Monte Carlo rendering

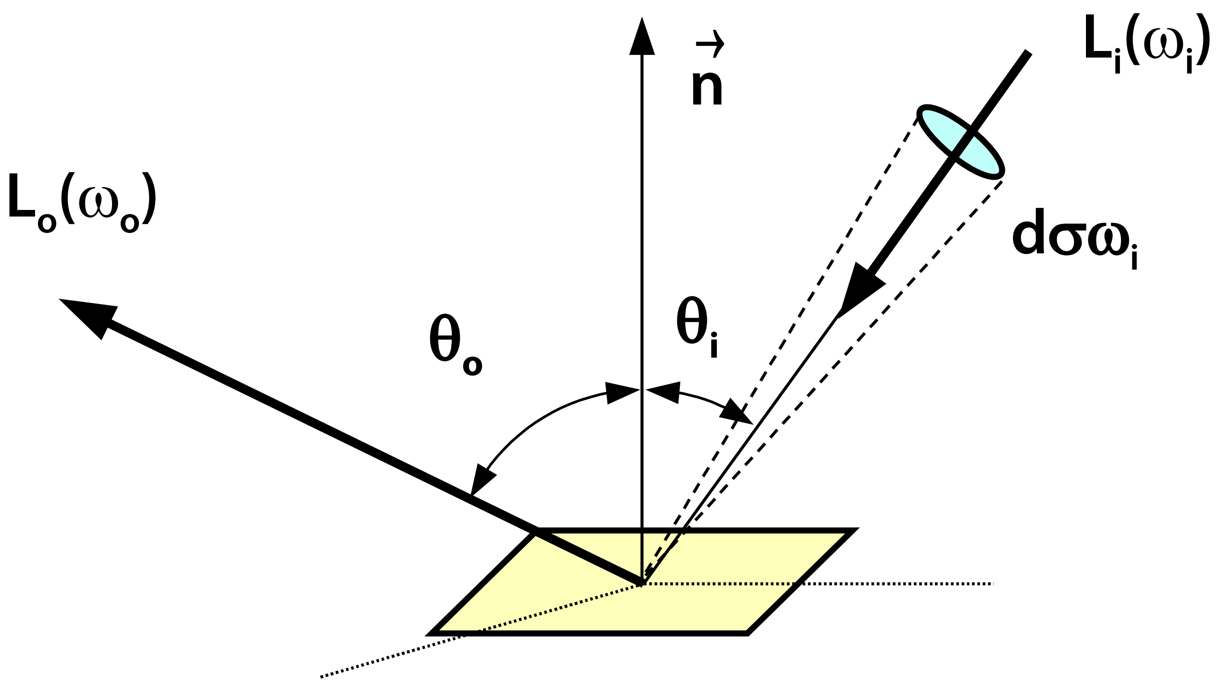
CGG MFF UK Praha

pepca@cgg.mff.cuni.cz

https://cgg.mff.cuni.cz/~pepca/

Monte-Carlo in rendering (BSDF)

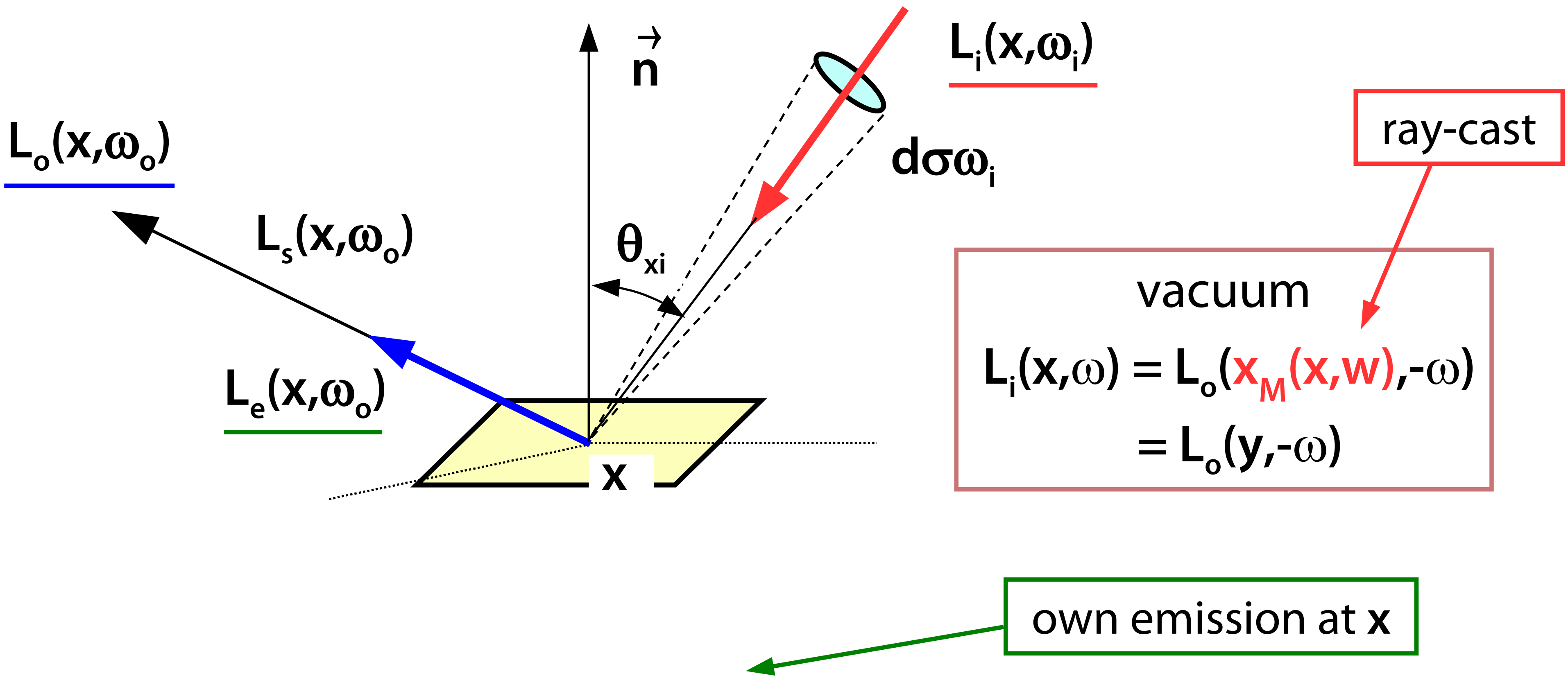
(„Bidirectional Scattering Distribution Function“, older term: BRDF)



$$f_s(\omega_i \to \omega_o) = \frac{d\,L_o(\omega_o)}{d\,E(\omega_i)} = \frac{d\,L_o(\omega_o)}{L_i(\omega_i)\,\cos\theta_i\,d\,\sigma^{\perp}(\omega_i)}$$

Local rendering equation (OVTIGRE)

("Outgoing, Vacuum, Time-Invariant, Gray Radiance Equation")



$$L_o(x,\omega_o) = L_e(x,\omega_o) + \int L_o(y,-\omega_i)\cdot f_s(x,\omega_i \rightarrow \omega_o)\cdot d\sigma_x^{\perp}(\omega_i)$$

Light propagation operators

Rendering equation for **radiance** (operators)

$$\mathbf{L} = \mathbf{e} + \mathbf{T}\mathbf{L}$$

$$\mathbf{L} = \mathbf{e} + \mathbf{T}\mathbf{e} + \mathbf{T}^2\mathbf{e} + \mathbf{T}^3\mathbf{e} + \ldots$$

Integral **operator T** can be decomposed into diffuse (**D**) and specular (**S**) components

$$\mathbf{T} = \mathbf{D} + \mathbf{S}$$

$$\mathbf{L} = \mathbf{e} + (\mathbf{D} + \mathbf{S})\mathbf{e} + (\mathbf{D} + \mathbf{S})^2\mathbf{e} + \ldots$$

$$\mathbf{L} = \mathbf{e} + \mathbf{De} + \mathbf{Se} + \mathbf{DDe} + \mathbf{DSe} + \mathbf{SDe} + \mathbf{SSe} + \ldots$$

Regular expression alphabet

Light source L

Diffuse reflection D

- Lambertian reflection (omnidirectional)

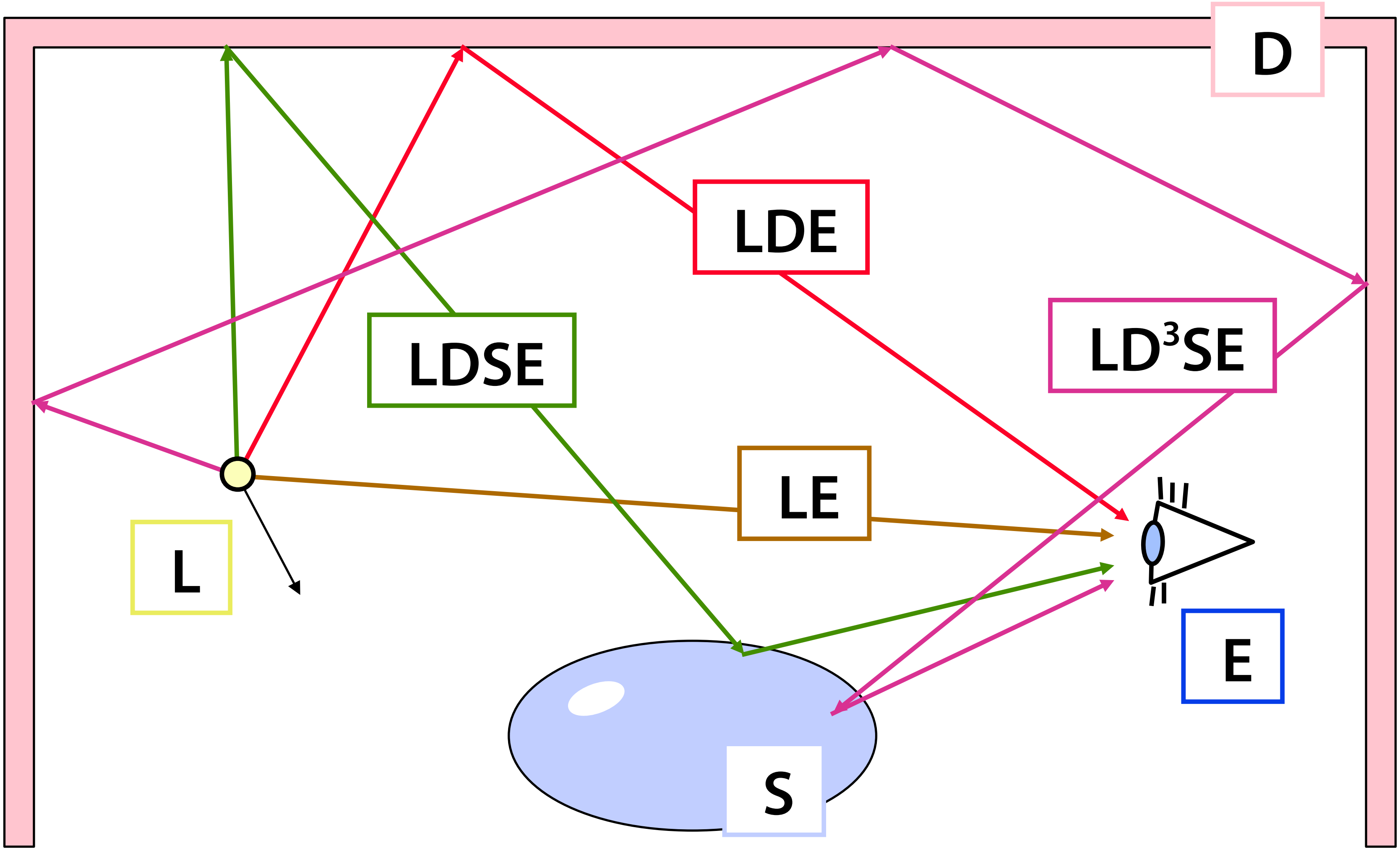
Specular reflection S

- directional reflection, highlight – directional part of a BRDF
- idealized **mirror reflection S_M**

Viewer's eye E

- contribution to the result image

Light propagation paths



Classical rendering methods I

Shading with highlights and **shadows** (e.g. Phong model on GPU): **L(D|S)E**

– shadow casting is often ignored

Recursive ray-tracing (Whitted): $\mathbf{L[D|S]S_M{}^{*}E}$

– the first specular reflection is accurate (reflectance model from a light source), the rest is replaced by mirror reflections

Classical rendering methods II

Distributed ray-tracing (Cook): **L[D]S*E**

– all specular reflections are estimated correctly

Basic **radiosity**: **LD*E**

– diffuse materials (reflections) only

All possible light paths: **L(D|S)*E**

– correct solution of rendering equation (Kajiya – Path tracing)

Monte-Carlo rendering

The integral in the rendering equation is often **multi-dimensional**

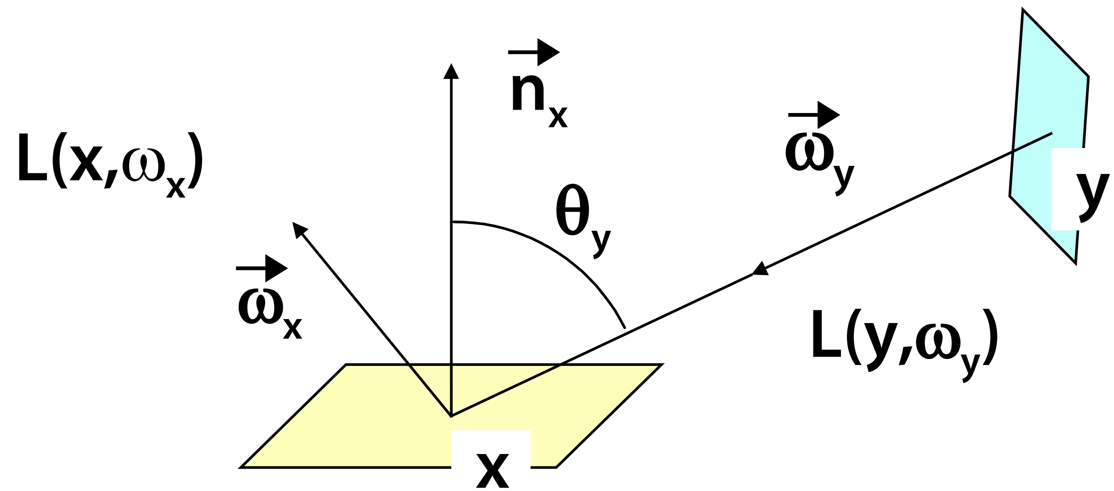
- anti-aliasing, depth of field, glossy reflection, motion blur…
- Monte-Carlo methods are not sensitive to higher dimensions

Integrands have many **discontinuities**

No high precision is required

- human visual systém is not absolutely sensitive
- precision of about 0.1-1.0 % is sufficient in most cases

Rendering equation for radiance



$$L(x,\omega_x) =$$

$$= L_e(x,\omega_x) + \int_{\Omega_x^{-1}} f(x,\omega_y \to \omega_x) \cdot L(y,\omega_y) \cdot \cos\theta_y \; d\omega_y$$

Radiant flux through a set **S** (e.g. a pixel)

$$\Phi_o(S) = \int_A \int_{\Omega_x} L(x,\omega_x) \cdot W_e(x,\omega_x,S) \cdot \cos\theta_x \; d\omega_x \, dA_x$$

Path Tracing

Radiant flux through the pixel (including anti-aliasing)

$$\left\langle \Phi(\mathbf{S}) \right\rangle_{\mathbf{path}} = \frac{\mathbf{W_e}(\mathbf{x_0}, \omega_0, \mathbf{S}) \cdot \mathbf{\cos\psi}}{\mathbf{p_0}(\mathbf{x_0}, \omega_0)} \cdot \left\langle \mathbf{L}(\mathbf{x_0}, \omega_0) \right\rangle_{\mathbf{path}}$$

($\mathbf{x_0}$, ω_0) is the 1st intersection point on the scene surface

$\mathbf{p_0}$ is the associated PDF

Sampling on the lens surface **$\mathbf{p_{-1}}$** (depth of field)

$$\mathbf{p_0}(\mathbf{x_0}, \omega_0) = \frac{\mathbf{p_{-1}}(\mathbf{x_{-1}}, \omega_0) \cdot \mathbf{\cos\psi}}{\left\| \mathbf{x_{-1}} - \mathbf{x_0} \right\|^2}$$

Path Tracing

Estimate of $\mathbf{L(x_0,\omega_0)}$ using **Monte-Carlo** with Russian roulette

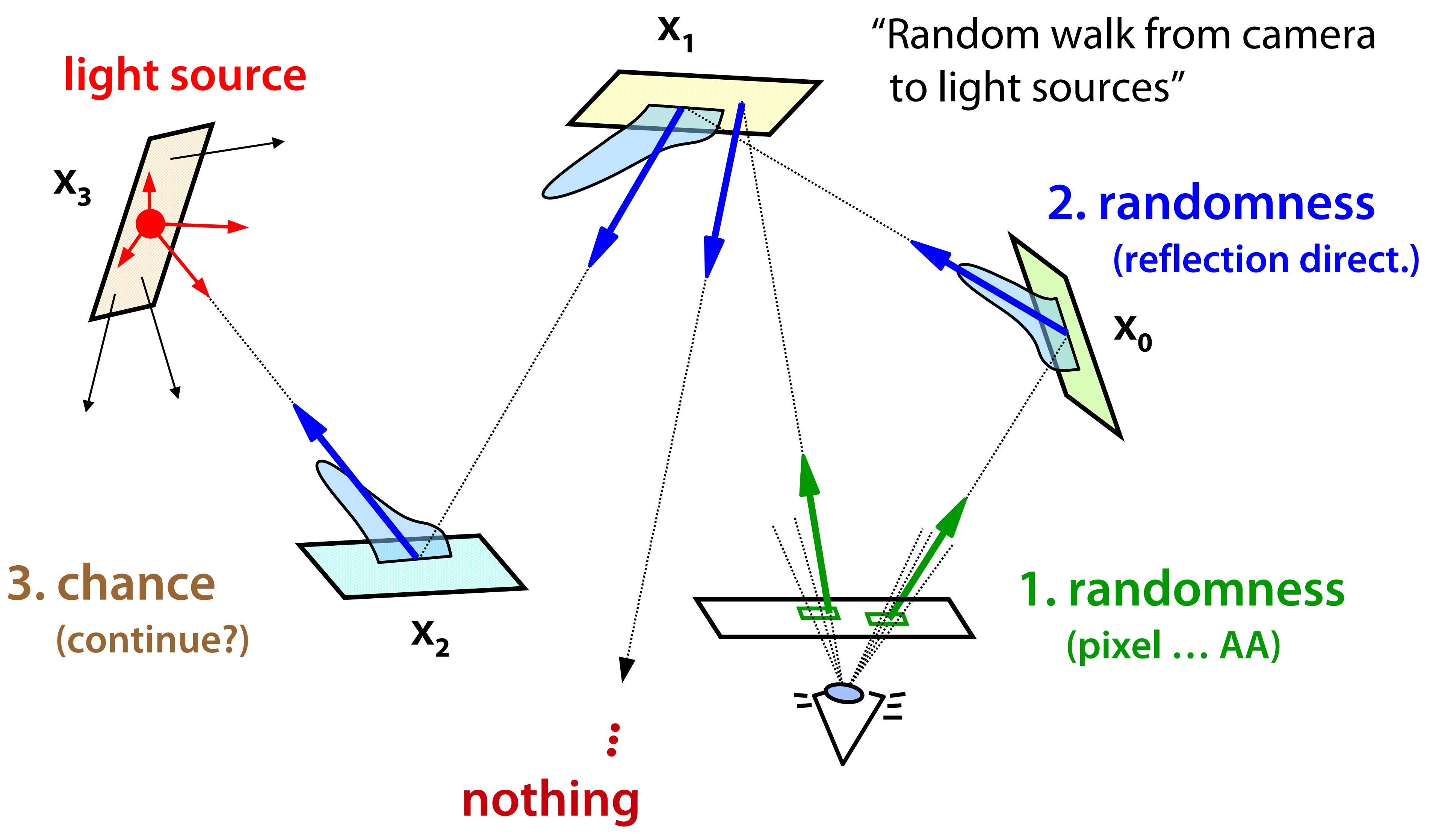
$$\langle \Phi(S) \rangle_{path} = \frac{W_e(x_0,\omega_0,S)\cdot\cos\psi}{p_0(x_0,\omega_0)}\cdot$$

$$\cdot\sum_{i=0}^{k}\left[\prod_{j=1}^{i}\frac{f(x_{j-1},\omega_j\to\omega_{j-1})\cdot\cos\theta_{j-1}}{P_j\cdot p_j(\omega_j)}\right]\cdot L_e(x_i,\omega_i)$$

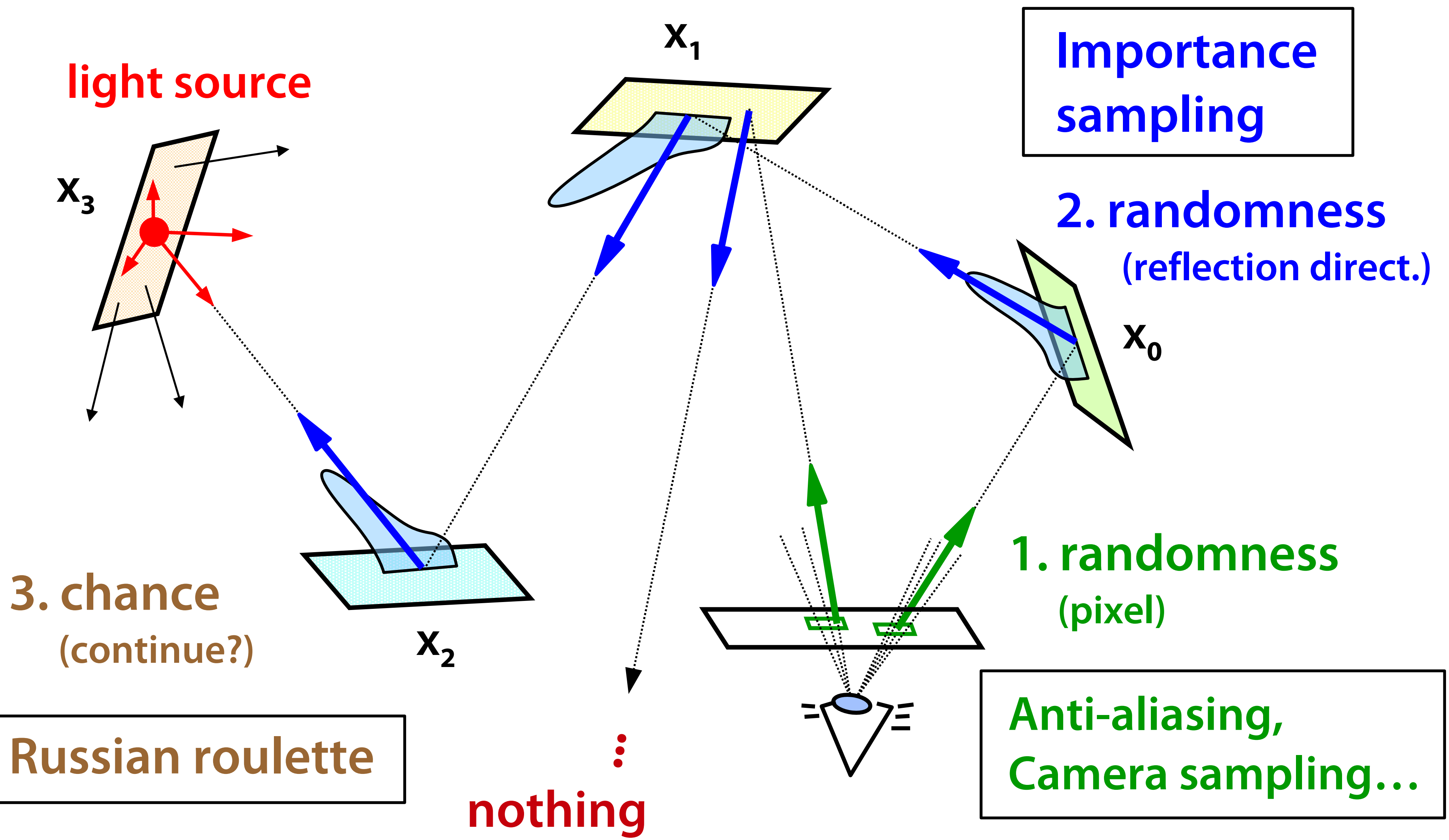
probability of the next step **j**

PDF of the incoming direction ω_j

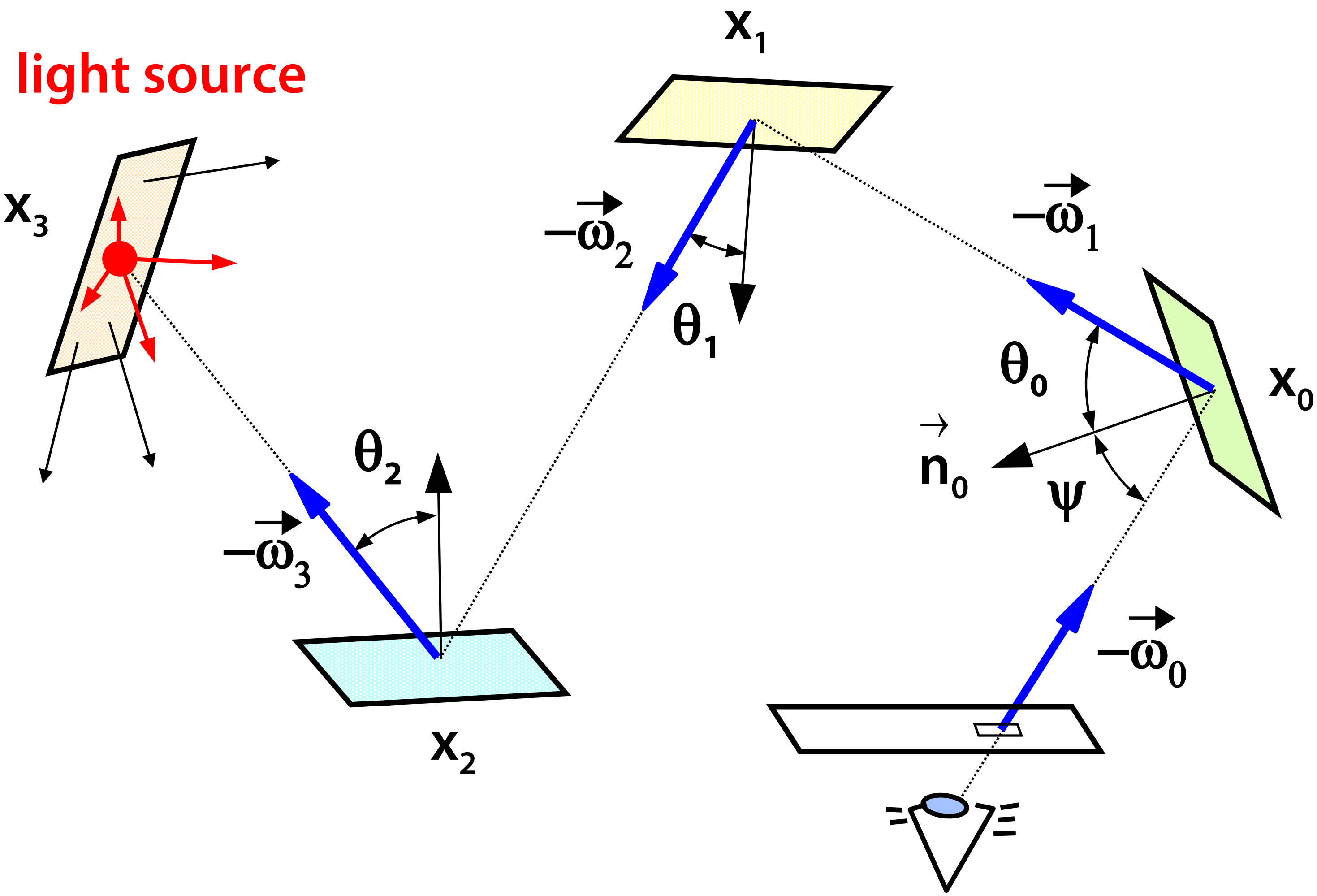
Path Tracing – principle



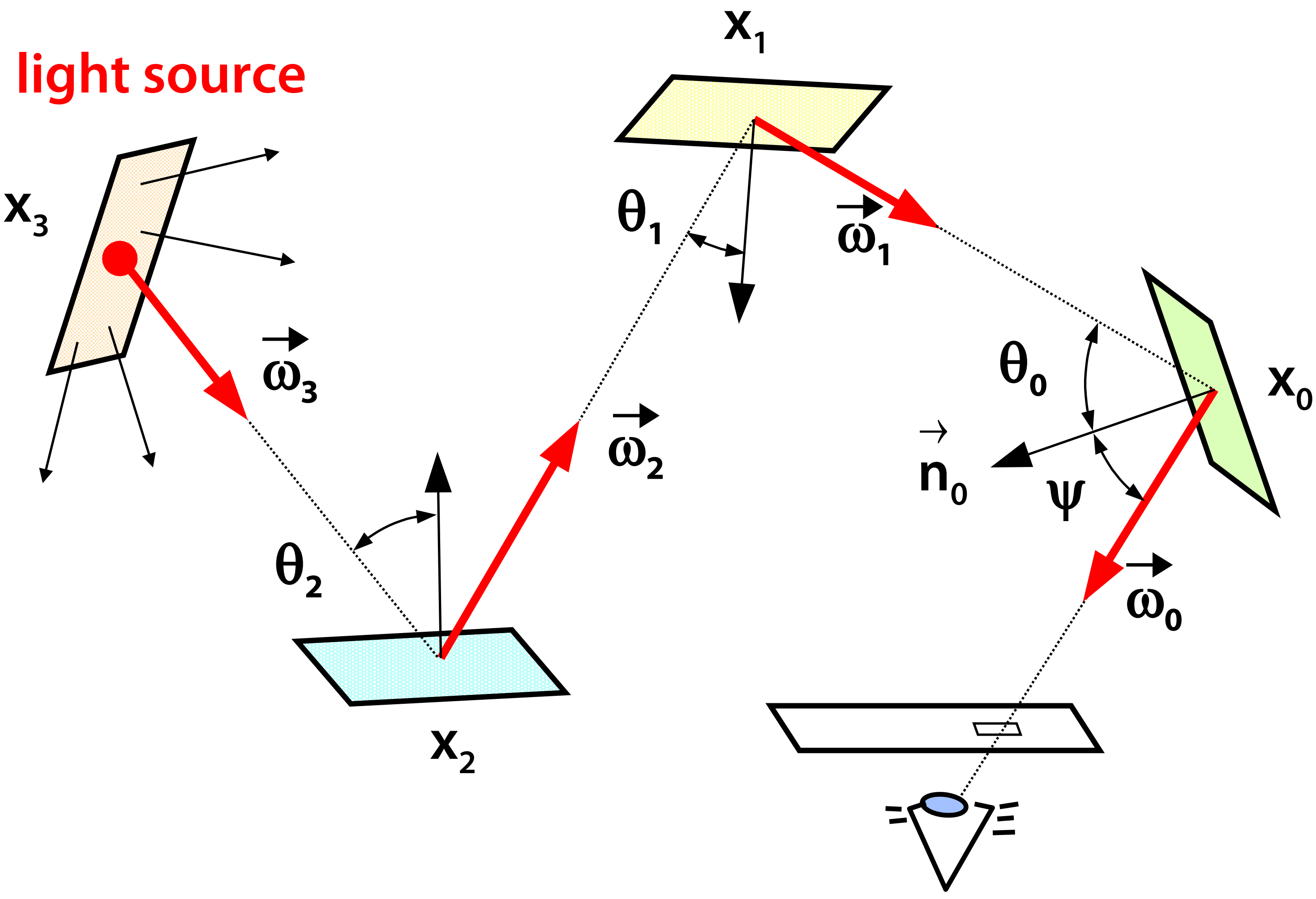
The role of randomness (Monte-Carlo)



Path Tracing – walk from camera

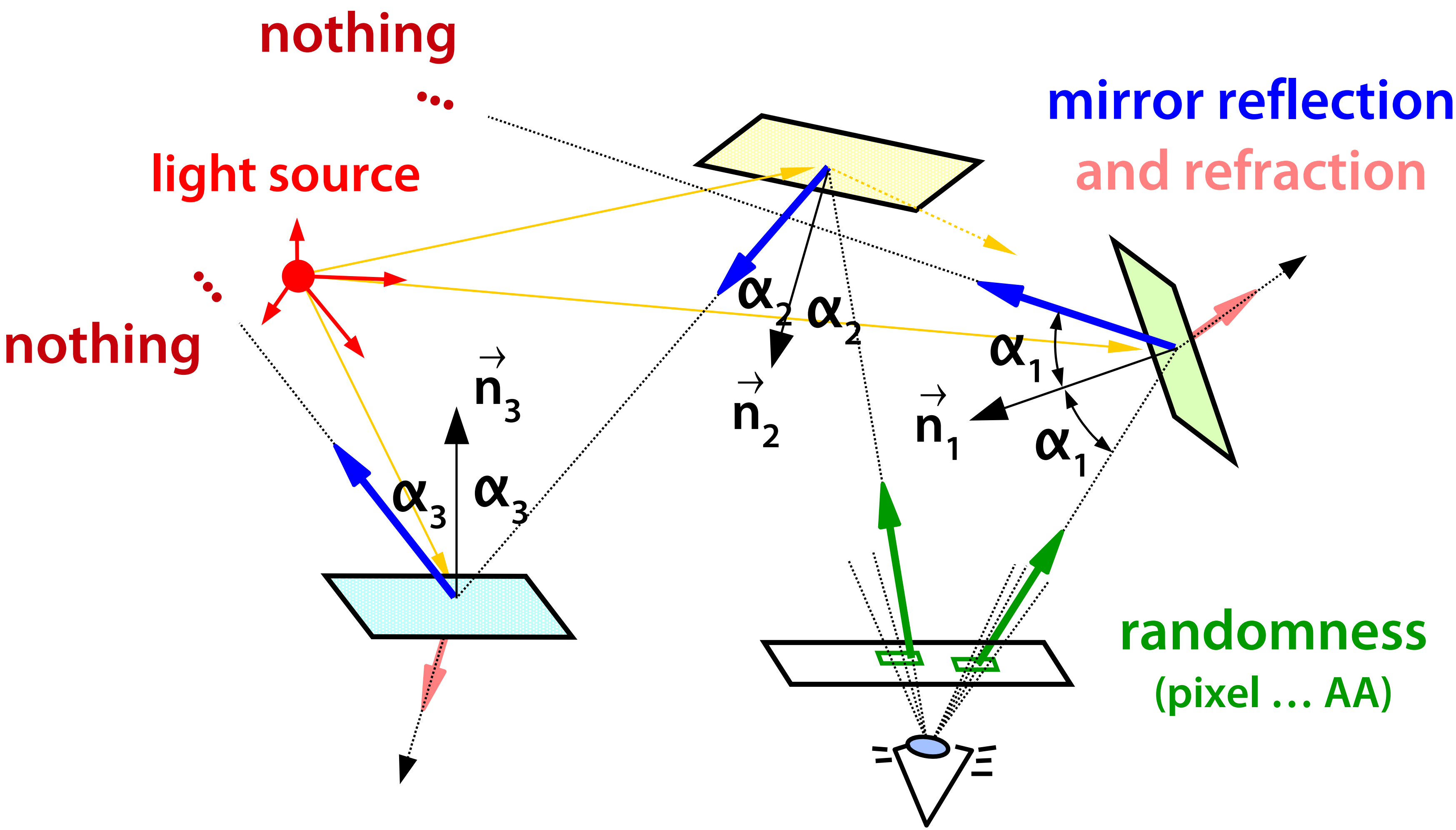


Path Tracing – light propagation



Raytracing for comparison

Deterministic walk from camera to scene



Importance sampling

For **radiant flux through the pixel** (the 2nd integral)

$$\mathbf{p_0(x_0,\omega_0)} = \frac{W_e(x_0,\omega_0,S)\cdot\cos\psi}{W(S)}, \text{ where}$$

$$W(S) = \int_A \int_{\Omega_x} W_e(x,\omega_x,S)\cdot\cos\theta_x \, d\omega_x \, dA_x$$

In the **rendering equation** (the 1st integral) we know the term **$f(x, \omega_y \rightarrow \omega_x) \cos \theta_y$**

It is **less than one** (physics), so it can be used for the **subcritical probability setup**

Sampling controlled by the BRDF

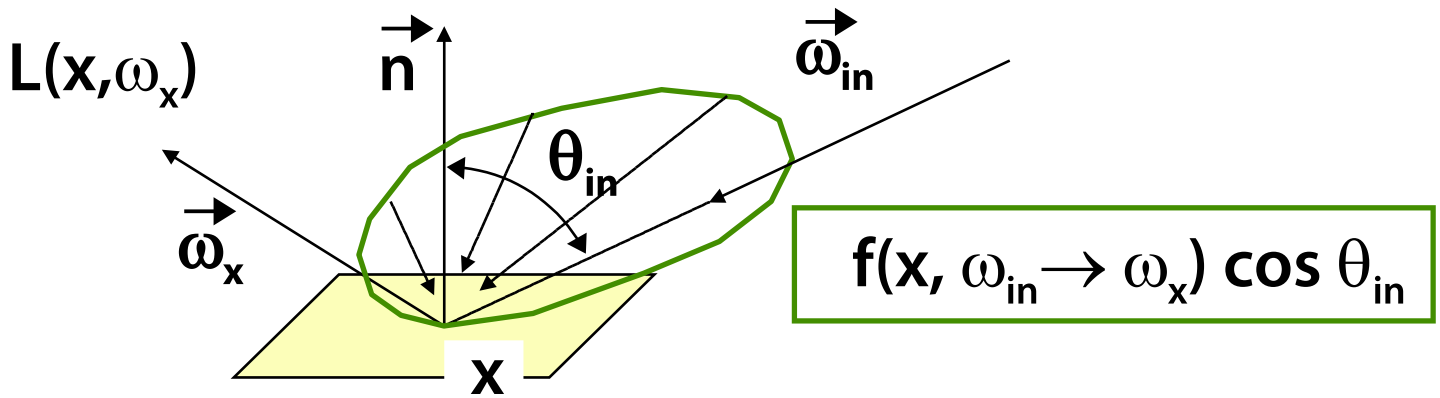
Probability of the next step **j**

$$P_j = \int_{\Omega^{-1}} f(x_{j-1}, \omega_{in} \to \omega_{j-1}) \cdot \cos\theta_{in} \, d\omega_{in}$$

Probability density (PDF) of the next direction ω_j

$$p_j(\omega_j) = \frac{f(x_{j-1}, \omega_j \to \omega_{j-1}) \cdot \cos\theta_{j-1}}{P_j}$$

Sampling controlled by the BRDF



The complete primary estimate using all the mentioned probabilities

$$\left\langle \Phi\left(S\right)\right\rangle_{path,imp} = W\left(S\right) \cdot \sum_{i=0}^{k} L_e\left(x_i, \omega_i\right)$$

Next event estimation (NEE)

Indirect light is divided into two components

$$L(x,\omega_x) = L_e(x,\omega_x) + L_r(x,\omega_x)$$

$$\underline{L_r(x,\omega_x)} = \int_{\Omega_x^{-1}} f(x,\omega_y \to \omega_x) \cdot L(y,\omega_y) \cdot \cos\theta_y \, d\omega_y =$$

$$= \int_A f(x,\omega_y \to \omega_x) \cdot L_e(y,\omega_y) \cdot G(y,x) \, dA_y +$$

$$+ \int_{\Omega_x^{-1}} f(x,\omega_y \to \omega_x) \cdot L_r(y,\omega_y) \cdot \cos\theta_y \, d\omega_y$$

Direct illumination component

Geometric term **G(y,x)**

$$\mathbf{G}(\mathbf{y},\mathbf{x}) = \mathbf{v}(\mathbf{y},\mathbf{x}) \cdot \frac{\cos\theta_{y,out} \cdot \cos\theta_{x,in}}{\|\mathbf{x}-\mathbf{y}\|^2}$$

visibility factor

Direct illumination contribution = the 1st integral

- domain is the area of all the light sources

Probability density for this part uses local radiosity [W/m_2] of the light source

Light source sampling

Probability density for **direct light contribution**

$$p(y) = \frac{L(y)}{L}$$

$$L(y) = \int_{\Omega_y} L_e(y, \omega_y) \cdot \cos\theta_y \, d\omega_y$$

radiosity emmited from point **y**

$$L = \int_A \int_{\Omega_y} L_e(y, \omega_y) \cdot \cos\theta_y \, d\omega_y \, dA_y$$

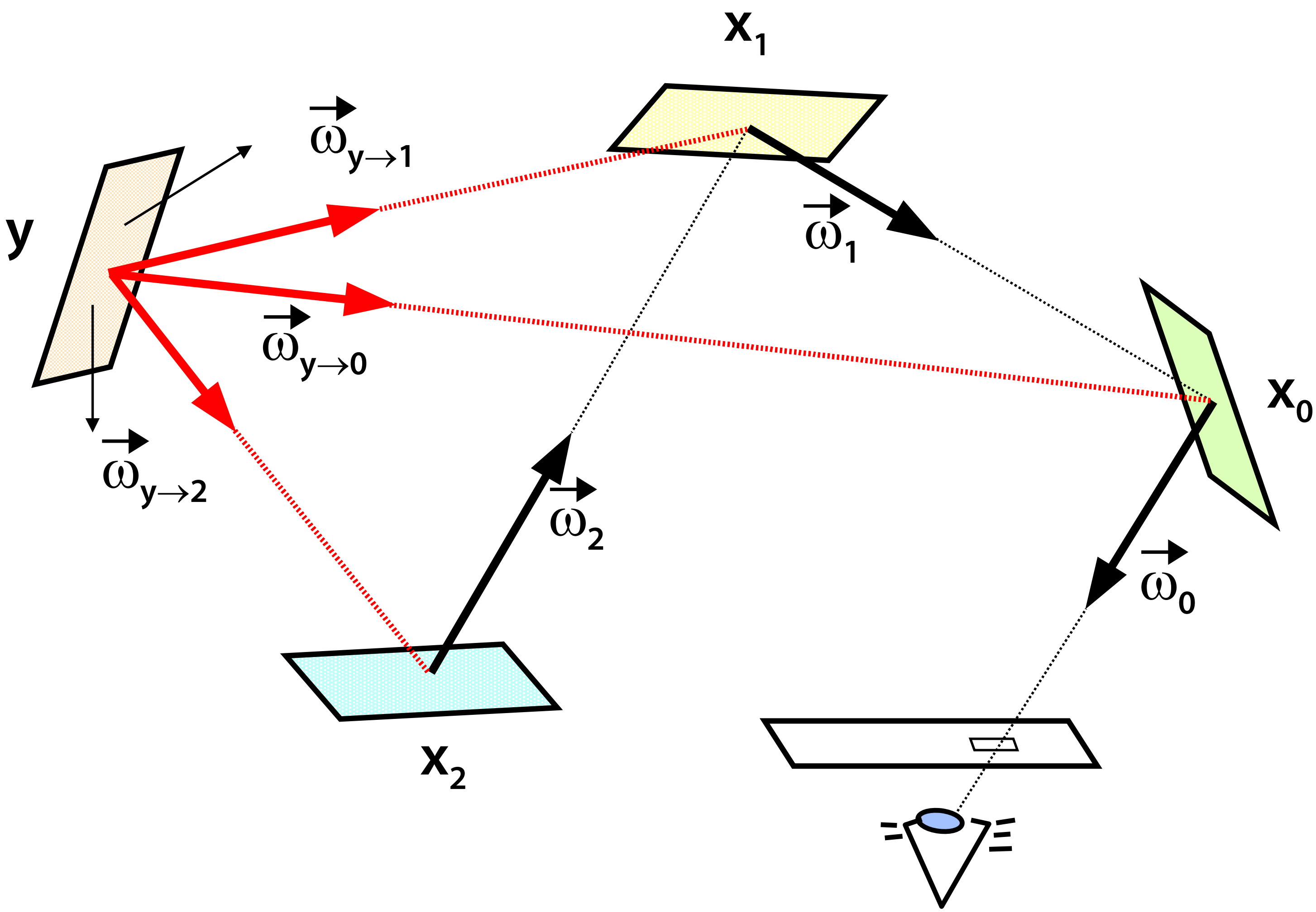
emmited power total

Next event estimation

BRDF-based sampling (subcritical probability)
with **Russian roulette** and **Next event estimation**

$$\left\langle \Phi(S) \right\rangle_{path,imp,nee} = W(S) \cdot \Big[L_e(x_0, \omega_0) + \\ + \frac{L}{L(y)} \sum_{i=0}^{k} L_e(y, \omega_{y \to i}) \cdot f(x_i, \omega_{y \to i}, \omega_i) \cdot G(y, x_i) \Big]$$

Light propagation (Path Tracing + NEE)



Path Tracing + Next Event Estimation

Best for scenes with **small** but **good visible** light sources

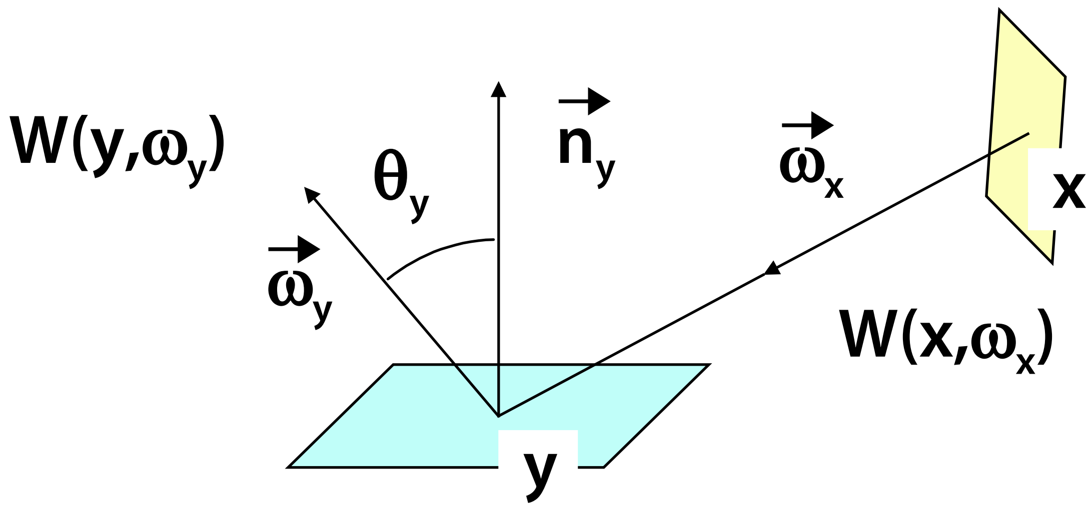
- sampling of light sources is dominant

Light source sampling doesn't consider **their visibility**

- not visible light source $\Rightarrow$ waste of effort!
- more advanced methods consider **BRDFs** and/or geometric terms **G(y,$\mathbf{x}_i$)**

Light source sampling could be done in every step $\mathbf{x}_i$

Rendering equation for importance



$$W(x,\omega_x) =$$

$$= W_e(x,\omega_x) + \int_{\Omega_y} f(y,\omega_x \to \omega_y) \cdot W(y,\omega_y) \cdot \cos\theta_y \, d\omega_y$$

$$\Phi_o(S) = \int_A \int_{\Omega_x} L_e(x,\omega_x) \cdot W(x,\omega_x,S) \cdot \cos\theta_x \, d\omega_x \, dA_x$$

Light tracing

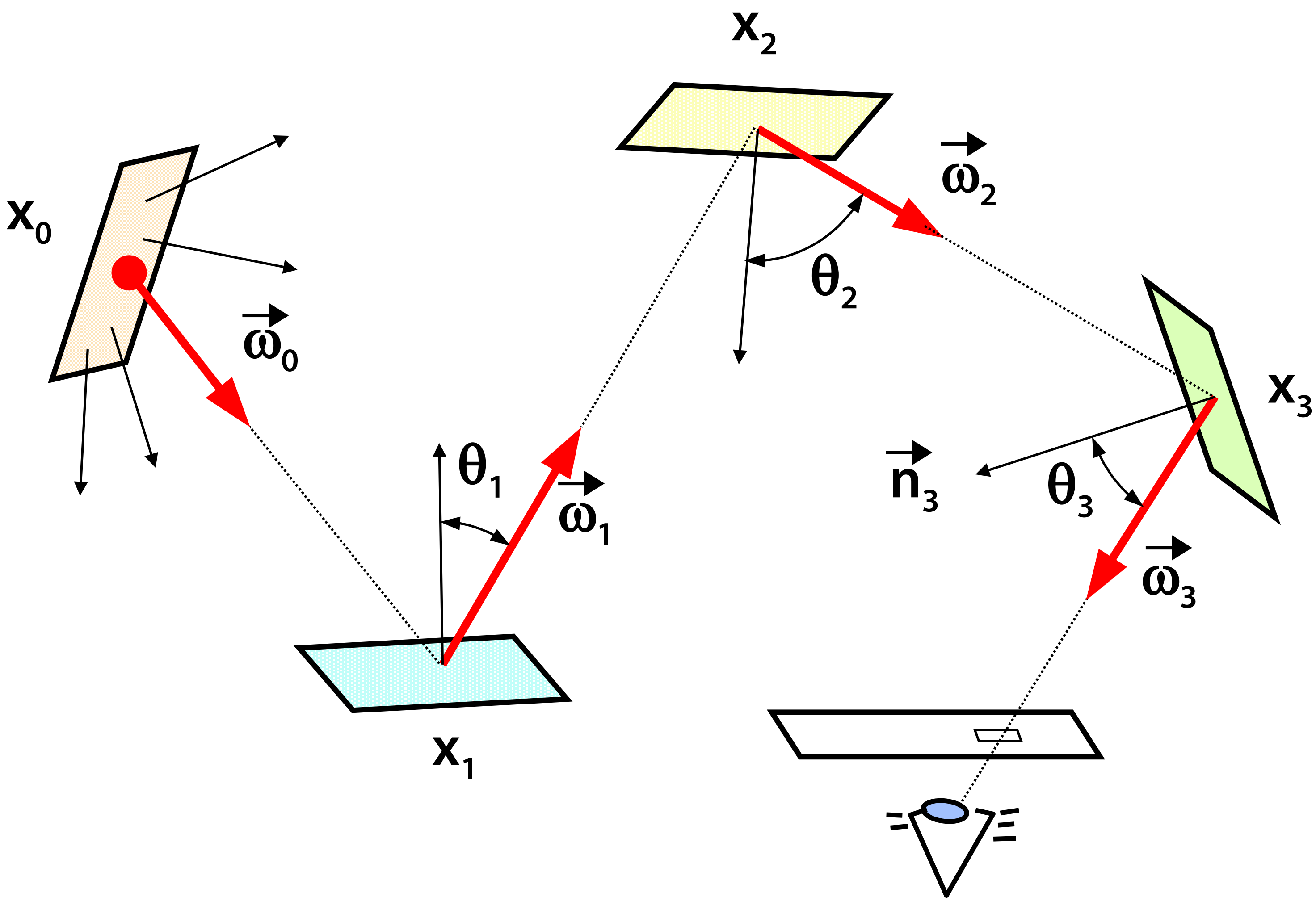
Ray coming **from the source** (radiation characteristics of the source)

$$\langle \Phi(S) \rangle_{light} = \frac{L_e(x_0, \omega_0) \cdot \cos\theta_0}{p_0(x_0, \omega_0)} \cdot \langle W(x_0, \omega_0, S) \rangle_{light}$$

$$\langle \Phi(S) \rangle_{light} = \frac{L_e(x_0, \omega_0) \cdot \cos\theta_0}{p_0(x_0, \omega_0)} \cdot \sum_{i=0}^{k} \left[\prod_{j=1}^{i} \frac{f(x_j, \omega_{j-1} \to \omega_j) \cdot \cos\theta_j}{P_j \cdot p_j(\omega_j)} \right] \cdot W_e(x_i, \omega_i, S)$$

total estimate

Light Tracing – light propagation



Next Event Estimation (NEE)

Reflected light is divided into two parts (not considering S)

$$W(x,\omega_x) = W_e(x,\omega_x) + W_r(x,\omega_x)$$

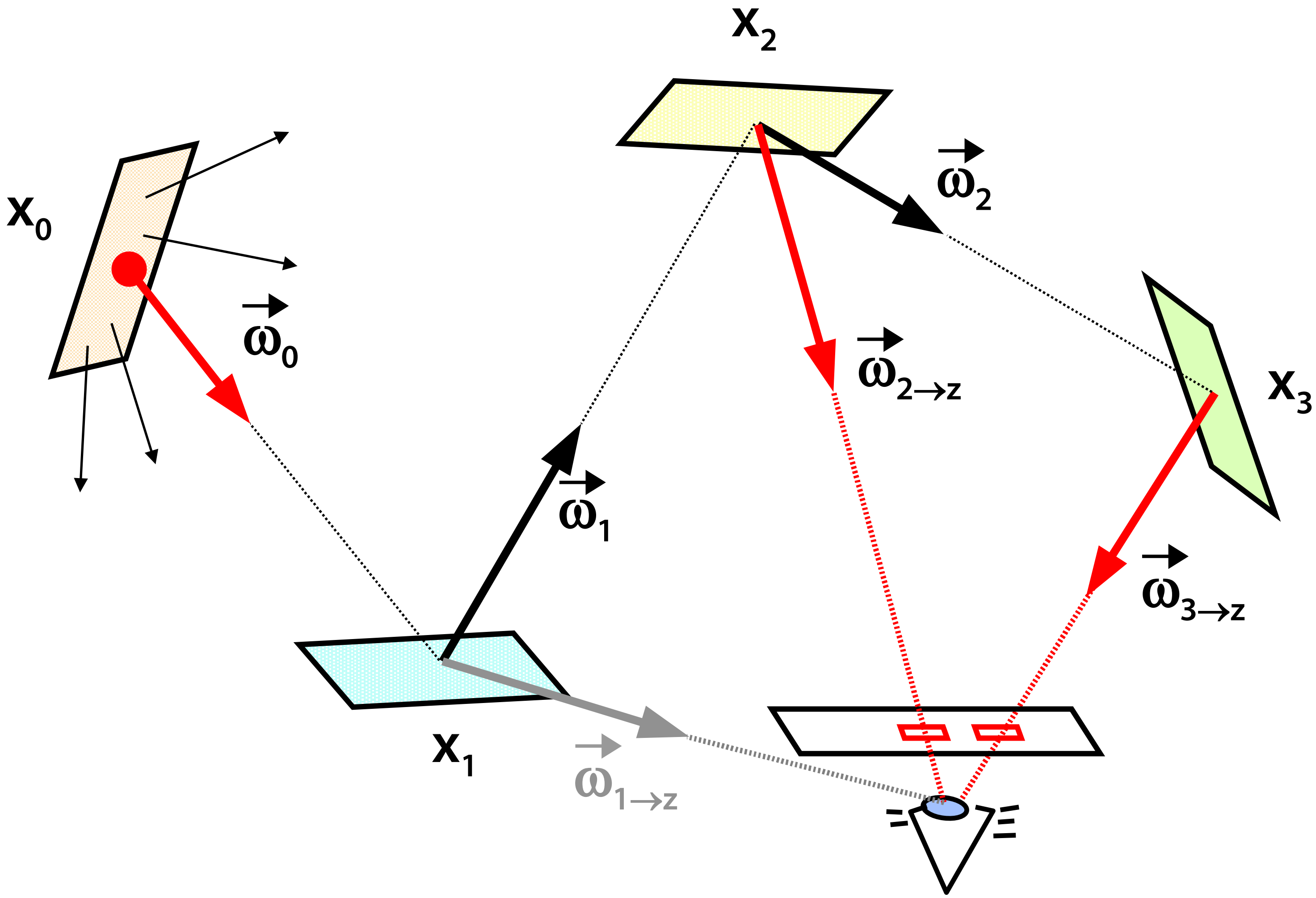
$$\underline{W_r(x,\omega_x)} = \int_{\Omega_y} f(y,\omega_x \to \omega_y) \cdot W(y,\omega_y) \cdot \cos\theta_y \, d\omega_y =$$

$$= \int_{\underline{Ape}} f(y,\omega_x \to \underline{\omega_z}) \cdot W_e(y,\underline{\omega_z}) \cdot G(y,\underline{z}) \, \underline{dA_z} +$$

$$+ \int_{\Omega_y} f(y,\omega_x \to \omega_y) \cdot W_r(y,\omega_y) \cdot \cos\theta_y \, d\omega_y$$

lens aperture → Ape

Light propagation (Light Tracing + NEE)



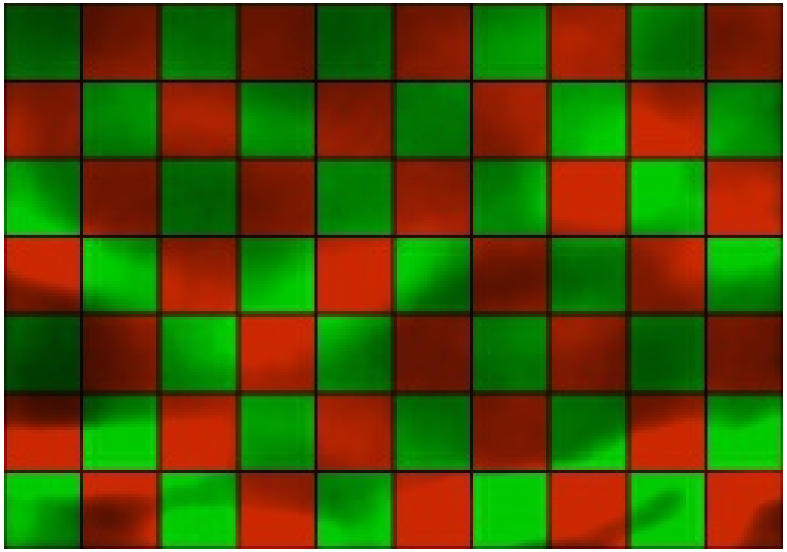
Use of Light Tracing

Direct **realistic image rendering**

- light is collected by the camera and stored in the projecting plane

Supporting algorithm for some “hybrid” method

- light is stored in “light maps” (photon maps)
- the higher amount of the $\mathbf{W}_e$ importance/potential leads to a more efficient calculation



Bidirectional Path Tracing – theory

Combined **global rendering equation**

own emitted radiance

discrete importance/potential

GRDF

$$\Phi(S) =$$

$$= \iint_{A,\Omega_x} \iint_{A,\Omega_y} L_e(x,\omega_x)\, W_e(y,\omega_y,S)\, F(x,\omega_x \rightarrow y,\omega_y)$$

$$\cos\theta_y \cos\theta_x \, d\omega_y \, dA_y \, d\omega_x \, dA_x$$

integrals over all the light source areas and directions and all the receptor areas and directions

Recursive definition of GRDF

The 1st reflection/bounce

$$F(x,\omega_x \to \underline{y,\omega_y}) = \delta(x,\omega_x,y,\omega_y) + \int_{\Omega_z} f(z,\omega_x \to \omega_z) \cdot F(z,\omega_z \to \underline{y,\omega_y}) \cdot \cos\theta_z \, d\omega_z$$

The last reflection/bounce

$$F(\underline{x,\omega_x} \to y,\omega_y) = \delta(x,\omega_x,y,\omega_y) + \int_{\Omega_y^{-1}} f(y,\omega_z \to \omega_y) \cdot F(\underline{x,\omega_x} \to z,\omega_z) \cdot \cos\theta_y \, d\omega_z$$

GRDF estimate

Linear combination of both recursive formulas

$$\mathbf{F} = \delta + \mathbf{w}^* \mathbf{T}^* \mathbf{F} + \mathbf{w}\,\mathbf{T}\,\mathbf{F}, \qquad \mathbf{w} + \mathbf{w}^* = 1$$

Infinite Neumann series

$$\mathbf{F} = \sum_{i=0}^{\infty} \sum_{j=0}^{\infty} \mathbf{w}_{ij} \mathbf{T}^{*i} \mathbf{T}^{j} \delta, \qquad \sum_{i=0}^{N} \mathbf{w}_{i,N-i} = 1$$

T and **T*** are estimated using random walks with Roussian roulette

Next event estimation has to be used to **reduce variance**

Bidirectional Path Tracing

$\mathbf{T^*}$ is estimated by <u>forward</u> tracking light <u>from the sources</u> ("Light Tracing")

$\mathbf{x_0, x_1 \ldots x_{k^*}}$ – direction $\omega_{\mathbf{xi}}$ is controlled by PDF $\mathbf{p_i}(\omega_{\mathbf{xi}})$, Russian roulette probability is $\mathbf{P_i}$

$\mathbf{T}$ is estimated by <u>backward</u> tracking light <u>from the observer</u> ("Path Tracing")

$\mathbf{y_0, y_1 \ldots y_k}$ – direction $\omega_{\mathbf{yi}}$ is controlled by PDF $\mathbf{q_i}(\omega_{\mathbf{yi}})$, Russial roulette probability is $\mathbf{Q_i}$

Next event estimation (NEE)

Including open paths

$$\langle \Phi(S) \rangle_{\mathbf{bipath,nee}} = \sum_{i=-1}^{k} \sum_{j=-1}^{k^*} w_{ij} C_{ij}$$

$\mathbf{i = -1, j > 0}$	open path from the viewer (w/o NEE)	
$\mathbf{i = 0, j \geq 0}$	path from the viewer to a sample on a light source	PT+NEE
$\mathbf{i > 0, j > 0}$	light i-times bounced from a light source and j-times from the viewer	
$\mathbf{i \geq 0, j = 0}$	path from a light source to a sample on the viewer (front lens of the camera)	LT+NEE
$\mathbf{i > 0, j = -1}$	open path from a light source (w/o NEE … inefficient)	

Sampling overview

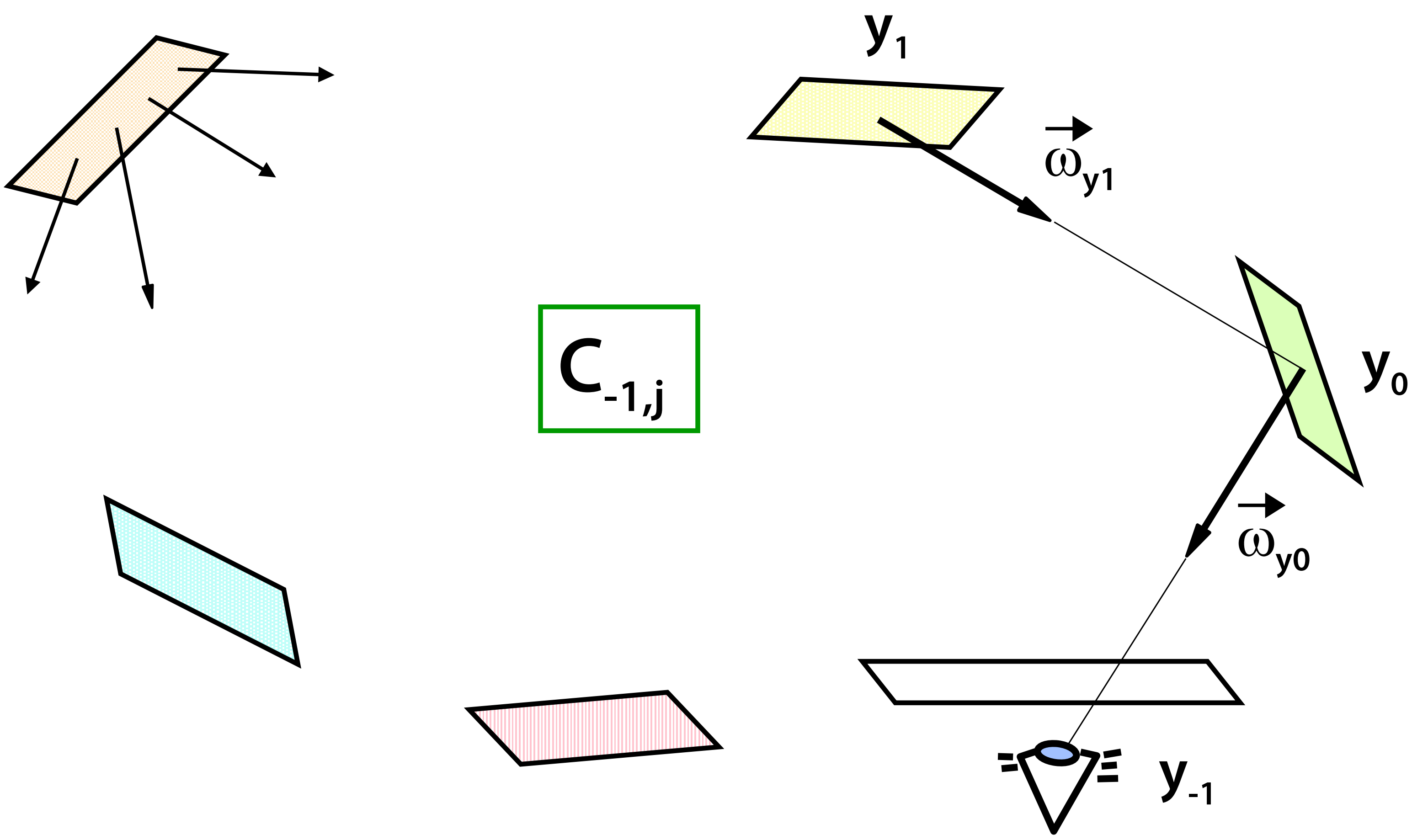
dependent on		sample on a path from a light source			
		x_0	x_1	x_2	x_3
			$C_{1,-1}$	$C_{2,-1}$	$C_{3,-1}$
y_{-1}		$C_{0,0}$	$C_{1,0}$	$C_{2,0}$	$C_{3,0}$
y_0	$C_{-1,1}$	$C_{0,1}$	$C_{1,1}$	$C_{2,1}$	$C_{3,1}$
y_2	$C_{-1,2}$	$C_{0,2}$	$C_{1,2}$	$C_{2,2}$	$C_{3,2}$

sample on a path from the viewer

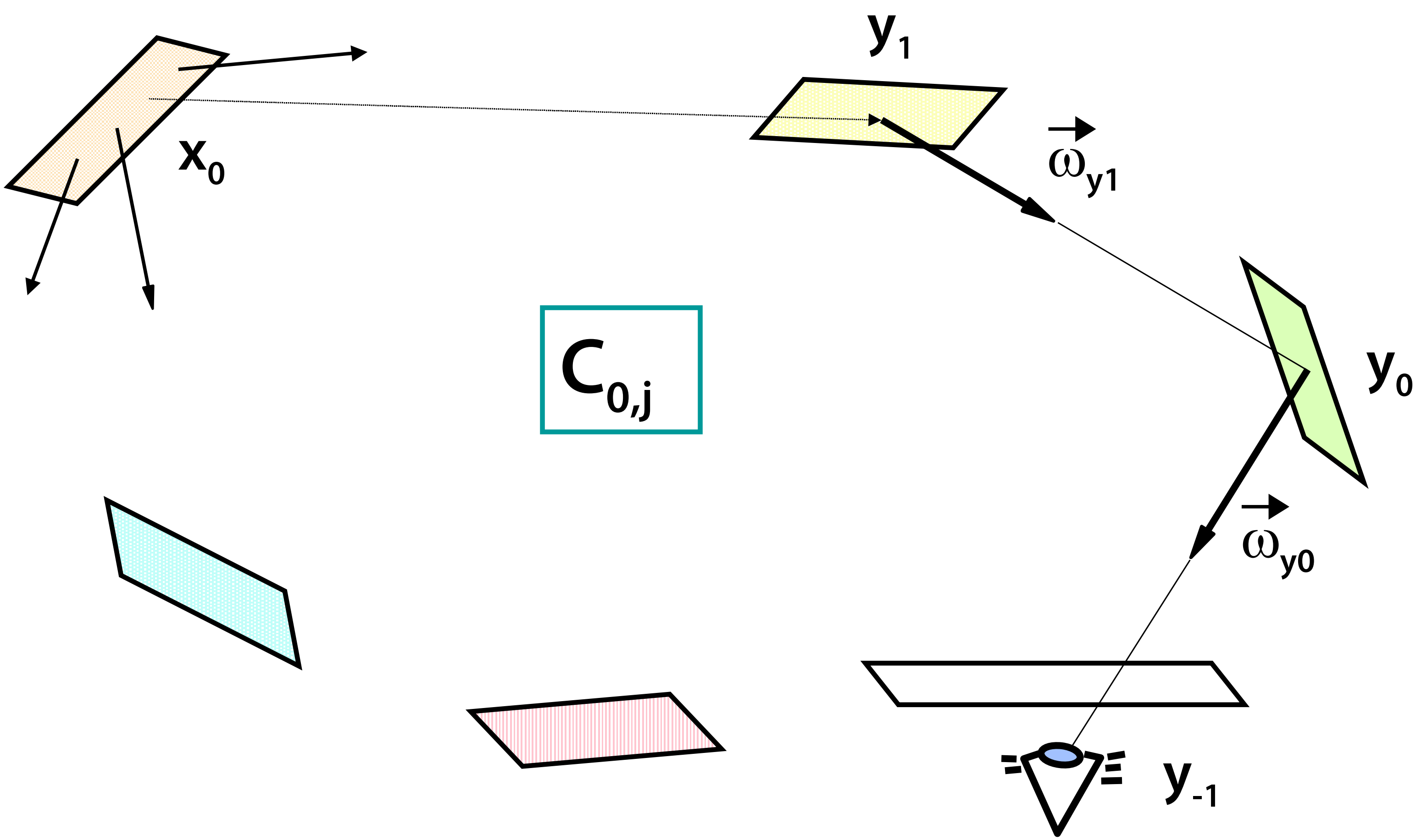
PT

LT

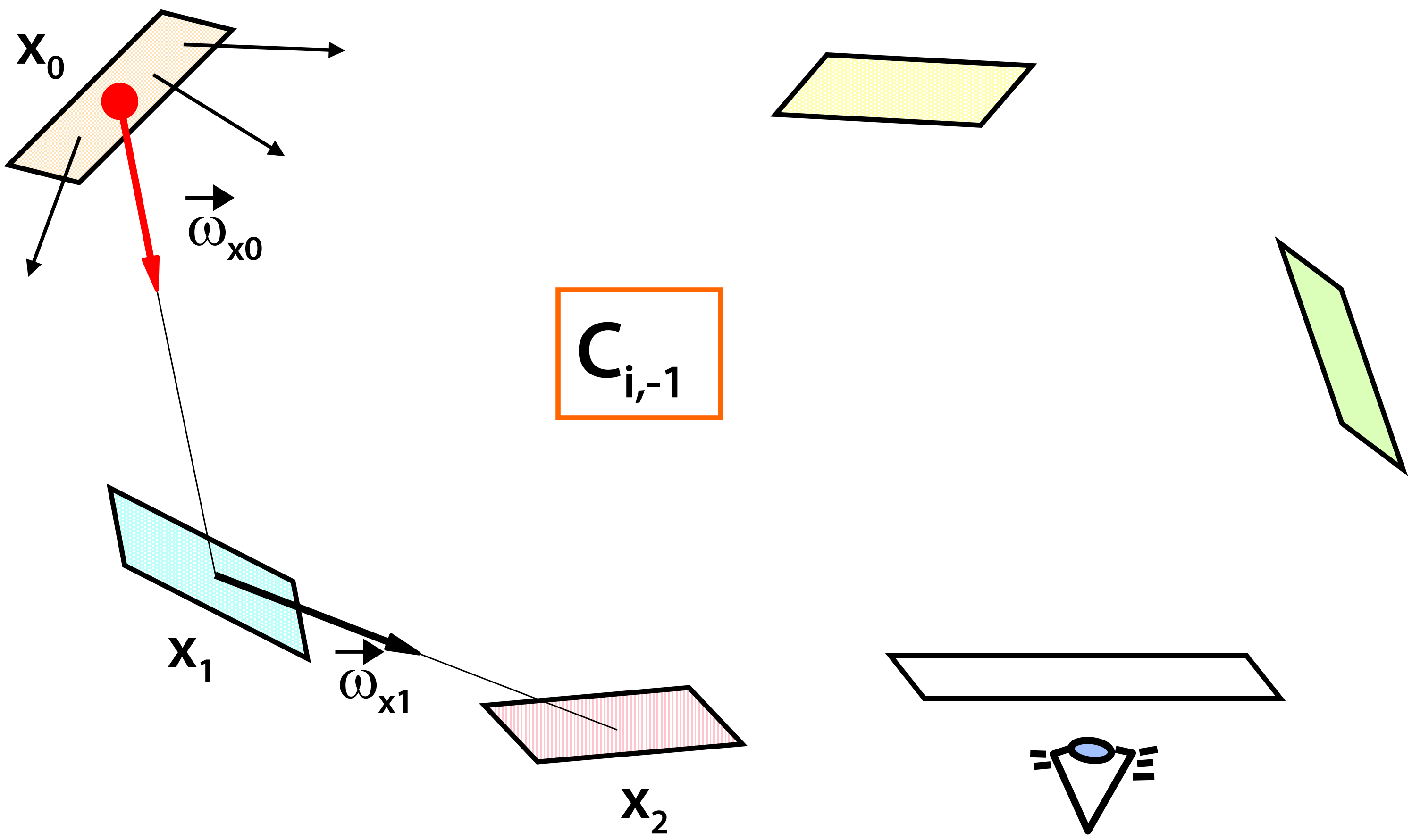
BidirPT – open path from the viewer



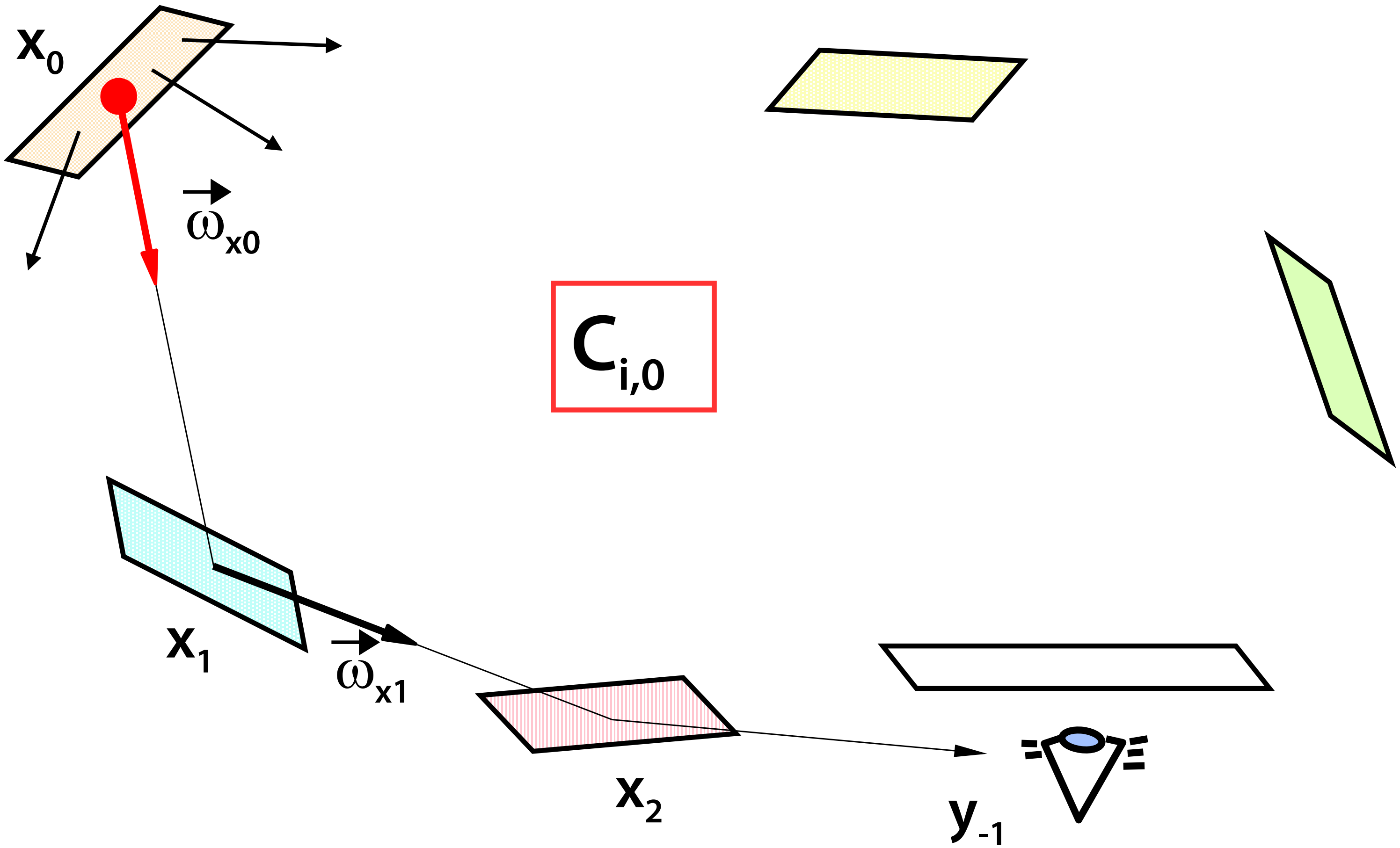
BidirPT – Path Tracing + NEE



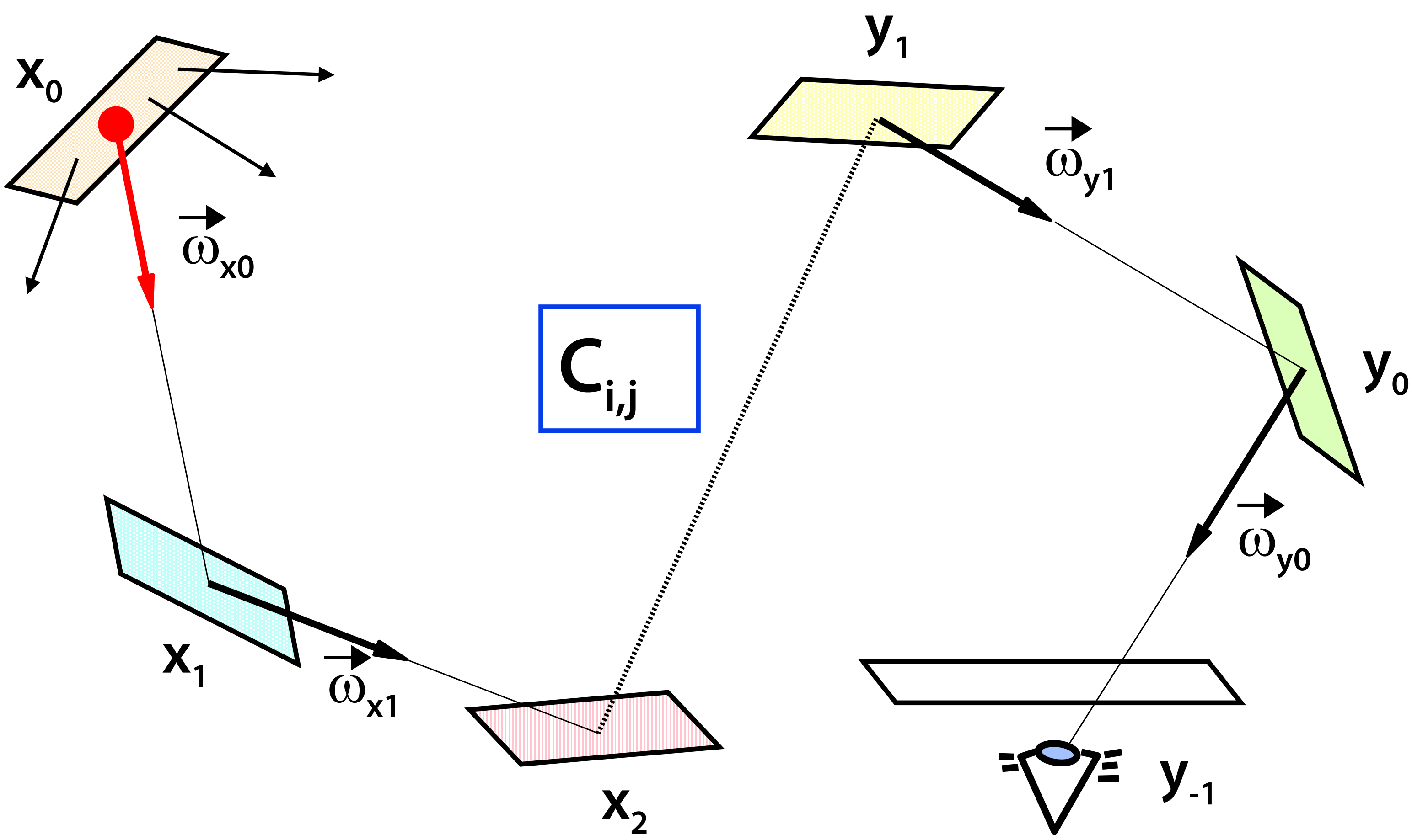
BidirPT – open path from a light source



BidirPT – Light Tracing + NEE



BidirPT – general combined path



Efficient implementation

Two **independent random walks** (with Russian roulette)

- from a light source (length **k***) and from the viewer (**k**)
- or single random path from a light source to the viewer (**K**)

Blending of **all path prefixes** (both directions)

- beware of systematic errors (biased estimate)!

K+2 combinations for fixed total path-length **K**

- combined estimate – blend of all estimates for allvalues of **K**

Bidirectional Path Tracing example

© 1995 Eric Veach,
Leonidas J. Guibas

References

E. Lafortune: ***Mathematical Models and Monte Carlo Algorithms for Physically Based Rendering***, PhD thesis, KU Leuven, 29-102

A. Glassner: ***Principles of Digital Image Synthesis***, Morgan Kaufmann, 1995, 1037-1049

E. Veach, L. Guibas: ***Optimally Combining Sampling Techniques for Monte Carlo Rendering***, SIGGRAPH '95