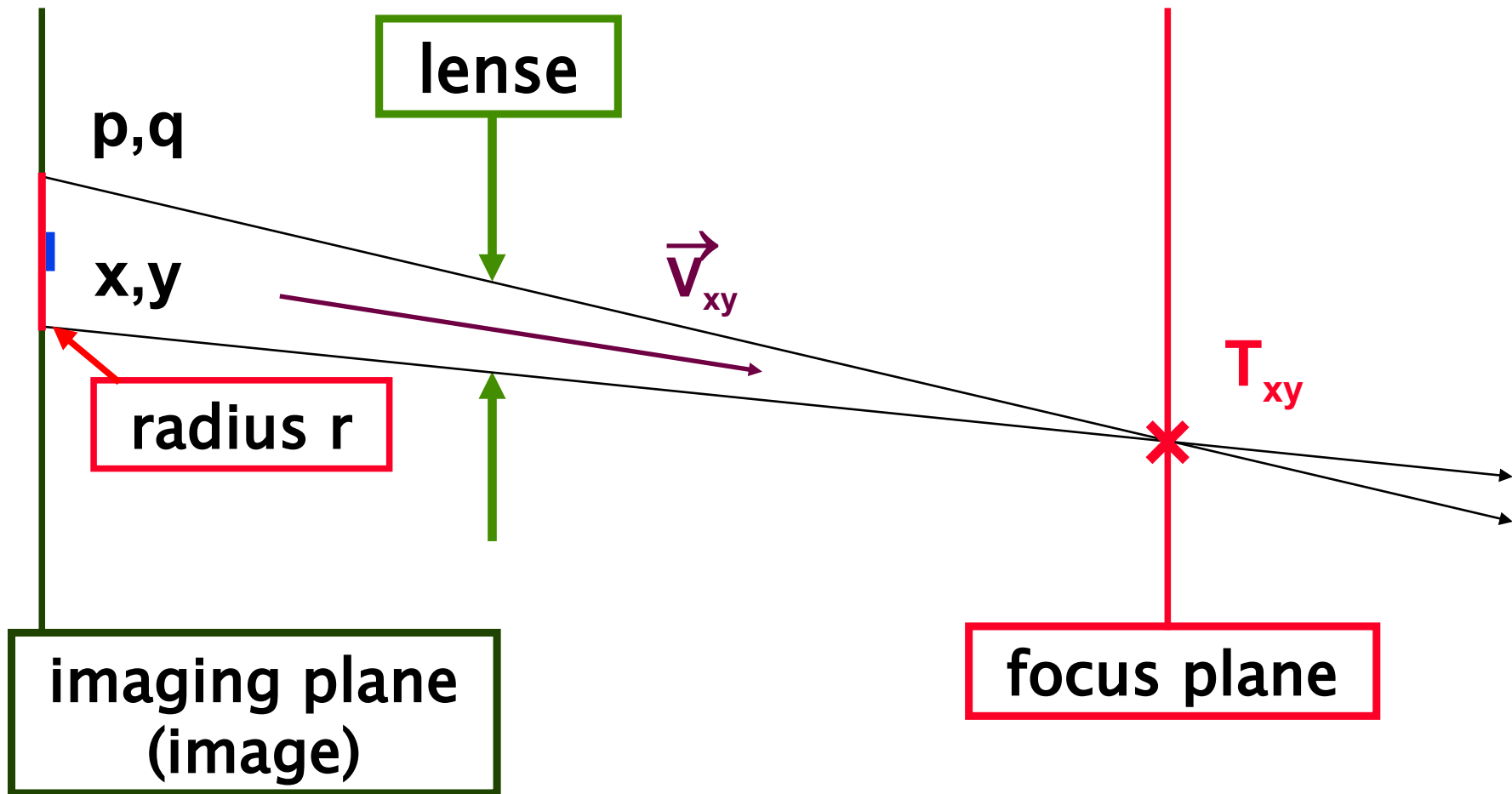


Depth of field





Geometric simplification





Depth of field computation

Pinhole camera model:

$$f(\mathbf{x}, \mathbf{y}) = I(\mathbf{V}_{xy})$$

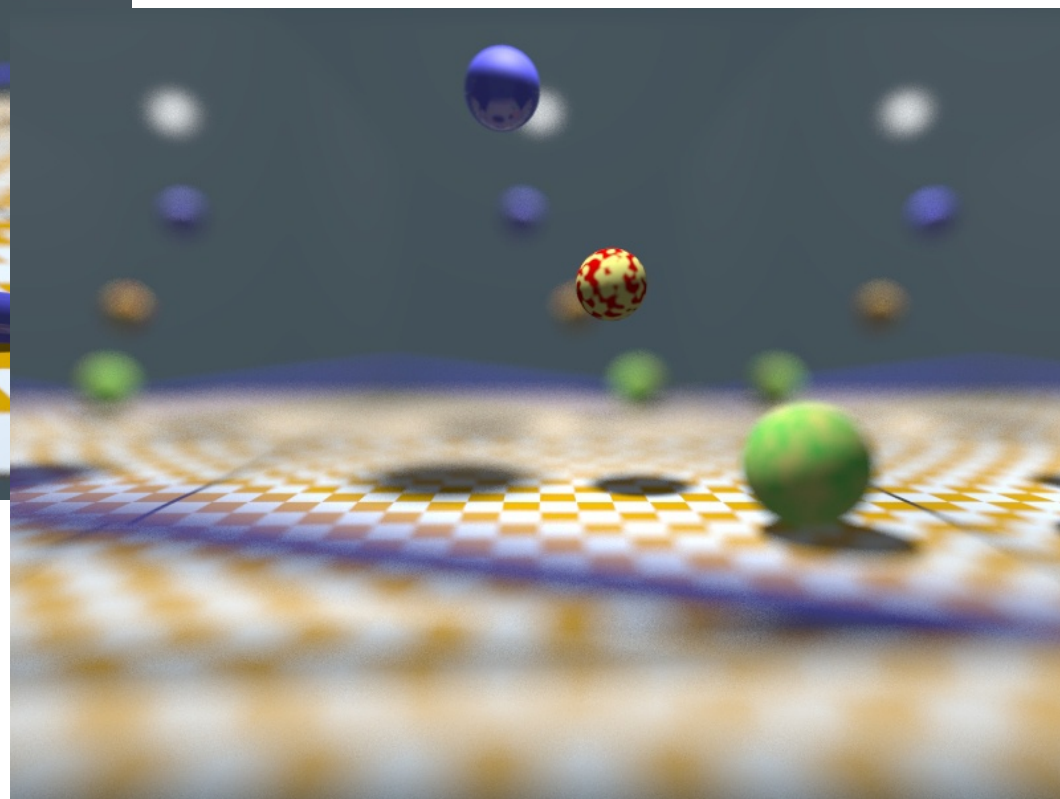
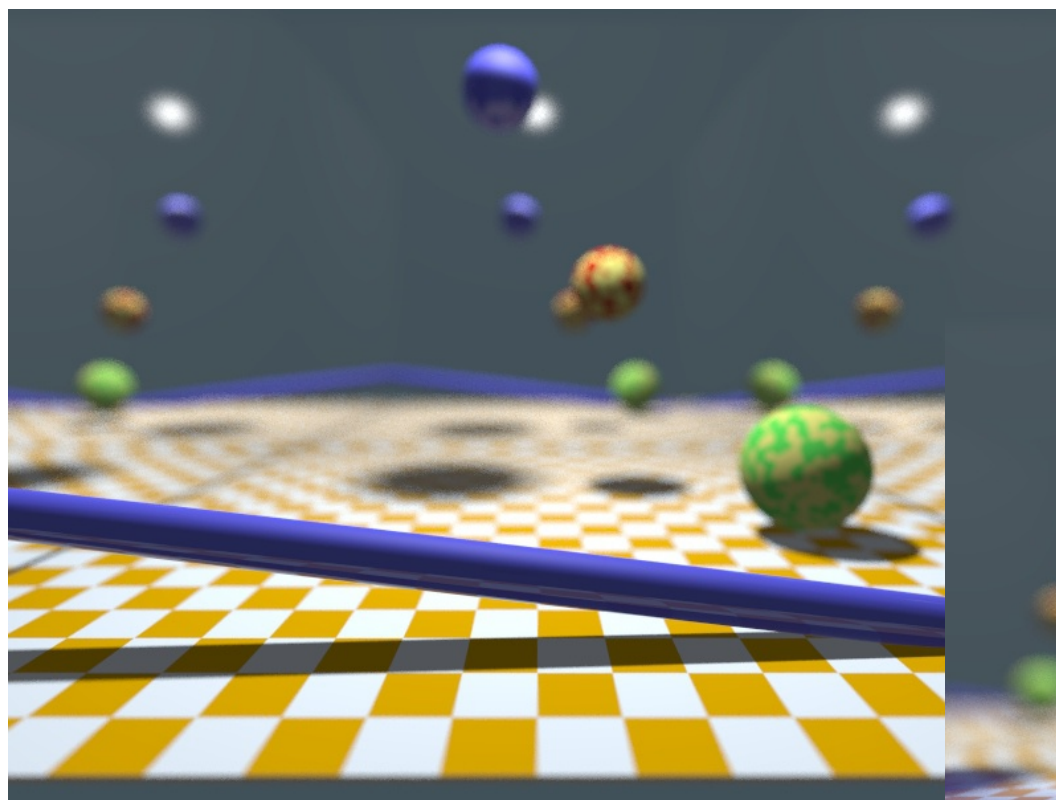
$$\mathbf{V}_{xy} = \mathbf{T}_{xy} - [\mathbf{x}, \mathbf{y}, 0]$$

$$\mathbf{V}_{pq} = \mathbf{T}_{xy} - [\mathbf{p}, \mathbf{q}, 0]$$

Lens with finite aperture:

$$f(\mathbf{x}, \mathbf{y}) = \int_{\text{circle around } [\mathbf{x}, \mathbf{y}]} I(\mathbf{V}_{pq}) \, dp \, dq$$

Depth of field – examples





Light dispersion

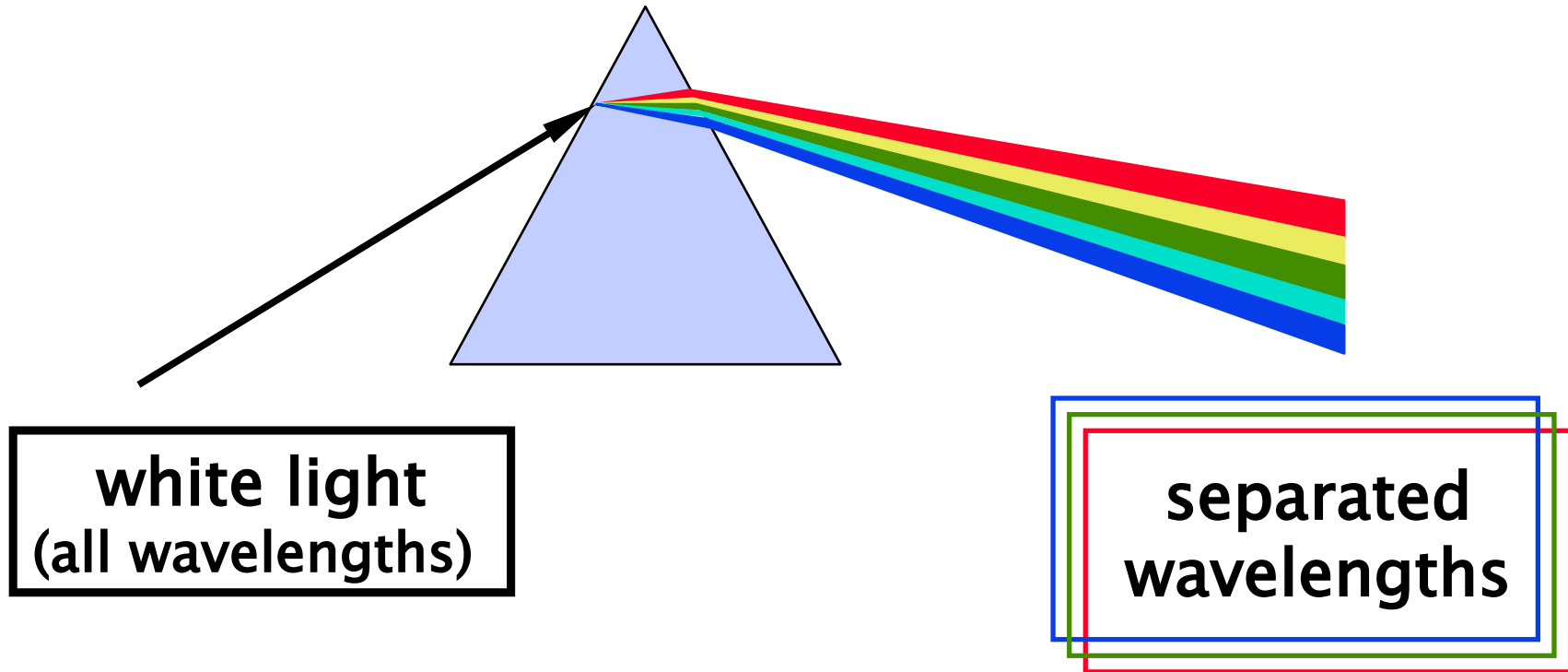


image function: $f(\lambda) = f(x, y, \lambda)$



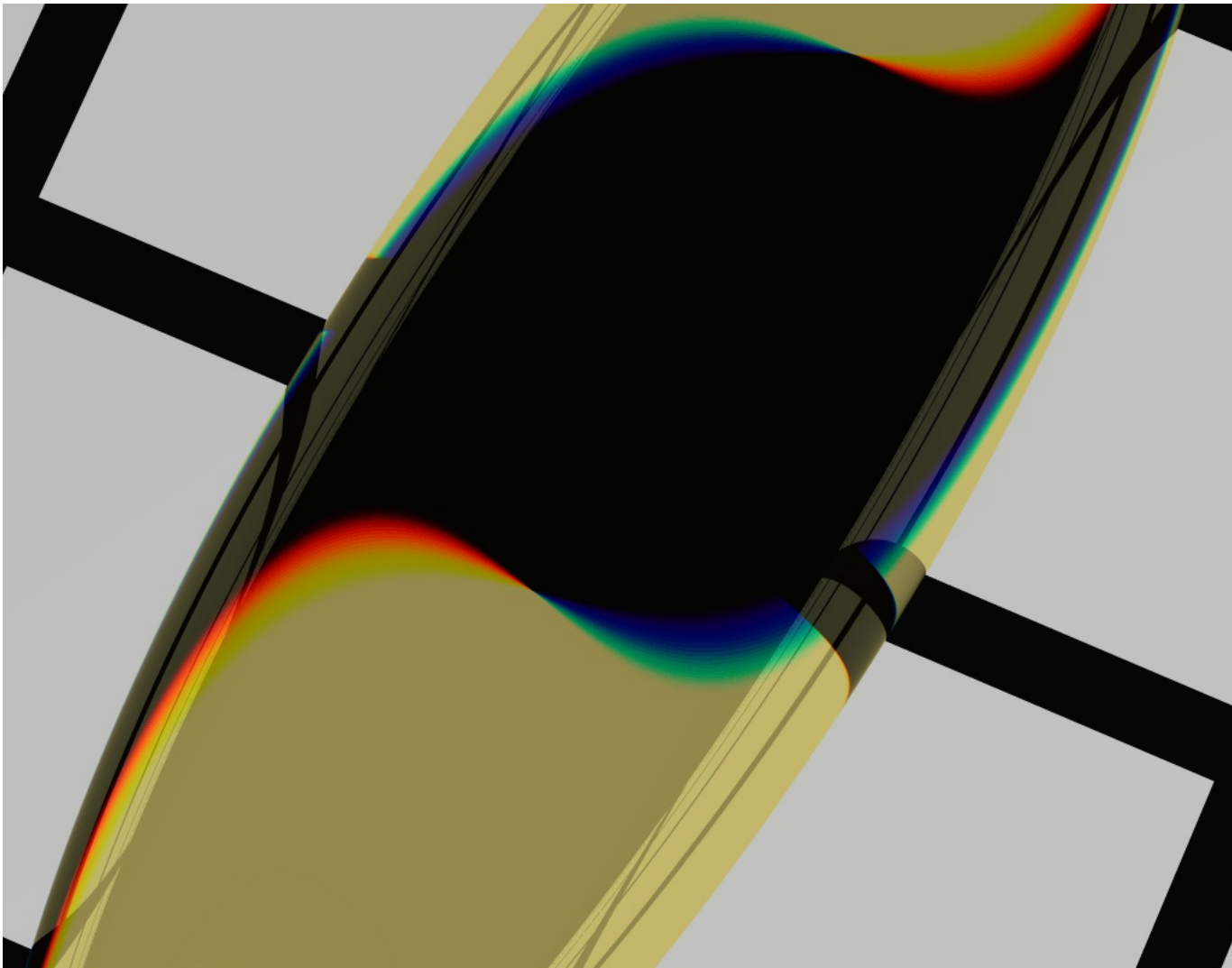
Light dispersion computation

Pixel RGB color from spectral distribution:

$$\begin{aligned} \mathbf{R}(\mathbf{x}, \mathbf{y}) &= \int_{\text{spectrum}} \mathbf{f}(\mathbf{x}, \mathbf{y}, \lambda) \cdot \mathbf{R}(\lambda) \, d\lambda \\ \mathbf{G}(\mathbf{x}, \mathbf{y}) &= \int_{\text{spectrum}} \mathbf{f}(\mathbf{x}, \mathbf{y}, \lambda) \cdot \mathbf{G}(\lambda) \, d\lambda \\ \mathbf{B}(\mathbf{x}, \mathbf{y}) &= \int_{\text{spectrum}} \mathbf{f}(\mathbf{x}, \mathbf{y}, \lambda) \cdot \mathbf{B}(\lambda) \, d\lambda \end{aligned}$$

trichromatic spectral coefficients

Light dispersion - example





Implementation

- integral averaging is done **stochastically** (Monte-Carlo methods)
 - finite number of point samples (rays)
 - integral is estimated by a [weighted] sum
- **weighted** integral average
 - uniform sampling and appropriate weight function
 - nonuniform sampling (using the right density/PDF)



Combining methods

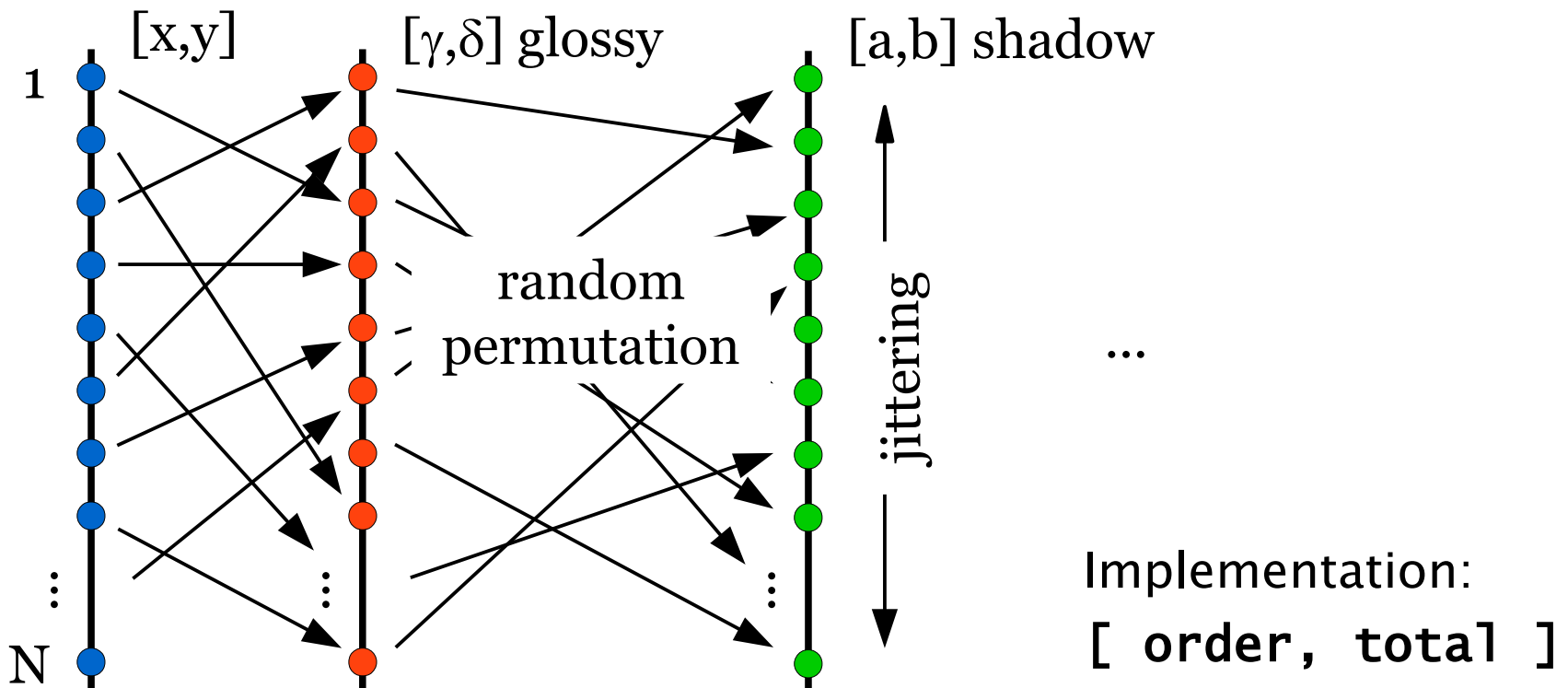
- any methods can be **combined**
 - with anti-aliasing as well
 - higher order integrals – e.g. dimension 10:
 - anti-aliasing (2), depth of field (2), glossy reflection (2), soft shadows (2), motion blur (1), light dispersion (1)

- **sampling method:**
 - jittering
 - independent jittering (“N rooks”) in form of **hidden sampling**
 - adaptive sampling



Hidden sampling

- number of **samples per pix** (primary rays) is defined
 - every inner component is sampling independently
 - arbitrary number of additional (sampled) dimensions





References

**A. Glassner: *An Introduction to Ray Tracing*,
Academic Press, London 1989, 171-199**

**A. Watt, M. Watt: *Advanced Animation and
Rendering Techniques*, Addison-Wesley,
Wokingham 1992, 262-265**