

IFT3355: Infographie

Sujet 7: illumination globale

Derek Nowrouzezahrai, Pierre Poulin

Département d'informatique et de recherche opérationnelle

Université de Montréal

[Plusieurs images © Wojciech Jarosz]



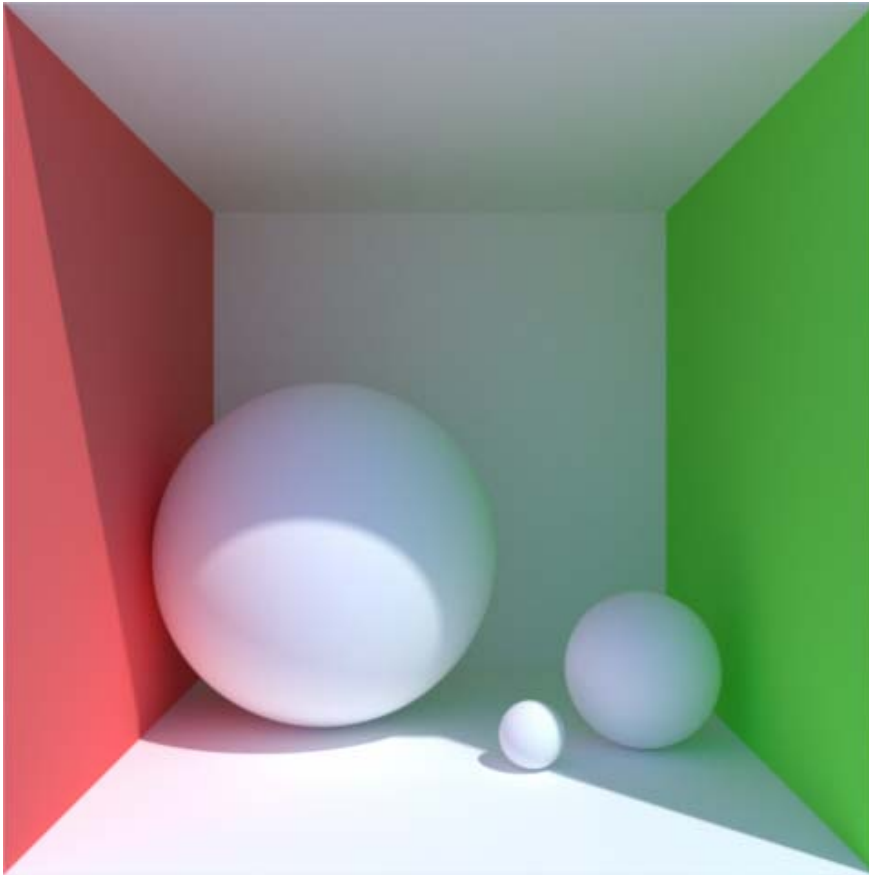
Illumination globale: motivation



Illumination globale: motivation



Illumination globale: motivation



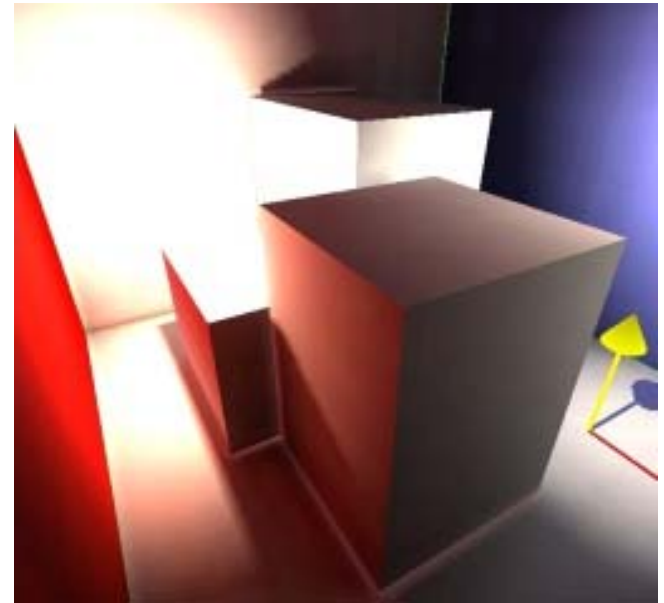
Illumination globale: motivation



Illumination globale: motivation



Illumination directe vs. globale



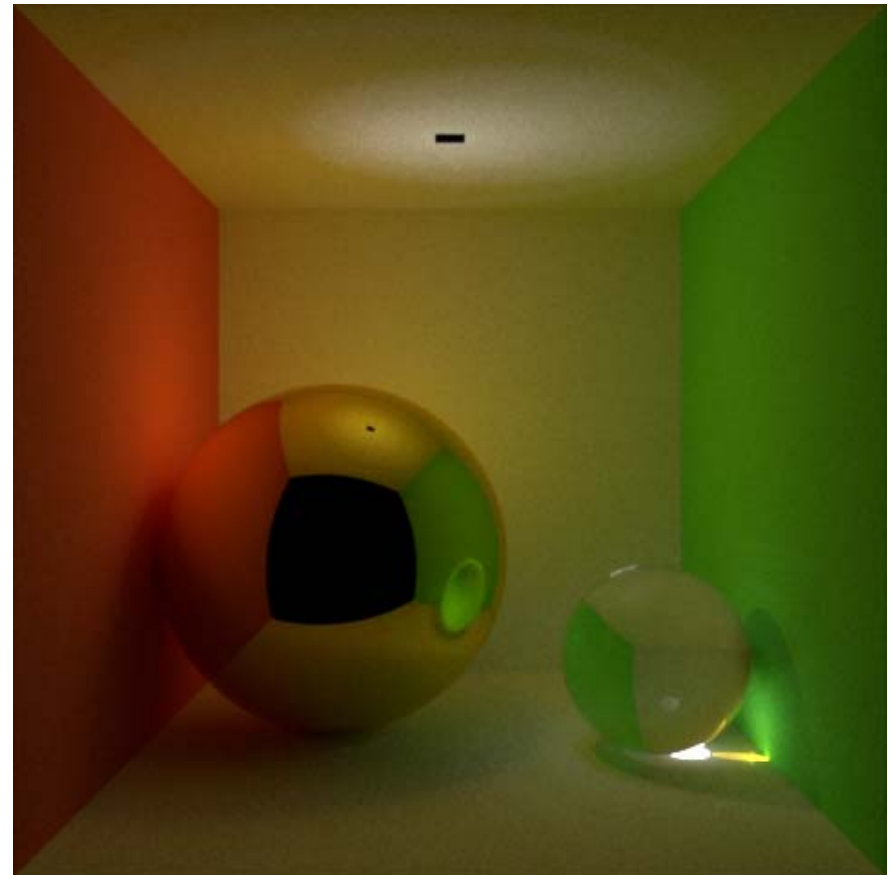
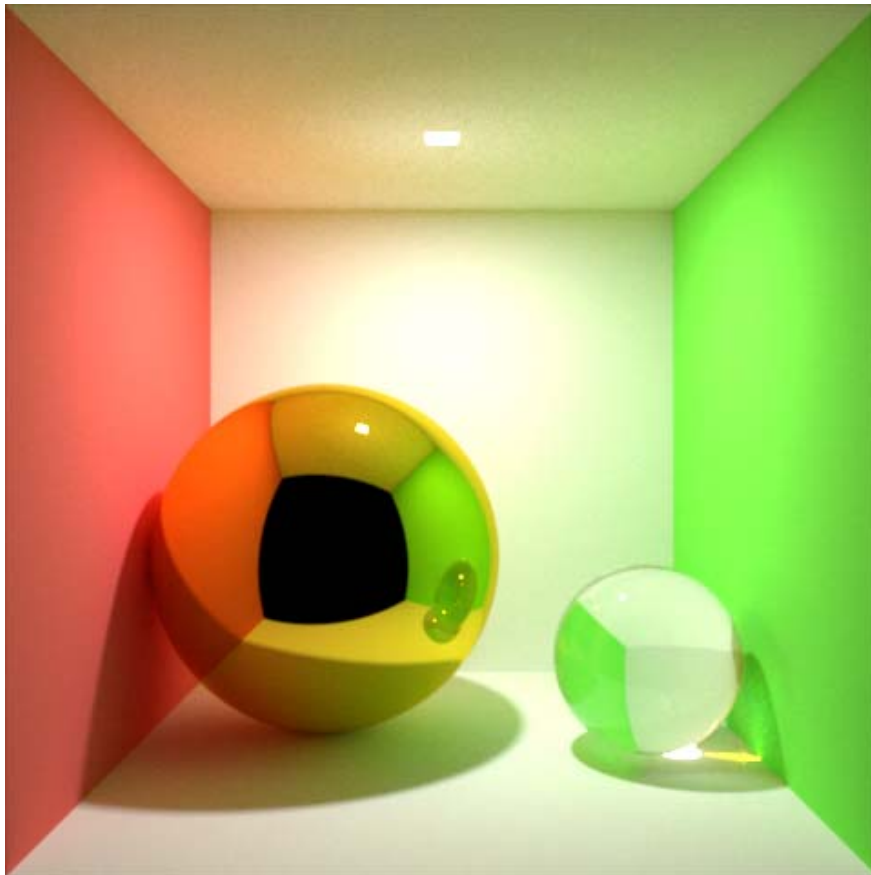
Illumination directe vs. globale



Illumination directe vs. globale



Illumination directe vs. globale

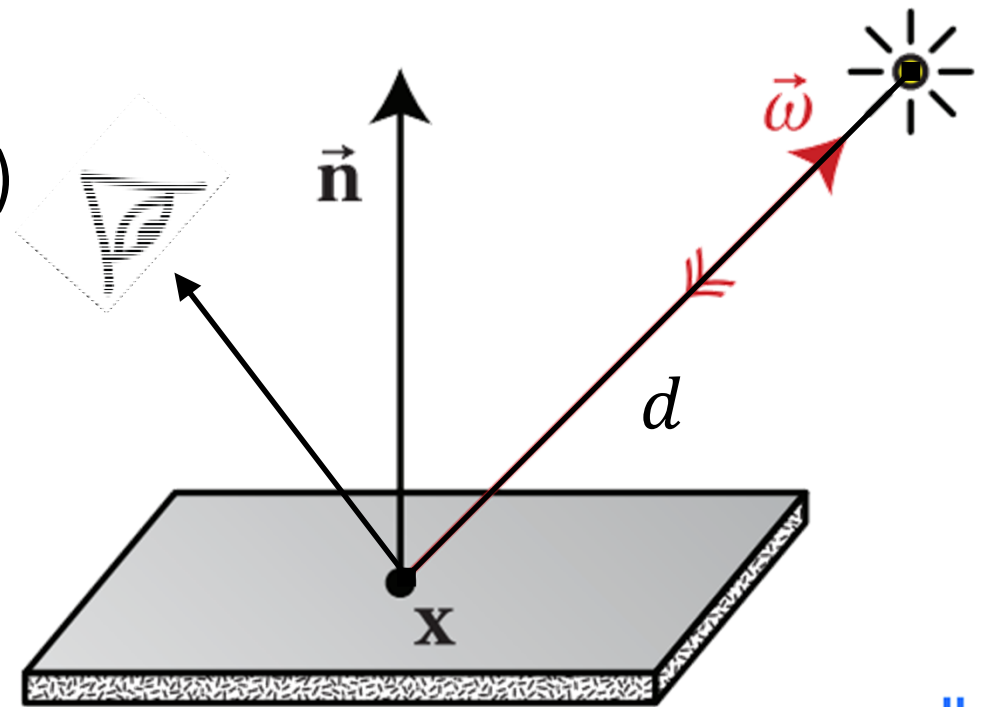


Illumination **directe** vs. globale

- Illumination directe
 - Lumière provient directement des sources de lumière et réfléchit vers la caméra

ex: lumière ponctuelle,
surface diffus (aucun ombre)

$$L_o(x) = \left(\frac{I}{d^2}\right) \frac{\rho}{\pi} [\vec{n} \cdot \omega]$$

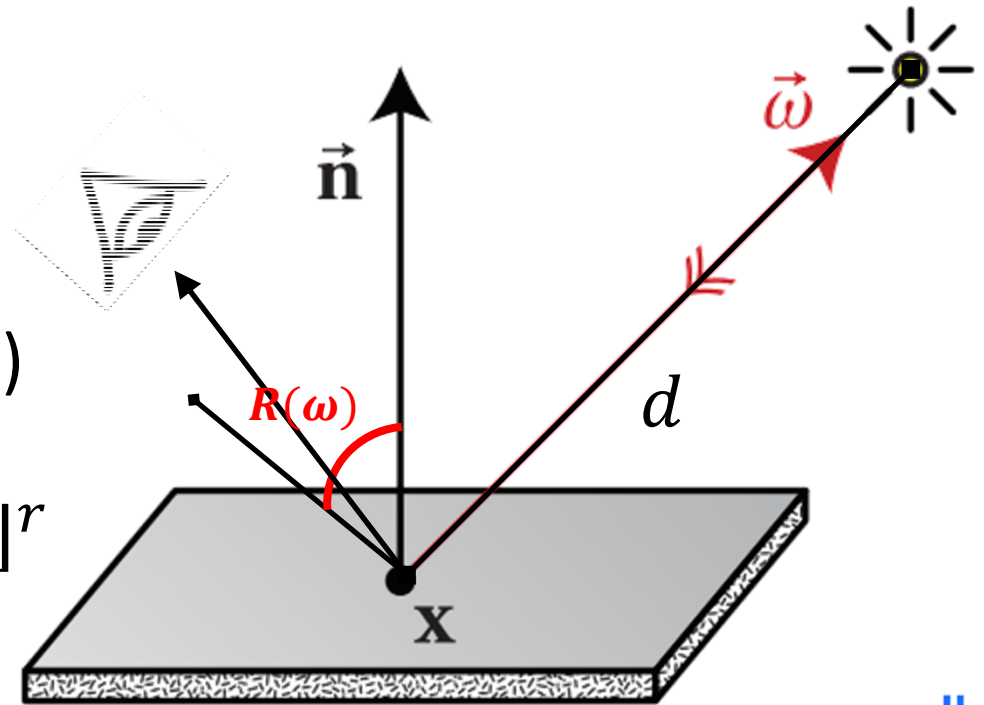


Illumination **directe** vs. globale

- Illumination directe
 - Lumière provient directement des sources de lumière et réfléchit vers la caméra

ex: lumière ponctuelle,
surface Phong (aucun ombre)

$$L_o = \left(\frac{I}{d^2}\right) \frac{r + 1}{2\pi} [\omega_o \cdot R(\omega)]^r$$

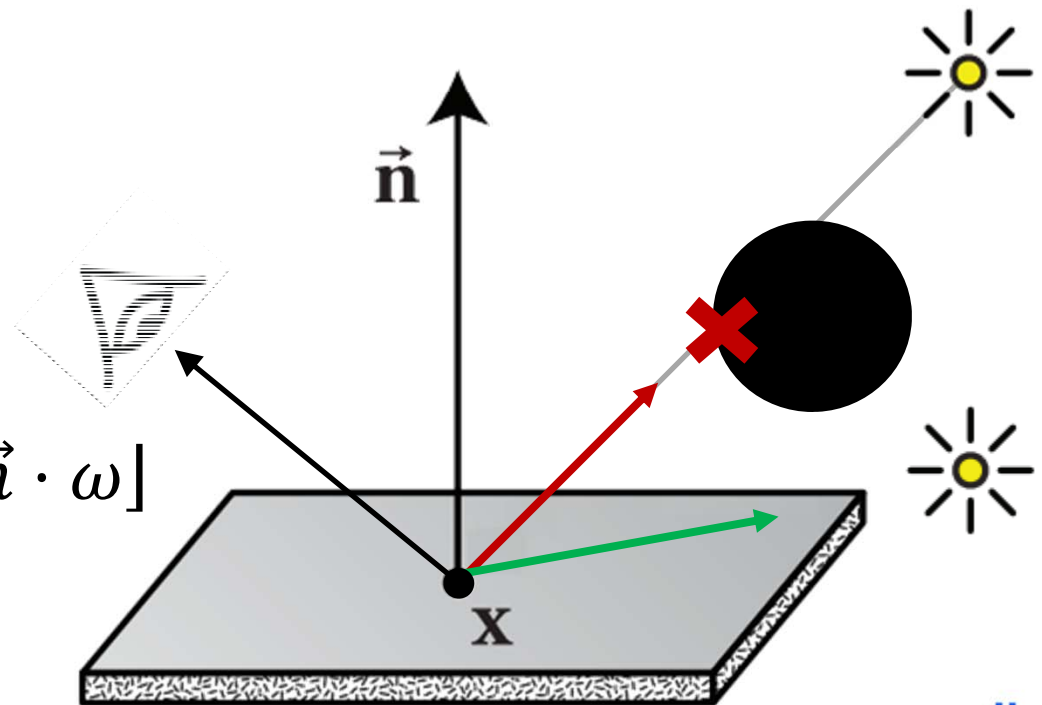


Illumination **directe** vs. globale

- Illumination directe
 - Lumière provient directement des sources de lumière et réfléchit vers la caméra

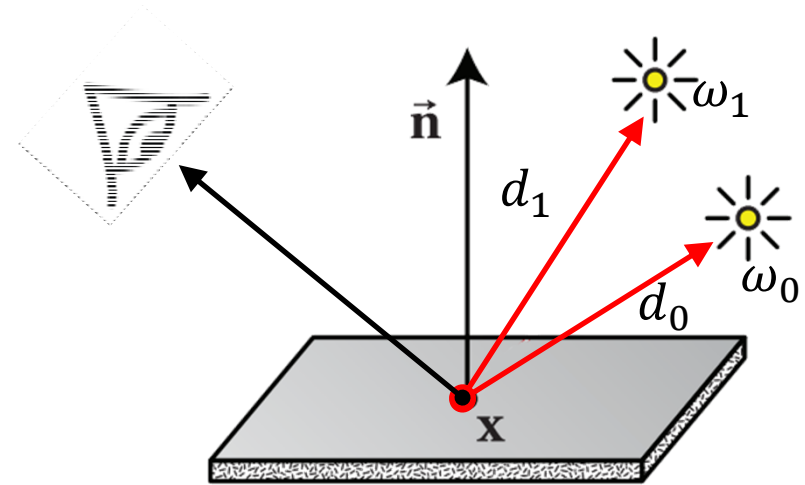
ex: lumière ponctuelle,
surface diffus

$$L_o(x) = \left(\frac{I}{d^2}\right) \frac{\rho}{\pi} V(x, \omega) [\vec{n} \cdot \omega]$$



Illumination **directe** vs. globale

- Illumination directe
 - Linéarité d'illuminationex: plusieurs lumières ponctuelles, surface diffus



$$L_o(x) = \left(\frac{I_0}{d_0^2}\right) \frac{\rho}{\pi} V(x, \omega_0) [\vec{n} \cdot \omega_0] + \left(\frac{I_1}{d_1^2}\right) \frac{\rho}{\pi} V(x, \omega_1) [\vec{n} \cdot \omega_1]$$

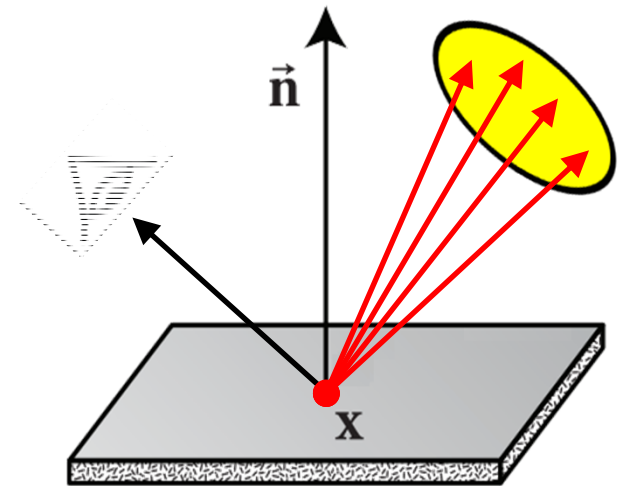
$$L_o = \sum_{i=0}^{N-1} [L_o]_i$$



Illumination **directe** vs. globale

- On peut similairement traiter/généraliser l'illumination directe venant des lumières étendues

ex: lumière **étendue**, surface diffus

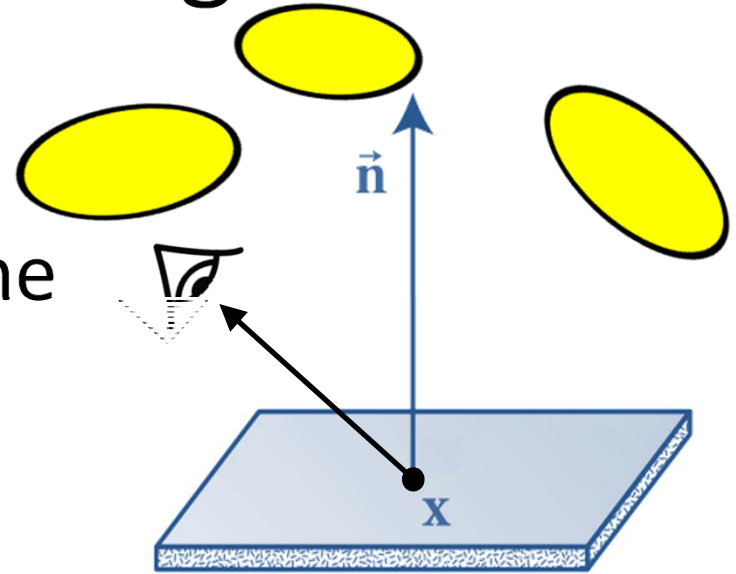


$$L_o(x) = \lim_{n \rightarrow \infty} \sum_{i=0}^{n-1} \left(\frac{I_i}{d_i^2} \right) \frac{\rho}{\pi} V(x, \omega_i) [\vec{n} \cdot \omega_i]$$



Illumination **directe** vs. globale

- En générale, l'illumination directe peut être écrit comme une intégrale:



$$L_o = \lim_{n \rightarrow \infty} \sum_{i=0}^{n-1} \left(\frac{I_i}{d_i^2} \right) \frac{\rho}{\pi} V(x, \omega_i) [\vec{n} \cdot \omega_i]$$

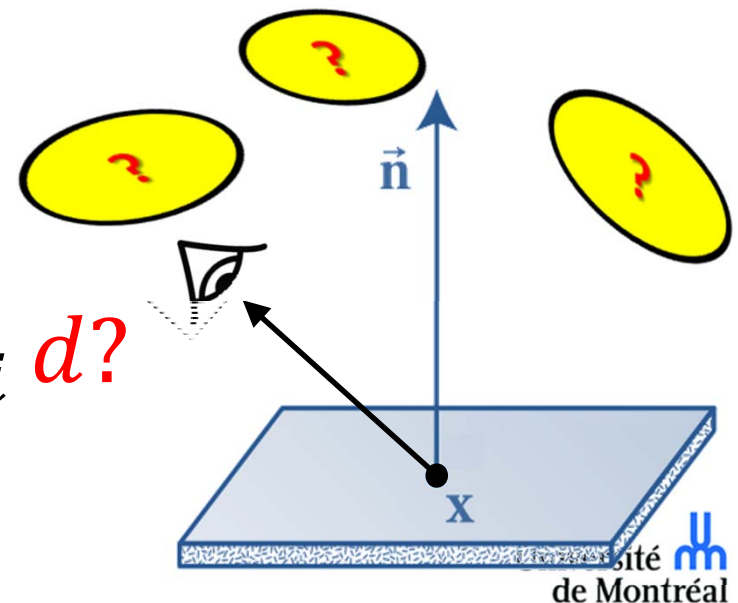
$$L_o = \int L_e \times f_r \times V \times \cos \theta_i$$



Paramétrisation des intégrales en rendu

- On fait l'intégrale d'illumination directe à travers tous les lumières dans notre scène = tous l'illumination *incident* à x à cause des lumières
- Comment peut-on paramétrer les lumières mathématiquement?

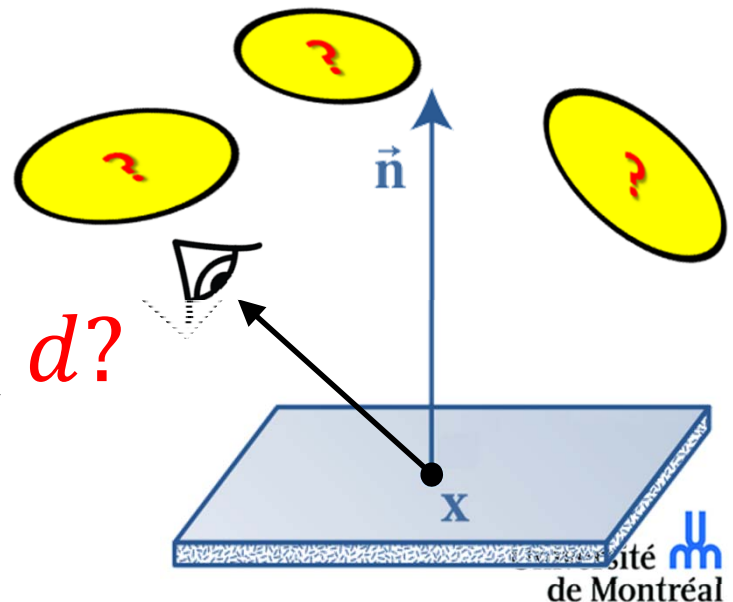
$$L_o = \int_{?} L_e \times f_r \times V \times \cos \theta_i \, d?$$



Paramétrisation des intégrales en rendu

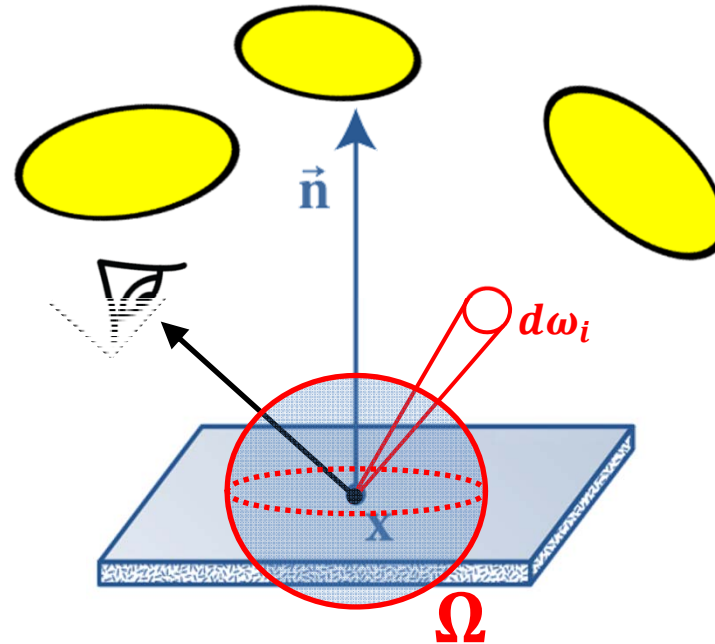
- Comment peut-on paramétrer les lumière mathématiquement?
- Plusieurs options... on va étudier deux entre eux:
 - Paramétrer les **directions** incidents
 - Paramétrer les **surfaces** (aires) des lumières

$$L_o = \int_{?} L_e \times f_r \times V \times \cos \theta_i \, d?$$



Paramétrisation des intégrales en rendu

- Paramétrer les **directions** incidents

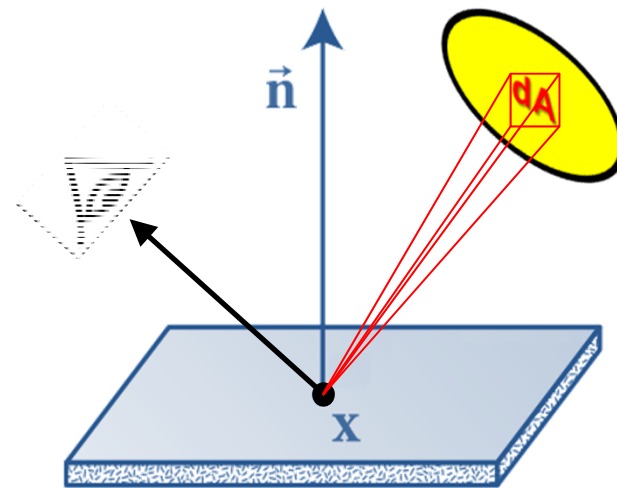


$$L_o = \int_{\Omega} L_e \times f_r \times V \times \cos \theta_i d\omega_i$$



Paramétrisation des intégrales en rendu

- Paramétrer les **surfaces** (aires) des lumières



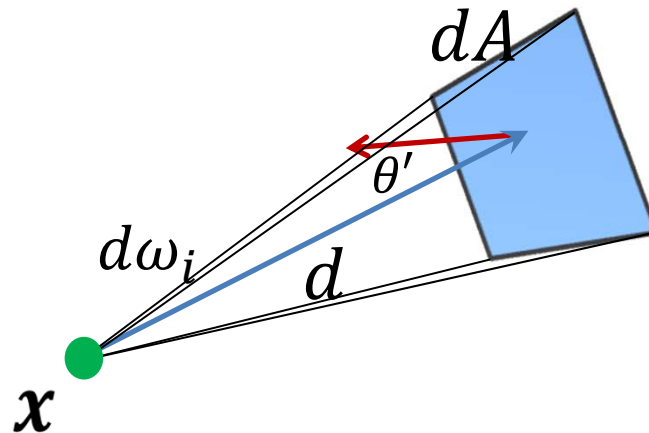
$$L_o = \int_{\mathbf{A}} L_e \times f_r \times V \times \cos \theta_i \, dA$$



Paramétrisation des intégrales en rendu

- On peut facilement convertir entre ces deux paramétrisations:

$$d\omega_i = \frac{[\cos \theta']}{d^2} dA$$



Équation d'illumination directe finale (équation de réflexion)

- Voici la formulation complet de l'équation d'illumination directe sous forme directionnelle, y compris tous les paramètres des fonctions:

$$L_o(x, \omega_o) = \int_{\Omega} L_e(x, \omega_i) f_r(x, \omega_i, \omega_o) V(x, \omega_i) [\cos \theta_i] d\omega_i$$

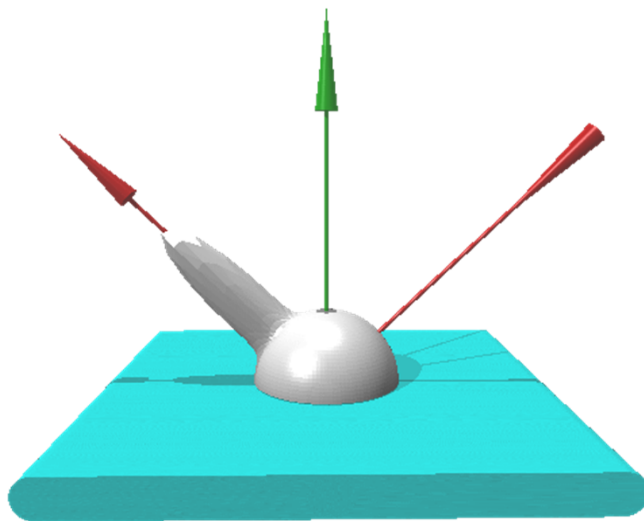


Définition formelle de $f_r(x, \omega_i, \omega_o)$

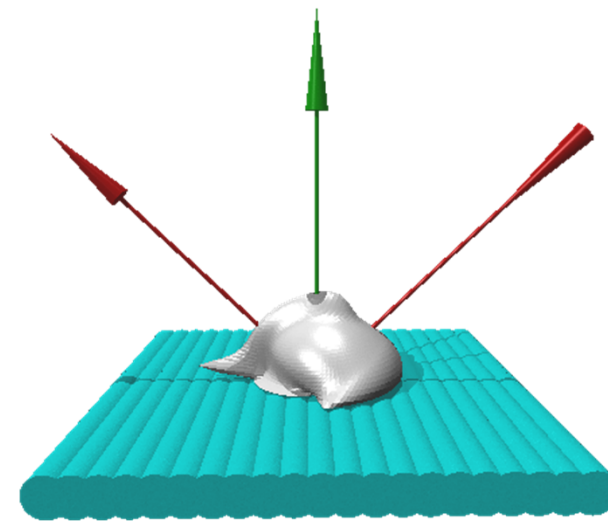
- f_r est le **bidirectional reflectance distribution function (BRDF)**, une fonction qui modèle les propriétés de réflectance d'une matérielle (où de plusieurs matériaux combinées)
- À chaque point x le BRDF f_r est une fonction de 4 dimensions:
 - Deux pour les directions de lumière *incident*, et
 - Deux pour les directions de lumière *sortant*
- Les propriétés mathématiques les plus importantes des BRDFs sont:
 1. réciprocity: $f_r(x, \omega_i, \omega_o) = f_r(x, \omega_o, \omega_i)$
 2. conservation d'énergie: $\int_{\Omega_{in}} f_r(x, \omega_i, \omega_o) \cos \omega_i d\omega_i \leq 1, \forall \omega_o$
 3. positivité: $f_r(x, \omega_i, \omega_o) \geq 0, \forall \omega_i \text{ et } \forall \omega_o$
- Plusieurs équation analytiques existes pour f_r pour des différentes types de réflexions: diffus, spéculaire (miroir, Phong, etc.), etc.
- On peut aussi utiliser des données tabulées capturées à partir du monde réel: `float f_r[theta_i][phi_i][theta_o][phi_o][couleurs]`



Exemples de $f_r(x, \omega_i, \omega_o)$



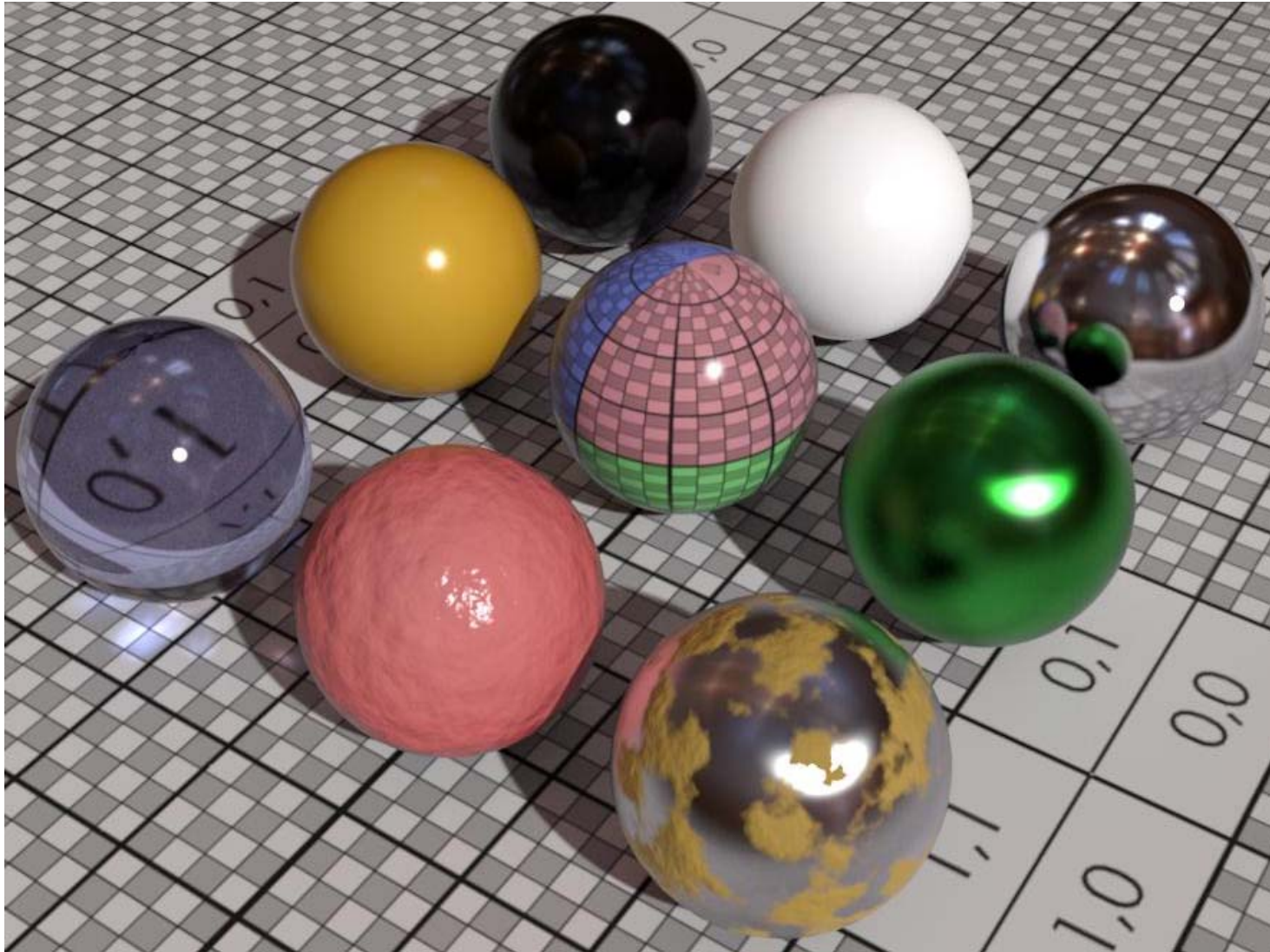
Modèle simple
diffus-spéculaire
isotrope



BRDF d'un modèle
plus complexe
anisotrope



Exemples de $f_r(x, \omega_i, \omega_o)$



BRDF Analytique #1: Diffus

- Le modèle le plus simple:

$$f_r(x, \omega_i, \omega_o) = \frac{\rho_x}{\pi}$$

- Ici, la seule variation dans le shading *d'un point illuminé* sera à cause du terme cosinus entre la direction d'illumination et la normale à x , $(\vec{n} \cdot \omega_i)$, et la variation spatiale de la paramètre de réflectance diffus ρ_x



BRDF Analytique: Spéculaire

- Contrairement à la réflexion diffuse, il y a plusieurs différents modèles conçus pour la réflexion spéculaire et/ou *glossy*
- Chaque modèle est différemment motivés et/ou basé sur des théories de réflexion spéculaire différent
- Nous (ré)étudierons les modèles de Phong et Blinn-Phong dans le cadre de notre définition plus mathématique/théorique du BRDF



BRDF Analytique #2a: Phong

- Le modèle de réflexion le plus simple est celle de **Phong** qui reflète la lumière autour la direction de réflexion miroir selon un cône de réflexion avec un taille ajustable

$$f_r(x, \omega_i, \omega_o) = \frac{r + 1}{2\pi} \frac{(\omega_r(\omega_i) \cdot \omega_o)^r}{\omega_i \cdot \vec{n}}$$

- Il y a deux interprétation de ce modèle:
 - Un flou de la réflexion miroir parfaite, ou
 - Une réflexion miroir parfaite d'un illumination incident floue



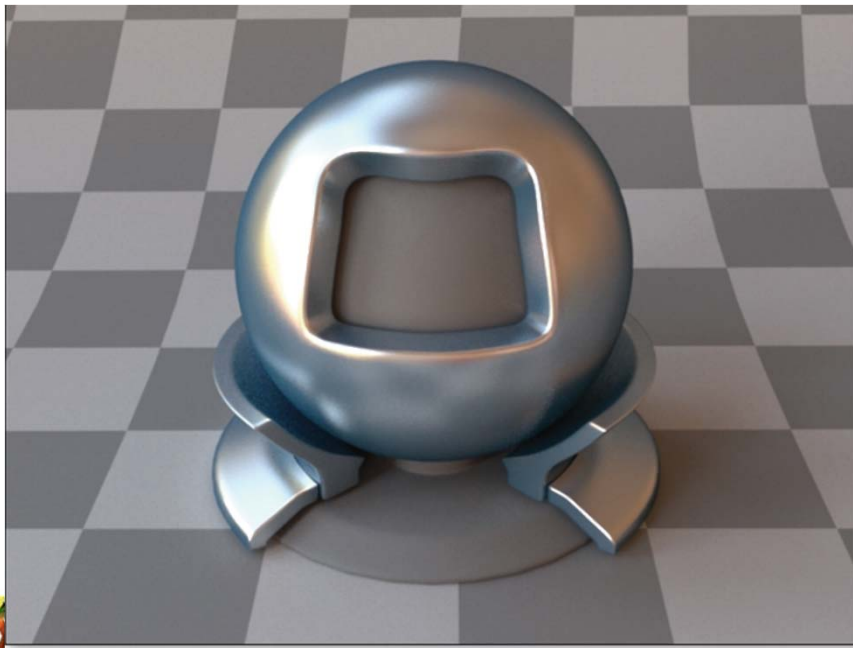
BRDF Analytique #2a: Phong

- Il y a deux interprétations de ce modèle:
 - Un flou de la réflexion miroir parfaite, ou
 - Une réflexion miroir parfaite d'une illumination incidente floue

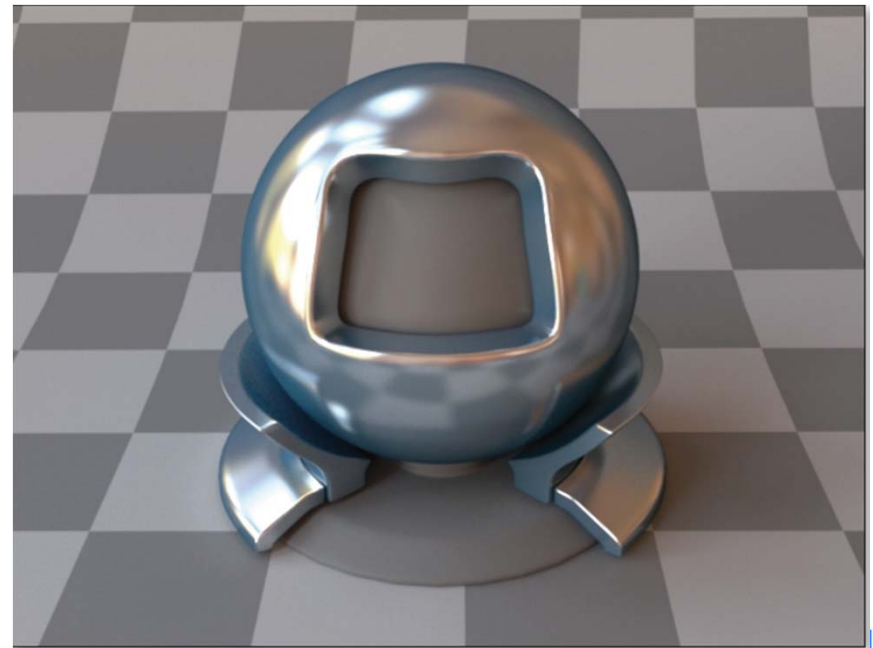


BRDF Analytique #2a: Phong

- Il y a deux interprétations de ce modèle:
 - Un flou de la réflexion miroir parfaite, ou
 - Une réflexion miroir parfaite d'un illumination incident floue



Exponent = 60



Exponent = 300



BRDF Analytique #2b: Blinn-Phong

- Un autre modèle de réflexion isotrope phénoménologique est ce de Blinn-Phong:

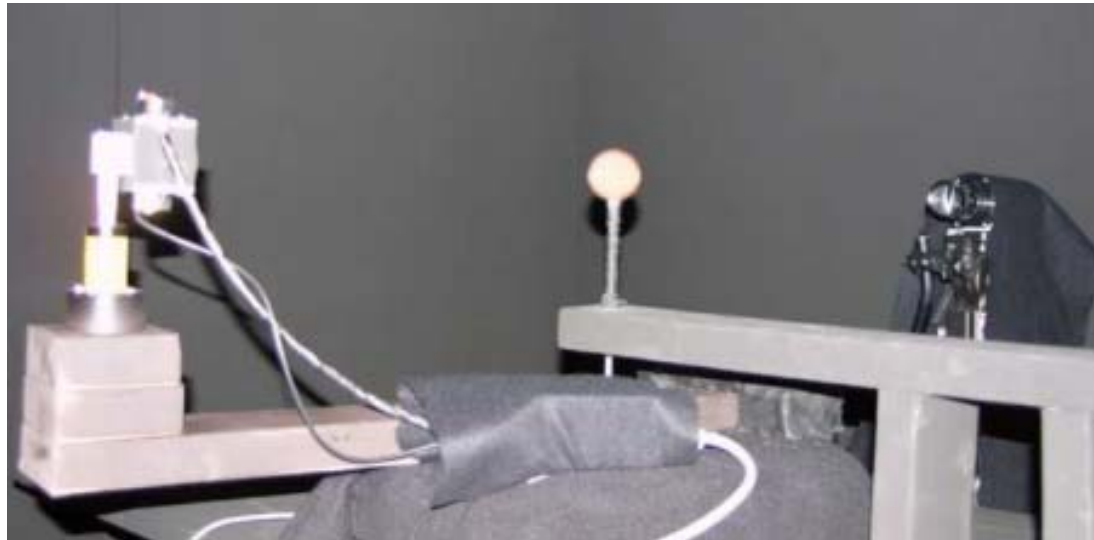
$$f_r(x, \omega_i, \omega_o) = \frac{r + 1}{2\pi} (\omega_h(\omega_i, \omega_o) \cdot \vec{n})^r$$

- Le vecteur *half-way* $\omega_h(\omega_i, \omega_o) = \frac{\omega_i + \omega_o}{\|\omega_i + \omega_o\|}$ est le bissecteur normalisé entre la (les) direction(s) d'illumination et la direction sortant
- La réflexion miroir se produit quand $\vec{n} = \omega_h$ et $r \rightarrow \infty$

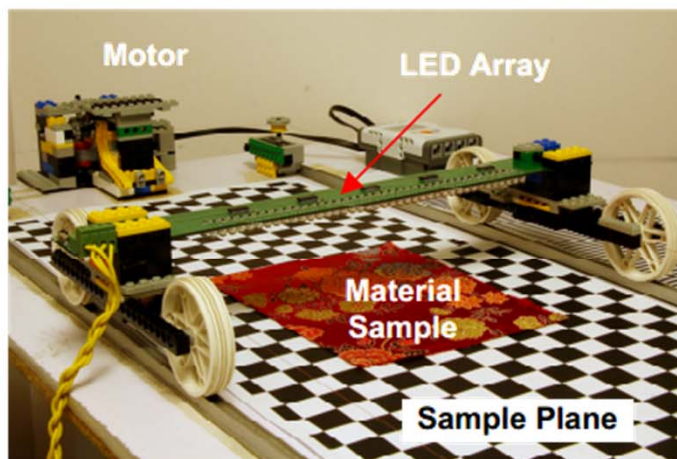
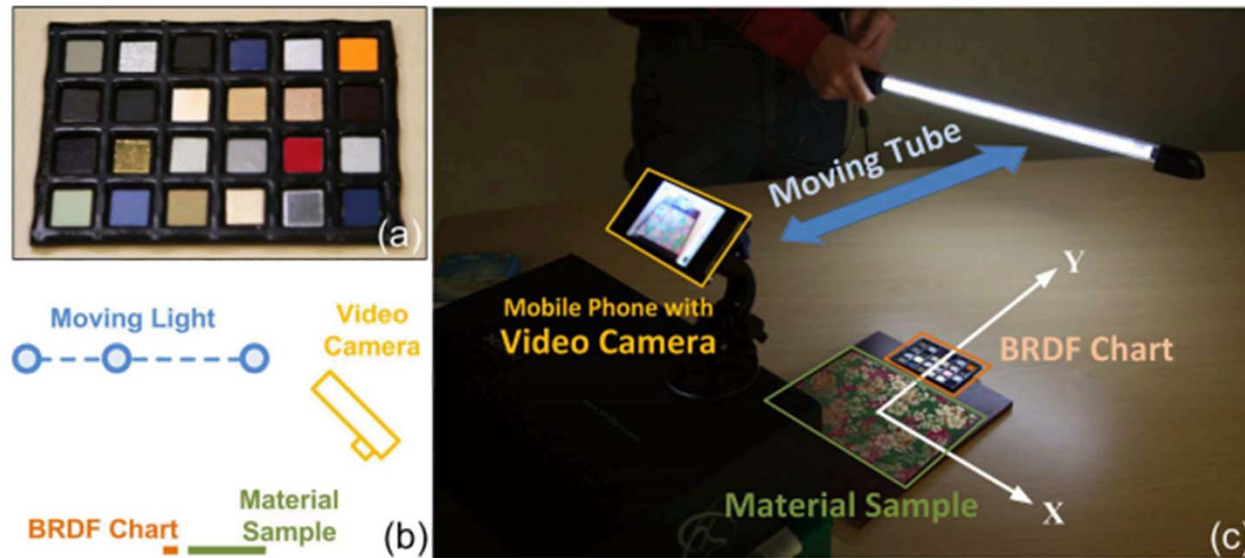


BRDFs Capturés

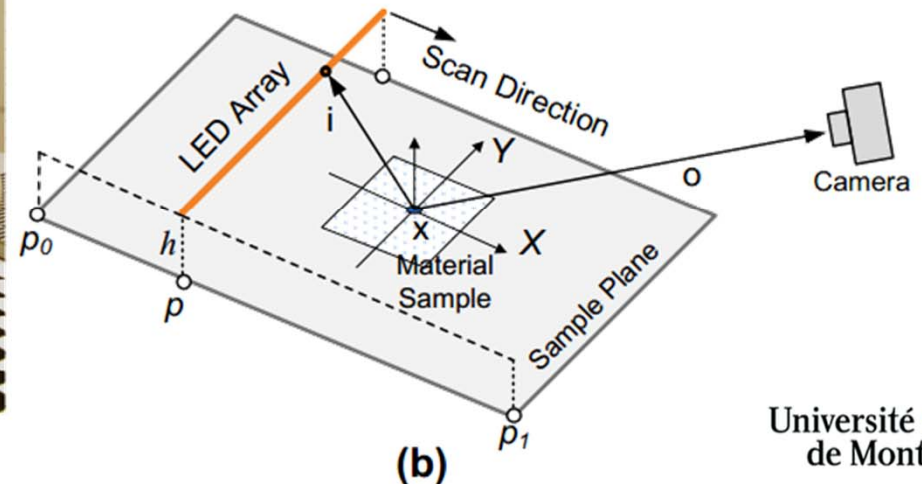
- Au lieu de modéliser l'apparence des objets selon des théories et des modèles mathématiques, une autre solution sera de “simplement” capturer les données 4D (ou 6D) du monde réel



BRDFs Capturés



(a)

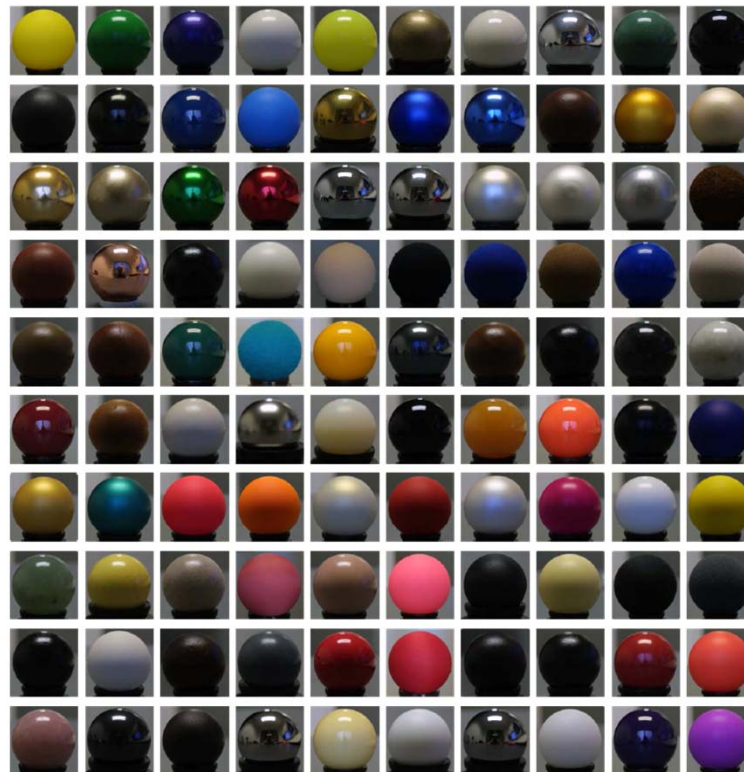


(b)



BRDFs Capturés

- Il reste plusieurs défis avec ces *data-driven BRDFs*:
 - Résolution: comment effectivement échantillonner l'espace 4D (ou 6D)?
 - Rapport signal sur bruit (*SNR*)
 - Comment combiner et/ou interpoler entre les différents BRDFs
 - Coût de stockage
 - Temps de capture



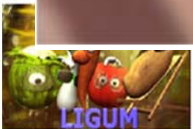
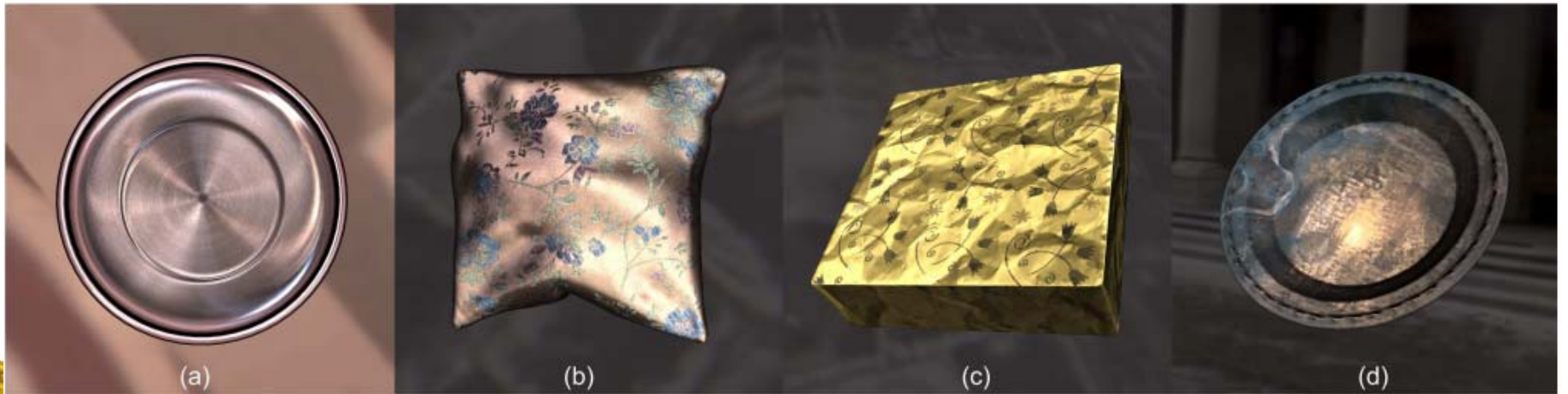
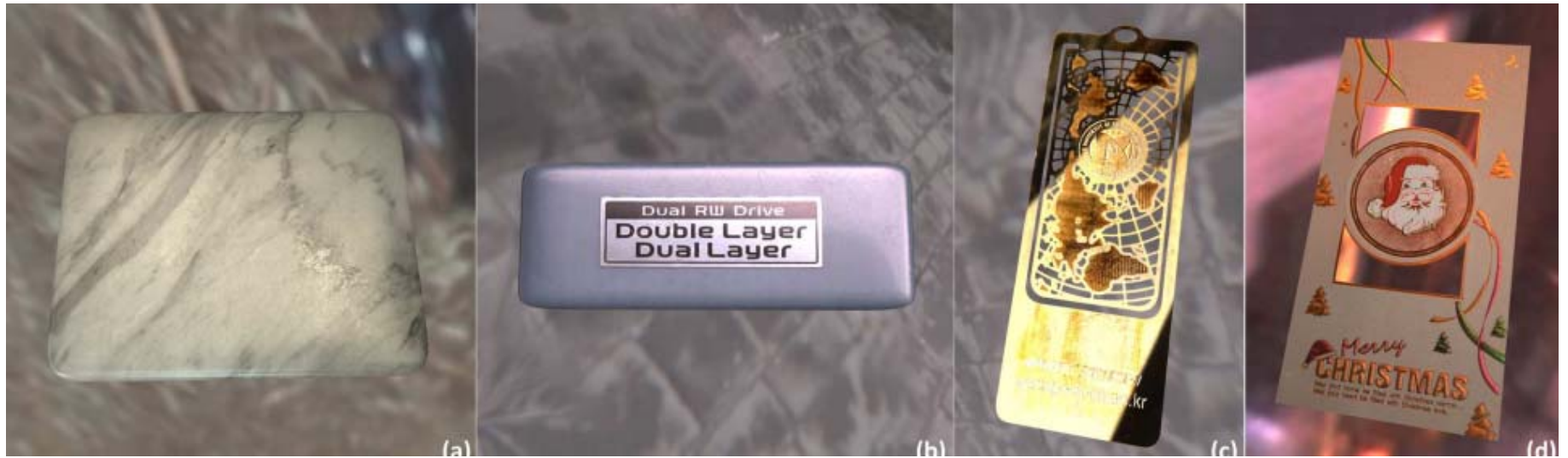
BRDFs Capturés



BRDFs Capturés



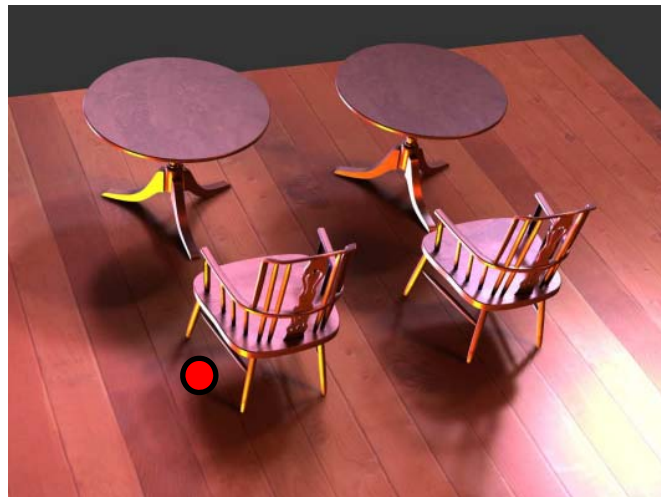
BRDFs Capturés



Visualisation des termes de l'équation d'illumination directe

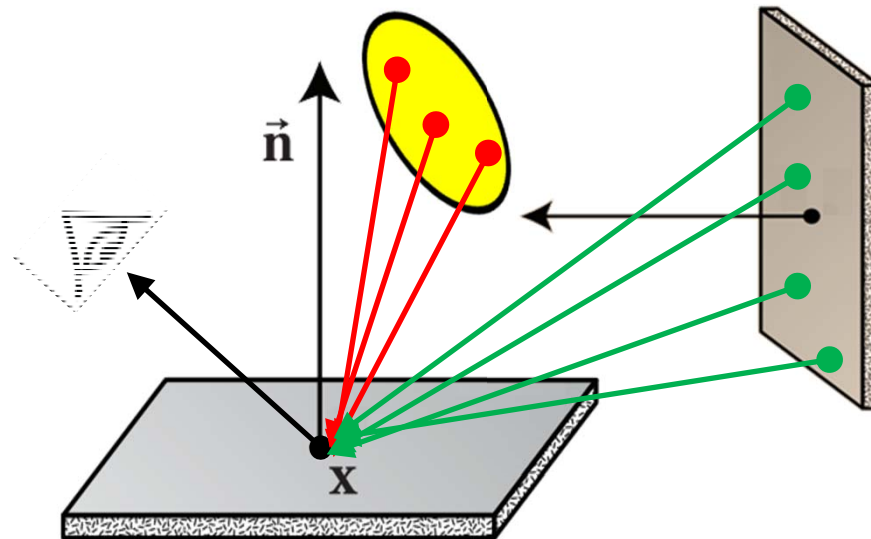
$$\int_{\Omega} L_e(x, \omega_i) V(x, \omega_i) f_r(x, \omega_o, \omega_i) (\vec{n} \cdot \omega_i) d\omega_i$$

$L_e(x, \omega_i)$ $V(x, \omega_i)$ $f_r(x, \omega_o, \omega_i) (\vec{n} \cdot \omega_i)$



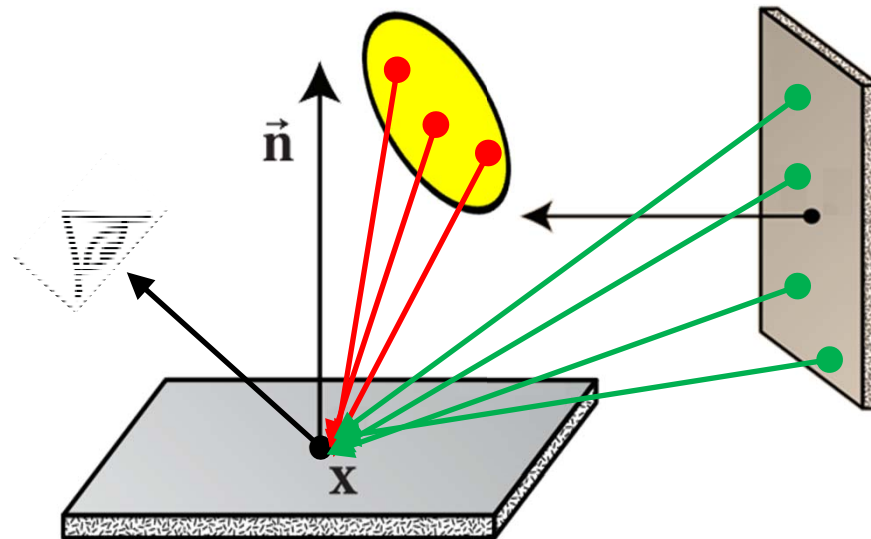
Illumination directe vs. globale

- Illumination globale
 - Lumière provient directement des sources de lumière **ainsi que de la lumière distribuée entre les surfaces**



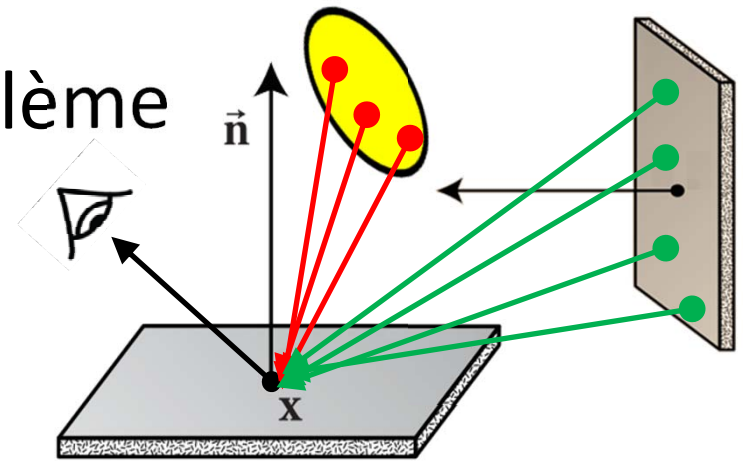
Illumination directe vs. globale

- Comment formuler ce problème mathématiquement?
 - Combinaison conceptuelle de la définition d'une surface et d'une lumière...



Illumination directe vs. globale

- Comment formuler ce problème mathématiquement?
 - Traitement récursif...



L'illumination globale à x ($\equiv L(x)$) est composé de deux composants:

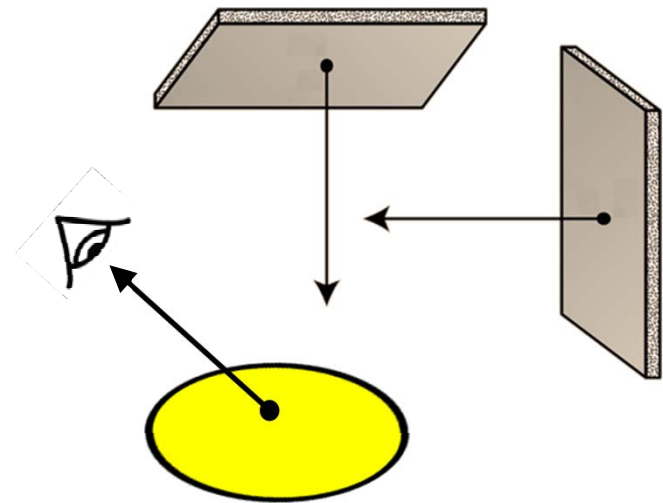
1. Composant émit $\equiv L_e$
2. Composant réfléchi $\equiv L_r$



$$L(x) = L_e(x) + L_r(x)$$

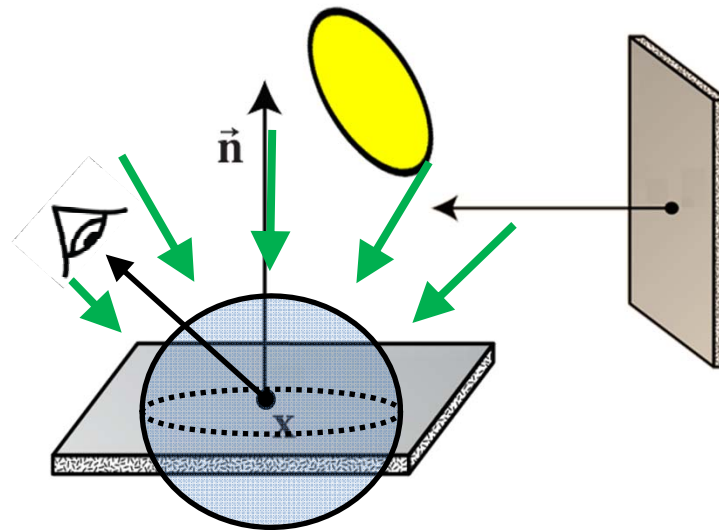
Illumination directe vs. globale

1. Composant émit $\equiv L_e$: l'illumination émit à x
 - est-ce que x demeure sur une surface émissive (ex: une lumière)?
 - Si oui, $L_e \neq 0$ (et $L_r = 0$) et x réside sur une surface lumineuse
 - Si non, $L_e = 0$ (et $L_r \neq 0$) ...



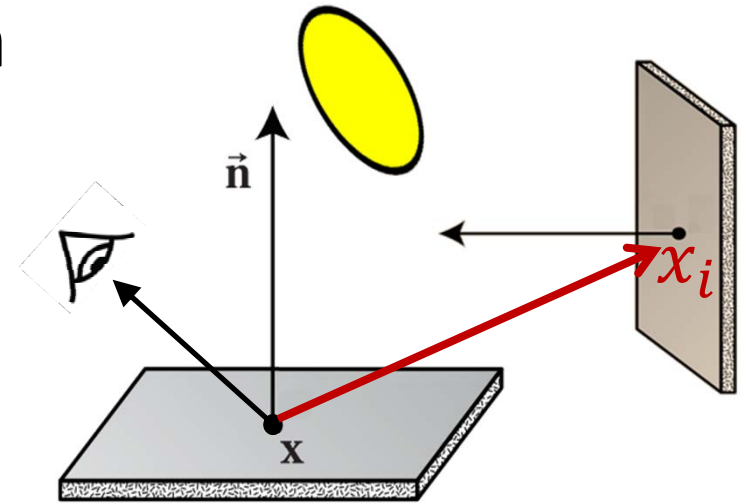
Illumination directe vs. globale

2. Composant réfléchi $\equiv L_r$: l'illumination réfléchi
- si x n'est pas sur une lumière, il réside sur une surface réfléchissante et on a besoin de réfléchir la lumière de tous les directions/surfaces possibles (vers l'œil)



Illumination directe vs. globale

- On a déjà étudié la réflexion à cause des lumières
 - on a besoin de traiter chaque surface non-émissive comme “lumière” secondaire



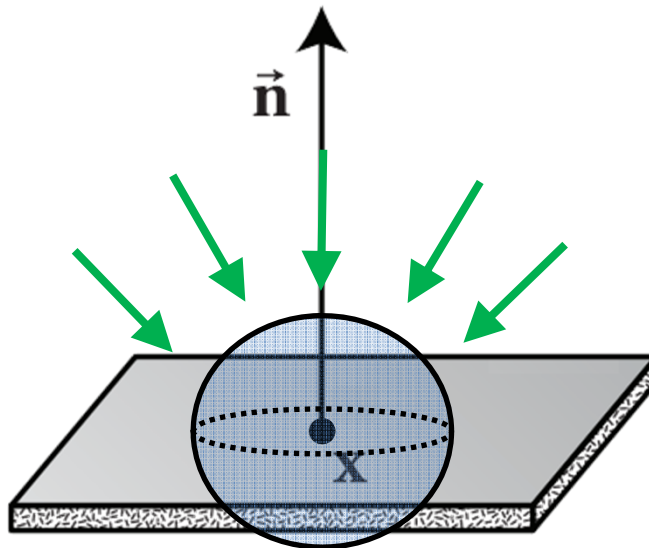
$$L_r(x) = \int_{\Omega} L(x_i) \times f_r \times \cos \theta_i d\omega_i$$



Illumination directe vs. globale

Formulation Finale: l'équation de rendu

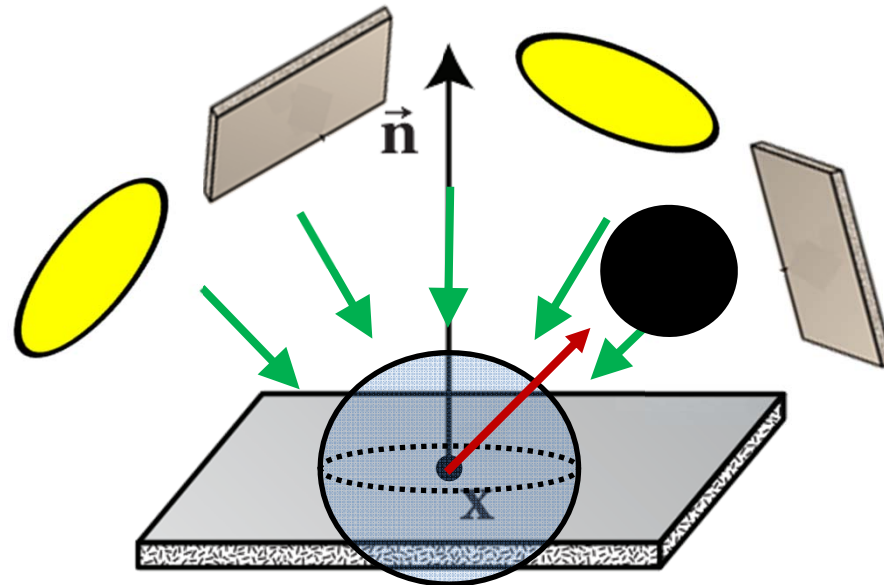
$$L(x, \omega_o) = L_e(x, \omega_o) + L_r(x, \omega_o)$$
$$= L_e(x, \omega_o) + \int_{\Omega} L(x', -\omega_i) f_r(x, \omega_o, \omega_i) \cos \theta_i d\omega_i$$



Illumination directe vs. globale

Formulation Finale: l'équation de rendu

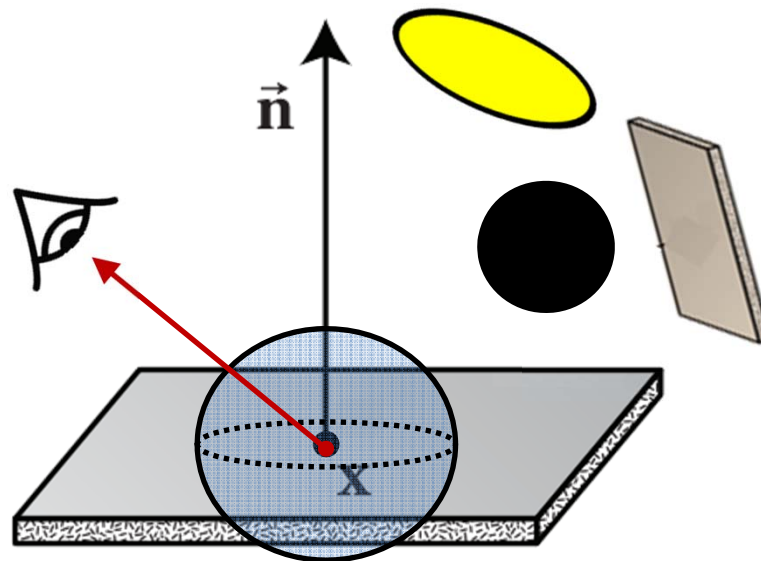
$$L(x, \omega_o) = L_e(x, \omega_o) + L_r(x, \omega_o)$$
$$= L_e(x, \omega_o) + \int_{\Omega} \mathbf{L}(x', -\omega_i) f_r(x, \omega_o, \omega_i) \cos \theta_i d\omega_i$$



Illumination directe vs. globale

Formulation Finale: l'équation de rendu

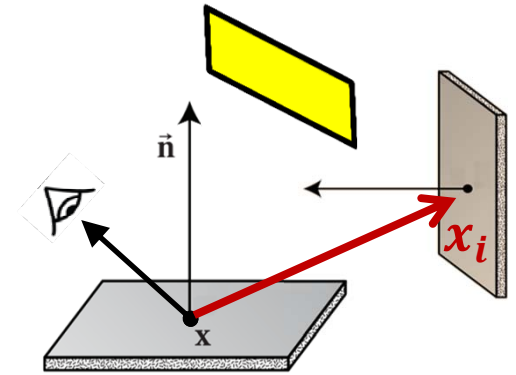
$$L(x, \omega_o) = L_e(x, \omega_o) + L_r(x, \omega_o)$$
$$= L_e(x, \omega_o) + \int_{\Omega} L(x', \omega_i) \mathbf{f}_r(\mathbf{x}, \omega_o, \omega_i) \cos \theta_i d\omega_i$$



Techniques



Radiosité



- Supposons que:
 - toutes les surfaces non-émisives sont diffuses,
 - les lumières ont une émission diffuse ($L_e(x_l, \omega) = L_e(x_l)$)
 - on a seulement des objets/lumières planaires dans la scène
- L'équation de rendu peut être simplifiée dans ce cas:

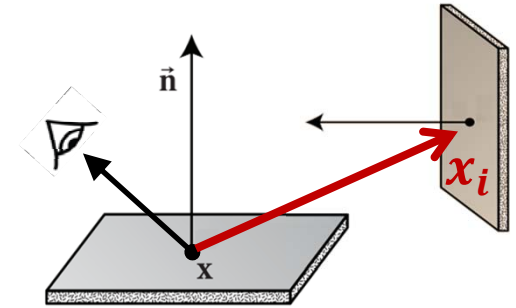
$$L(x, \omega_o) = L_e(x, \omega_o) + \int_{\Omega} L(x', \omega_i) f_r(x, \omega_o, \omega_i) \cos \theta_i d\omega_i$$



$$L(x) = L_e(x) + \frac{\rho_x}{\pi} \int_{\Omega} L(x') \cos \theta_i d\omega_i$$



Radiosité



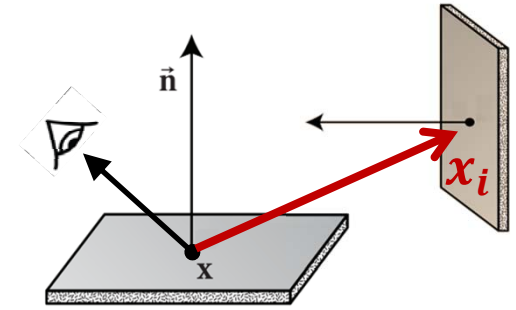
- Si la géométrie de la scène ne change pas, nous pouvons discrétiser et précalculer la solution à cette équation de radiosité pour tous les x comme fonction de l'intensité de toutes les lumières
- Premièrement on va discrétiser l'équation (en discrétisant la scène en N éléments planaires)

$$L(x) = L_e(x) + \frac{\rho_x}{\pi} \int_A L(x') \cos \theta_i \frac{\cos \theta_{x \rightarrow i}}{d_{x \rightarrow i}^2} dA$$

$$L[x] = L_e[x] + \frac{\rho[x]}{\pi} \sum_{i=1}^N L[i^*] \cos \theta_i \frac{\cos \theta_{x \rightarrow i}}{d_{x \rightarrow i}^2} A_i$$



Radiosité



$$L[x] = L_e[x] + \underbrace{\sum_{i=1}^N L[i] \rho[x] V(x \rightarrow i) \cos \theta_i \frac{\cos \theta_{x \rightarrow i}}{\pi d_{x \rightarrow i}^2} A_i}_{F[x][i]}$$

- On peut re-écrire l'équation sous forme matricielle:

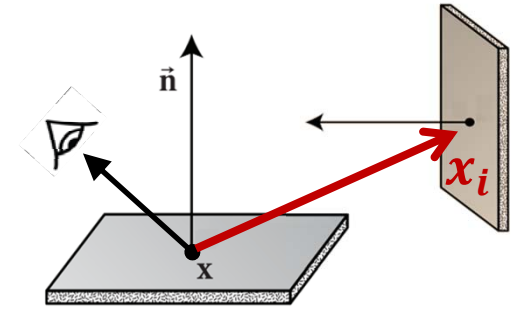
$$\underbrace{\begin{bmatrix} L[1] \\ \vdots \\ L[N] \end{bmatrix}}_{\mathbf{L}} = \underbrace{\begin{bmatrix} L_e[1] \\ \vdots \\ L_e[N] \end{bmatrix}}_{\mathbf{L}_e} + \underbrace{\begin{bmatrix} F[1][1] & \cdots & F[1][N] \\ \vdots & \ddots & \vdots \\ F[N][1] & \cdots & F[N][N] \end{bmatrix}}_{\mathbf{T}} \underbrace{\begin{bmatrix} L[1] \\ \vdots \\ L[N] \end{bmatrix}}_{\mathbf{L}}$$

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



Radiosité

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

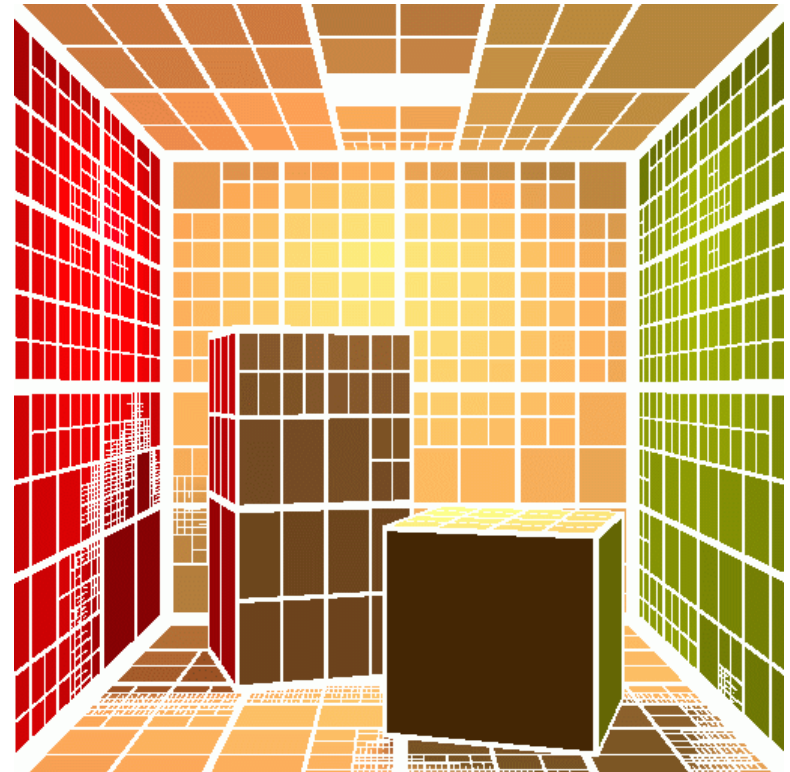
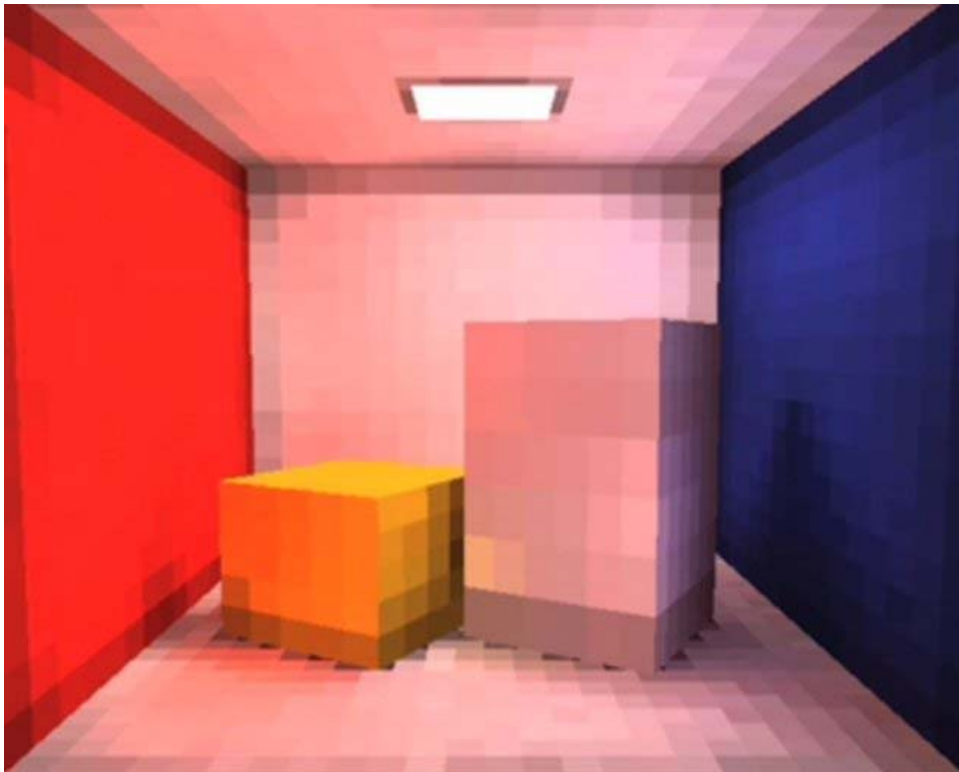
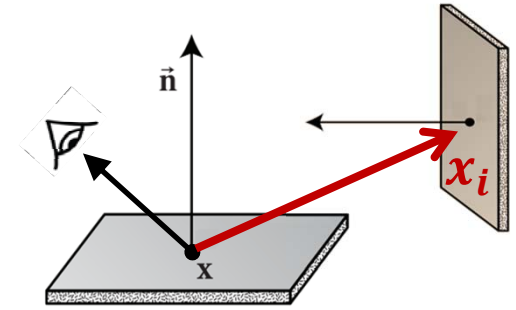


- Comment résoudre pour \mathbf{L} ?
 1. Pour une scène fixe, on peut précalculer les éléments de \mathbf{T} en utilisant un lanceur de rayon pour résoudre le terme $V(x \rightarrow i)$
 2. Après un peu d'algèbre linéaire: $\mathbf{L} = (\mathbf{I} - \mathbf{T})^{-1} \mathbf{L}_e$
 3. Précalculer* $(\mathbf{I} - \mathbf{T})^{-1}$
 4. Pour calculer les couleurs à tous les éléments planaires de la scène on a seulement à déterminer les valeurs émissif de chaque élément (\mathbf{L}_e) et calculer $\mathbf{L} = (\mathbf{I} - \mathbf{T})^{-1} \mathbf{L}_e$



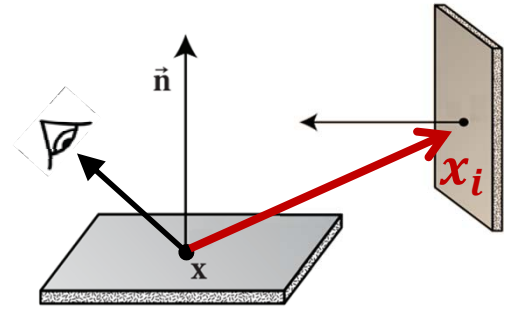
Radiosité

$$\mathbf{L} = (\mathbf{I} - \mathbf{T})^{-1} \mathbf{L}_e$$



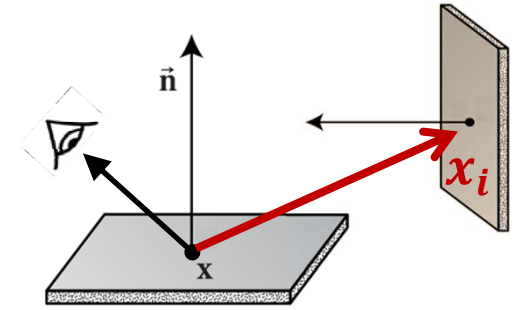
Radiosité

$$\mathbf{L} = (\mathbf{I} - \mathbf{T})^{-1} \mathbf{L}_e$$



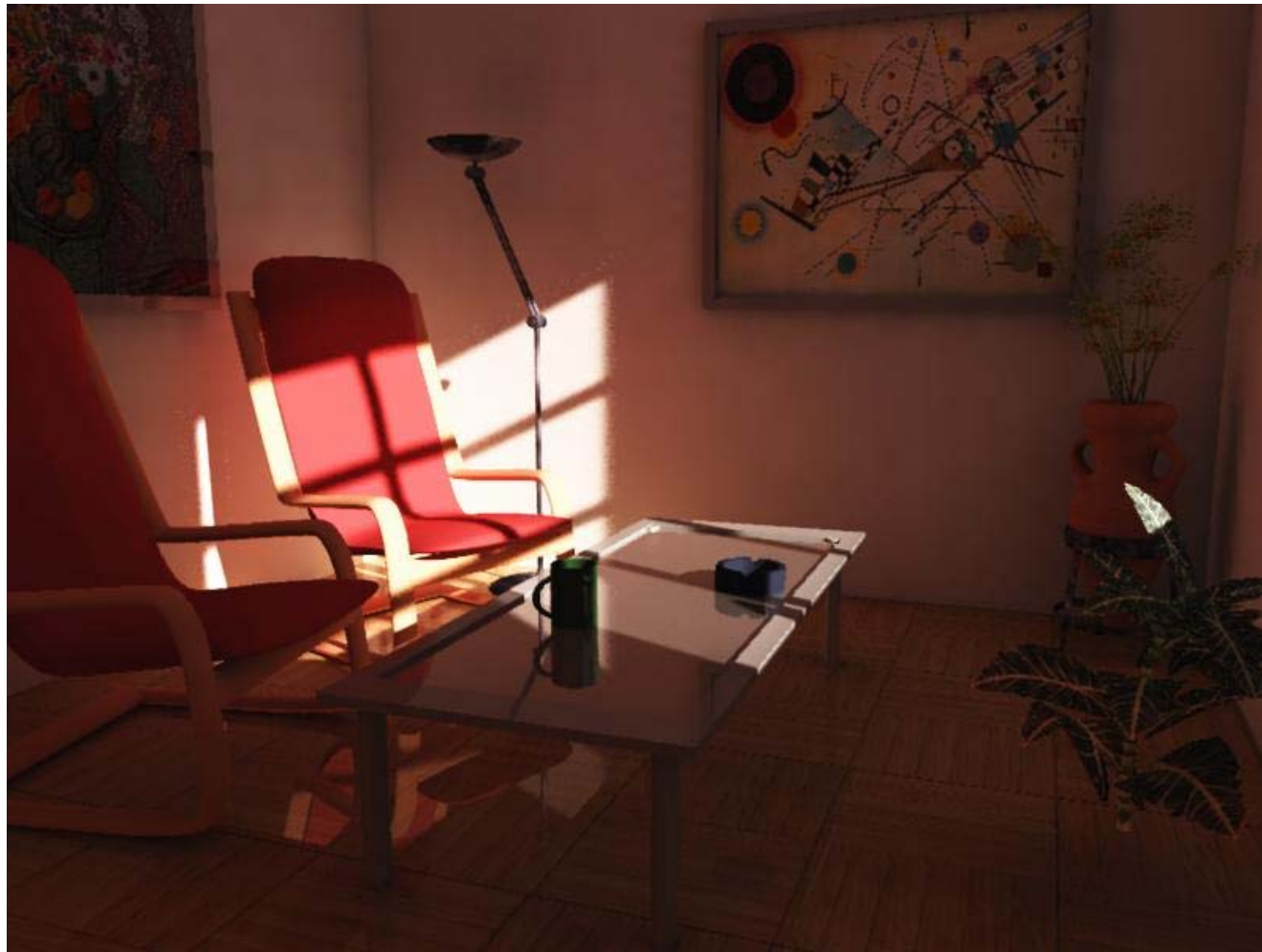
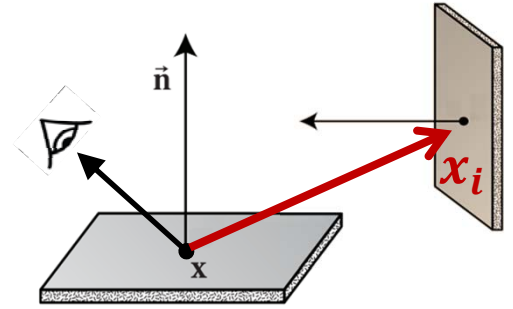
Radiosité

$$\mathbf{L} = (\mathbf{I} - \mathbf{T})^{-1} \mathbf{L}_e$$

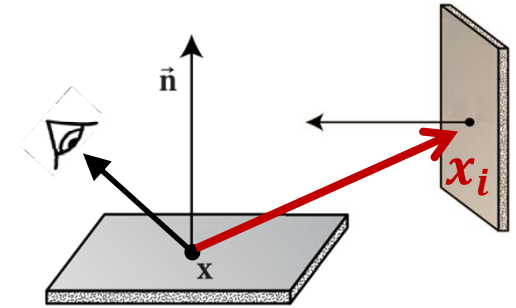


Radiosité

$$\mathbf{L} = (\mathbf{I} - \mathbf{T})^{-1} \mathbf{L}_e$$



Radiosité



- Avantages de radiosity:
 - Seulement besoin de calculer la valeur quand L_e change
 - Espace objet
- Désavantages de radiosity:
 - Pour N éléments, T à une taille de N^2
 - Temps de précalcul
 - Stockage
 - L'inversion d'une matrice est difficile
 - Limitations sur la scène: taille, diffus, statique



Comment peut-on résoudre
l'équation de rendu plus générale?



L'équation de rendu

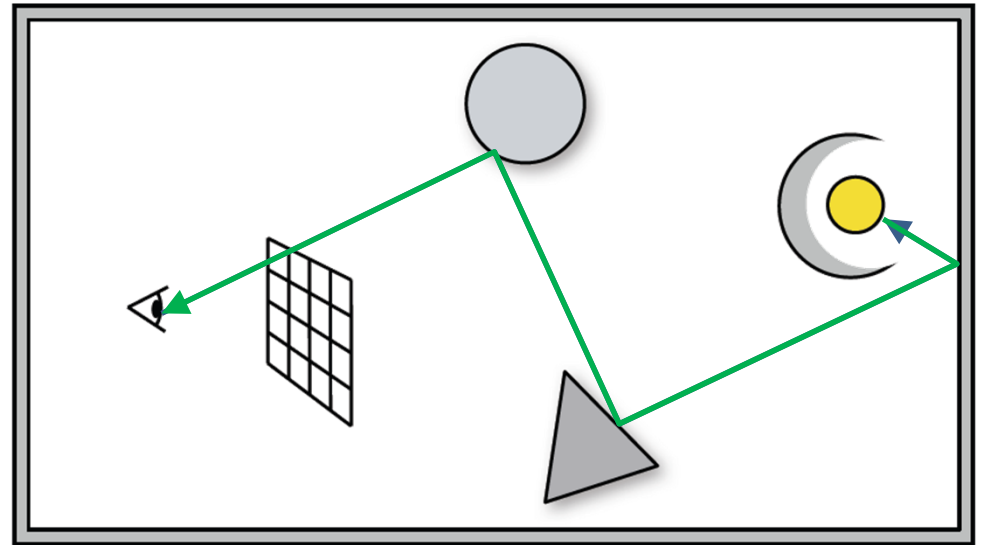
- Comment résoudre une équation d'intégrale multidimensionnelle comme l'équation de rendu?
- Conceptuellement, on peut étendre l'idée de lancer de rayons pour résoudre l'équation *numériquement*
- On veut connecter des points sur le plan d'image (pixels) avec les points de lumière
 - Cela correspond à un échantillonnage de l'espace multidimensionnelle



L'équation de rendu

- Plusieurs questions:

- Est-ce qu'on commence de l'œil?
Est-ce qu'on commence de la lumière?



- Comment est-ce qu'on génère nos chemins? Comment est-ce qu'on peut garantir une solution précise (même, qu'est-ce qui constitue une solution précise?)
- On a besoin d'une théorie d'intégration numérique robuste à notre problème....



Monte Carlo

- Les techniques de Monte Carlo nous permettrons de résoudre les intégrales multidimensionnelles en utilisant des bases probabiliste
 - Voici un révision de certaines notions et propriétés probabilistes importantes
- Supposons que X est une variable aléatoire
- La fonction de distribution cumulative (CDF) de X est

$$cdf(x) = \Pr[X \leq x]$$

- La fonction de densité de probabilité correspondante est

$$pdf(x) = \frac{d}{dx} cdf(x)$$

- et on voit que la probabilité que X prendra une valeur entre a et b est

$$\Pr[a \leq X \leq b] = \int_a^b pdf(x) dx$$



Monte Carlo

- L'espérance mathématique d'un variable aléatoire $Y = f(X)$ sur une domaine Z est

$$E[Y] = \int_Z f(z) pdf(z) dz$$

- et ça variance est

$$\sigma^2[Y] = E[(Y - E[Y])^2] = E[Y^2] - E[Y]^2$$

- Voici quelques propriétés qui nous aiderons d'analyser le comportement de convergence de nos intégrateurs Monte Carlo

$$E[aY] = a E[Y]$$

$$E\left[\sum_i Y_i\right] = \sum_i E[Y_i]$$

$$\sigma^2[aY] = a^2 \sigma^2[Y]$$

$$\sigma^2\left[\sum_i Y_i\right] = \sum_i \sigma^2[Y_i]$$



Intégration Monte Carlo

- Un *estimateur Monte Carlo* pour une intégrale (avec une dimensionnalité arbitraire)

$$F = \int_X f(x) dx$$

- est défini comme

$$\bar{F} = \frac{1}{N} \sum_{i=0}^{N-1} \frac{f(X_i)}{pdf(X_i)} \approx \int_X f(x) dx$$

- On peut montrer mathématiquement que $E[\bar{F}] = F$



Intégration Monte Carlo

$$\bar{F} = \frac{1}{N} \sum_{i=0}^{N-1} \frac{f(X_i)}{pdf(X_i)}$$

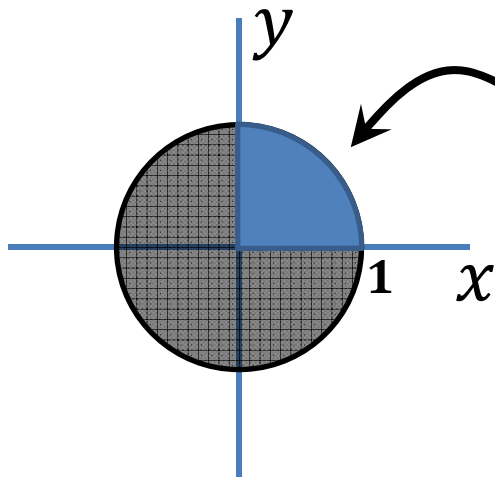
- Alors compte tenu de la définition de l'estimateur Monte Carlo, le processus pour évaluer \bar{F} est divisé en 3 étapes:
 1. Choisir une distribution* pdf selon laquelle nous distribuons nos points d'échantillonnage aléatoire
 2. Générer N points aléatoire (les X_i)
 3. Évaluer $\bar{F} = \frac{1}{N} \sum_{i=0}^{N-1} \frac{f(X_i)}{pdf(X_i)}$
- Noter que les points X_i peuvent aussi être générés à la volée pendant l'étape 3



Intégration Monte Carlo: Exemple en 1D

$$\bar{F} = \frac{1}{N} \sum_{i=0}^{N-1} \frac{f(X_i)}{pdf(X_i)}$$

- Prenons une exemple simple où on peut vérifier la valeur analytiquement:



$$\int_0^1 \sqrt{1-x^2} dx$$
$$= \frac{\pi}{4} \approx 0.785398$$

```
intégral = 0
for( i = 0 à N-1 )
  X[i] = rand();
  pdf[i] = 1;
  intégral += sqrt(1.-
  X[i]*X[i])/pdf[i]

intégral = intégral / N
```

1. Choisir pdf : on va utiliser une distribution uniforme

$$pdf(x) = \frac{1}{b-a} = \frac{1}{1-0} = 1$$

2. Générer les N points X_i
3. Évaluer \bar{F}



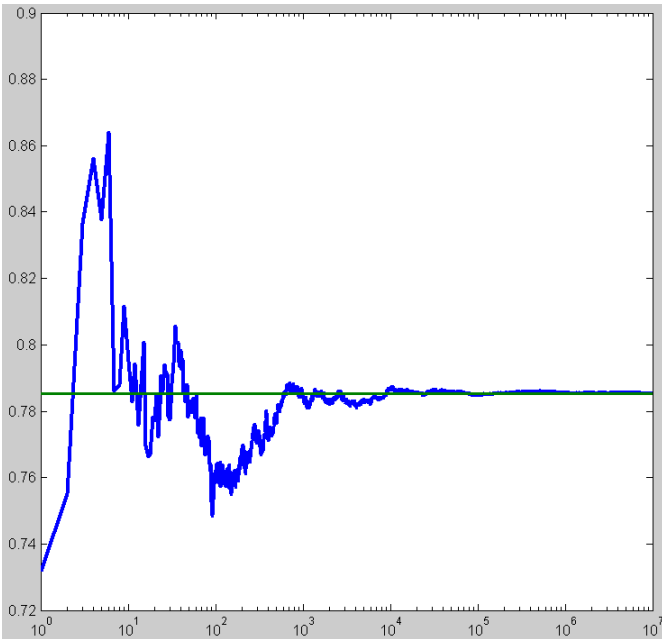
Intégration Monte Carlo

$$\bar{F} = \frac{1}{N} \sum_{i=0}^{N-1} \frac{f(X_i)}{pdf(X_i)}$$

- On peut déterminer la **convergence** de notre estimateur \bar{F} selon une analyse de la déviation standard:

$$\begin{aligned}\sigma^2[\bar{F}] &= \sigma^2 \left[\frac{1}{N} \sum_{i=0}^{N-1} \frac{f(X_i)}{pdf(X_i)} \right] \\ &= \frac{1}{N^2} \sum_{i=0}^{N-1} \sigma^2 \left[\frac{f(X_i)}{pdf(X_i)} \right] \\ &= \frac{1}{N^2} \sum_i^N \sigma^2[Y_i] \\ &= \frac{1}{N} \sigma^2[Y]\end{aligned}$$

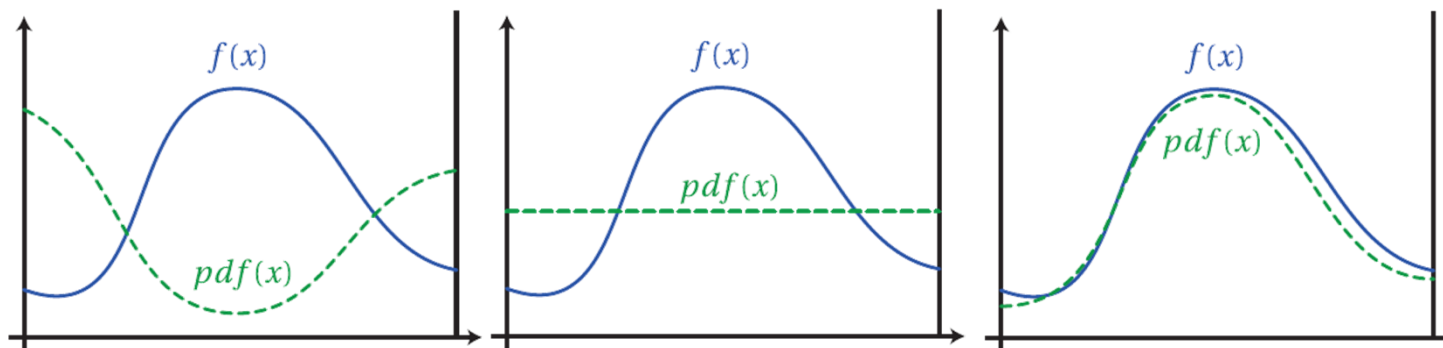
$$\sigma[Y] \propto \frac{1}{\sqrt{N}}$$



Intégration Monte Carlo

$$\bar{F} = \frac{1}{N} \sum_{i=0}^{N-1} \frac{f(X_i)}{pdf(X_i)}$$

- Un bût important quand on utilise l'intégration Monte Carlo est de réduire la variance de nos points X_i
- Deux manières typiques de réduire la variance sont:
 - Stratification, et
 - Importance sampling
- Stratification: divise la domaine d'intégration en plusieurs sous-domaines et effectue une intégration Monte Carlo sous chacun des sous-domaines
- Importance sampling: choisi $pdf(x)$ afin de placer les points X_i aux locations où $f(x)$ a une valeur plus élevée

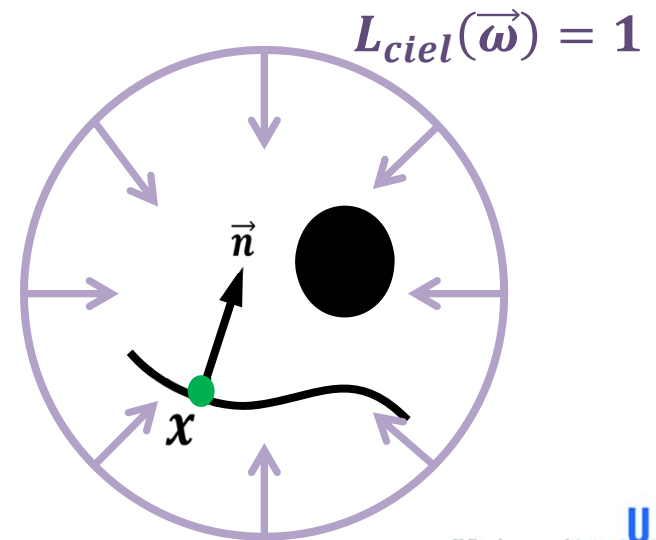


Exemple 1: Ambient Occlusion

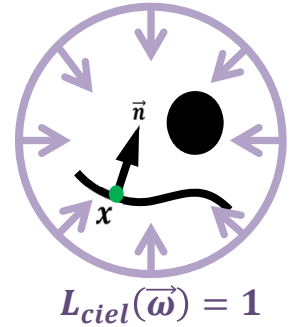
- Considère le prochain scénario: on veut déterminer le shading d'une scène composée complètement des objets **diffus** (tous avec $S_\lambda = k_d = 1$) grâce à l'**illumination directe** du ciel
 - ici on traite le ciel comme une source sphérique à distance infini de tous les points x avec une profile d'émission de $L_{in}(x, \vec{\omega}) = L_{ciel}(\vec{\omega}) = 1$

$$L_o(x, \vec{\omega}_o) = L_{ao}(x) = \int_{\Omega} \frac{\max(\vec{n}_x \cdot \vec{\omega}, 0)}{\pi} V(x, \vec{\omega}) d\vec{\omega}$$

- On va calculer cette intégrale avec un estimateur Monte Carlo (ou plusieurs différents estimateurs)



Exemple 1: Ambient Occlusion



- On écrit l'estimateur Monte Carlo (avec une *pdf* générale) pour

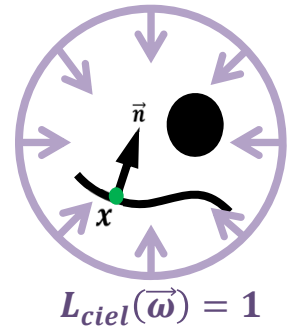
$$L_{ao}(x) = \int_{\Omega} \frac{\max(\vec{n}_x \cdot \vec{\omega}, 0)}{\pi} V(x, \vec{\omega}) d\vec{\omega}$$

comme

$$\overline{L}_{ao}(x) = \frac{1}{N} \sum_{i=0}^{N-1} \frac{\max(\vec{n}_x \cdot \vec{\omega}_i, 0) V(x, \vec{\omega}_i)}{\pi pdf(\vec{\omega}_i)}$$



Exemple 1: Ambient Occlusion



$$\overline{L}_{ao}(x) = \frac{1}{N} \sum_{i=0}^{N-1} \frac{\max(\vec{n}_x \cdot \vec{\omega}_i, 0) V(x, \vec{\omega}_i)}{\pi pdf(\vec{\omega}_i)}$$

- Et maintenant, comme avec tous les estimateur Monte Carlo, le problème se réduit à:
 - choisir $pdf(\vec{\omega}_i)$, et
 - distribuer les échantillons aléatoires $\vec{\omega}_i$ selon cette pdf
- On va discuter trois choix de $pdf(\vec{\omega}_i)$, comment distribuer les échantillons selon ces $pdfs$, et l'effet sur la convergence/variance de nos estimateurs
 - Rappelez-vous que un but pendant l'intégration Monte Carlo est de réduire la variance de nos estimateurs en utilisant des techniques comme ex: stratification, importance sampling, etc.



Ambient Occlusion – Estimateur 1

(pdf = direction sphérique uniforme)

$$\overline{L}_{ao}(x) = \frac{1}{N} \sum_{i=0}^{N-1} \frac{\max(\vec{n}_x \cdot \vec{\omega}_i, 0) V(x, \vec{\omega}_i)}{\pi \text{pdf}(\vec{\omega}_i)}$$

- Comme l'exemple en 1D la semaine dernière on peut commencer avec le choix de *pdf* le plus simple: une distribution uniforme
 - Dans le cas 1D on avait $\text{pdf}(x_i) = \frac{1}{\text{longueur}(X)} = \frac{1}{\int_X dx} = \frac{1}{b-a}$ mais maintenant nous échantillons les **directions sur Ω**
 - cela correspond à échantillonner les points sur une sphère de rayon = 1
 - Alors pour distribuer nos échantillons uniformément sur la surface de la sphère on aura $\text{pdf}(\vec{\omega}_i) = \frac{1}{\text{aire}(\Omega)} = \frac{1}{\int_{\Omega} d\vec{\omega}} = ?$



De côté:

géométrie et intégration sur la sphère (avec rayon = 1)

- Étant donné que nos équations de rendu contiennent des fonctions paramétrisés sur Ω , nous réviserons brièvement comment représenter et effectuer des calcul analytique dans cette domaine
- Plusieurs paramétrisations *mathématiques** existent pour représenter les points sur la surface d'une sphère

- Voici une représentation souvent utilisé:

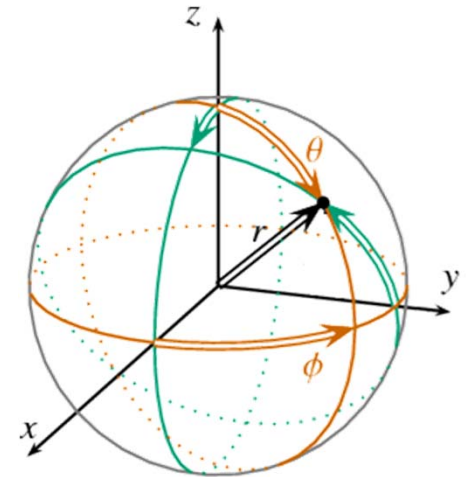
$$\Omega \rightarrow [0 \leq \theta \leq \pi] \times [0 \leq \phi \leq 2\pi]$$

$$\vec{\omega} \leftrightarrow (\theta, \phi) \leftrightarrow (x, y, z) \text{ où } x^2 + y^2 + z^2 = 1$$

$$(x, y, z) = (\cos \phi \sin \theta, \sin \theta \sin \phi, \cos \theta)$$

$$(\theta, \phi) = \left(\arccos(z), \arctan\left(\frac{y}{x}\right) \right)$$

$$d\vec{\omega} = \sin \theta d\theta d\phi$$



Ambient Occlusion – Estimateur 1

(pdf = direction sphérique uniforme)

$$\overline{L_{ao}}(x) = \frac{1}{N} \sum_{i=0}^{N-1} \frac{\max(\vec{n}_x \cdot \vec{\omega}_i, 0) V(x, \vec{\omega}_i)}{\pi \text{pdf}(\vec{\omega}_i)}$$

- Alors avec une *pdf* uniforme sur la sphère on avait:

$$\text{pdf}(\vec{\omega}_i) = \frac{1}{\text{aire}(\Omega)} = \frac{1}{\int_{\Omega} d\vec{\omega}} = \frac{1}{\int_{\theta=0}^{\pi} \int_{\phi=0}^{2\pi} \sin \theta d\theta d\phi} = \frac{1}{4\pi}$$

- Notre estimateur pour L_{ao} , avec une *pdf* uniforme, devient:

$$\overline{L_{ao}}(x) = \frac{4}{N} \sum_{i=0}^{N-1} \max(\vec{n}_x \cdot \vec{\omega}_i, 0) V(x, \vec{\omega}_i)$$

et on a seulement à déterminer comment générer ces échantillons...



Ambient Occlusion – Estimateur 1

(pdf = direction sphérique uniforme)

$$\overline{L_{ao}}(x) = \frac{4}{N} \sum_{i=0}^{N-1} \max(\vec{n}_x \cdot \vec{\omega}_i, 0) V(x, \vec{\omega}_i)$$

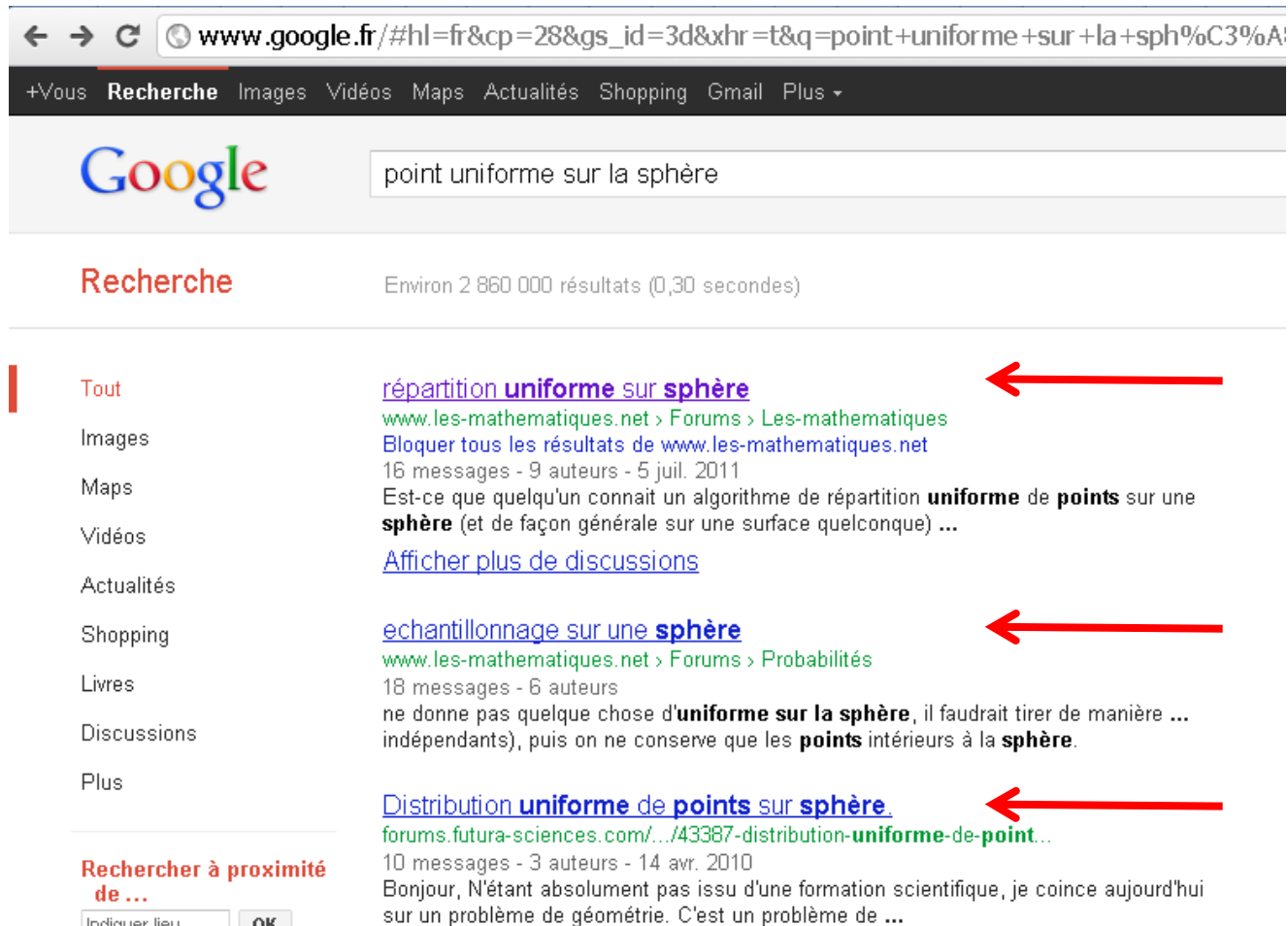
- Comment générer les $\vec{\omega}_i$ avec une distribution uniforme sur la sphère?
- Méthode 1: je vous explique beaucoup de maths supplémentaires.
- Ou...



Ambient Occlusion – Estimateur 1

(pdf = direction sphérique uniforme)

- Méthode 2: on demande Google



The screenshot shows a Google search results page for the query "point uniforme sur la sphère". The search bar contains the text "point uniforme sur la sphère". The results are displayed in a list format with a left-hand navigation menu. Three red arrows point to the first three search results.

Recherche Environ 2 860 000 résultats (0,30 secondes)

Tout [répartition uniforme sur sphère](#) ←

Images [www.les-mathematiques.net > Forums > Les-mathematiques](#)
[Bloquer tous les résultats de www.les-mathematiques.net](#)
16 messages - 9 auteurs - 5 juil. 2011

Maps Est-ce que quelqu'un connaît un algorithme de répartition **uniforme** de **points** sur une **sphère** (et de façon générale sur une surface quelconque) ...
[Afficher plus de discussions](#)

Vidéos

Actualités

Shopping [echantillonnage sur une sphère](#) ←

Livres [www.les-mathematiques.net > Forums > Probabilités](#)
18 messages - 6 auteurs

Discussions ne donne pas quelque chose d'**uniforme sur la sphère**, il faudrait tirer de manière ... indépendants), puis on ne conserve que les **points** intérieurs à la **sphère**.

Plus [Distribution uniforme de points sur sphère](#) ←

[forums.futura-sciences.com/.../43387-distribution-uniforme-de-point...](#)
10 messages - 3 auteurs - 14 avr. 2010

Bonjour, N'étant absolument pas issu d'une formation scientifique, je coince aujourd'hui sur un problème de géométrie. C'est un problème de ...

Rechercher à proximité de ...



Ambient Occlusion – Estimateur 1

(pdf = direction sphérique uniforme)

$$\overline{L}_{ao}(x) = \frac{4}{N} \sum_{i=0}^{N-1} \max(\vec{n}_x \cdot \vec{\omega}_i, 0) V(x, \vec{\omega}_i)$$

- Et on voit que:
 - au lieu de générer deux points aléatoires uniformément entre $0 \leq \theta \leq \pi$ et $0 \leq \phi \leq 2\pi$ selon:

$$\theta_{uniforme} = M_PI * rand();$$

$$\phi_{uniforme} = 2 * M_PI * rand();$$

- on doit effectivement utiliser cette formulation:

$$\theta_{uniforme} = \text{acos}(1 - 2 * rand());$$

$$\phi_{uniforme} = 2 * M_PI * rand();$$



Ambient Occlusion – Estimateur 1

(pdf = direction sphérique uniforme)

- Donc, en résumé, on a défini notre première estimateur Monte Carlo pour l'ambient occlusion en:

1. Choissant une *pdf* uniforme sur la surface de la sphère:

$$pdf(\vec{\omega}_i) = \frac{1}{4\pi}$$

2. Générant des échantillons $\vec{\omega}_i = (\theta_i, \phi_i)$ selon cette distribution:

$$\theta_i = \text{acos}(1 - 2 * \text{rand}());$$

$$\phi_i = 2 * \text{M_PI} * \text{rand}();$$

3. Calculant notre valeur d'ambient occlusion:

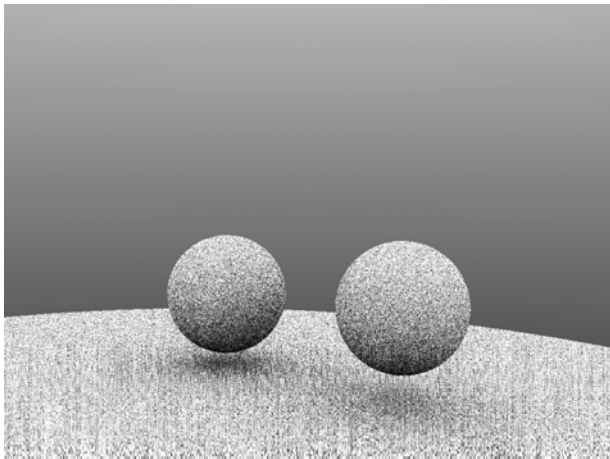
$$\overline{L}_{ao}(x) = \frac{4}{N} \sum_{i=0}^{N-1} \max(\vec{n}_x \cdot \vec{\omega}_i, 0) V(x, \vec{\omega}_i)$$



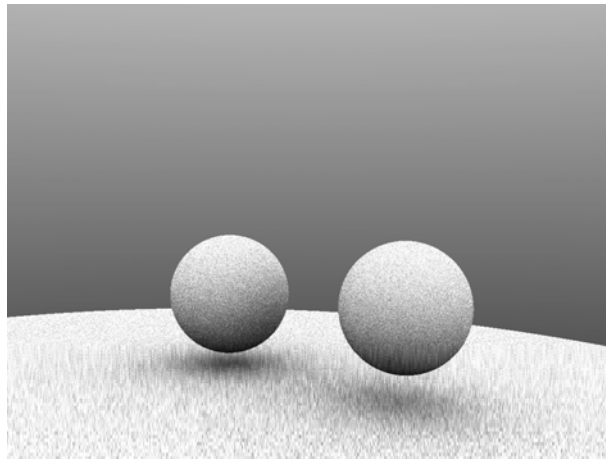
Ambient Occlusion – Estimateur 1

(pdf = direction sphérique uniforme)

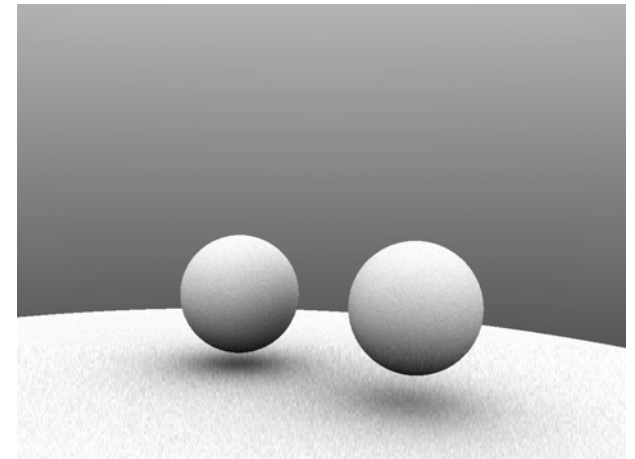
- Résultats (utilisant le `rand()` de `stdlib.h`):



$N = 10$



$N = 100$



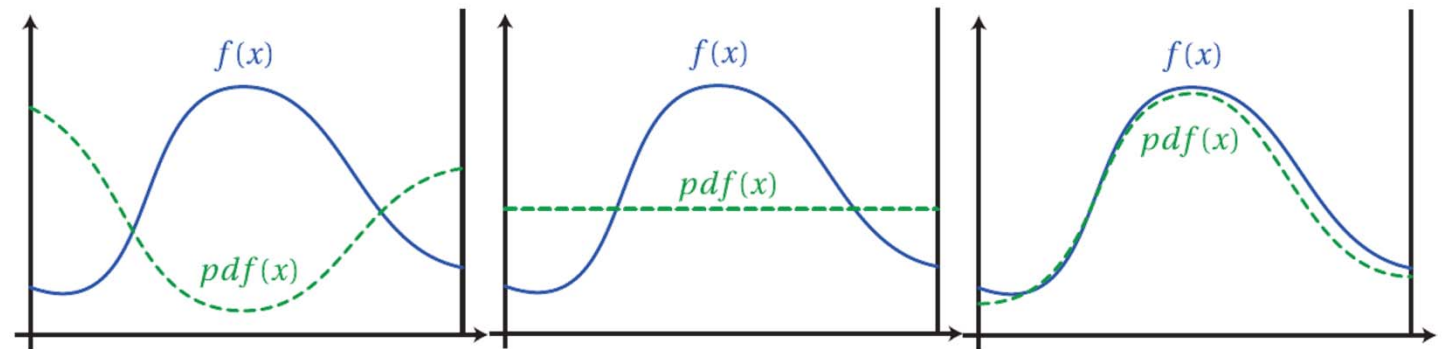
$N = 1000$



Ambient Occlusion: des estimateurs plus efficaces?

- Rappelez-vous qu'un des buts d'intégration Monte Carlo est de réduire la variance de l'estimateur
 - Un moyen de réduire la variance est d'utiliser une *pdf* qui distribue nos échantillons selon la forme de notre intégrand

$$\bar{F} = \frac{1}{N} \sum_{i=0}^{N-1} \frac{f(X_i)}{pdf(X_i)}$$



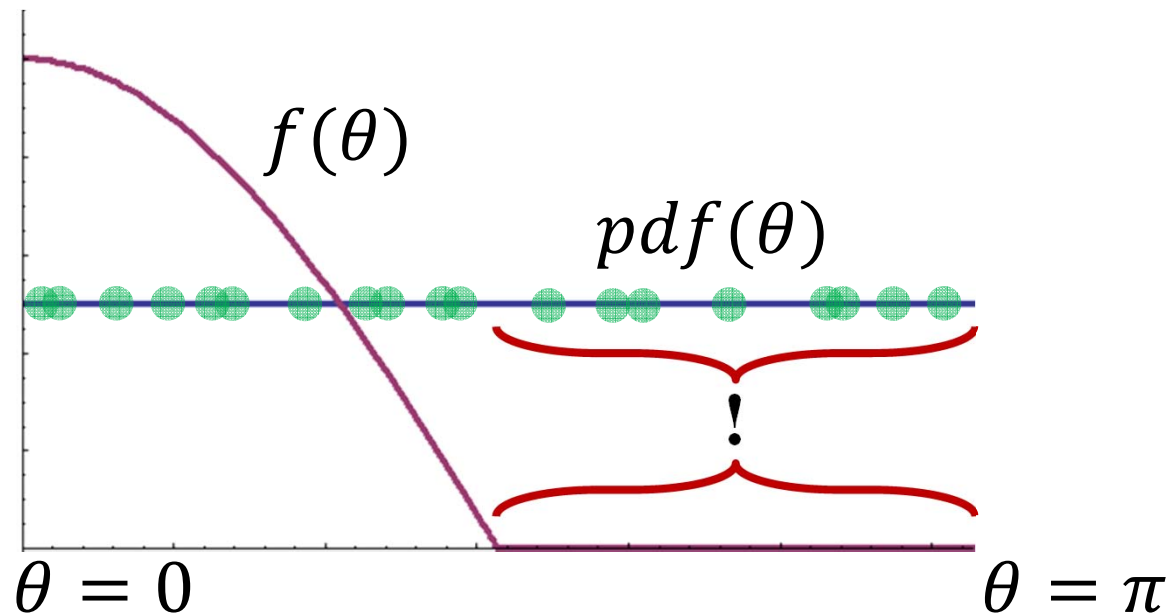
- Présentement, notre intégrand est $f(\omega) = \frac{\max(\vec{n}_x \cdot \vec{\omega}, 0) V(x, \vec{\omega})}{\pi}$ mais
notre *pdf* est $\frac{1}{4\pi}$



Ambient Occlusion: réduire la variance

- Si nous fixons $\vec{n}_x = \vec{z} = (0,0,1)$ et nous *ignorons la visibilité* nous pouvons étudier l'intégrande et la *pdf* en manière visuelle:

- $f(\omega)|_{\vec{n}_x=\vec{z}} = \frac{\max((0,0,1) \cdot (\cos \phi \sin \theta, \sin \theta \sin \phi, \cos \theta), 0)}{\pi} = \frac{\max(\cos \theta, 0)}{\pi} = f(\theta)$



- Clairement cette *pdf* n'est pas optimale

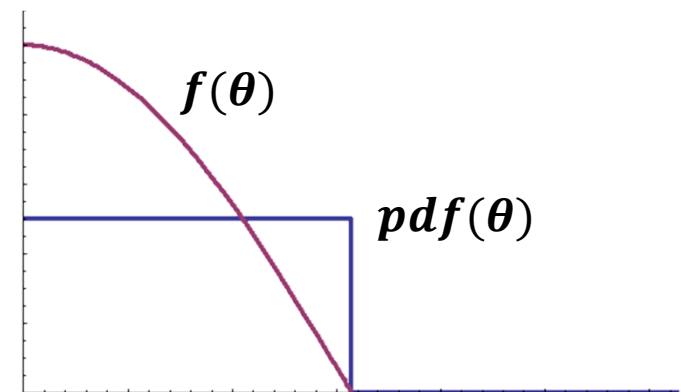


Ambient Occlusion – Estimateur 2

(pdf = direction **hémisphérique** uniforme)

- Nous pouvons améliorer notre estimateur en éliminant la distribution des échantillons où $f(\vec{\omega})$ est 0
 - ces régions contribue rien à notre intégrale finale!
 - la nouvelle distribution est uniforme sur l'hémisphère autour de la normale \vec{n}_x avec une $pdf(\vec{\omega}) = \frac{1}{\int_{H_{n_x}} d\vec{\omega}} = \frac{1}{2\pi}$
 - On génère nos échantillons $\vec{\omega}_i = (\theta_i, \phi_i)$ selon cette pdf avec:

```
 $\theta_i = \text{acos}(\text{sqrt}(\text{rand}()));$   
 $\phi_i = 2 * \text{M\_PI} * \text{rand}();$   
 $\text{if}(\text{dot}(\vec{n}_x, \omega_i) > 0) \text{accepter};$   
 $\text{else rejeter};$ 
```



- Notre nouveau estimateur est:

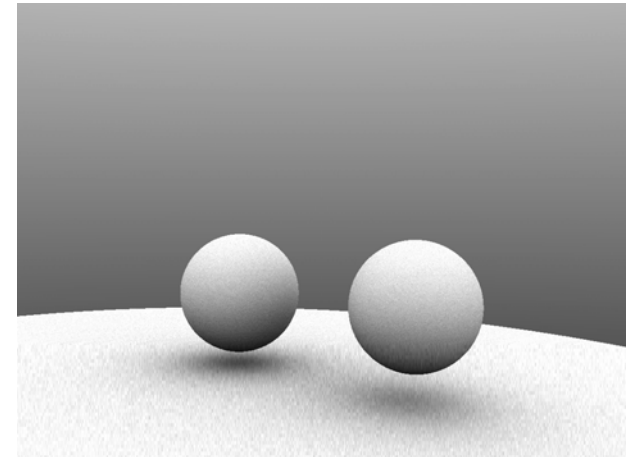
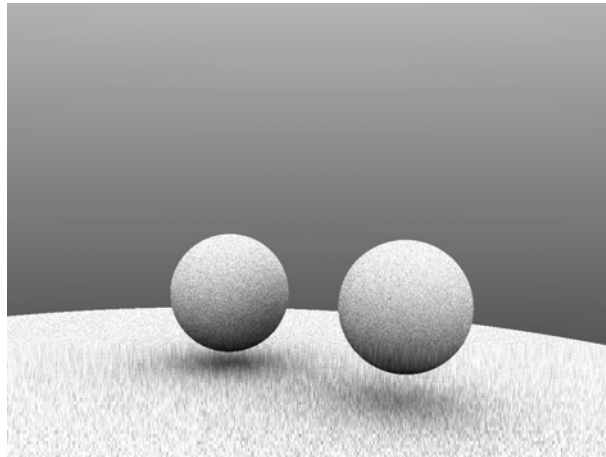
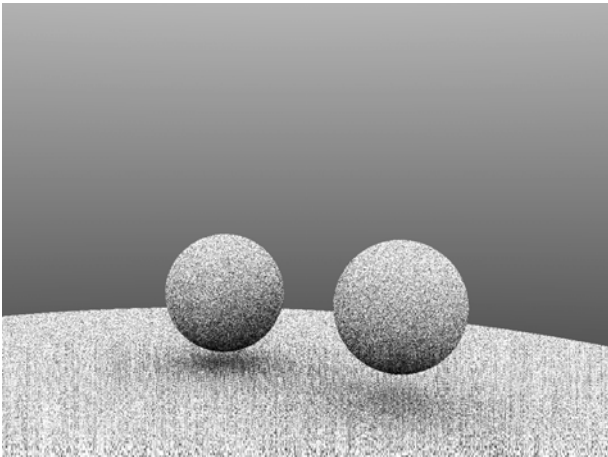
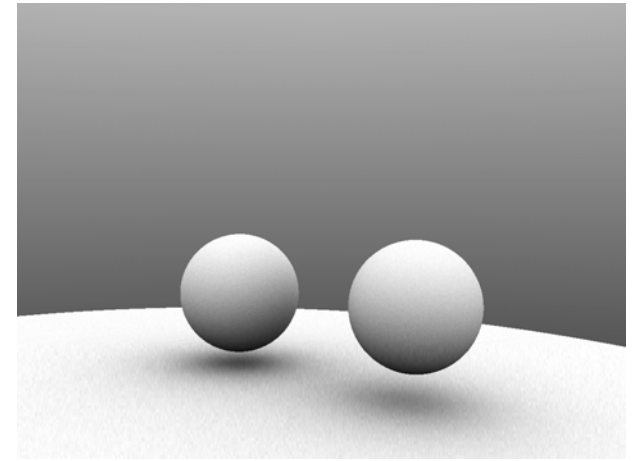
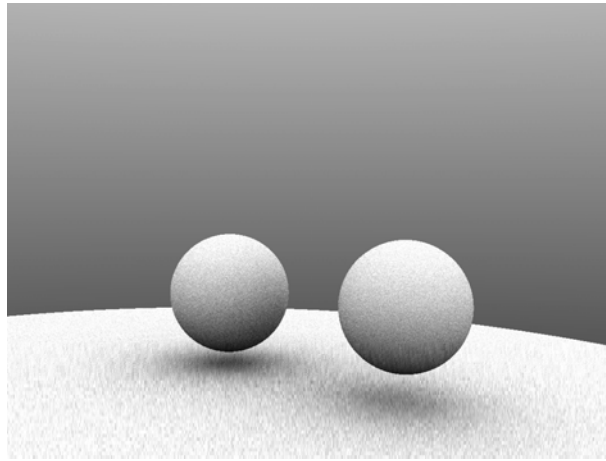
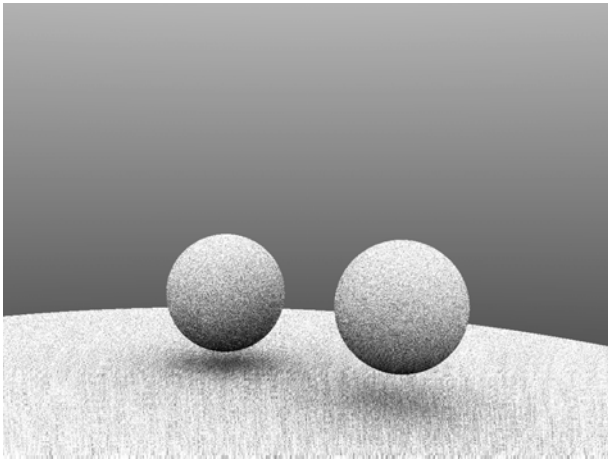
$$\overline{L_{ao}}(x) = \frac{2}{N} \sum_{i=0}^{N-1} \max(\vec{n}_x \cdot \vec{\omega}_i, \theta) V(x, \vec{\omega}_i)$$



Ambient Occlusion – Estimateur 2

(pdf = direction hémisphérique uniforme)

- Résultats avec la nouvelle *pdf* et le nouveau estimateur:



$N = 10$

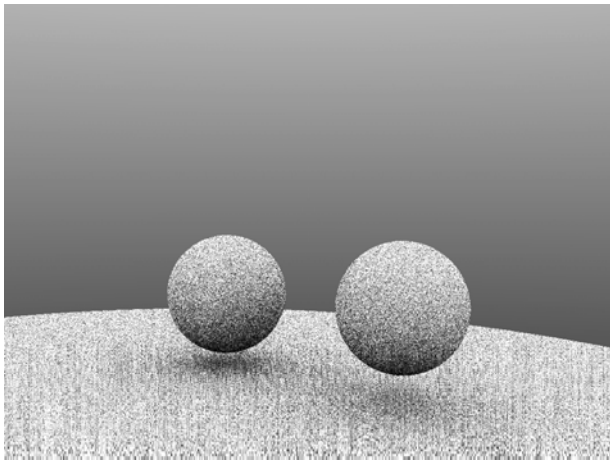
$N = 100$

$N = 1000$

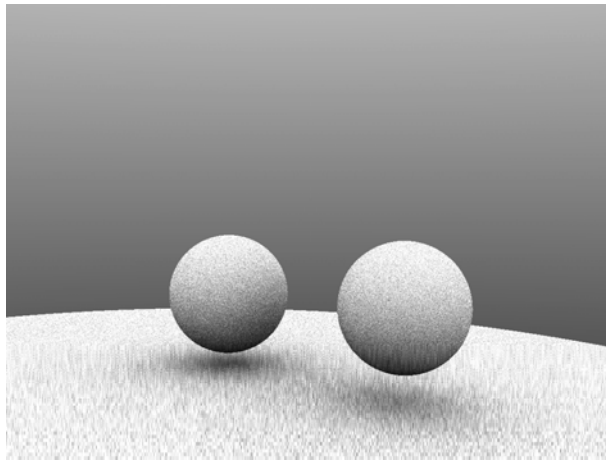
Ambient Occlusion – Estimateur 2

(pdf = direction hémisphérique uniforme)

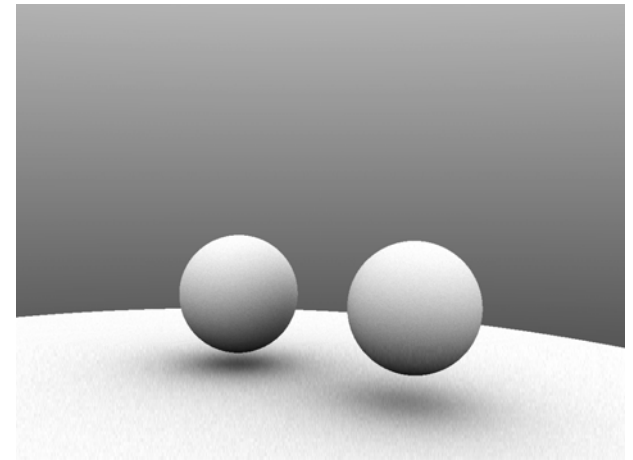
- Résultats avec la nouvelle *pdf* et le nouveau estimateur:



$N = 10$



$N = 100$

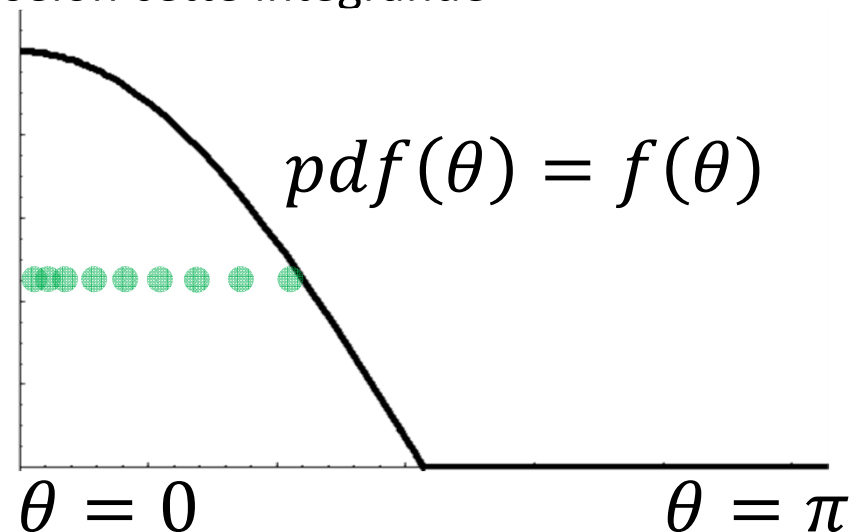
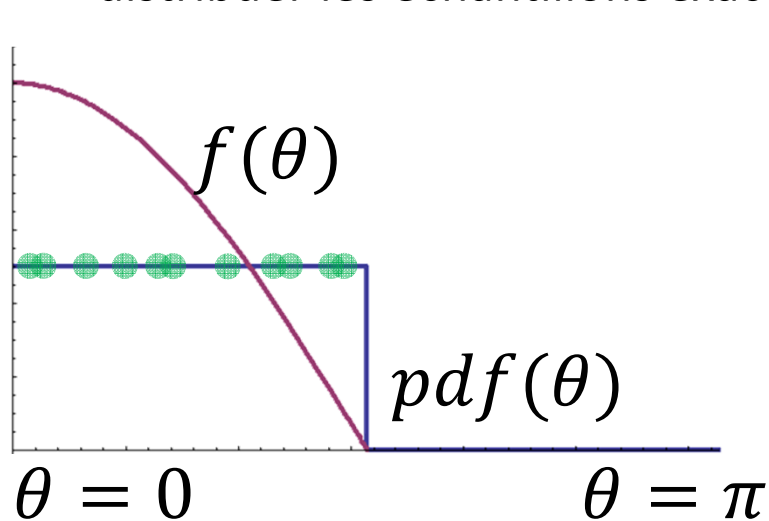


$N = 1000$



Ambient Occlusion: réduire la variance encore plus

- Même les petites modifications qu'on vient d'appliquer dans le deuxième estimateur ont améliorés nos résultats
- On peut encore améliorer (réduire) la variance de notre estimateur:
 - Si on continue d'ignorer la visibilité nous savons exactement la forme de notre intégrande, $f(\omega) = \frac{\max(\vec{n}_x \cdot \vec{\omega}, 0)}{\pi}$, et nous pouvons adapter notre *pdf* pour distribuer les échantillons exactement selon cette intégrande



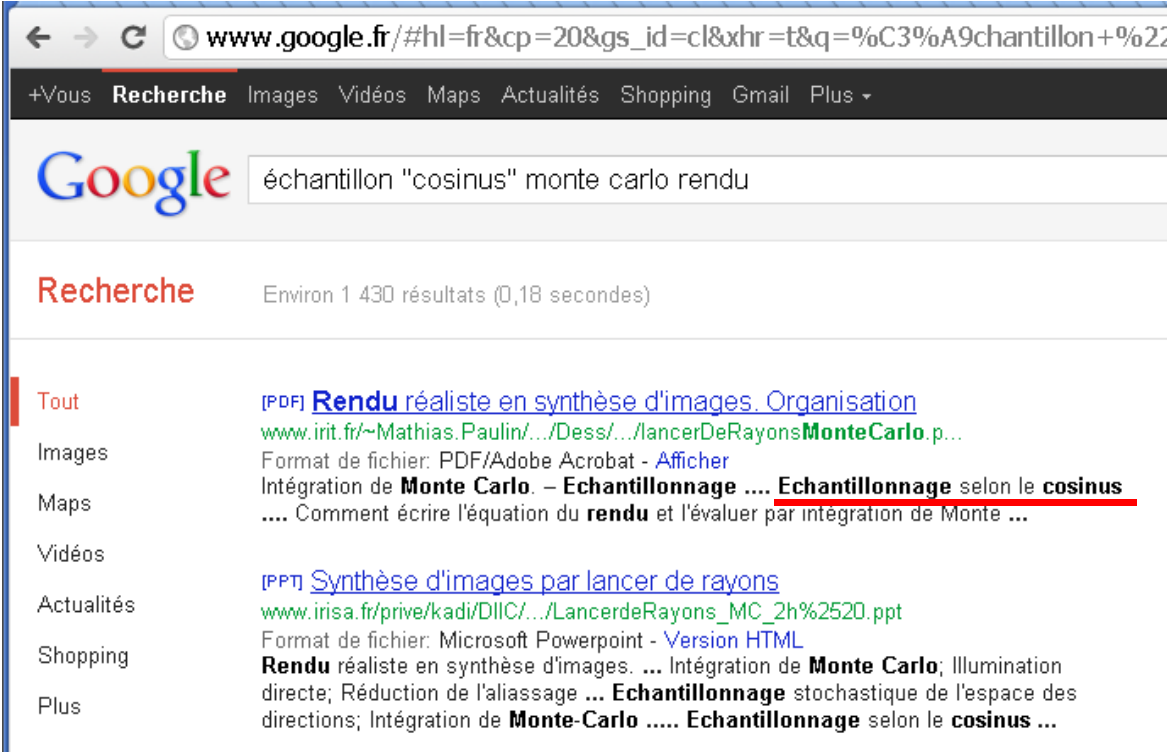
Ambient Occlusion – Estimateur 3

(pdf = distribution hémisphérique + cosinus autour de la normale)

- Donc, on peut suivre les 3 étapes standards:

1. Choisi pdf : $pdf(\vec{\omega}_i) = \frac{\max(\vec{n}_x \cdot \vec{\omega}_i, 0)}{\pi}$

2. Génère les échantillons $\vec{\omega}_i = (\theta_i, \phi_i)$ selon cette distribution:



The screenshot shows a Google search results page. The search query is "échantillon \"cosinus\" monte carlo rendu". The results are filtered by "Tout". The first result is a PDF document titled "Rendu réaliste en synthèse d'images. Organisation" from irit.fr/~Mathias.Paulin/.../Dess/.../lancerDeRayonsMonteCarlo.p... The second result is a Microsoft Powerpoint presentation titled "Synthèse d'images par lancer de rayons" from www.irisa.fr/privé/kadi/DIIC/.../LancerdeRayons_MC_2h%2520.ppt. Both results mention "Echantillonnage selon le cosinus".



Ambient Occlusion – Estimateur 3

(pdf = distribution hémisphérique + cosinus autour de la normale)

- Donc, on peut suivre les 3 étapes standards:

1. Choisi *pdf* : $pdf(\vec{\omega}_i) = \frac{\max(\vec{n}_x \cdot \vec{\omega}_i, 0)}{\pi}$

2. Génère les échantillons $\vec{\omega}_i = (\theta_i, \phi_i)$ selon cette distribution:

$$\theta_i = \text{acos}(\text{sqrt}(\text{rand}()));$$

$$\phi_i = 2 * \text{M_PI} * \text{rand}();$$

effectue une rotation du point autour de \vec{n}_x ;

3. Calcul de notre valeur d'ambient occlusion:

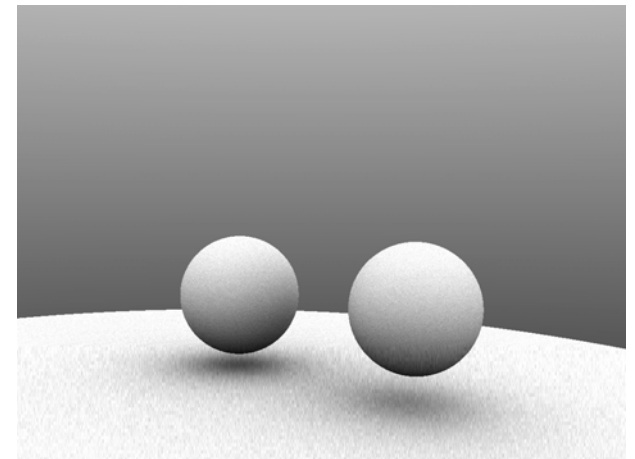
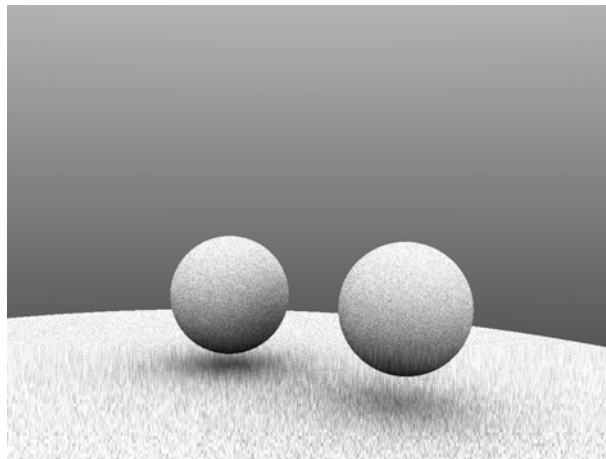
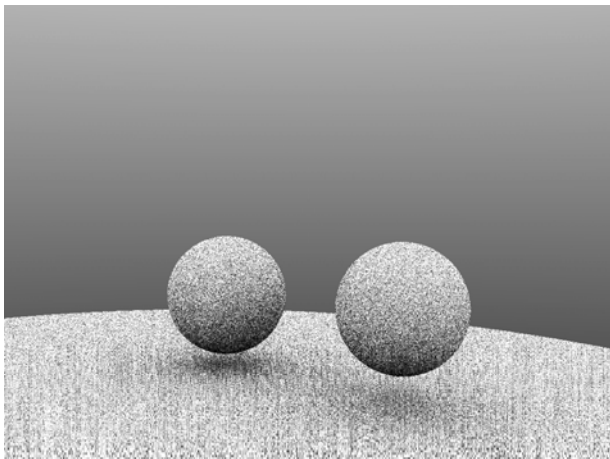
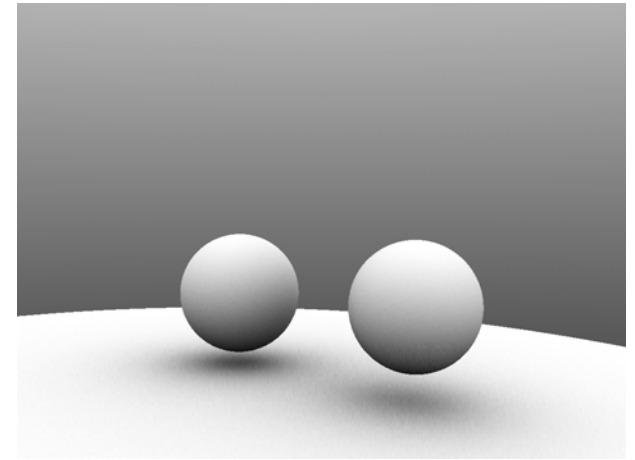
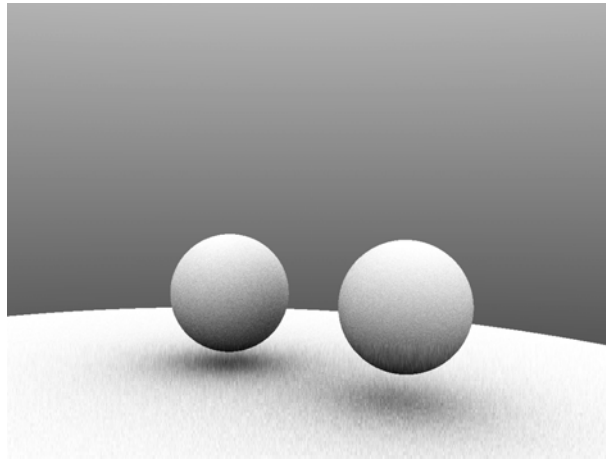
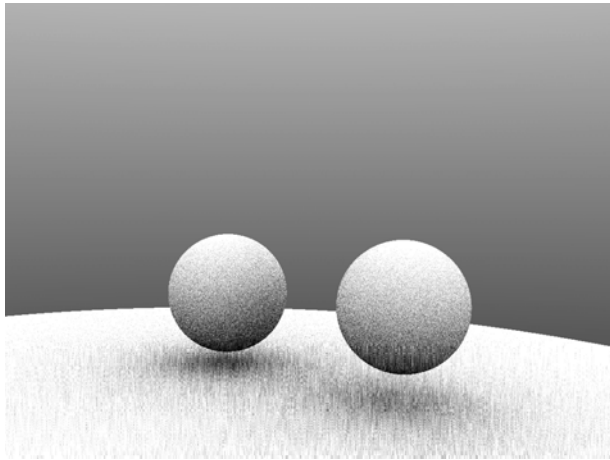
$$\overline{L}_{ao}(x) = \frac{1}{N} \sum_{i=0}^{N-1} \frac{\max(\vec{n}_x \cdot \vec{\omega}_i, 0) V(x, \vec{\omega}_i)}{\pi \left(\frac{\max(\vec{n}_x \cdot \vec{\omega}_i, 0)}{\pi} \right)}$$



Ambient Occlusion – Estimateur 2

(pdf = distribution hémisphérique + cosinus autour de la normale)

- Résultats avec la nouvelle *pdf* et le nouveau estimateur:



$N = 10$

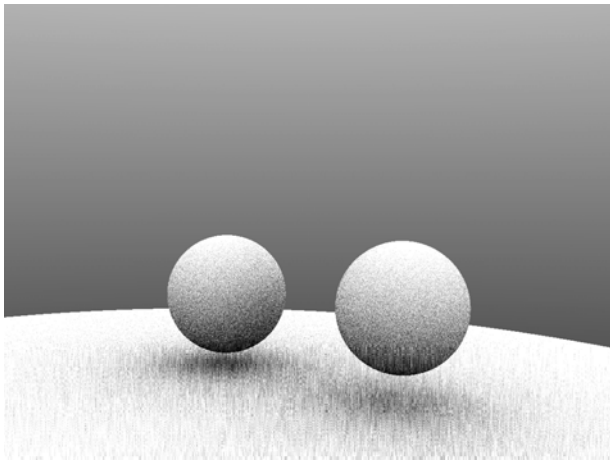
$N = 100$

$N = 1000$

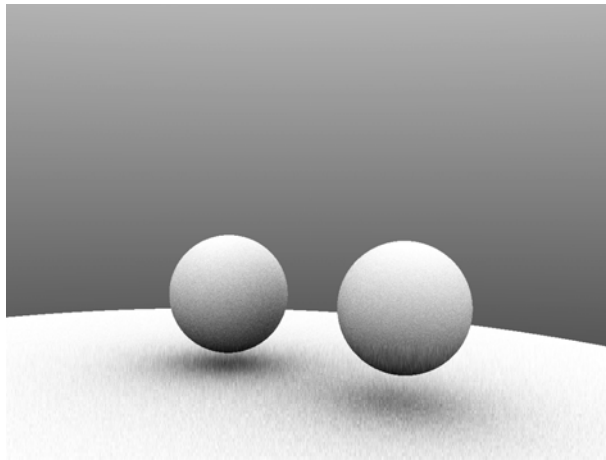
Ambient Occlusion – Estimateur 3

(pdf = distribution hémisphérique + cosinus autour de la normale)

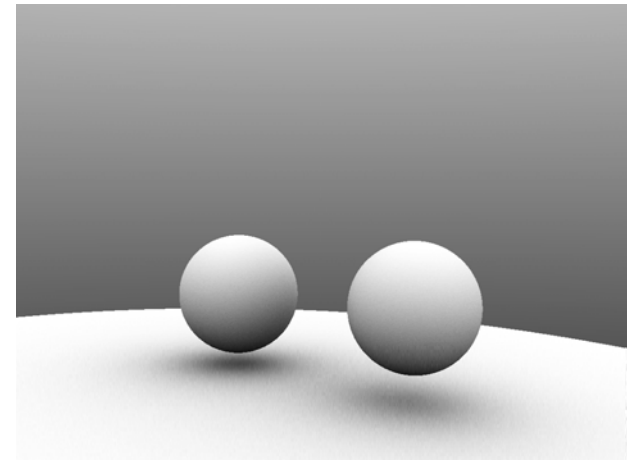
- Résultats avec la nouvelle *pdf* et le nouveau estimateur:



$N = 10$



$N = 100$



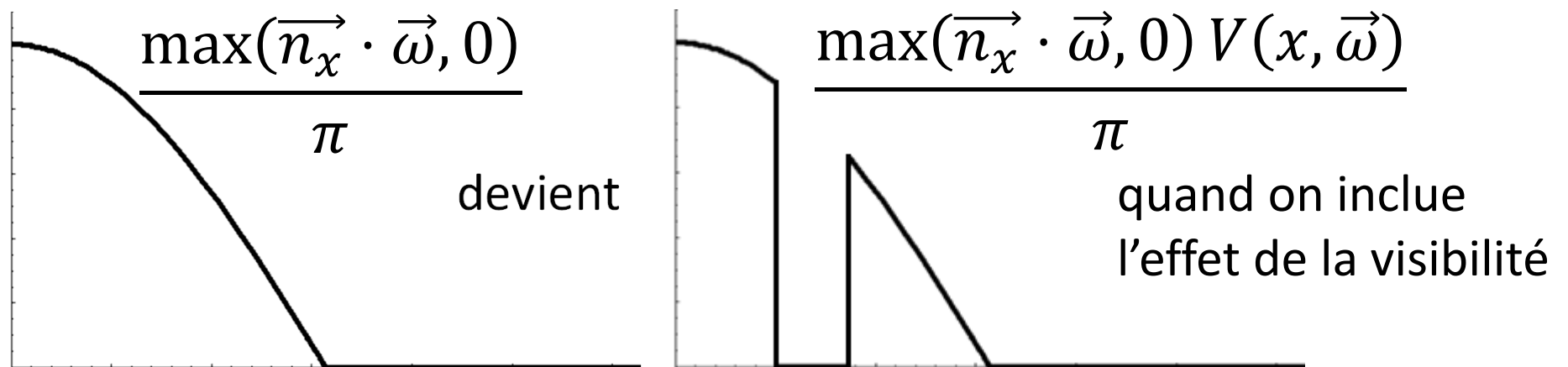
$N = 1000$



Ambient Occlusion:

qu'est-ce qui cause la variance (bruit) résiduelle?

- Si on distribue nos échantillons parfaitement selon le cosinus clampé, pourquoi reste-il du bruit dans nos images?
- C'est à cause de la **visibilité**! N'oublie-pas que nous avons ignoré le terme de visibilité dans notre intégrande. En réalité:

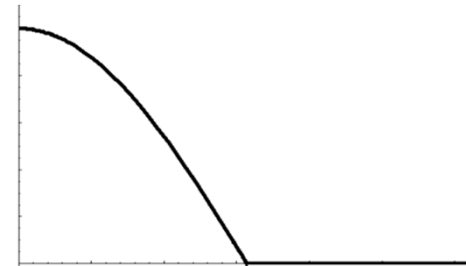
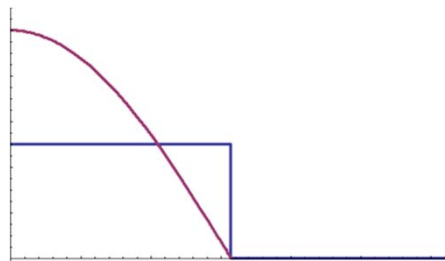
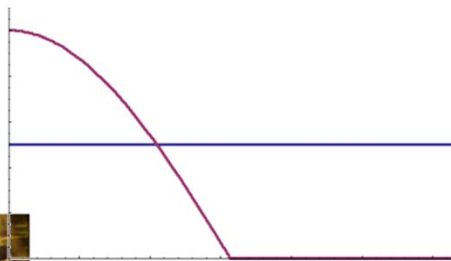
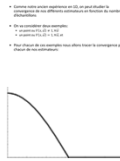


- Si on pourrait déterminer une fonction d'importance sampling *parfaite*, combien d'échantillons est-ce qu'on aura besoin pour résoudre notre intégrale?



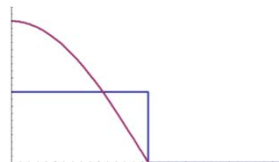
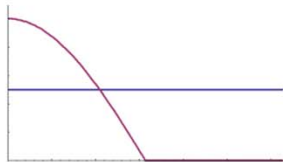
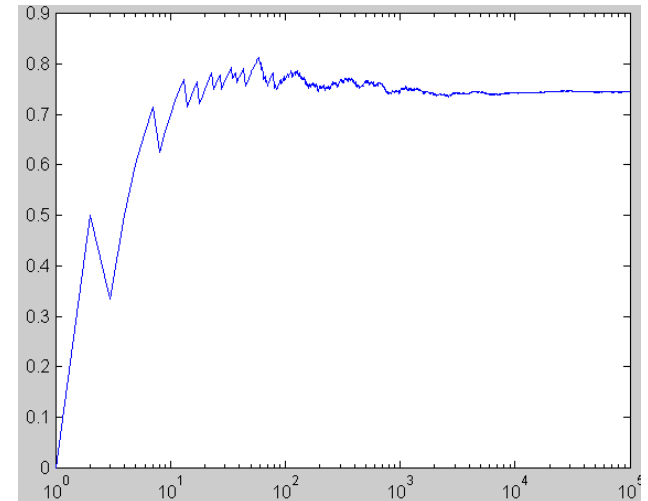
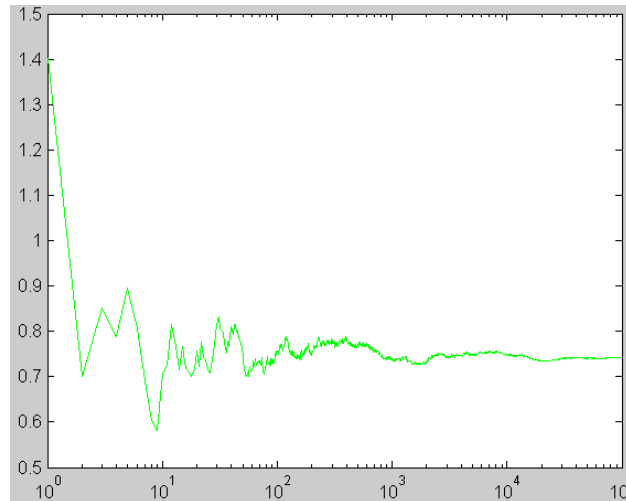
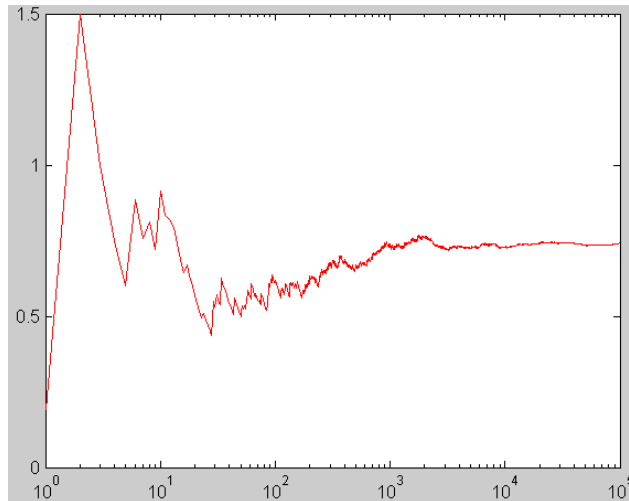
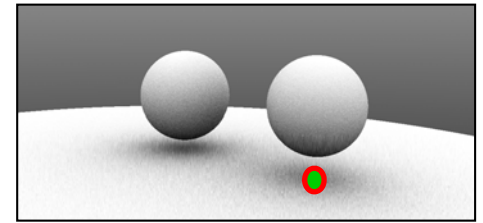
Ambient Occlusion – expérience numérique

- Comme notre ancienne expérience en 1D, on peut étudier la convergence de nos différents estimateurs en fonction du nombre d'échantillons
- On va considérer deux exemples:
 - un point où $V(x, \vec{\omega}) \neq 1, \forall \vec{\omega}$
 - un point où $V(x, \vec{\omega}) = 1, \forall \vec{\omega}$, et
- Pour chacun de ces exemples nous allons tracer la convergence pour chacun de nos estimateurs:



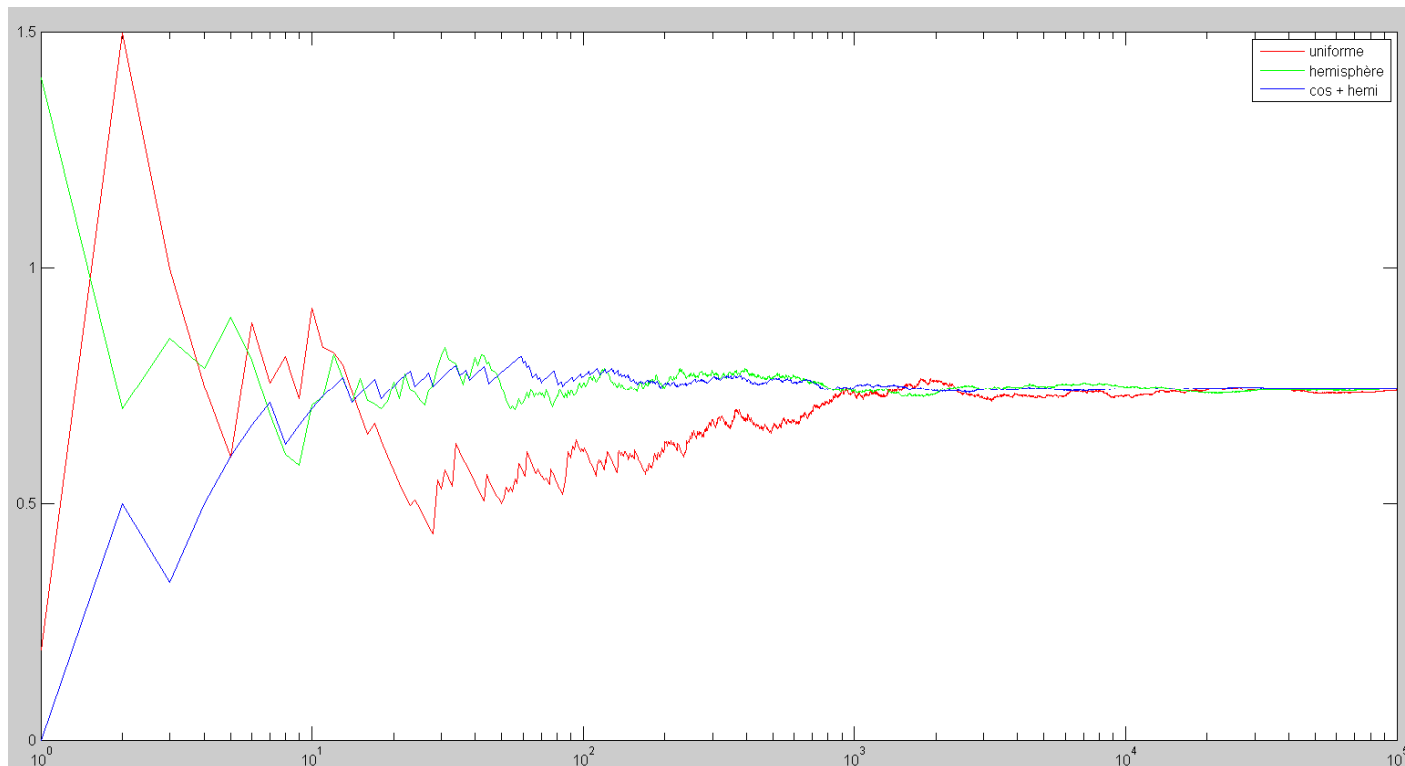
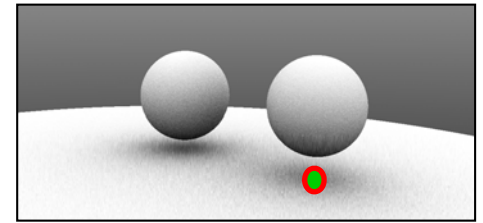
Ambient Occlusion – expérience 1: $V(x, \vec{\omega}) \neq 1$

- Pour cet premier expérience on peut simplement tracer la valeur d'ambient occlusion pour un point dans notre scène



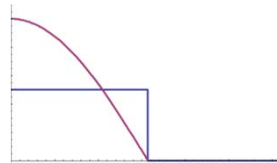
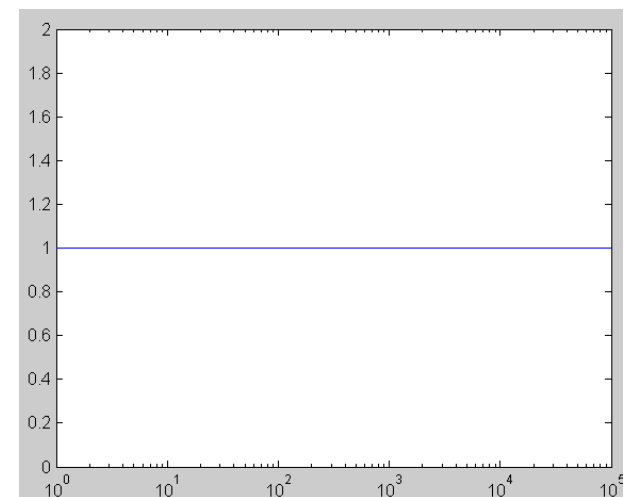
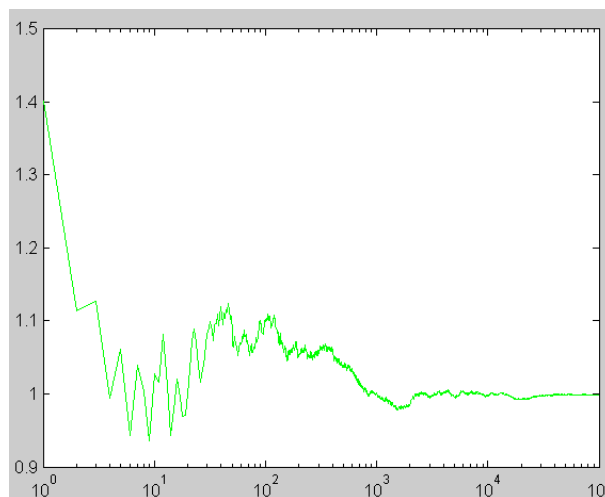
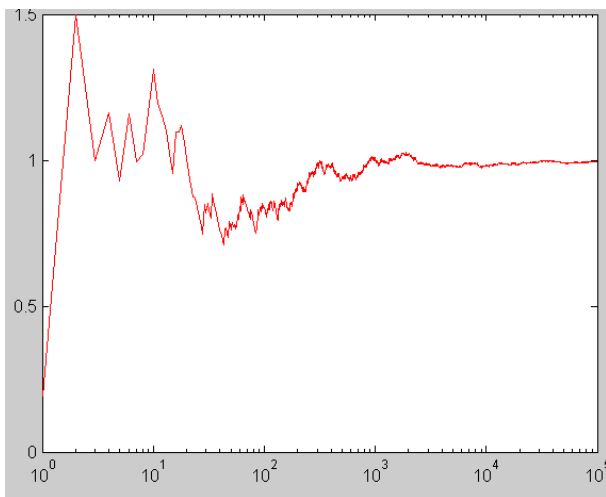
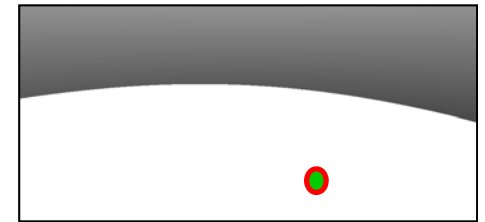
Ambient Occlusion – expérience 1: $V(x, \vec{\omega}) \neq 1$

- Pour cet premier expérience on peut simplement tracer la valeur d'ambient occlusion pour un point dans notre scène



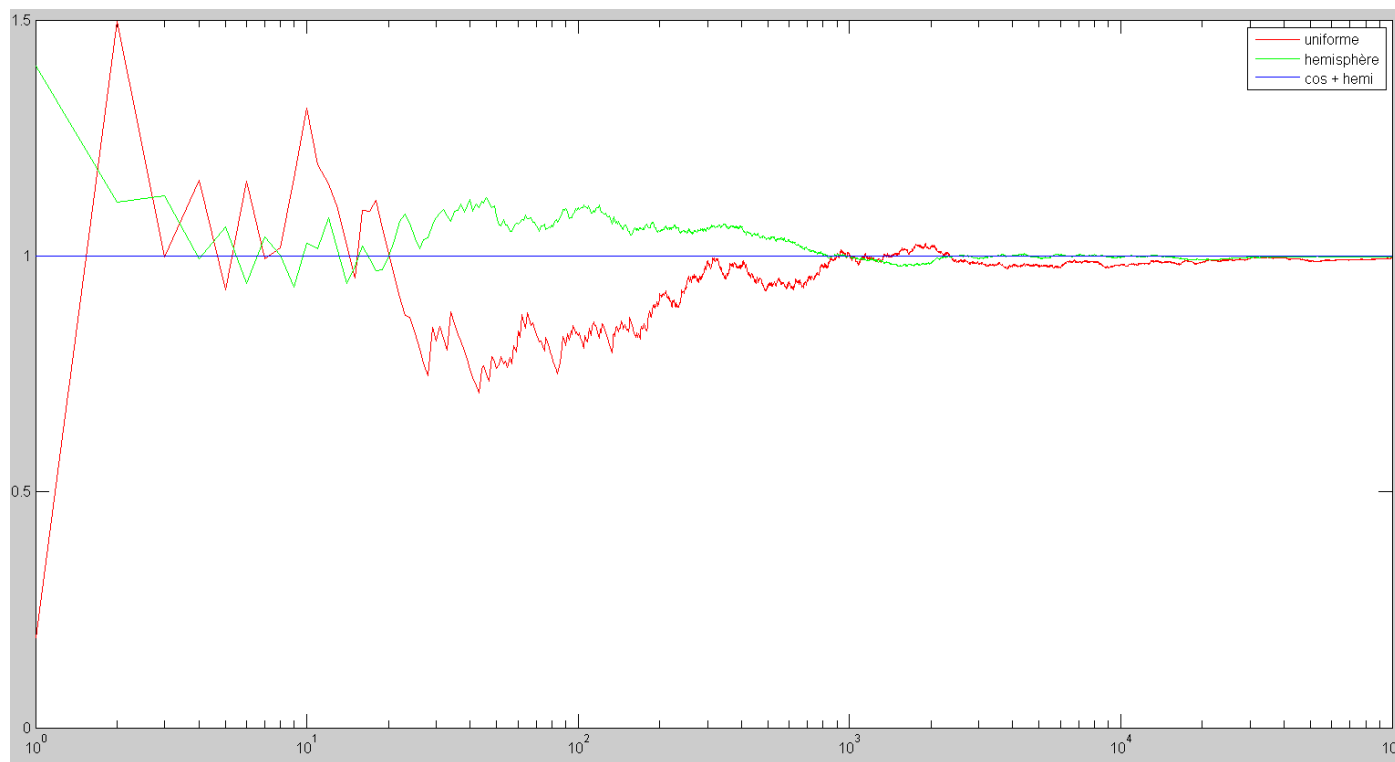
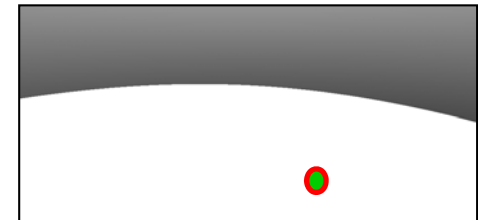
Ambient Occlusion – expérience 2: $v(x, \vec{\omega}) = 1$

- Pour cet premier expérience on peut simplement tracer la valeur d'ambient occlusion pour un point dans une scène simplifiée



Ambient Occlusion – expérience 2: $v(x, \vec{\omega}) = 1$

- Pour cet premier expérience on peut simplement tracer la valeur d'ambient occlusion pour un point dans une scène simplifiée



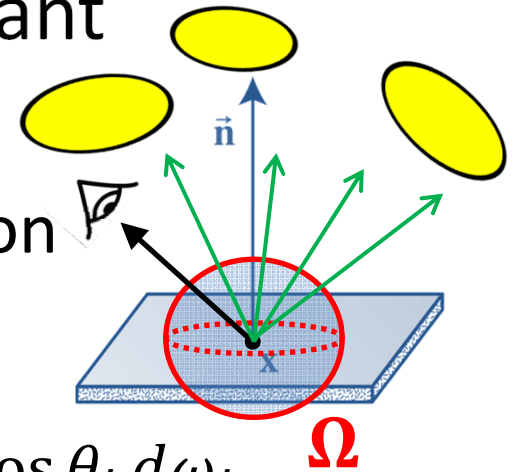
Résoudre l'équation de rendu avec MC



MC pour l'illumination directe

- Commençons rapidement en appliquant MC à l'illumination directe:

– Avec la paramétrisation directionnelle, on tombe sur un cas généralisé de l'AO...



$$L_o(x, \omega_o) = \int_{\Omega} L_e(x, \omega_i) f_r(x, \omega_i, \omega_o) V(x, \omega_i) \cos \theta_i d\omega_i$$

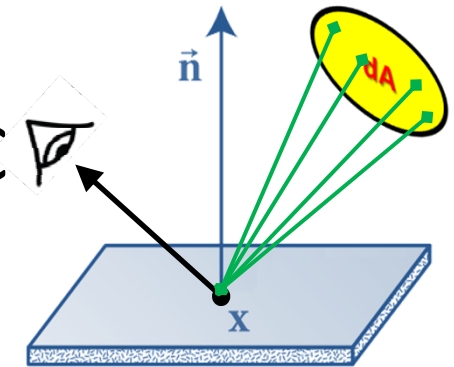
$$L_o(x, \omega_o) \approx \frac{1}{N} \sum_i \frac{L_e(x'_i, \omega_i) f_r(x, \omega_i, \omega_o) V(x, \omega_i) \cos \theta_i}{pdf(\omega_i)}$$

– Comme avec AO, il y a des techniques pour mieux designer la PDF pour réduire la variance



MC pour l'illumination directe

- On peut aussi écrire un estimateur MC dans la paramétrisation surfacique



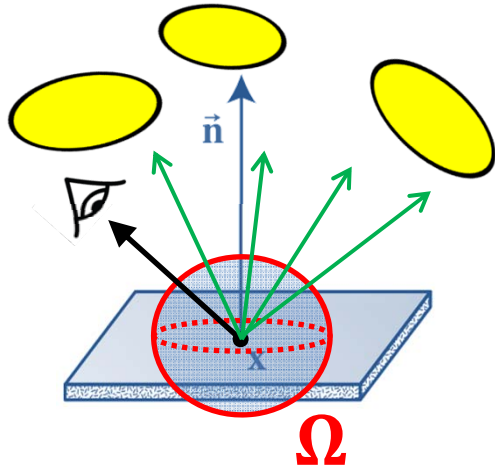
$$L_o(x, \omega_o) = \int_A L_e(x, x_i) f_r(x, x_i, \omega_o) V(x, x_i) \cos \theta_i \frac{\cos \theta_{x \rightarrow i}}{d_{x \rightarrow i}^2} dA(x_i)$$

$$L_o(x, \omega_o) \approx \frac{1}{N} \sum_i \frac{L_e(x'_i, \omega_i) f_r(x, \omega_i, \omega_i) V(x, \omega_i) \cos \theta_i \cos \theta_{x \rightarrow i}}{d_{x \rightarrow i}^2 \text{pdf}(x_i)}$$

- Maintenant notre distribution PDF est selon des points sur la (les) lumière(s)

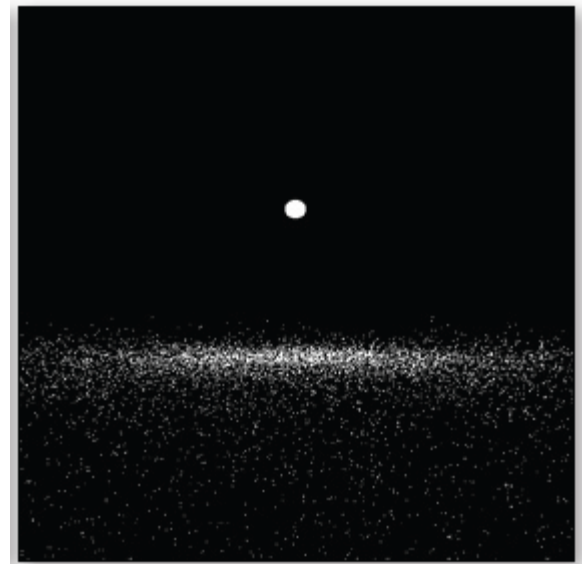
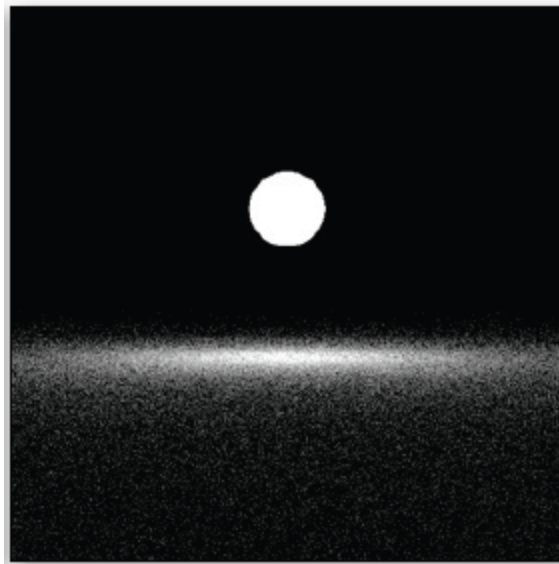
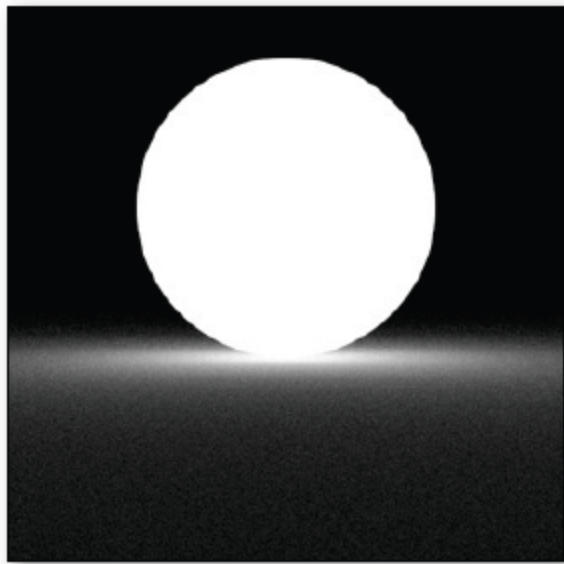


Différences entre paramétrisations pour l'illumination directe avec MC

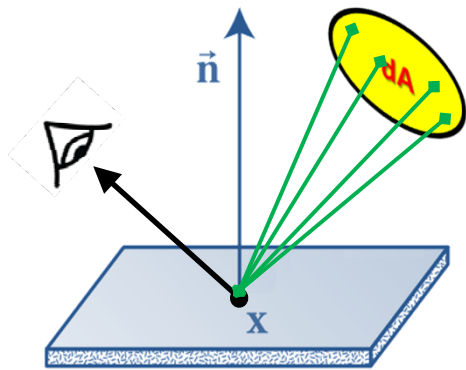


$$L_o(x, \omega_o) = \int_{\Omega} L_e(x, \omega_i) f_r(x, \omega_i, \omega_o) V(x, \omega_i) \cos \theta_i d\omega_i$$

$$L_o(x, \omega_o) \approx \frac{1}{N} \sum_i \frac{L_e(x'_i, \omega_i) f_r(x, \omega_i, \omega_o) V(x, \omega_i) \cos \theta_i}{pdf(\omega_i)}$$

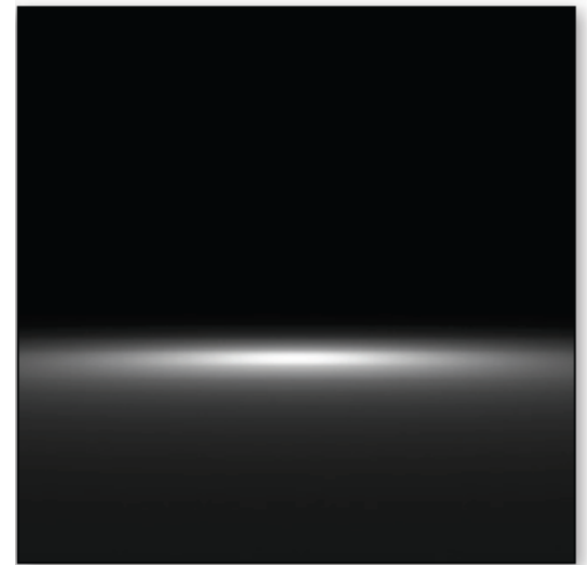
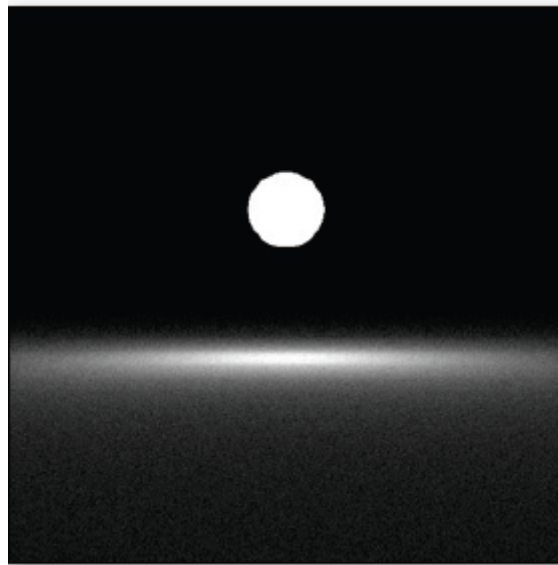
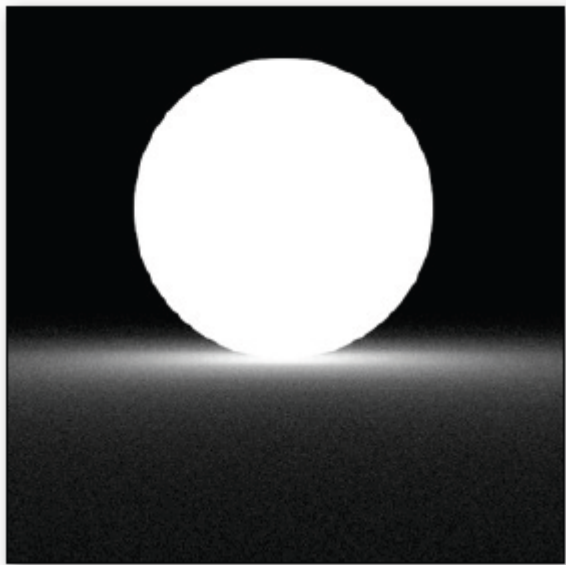


Différences entre paramétrisations pour l'illumination directe avec MC

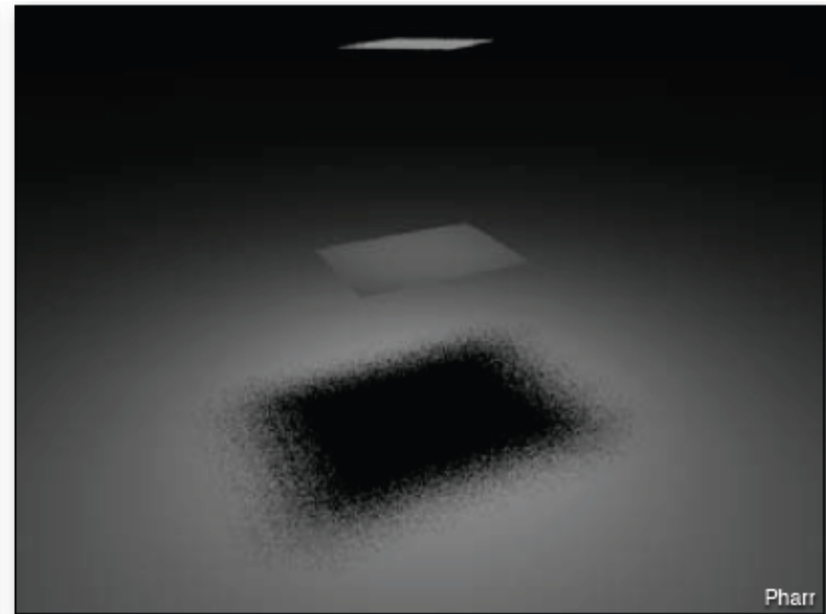
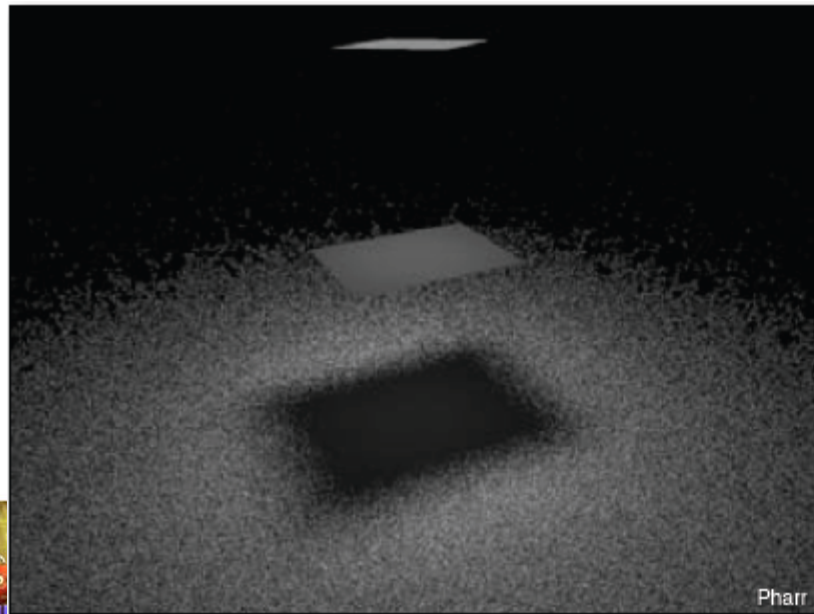
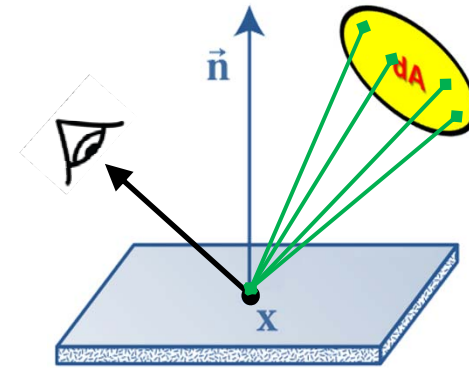
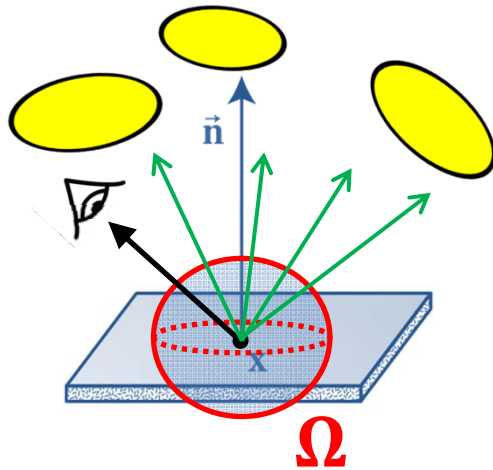


$$L_o = \int_A L_e(x, x_i) f_r(x, x_i, \omega_o) V(x, x_i) \cos \theta_i \frac{\cos \theta_{x \rightarrow i}}{d_{x \rightarrow i}^2} dA(x_i)$$

$$L_o \approx \frac{1}{N} \sum_i \frac{L_e(x, \omega_i) f_r(x, \omega_i, \omega_i) V(x, \omega_i) \cos \theta_i \cos \theta_{x \rightarrow i}}{d_{x \rightarrow i}^2 pdf(x_i)}$$

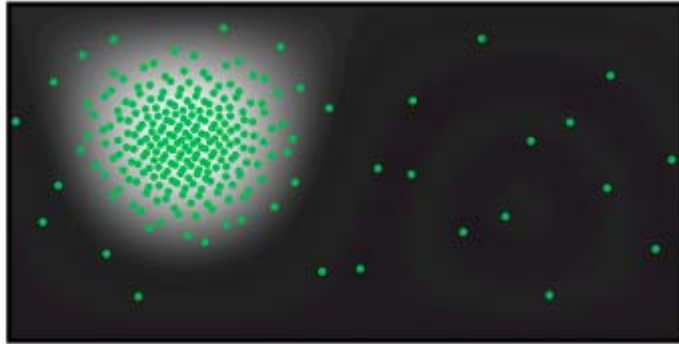


Différences entre paramétrisations pour l'illumination directe avec MC



Pharr

MC pour l'illumination directe



×



$$pdf(\omega_i) \propto f_r(x, \omega_i, \omega_o)$$

$$pdf(\omega_i) \propto L_e(x, \omega_i)$$

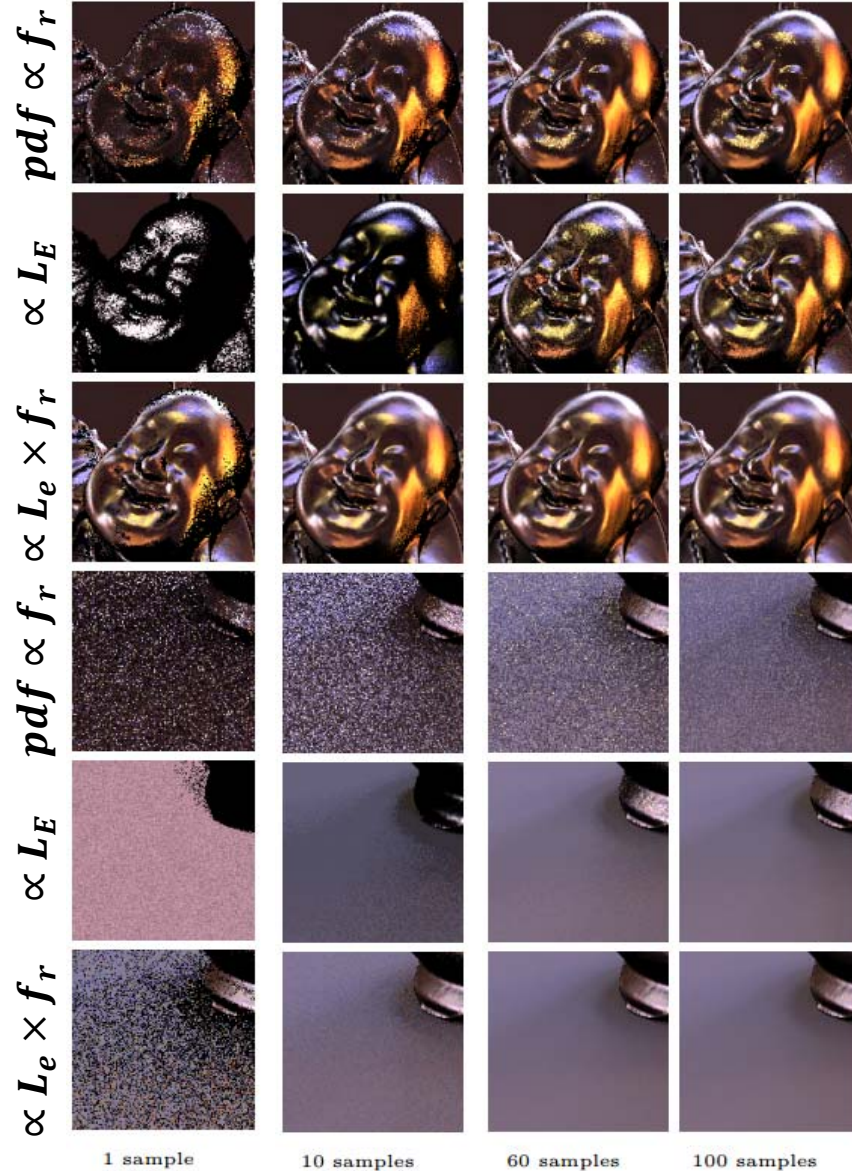
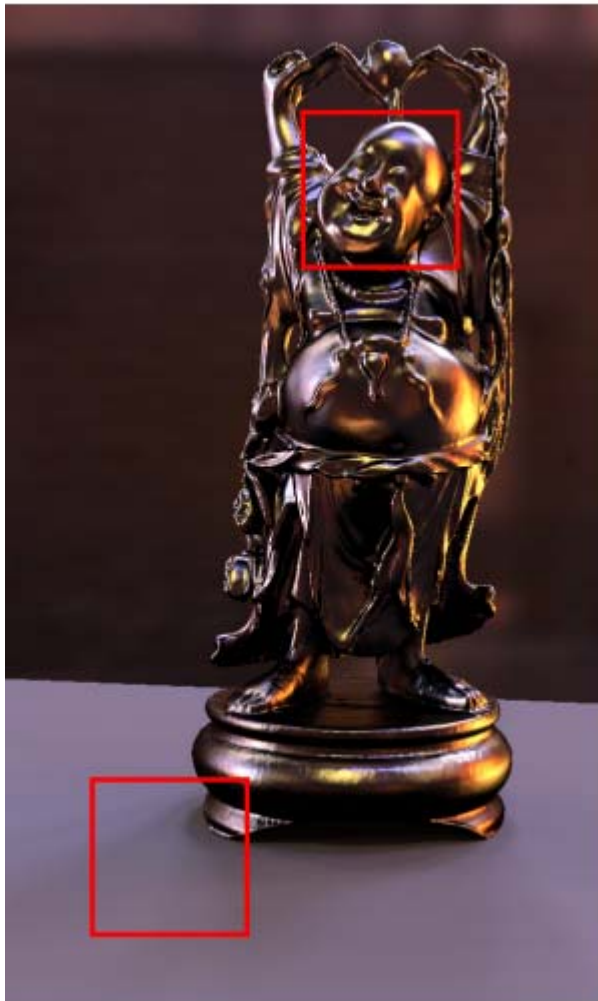
=



$$pdf(\omega_i) \propto L_e(x, \omega_i) \times f_r(x, \omega_i, \omega_o)$$



MC pour l'illumination directe



Monte Carlo pour l'illumination globale



MC pour l'équation de rendu

- Qu'est-ce qui arrive si on applique un estimateur MC à l'équation de rendu?

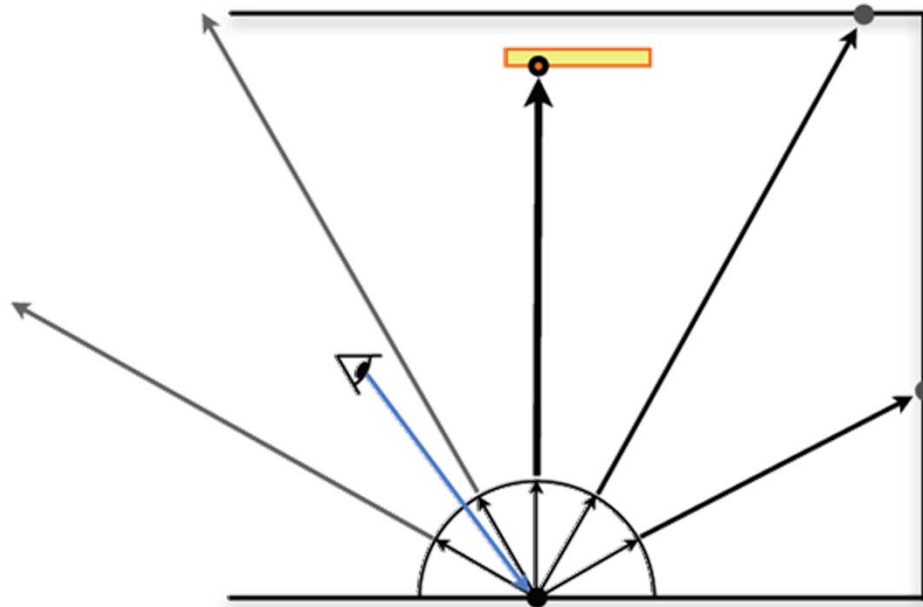
$$L(x, \omega_o) = L_e(x, \omega_o) + \int_{\Omega} L(x', \omega_i) f_r(x, \omega_i, \omega_o) \cos \theta_i d\omega_i$$

$$L(x, \omega_o) \approx L_e(x, \omega_o) + \frac{1}{N} \sum_i \frac{L(x'_i, \omega_i) f_r(x, \omega_i, \omega_o) \cos \theta_i}{pdf(\omega_i)}$$

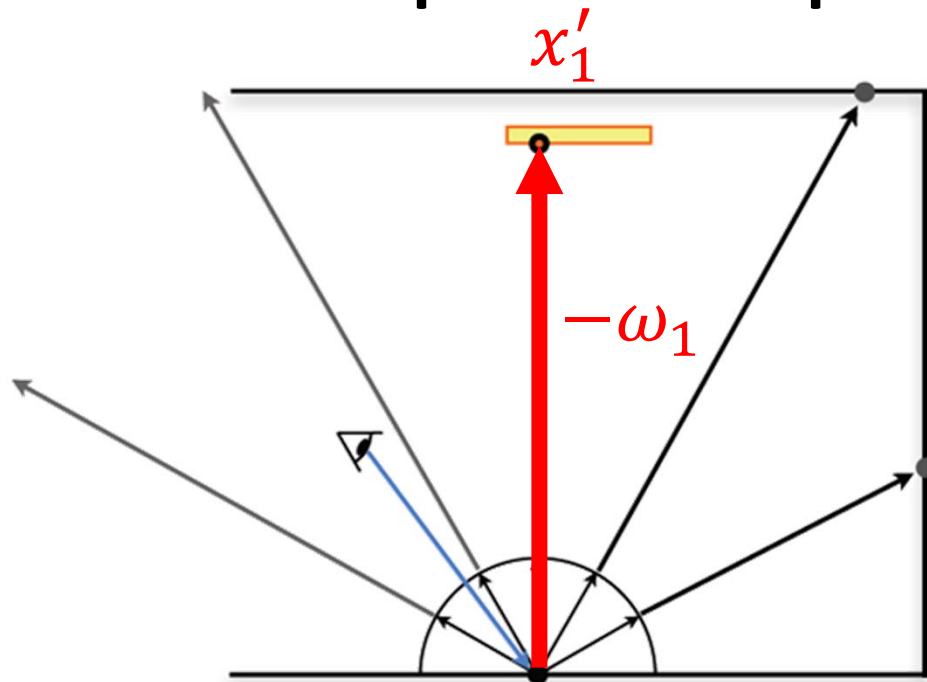


MC pour l'équation de rendu

$$L(x, \omega_o) \approx L_e(x, \omega_o) + \frac{1}{N} \sum_i \frac{L(x'_i, \omega_i) f_r(x, \omega_i, \omega_o) \cos \theta_i}{pdf(\omega_i)}$$



MC pour l'équation de rendu



$$f_r(x'_1) = 0$$

$$L(x'_1, \omega_1) = L_e(x'_1, \omega_1)$$

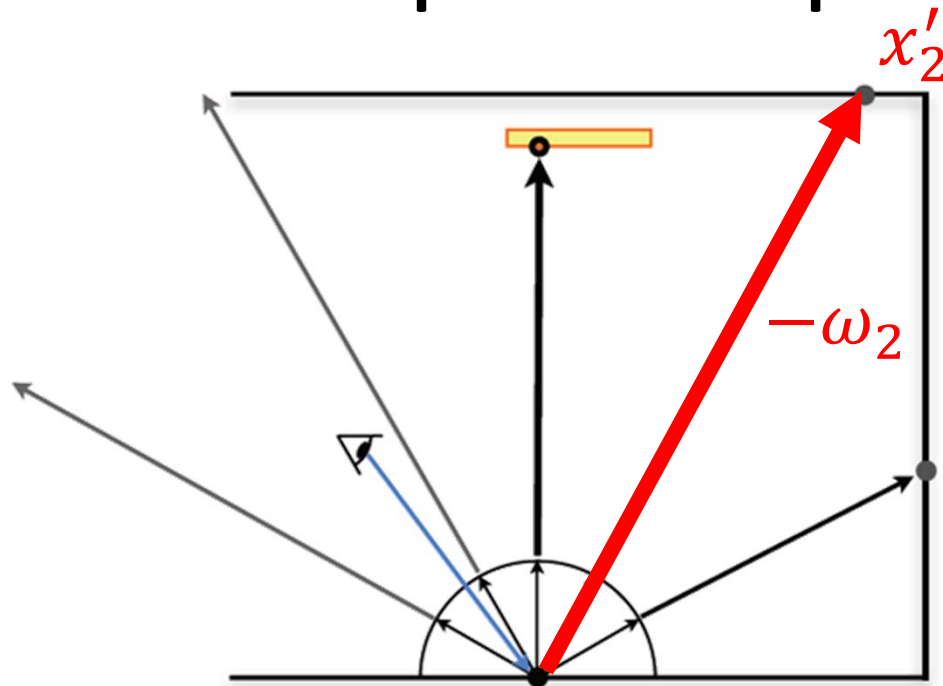
$$L(x'_1, \omega_1) = L_e(x'_1, \omega_1) + \frac{1}{N} \sum_j \frac{L(x'_j, \omega_j) f_r(x'_1, \omega_j, \omega_1) \cos \theta_j}{pdf(\omega_j)}$$

- Quel est la valeur de l'échantillon $L(x'_1, \omega_1)$?

$$L(x, \omega_o) = L_e(x, \omega_o) + \frac{1}{N} \sum_i \frac{L(x'_i, \omega_i) f_r(x, \omega_i, \omega_o) \cos \theta_i}{pdf(\omega_i)}$$



MC pour l'équation de rendu



$$f_r(x'_2) \neq 0$$

$$L_e(x'_2, \omega_2) = 0$$

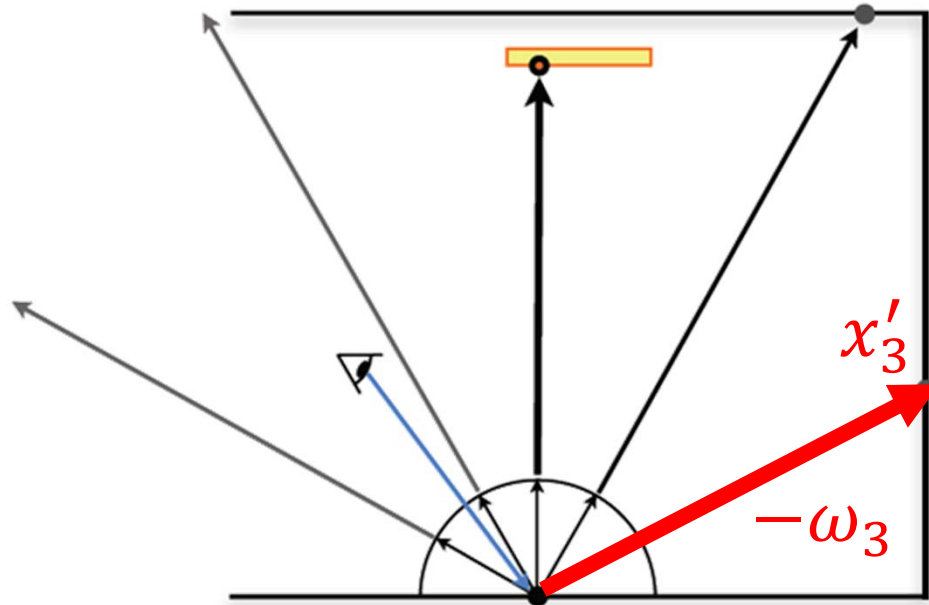
$$L(x'_2, \omega_2) = L_e(x'_2, \omega_2) + \frac{1}{N} \sum_k \frac{L(x'_k, \omega_k) f_r(x'_2, \omega_k, \omega_2) \cos \theta_k}{pdf(\omega_k)}$$

- Quel est la valeur de l'échantillon $L(x'_2, \omega_2)$?

$$L(x, \omega_o) = L_e(x, \omega_o) + \frac{1}{N} \sum_i \frac{L(x'_i, \omega_i) f_r(x, \omega_i, \omega_o) \cos \theta_i}{pdf(\omega_i)}$$



MC pour l'équation de rendu



$$f_r(x'_2) \neq 0$$

$$L_e(x'_2, \omega_2) = 0$$

$$f_r(x'_3) \neq 0$$

$$L_e(x'_3, \omega_3) = 0$$

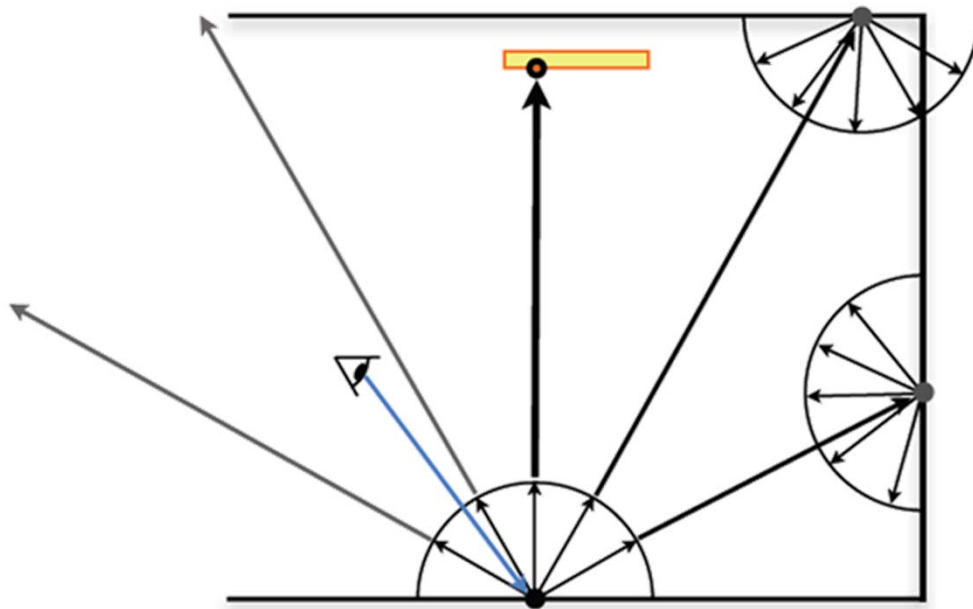
$$L(x'_3, \omega_3) = L_e(x'_3, \omega_3) + \frac{1}{N} \sum_l \frac{L(x'_l, \omega_l) f_r(x'_3, \omega_l, \omega_3) \cos \theta_l}{pdf(\omega_l)}$$

- Quel est la valeur de l'échantillon $L(x'_3, \omega_3)$?

$$L(x, \omega_o) = L_e(x, \omega_o) + \frac{1}{N} \sum_i \frac{L(x'_i, \omega_i) f_r(x, \omega_i, \omega_o) \cos \theta_i}{pdf(\omega_i)}$$



MC pour l'équation de rendu



$$L(x'_2, \omega_2) = \frac{1}{N} \sum_k \frac{L(x_k''', \omega_k) f_r(x'_2, \omega_k, \omega_2) \cos \theta_k}{pdf(\omega_k)}$$

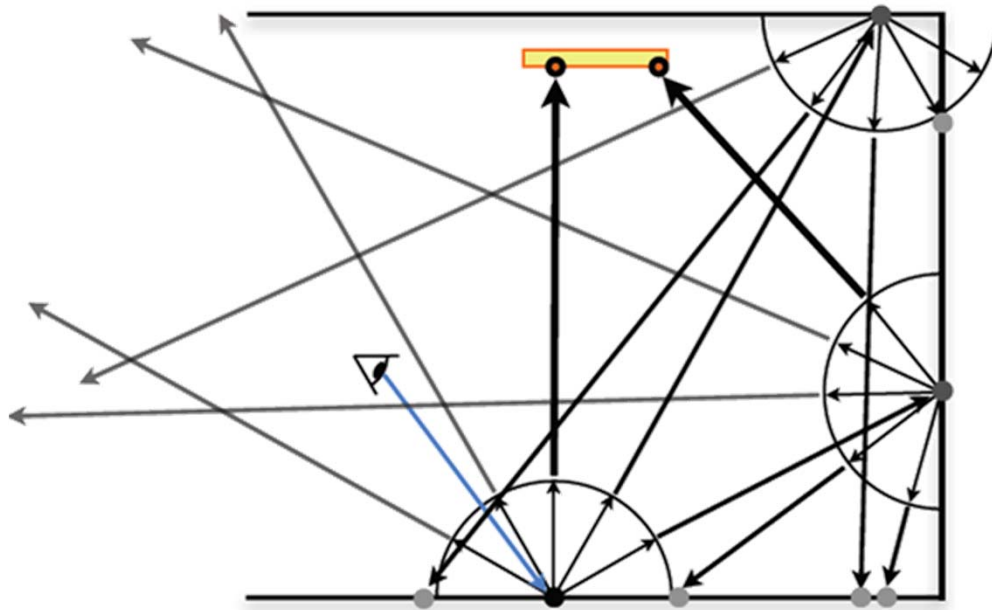
$$L(x'_3, \omega_3) = \frac{1}{N} \sum_l \frac{L(x_l''''', \omega_3) f_r(x'_3, \omega_l, \omega_3) \cos \theta_l}{pdf(\omega_l)}$$

- On peut procéder récursivement...

$$L(x, \omega_o) = L_e(x, \omega_o) + \frac{1}{N} \sum_i \frac{L(x'_i, \omega_i) f_r(x, \omega_i, \omega_o) \cos \theta_i}{pdf(\omega_i)}$$



MC pour l'équation de rendu



```
couleur shade( point x )
if( x.Le != 0 ) return x.Le;
couleur L = 0;
for( chaque échantillon i ) {
    omega_i = direction aléatoire
              distribué selon pdf;
    L += shade( trace(x, omega_i) )
          * brdf * cos / pdf(omega_i);
}
return L /= N;
```

- Quand est-ce la récursion se termine*?
- Autre problème?
- **Plus** de calcul pour les effets qui contribuent **moins** à l'image finale!



Complexité exponentielle

```
couleur shade( point x )
  if( x.Le != 0 ) return x.Le;
  couleur L = 0;
  for( chaque échantillon i ) {
    omega_i = direction aléatoire distribué selon pdf;
    L += shade( trace(x, omega_i) ) * brdf * cos / pdf(omega_i);
  }
  return L /= N;
```

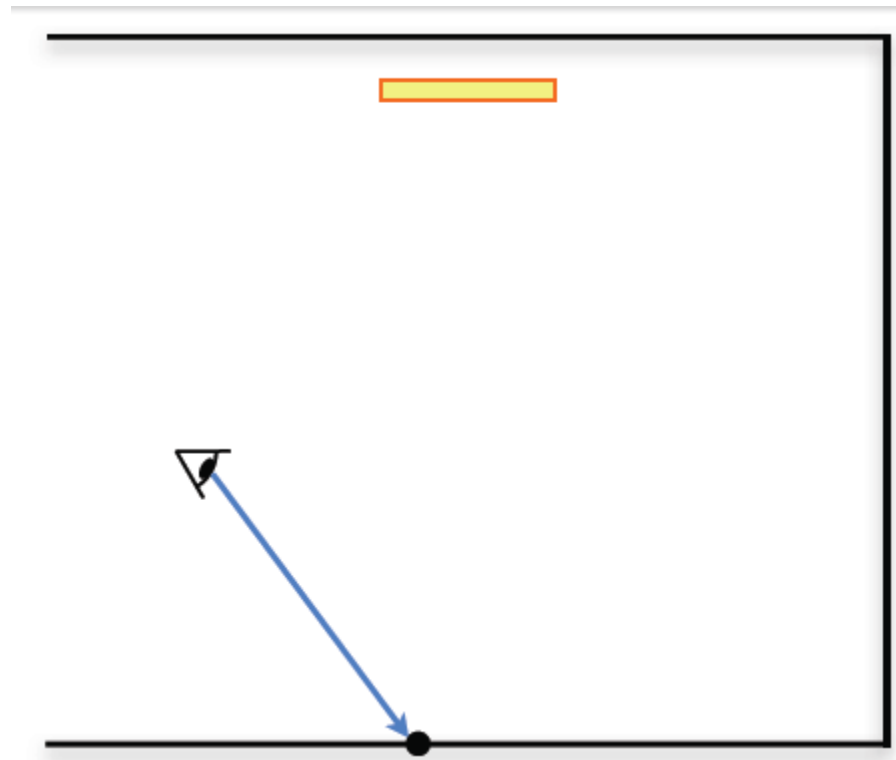
- Comment pouvons-nous éliminer l'explosion exponentielle de calcul?
- Utiliser un estimateur MC avec un seul échantillon!

$$L(x, \omega_o) \approx L_e(x, \omega_o) + \frac{L(x', \omega) f_r(x, \omega, \omega_o) \cos \theta}{pdf(\omega)}$$



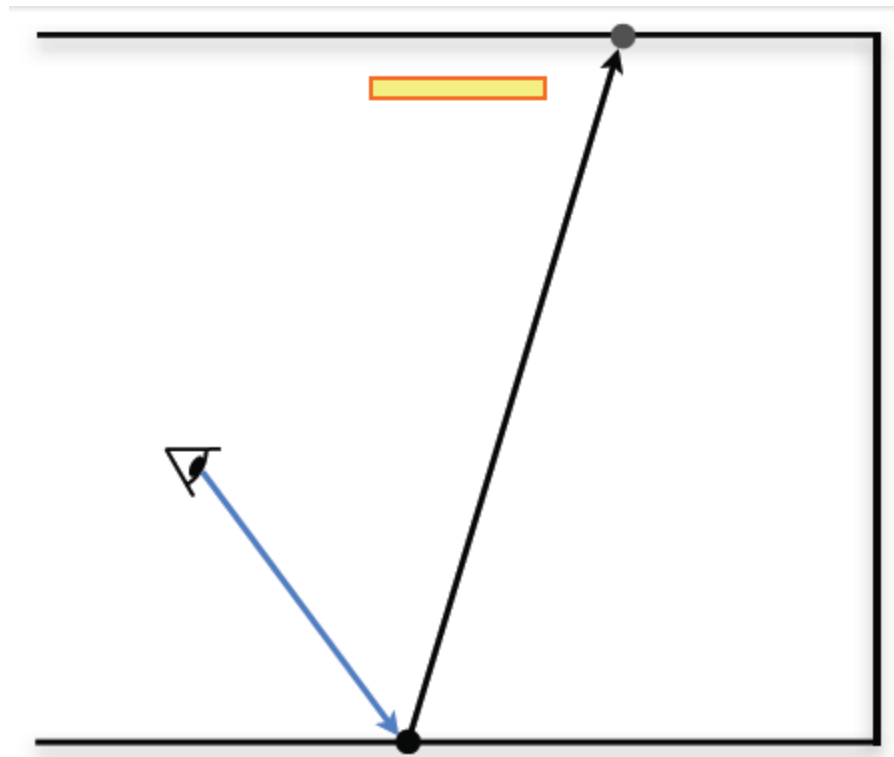
L'algorithme de *path-tracing*

$$L(x, \omega_o) \approx L_e(x, \omega_o) + \frac{L(x', \omega) f_r(x, \omega, \omega_o) \cos \theta}{pdf(\omega)}$$



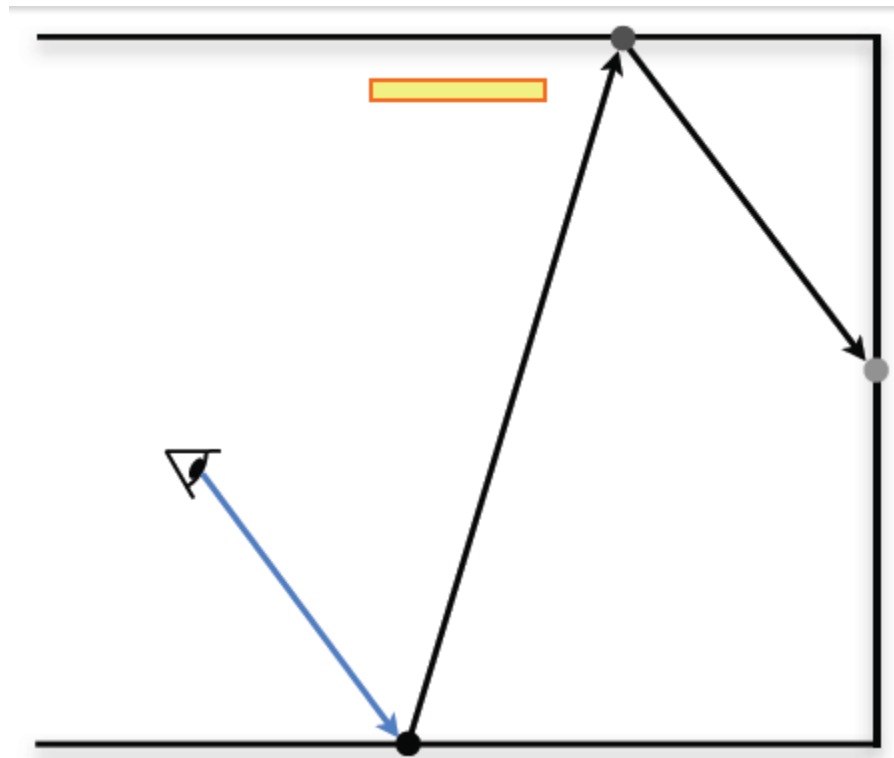
L'algorithme de *path-tracing*

$$L(x, \omega_o) \approx L_e(x, \omega_o) + \frac{L(x', \omega) f_r(x, \omega, \omega_o) \cos \theta}{pdf(\omega)}$$



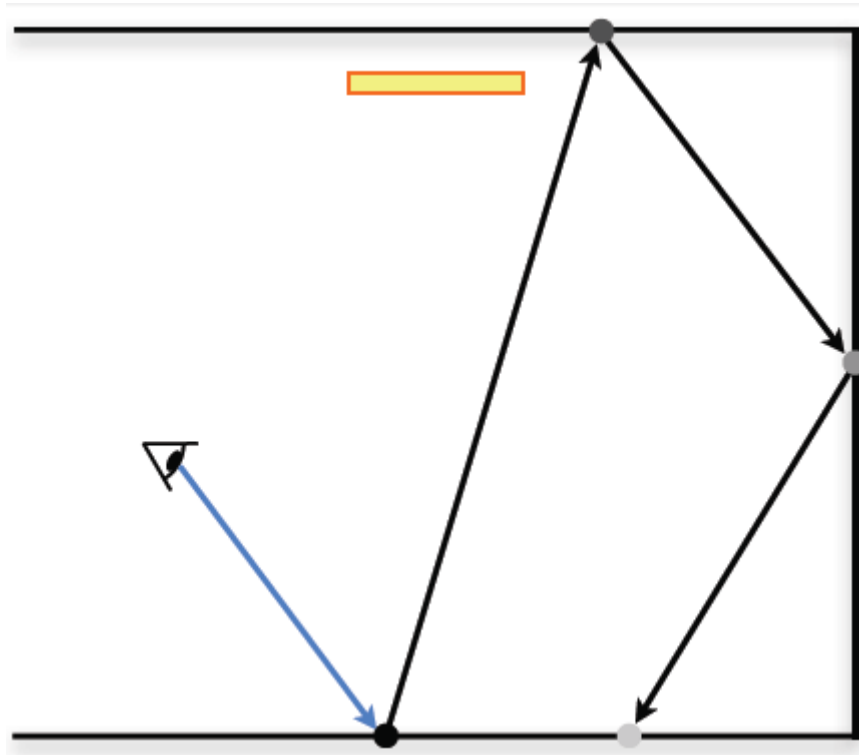
L'algorithme de *path-tracing*

$$L(x, \omega_o) \approx L_e(x, \omega_o) + \frac{L(x', \omega) f_r(x, \omega, \omega_o) \cos \theta}{pdf(\omega)}$$



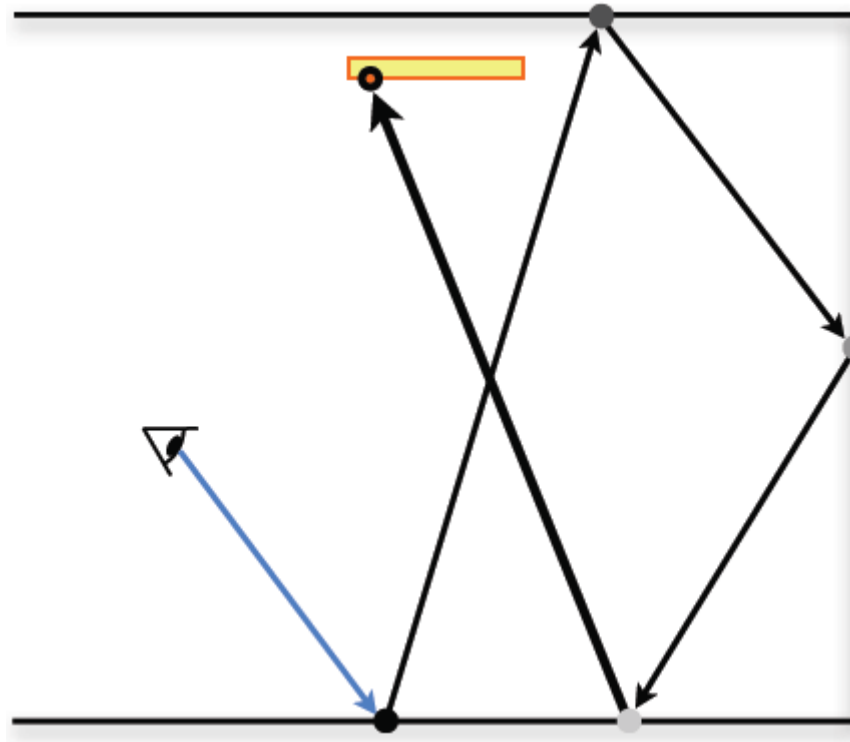
L'algorithme de *path-tracing*

$$L(x, \omega_o) \approx L_e(x, \omega_o) + \frac{L(x', \omega) f_r(x, \omega, \omega_o) \cos \theta}{pdf(\omega)}$$



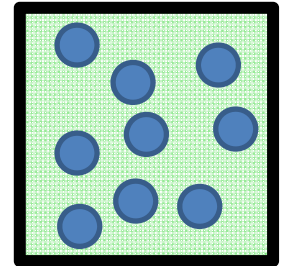
L'algorithme de *path-tracing*

$$L(x, \omega_o) \approx L_e(x, \omega_o) + \frac{L(x', \omega) f_r(x, \omega, \omega_o) \cos \theta}{pdf(\omega)}$$



Implicit path-tracing

```
couleur shade( point x )
  if( x.Le != 0 ) return x.Le;
  couleur L = 0;
  omega_i = direction aléatoire distribué selon pdf;
  L += shade( trace(x, omega_i) ) * brdf * cos / pdf(omega_i);
  return L;
```

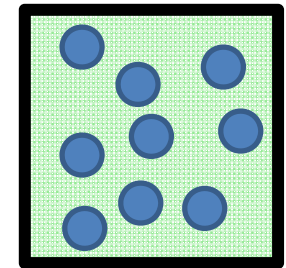
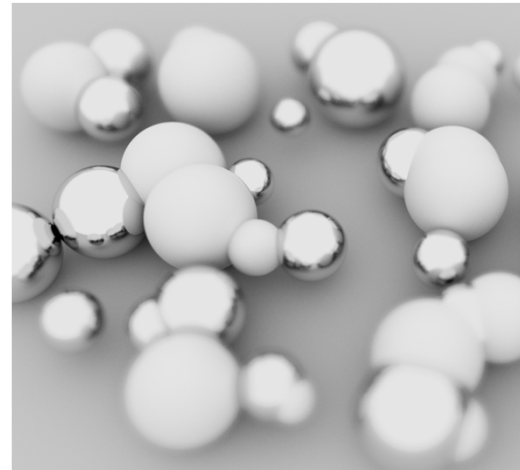
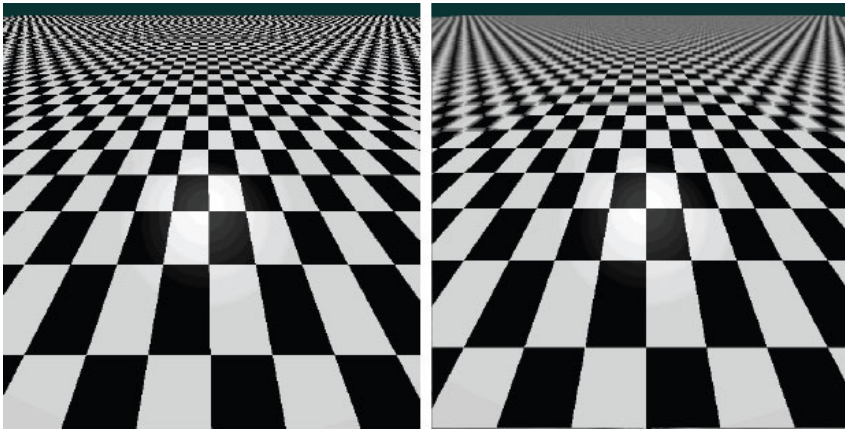


- Pour mieux approximer l'intégrale, on a seulement besoin de lancer/générer plus de *chemins*
 - Envoyez plusieurs rayons à travers chaque pixel, chacun crée un nouveau chemin qui contribue (potentiellement) à l'intensité finale du pixel
- On peut gagner des effets d'anti-aliasage, de flou de mouvement, et/ou de profondeur de champs



“gratuitement” en échantillonnant le carée du pixel

Effets de distribution

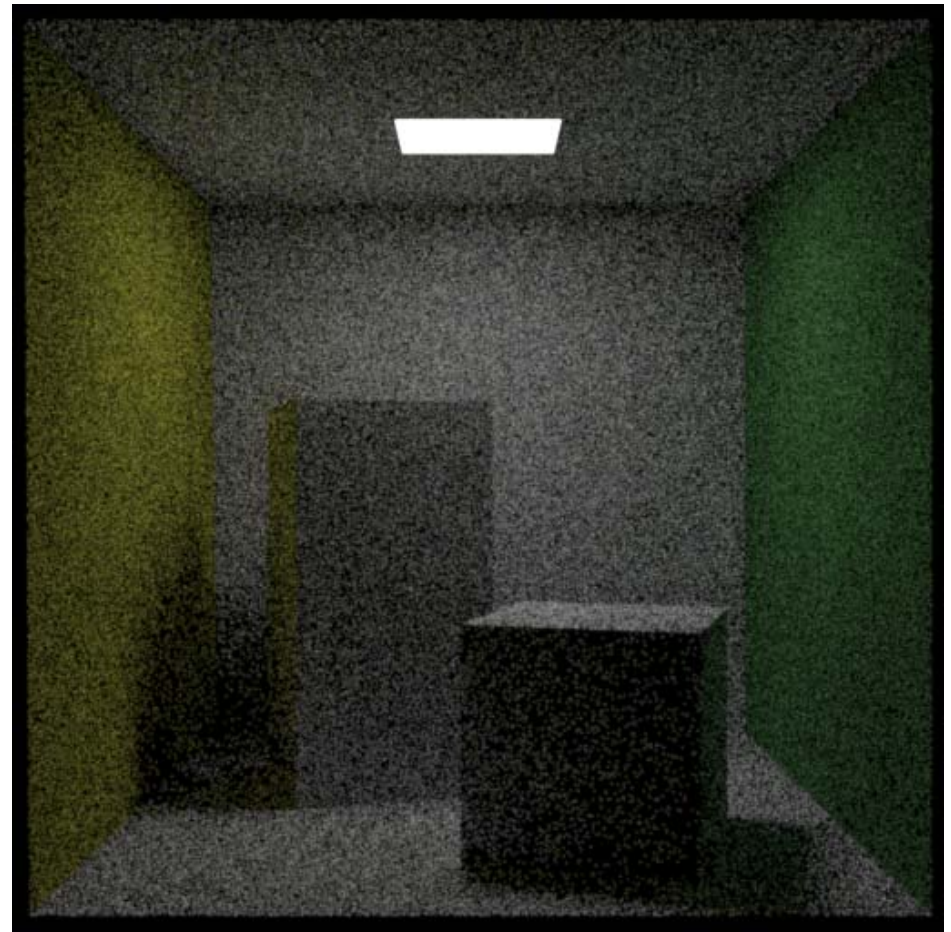


Implicit path-tracing

$$L(x, \omega_o) \approx \frac{1}{N} \sum_N \left[L_e(x, \omega_o) + \frac{L(x', \omega) f_r(x, \omega, \omega_o) \cos \theta}{pdf(\omega)} \right]$$

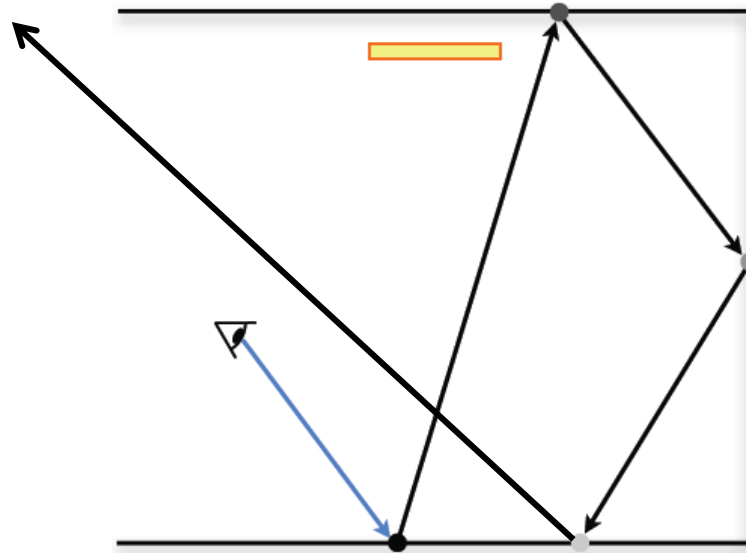
```
couleur shade( point x )  
  if( x.Le != 0 ) return x.Le;  
  couleur L = 0;  
  omega_i = direction aléatoire  
            distribué selon pdf;  
  L += shade( trace(x, omega_i) )  
        * brdf * cos / pdf(omega_i);  
  return L;
```

```
...  
couleur c = 0;  
for(chaque échantillon du pixel i)  
  c += shade(trace(œil, i))  
c /= N;  
...
```



Implicit path-tracing: Résultat bruité

- Pourquoi tellement de bruit?
- Récursion termine (avec une valeur non-zéro) seulement* si le (dernier sommet du) chemin frappe la lumière!
- Beaucoup de chemins gaspillés!



Segmentation de l'intégrale

- On va réduire le bruit en séparant les contributions d'illumination directe et indirecte

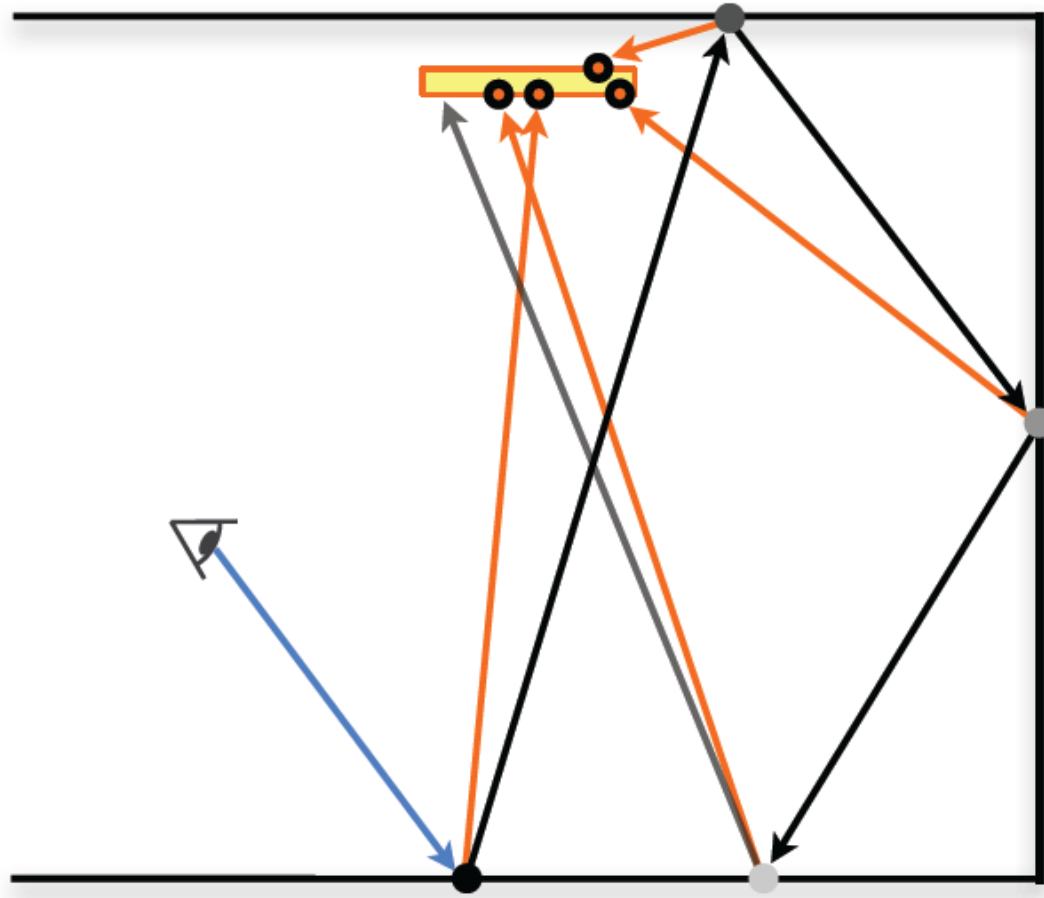
$$L \approx L_e(x, \omega_o) + L_d(x, \omega_o) + L_{ind}(x, \omega_o)$$

- Ici on va utiliser des estimateurs MC avec 1 échantillon pour chaque sous-intégrale
 - l'illumination directe sera *explicitement* échantillonné
 - On doit éviter le double comptage d'illumination

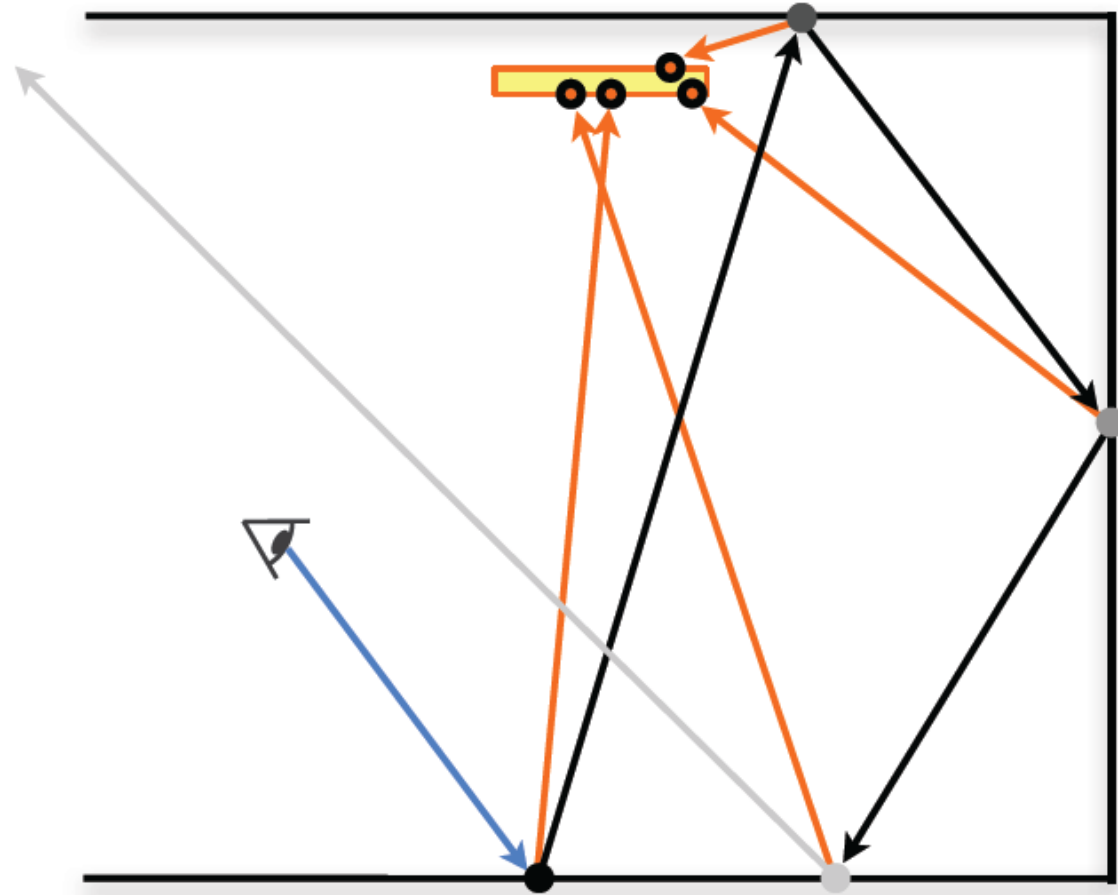
$$\approx L_e + \frac{L_e(x_d, \omega_d) f_r(x, \omega_i, \omega_d) V(x, \omega_d) \cos \theta_d \cos \theta_{x \rightarrow d}}{d_{x \rightarrow d}^2 pdf(x_d)} + \frac{L(x', \omega) f_r(x, \omega, \omega_o) \cos \theta}{pdf(\omega)}$$



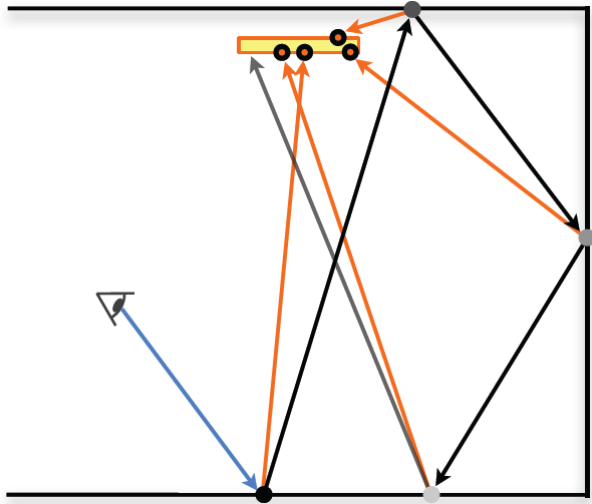
Explicit path-tracing



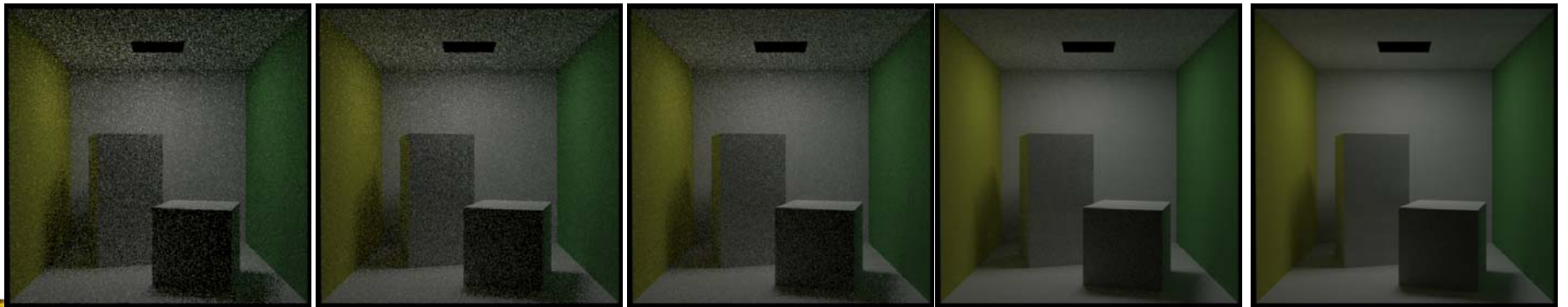
Explicit path-tracing



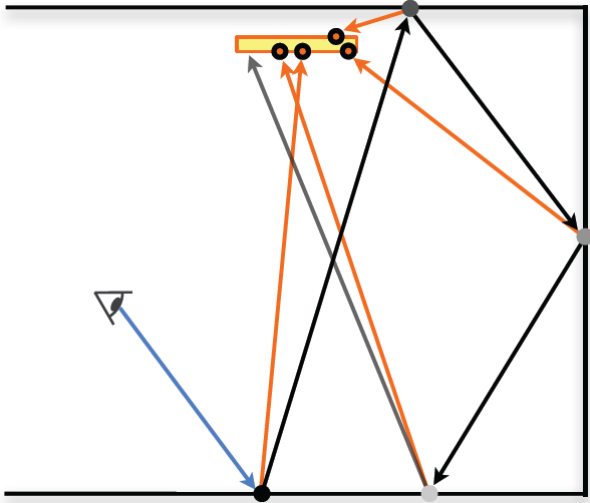
Explicit path-tracing: échantillonnage de l'illum directe



```
couleur shade( point x )  
    couleur L = x.Le;  
    point p = point aléatoire sur la lumière;  
    omega_i = dir aléatoire distribué selon pdf;  
    // L_d  
    if( trace(x, p) )  
        L += fr * cos * p.Le * cos' / (|x-p|^2 * pdf_x)  
    // L_ind  
    L += shade(trace(x,omega_i)) * fr' * cos' / pdf;  
    return L;
```



Réursion infini



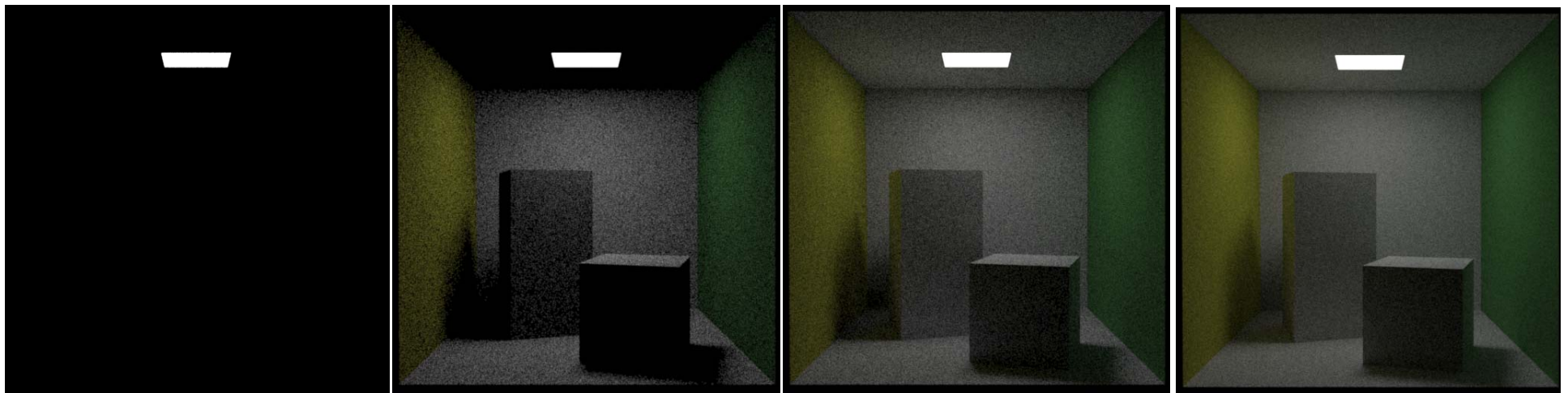
```
couleur shade( point x )
couleur L = x.Le;
point p = point aléatoire sur la lumière;
omega_i = dir aléatoire distribué selon pdf;
// L_d
if( trace(x, p) )
    L += fr * cos * p.Le * cos' / (|x-p|^2 * pdf_x)
// L_ind
L += shade(trace(x,omega_i)) * fr' * cos' / pdf;
return L;
```

Encore une récursion (potentiellement) infini!



Comment terminer la récursion?

- Quand/comment terminer la récursion?
 - Clamper avec une profondeur maximale?
 - Biaisée



p=1

p=2

p=3

p=4



- Solution non-biaisée?

Roulette Russe

- Roulette Russe: un approche probabiliste pour terminer un processus MC sans changer sa valeur d'expectation / moyenne (ie: sans introduire du biais dans le résultat)
1. Choisissez un seuil / probabilité de termination χ
 2. Générez une valeur aléatoire ε entre (0,1]
 3. Retournez une valeur $V' = \frac{V}{1-\chi}$ quand $\varepsilon \leq \chi$ ou 0 sinon

$$E[V'] = (1 - \chi) \left[\frac{E[V]}{1 - \chi} \right] = E[V]$$



Roulette Russe

- Comment choisir la probabilité de termination χ ?
 - Peu importe notre choix, l'estimateur sera "correcte"
 - Mais, comme avec le choix de notre PDF, nous pouvons réduire la variance en choisissant un seuil intelligemment
- Ex: pour f_r diffus, un choix typique est $\chi = \rho_x$
 - Interpretation: plus que ρ_x est grand, plus grand la valeur de reflection d'une surface, plus qu'on veut promouvoir des évènements de réflexion



Explicit Path Tracing w/ Russian Roulette

- Alors, notre algorithme finale de *path tracing* est:

```
couleur shade( point x )
// L_d
if( trace(x, p) )
    L += fr * cos * p.Le * cos' / (|x-p|^2 * pdf_x)

// L_ind
if( rand() < X )
    omega_i = dir aléatoire distribué selon pdf;
    L += shade(trace(x,omega_i)) * fr' * cos' / (pdf * (1-X));

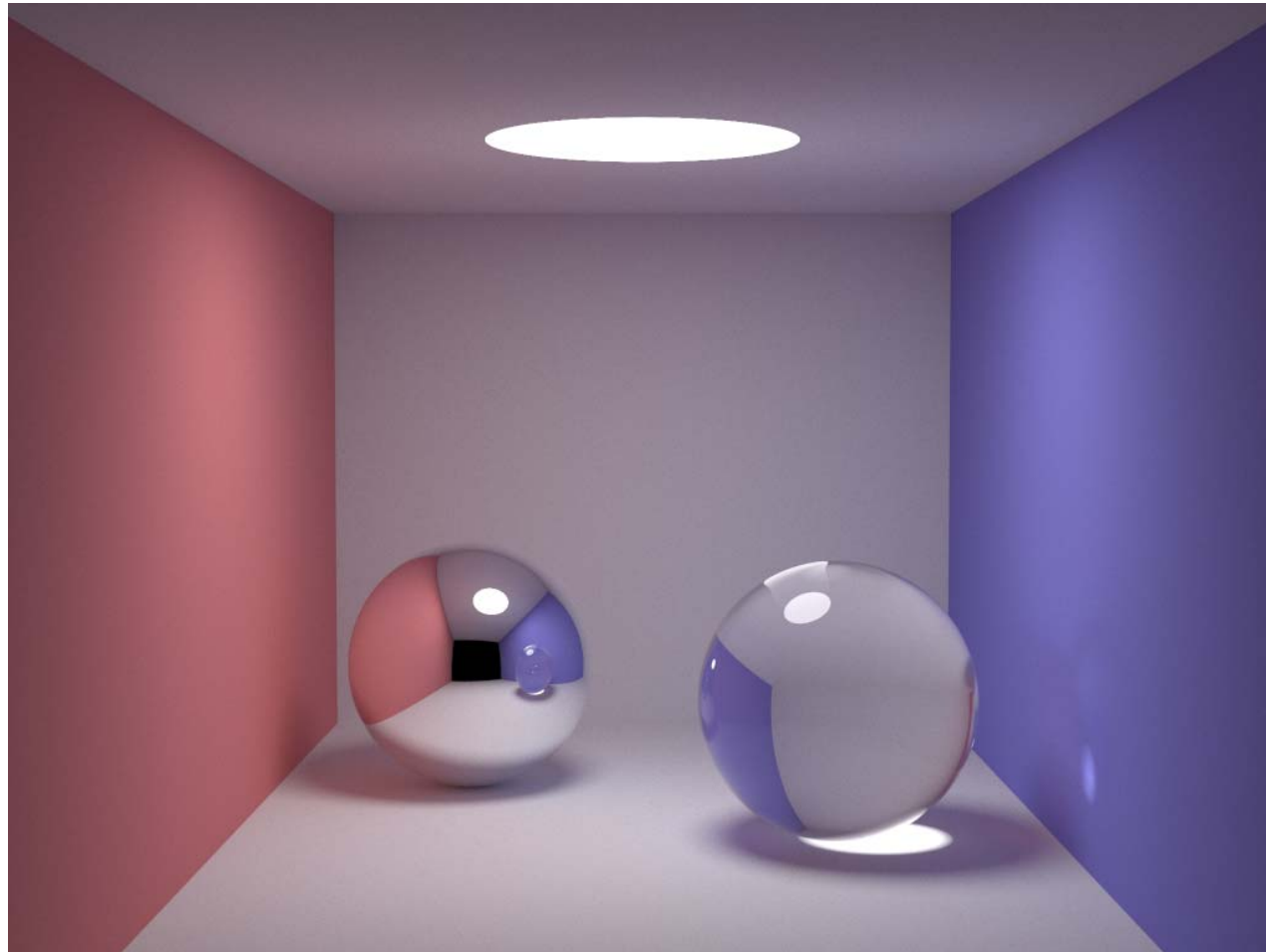
return L;
```

- Avec un lanceur de rayon en place, on peut coder un *path tracer* de base en moins de 50 lignes de code



• Voir *smallpt* en ligne

Explicit Path Tracing w/ Russian Roulette



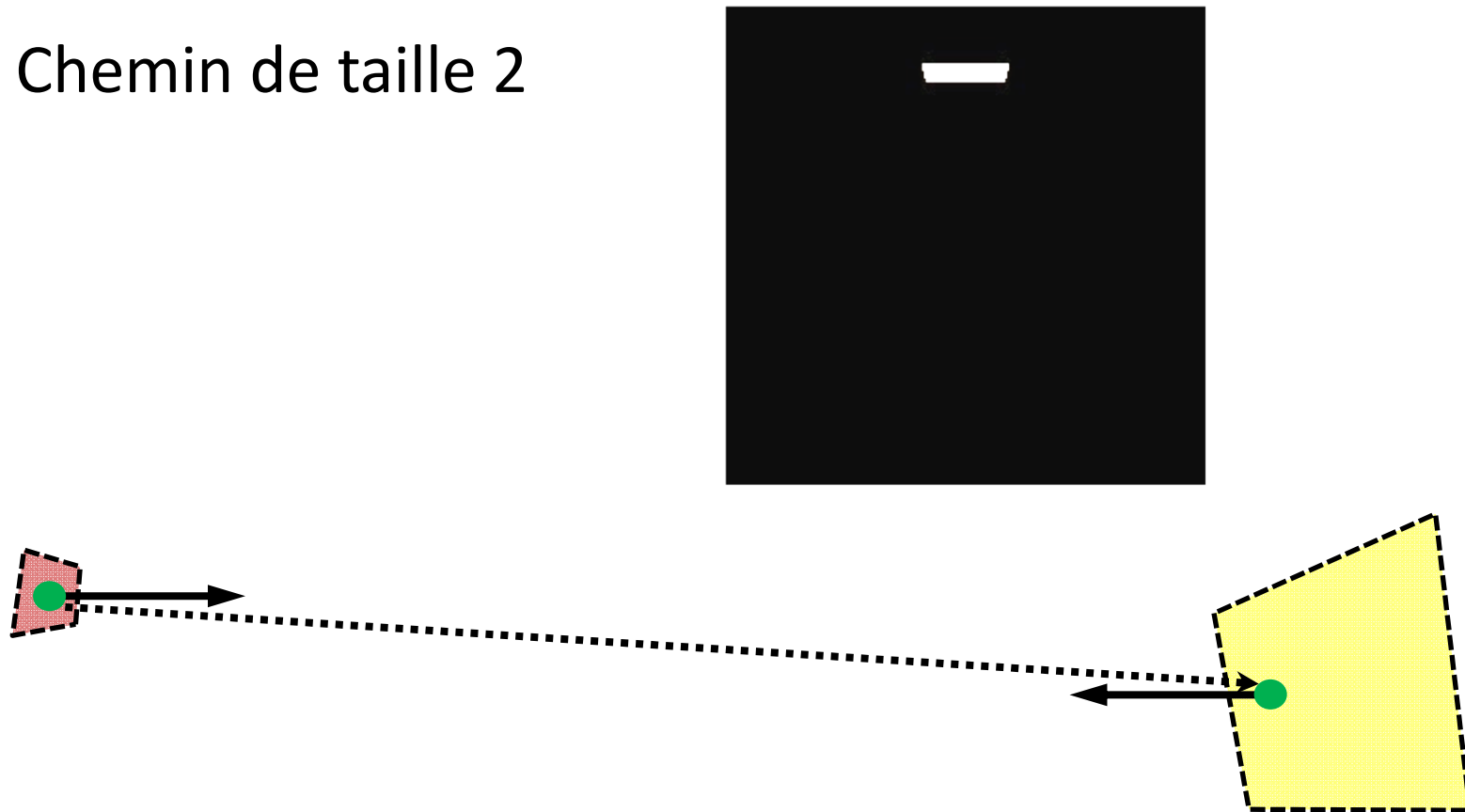
<http://www.kevinbeason.com/smallpt/>

99 lignes de C++



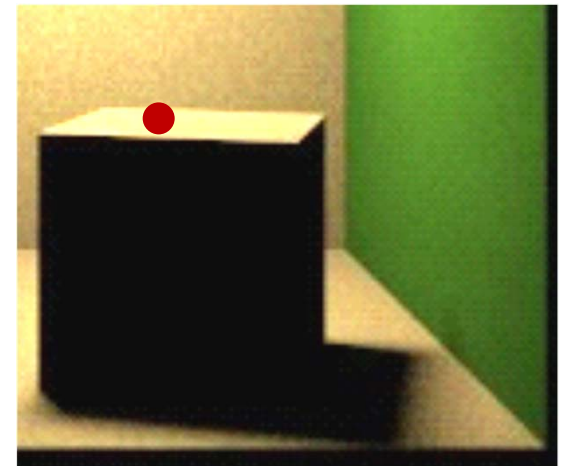
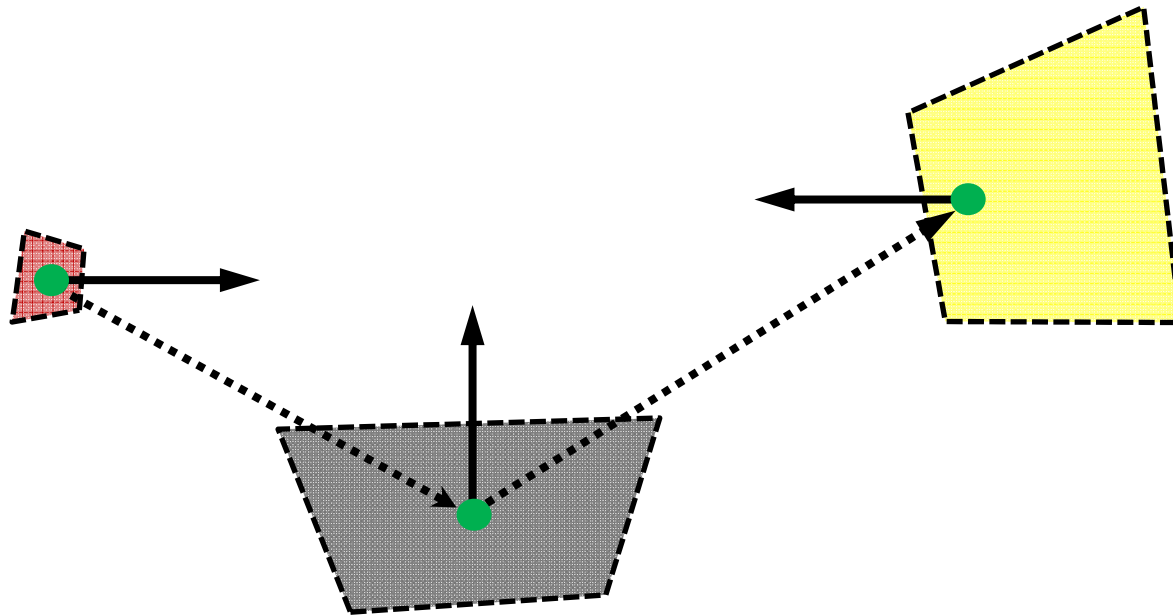
Quelques exemples de chemin particulier

- Lumière émissive
 - Chemin de taille 2



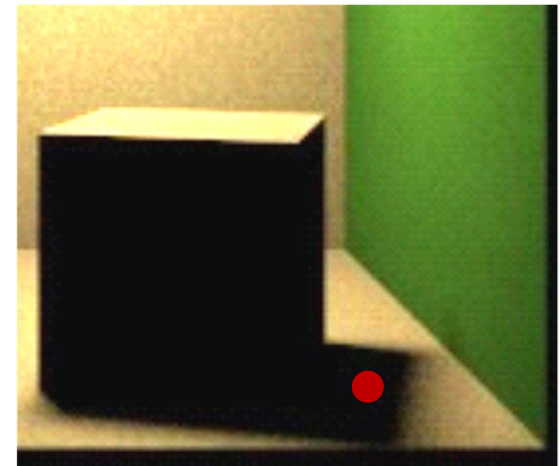
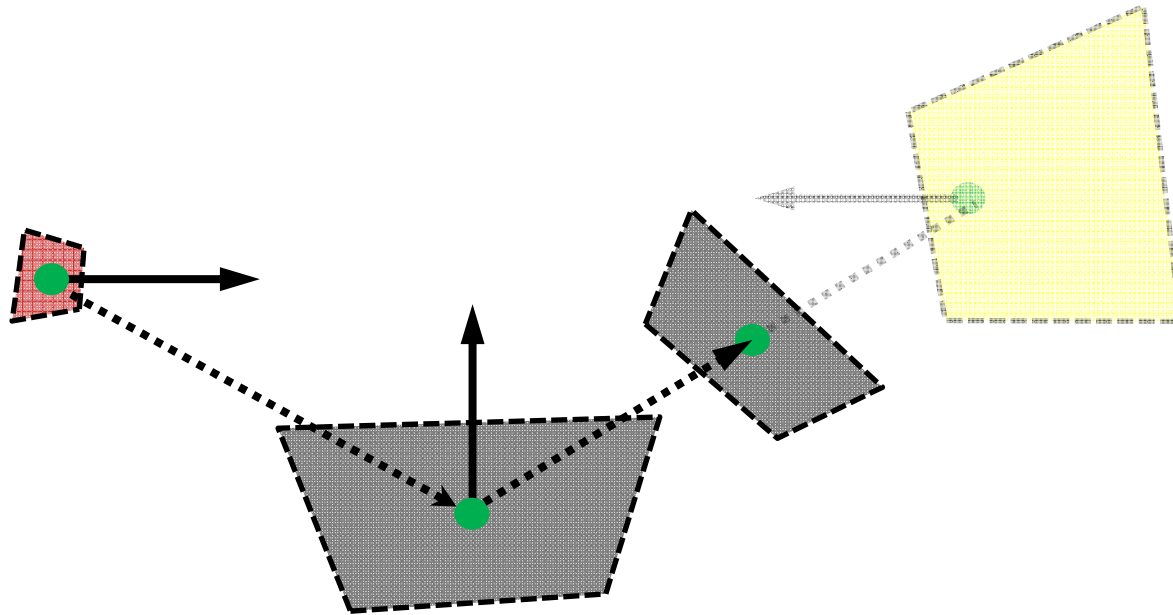
Quelques exemples de chemin particulier

- Illumination directe diffus
 - Chemin de taille 3



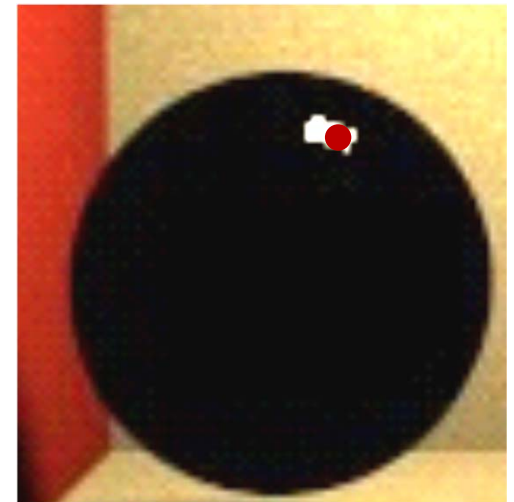
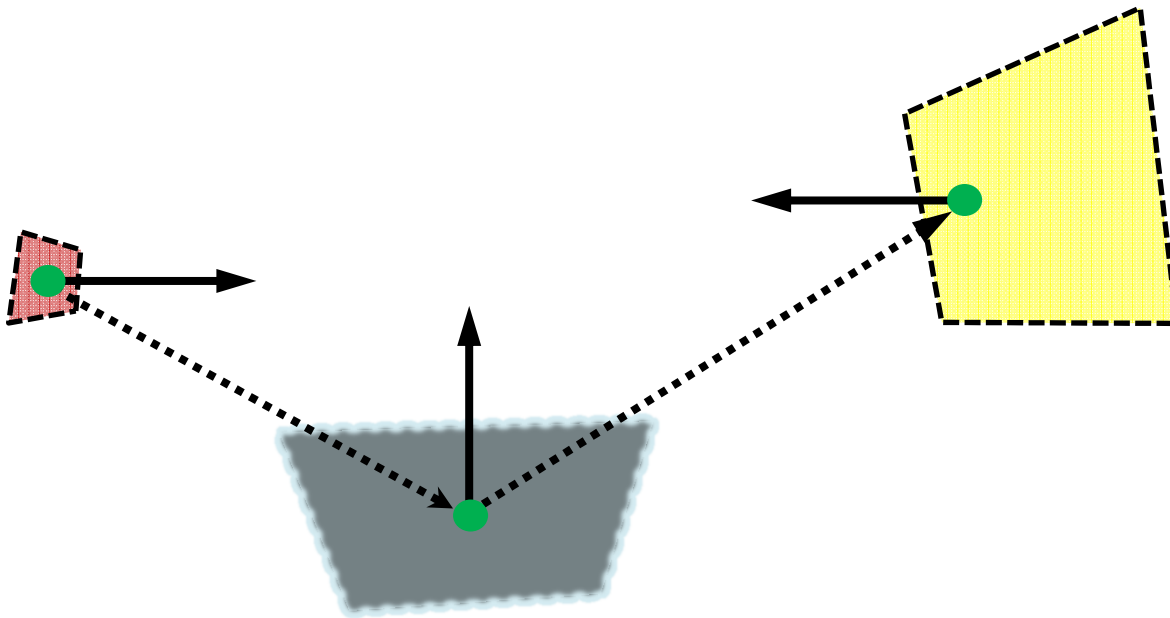
Quelques exemples de chemin particulier

- Illumination directe diffus
 - Chemin de taille 3



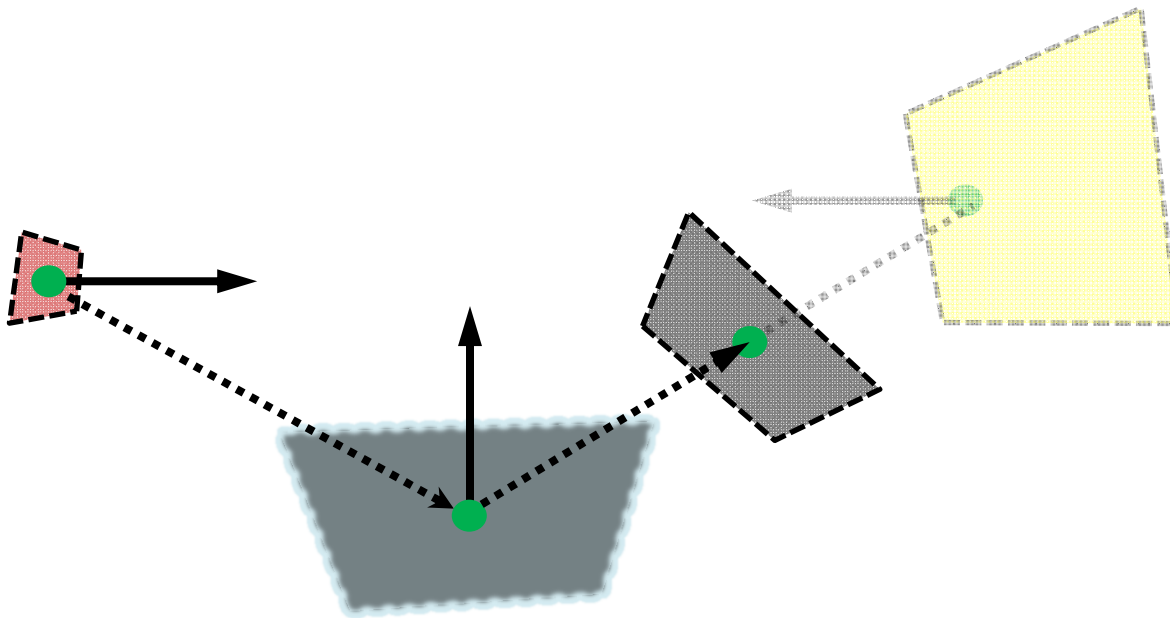
Quelques exemples de chemin particulier

- Illumination directe spéculaire
 - Chemin de taille 3



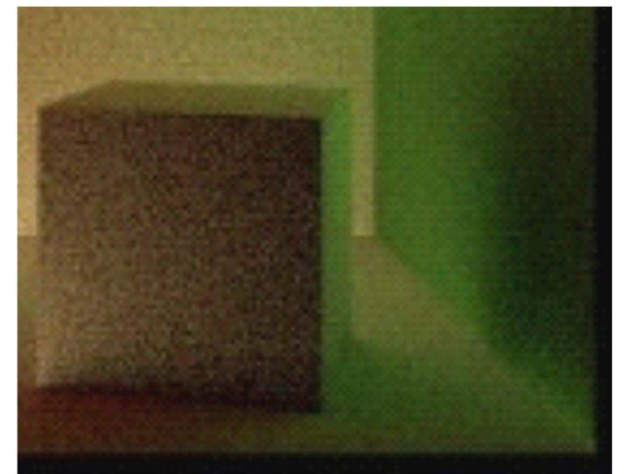
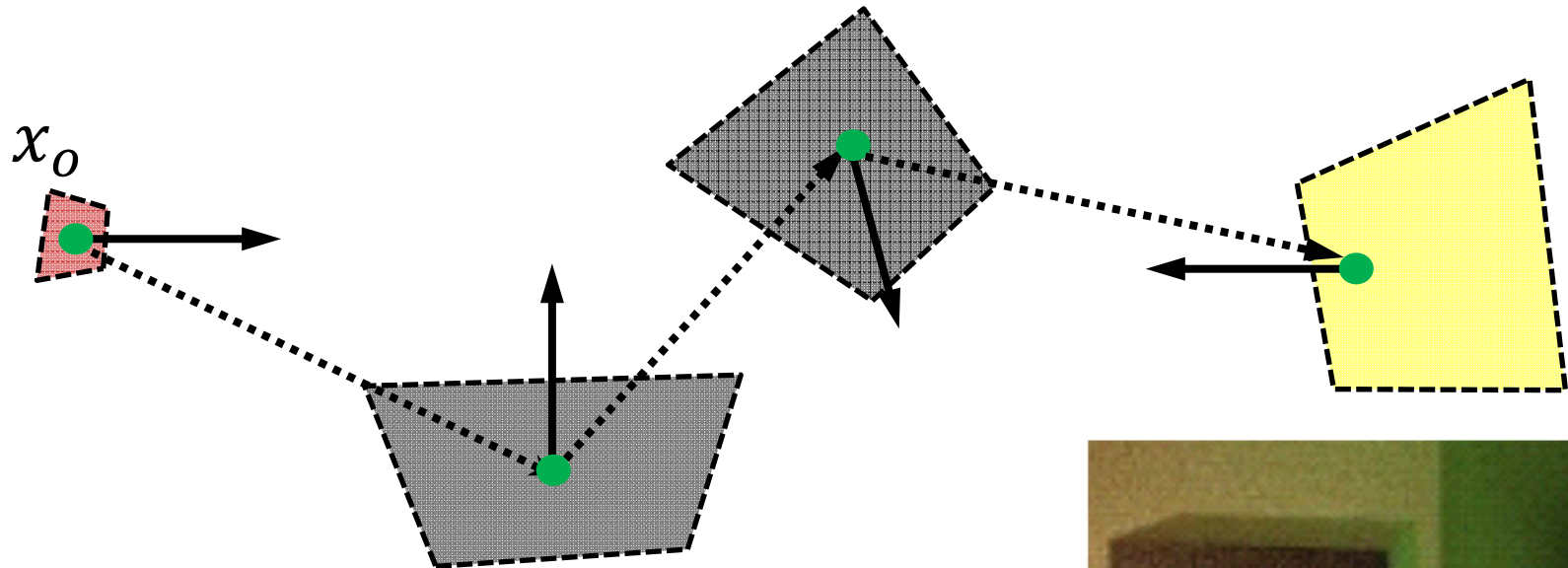
Quelques exemples de chemin particulier

- Illumination directe spéculaire
 - Chemin de taille 3



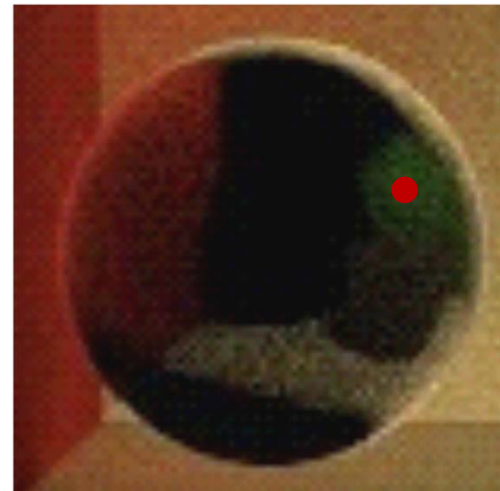
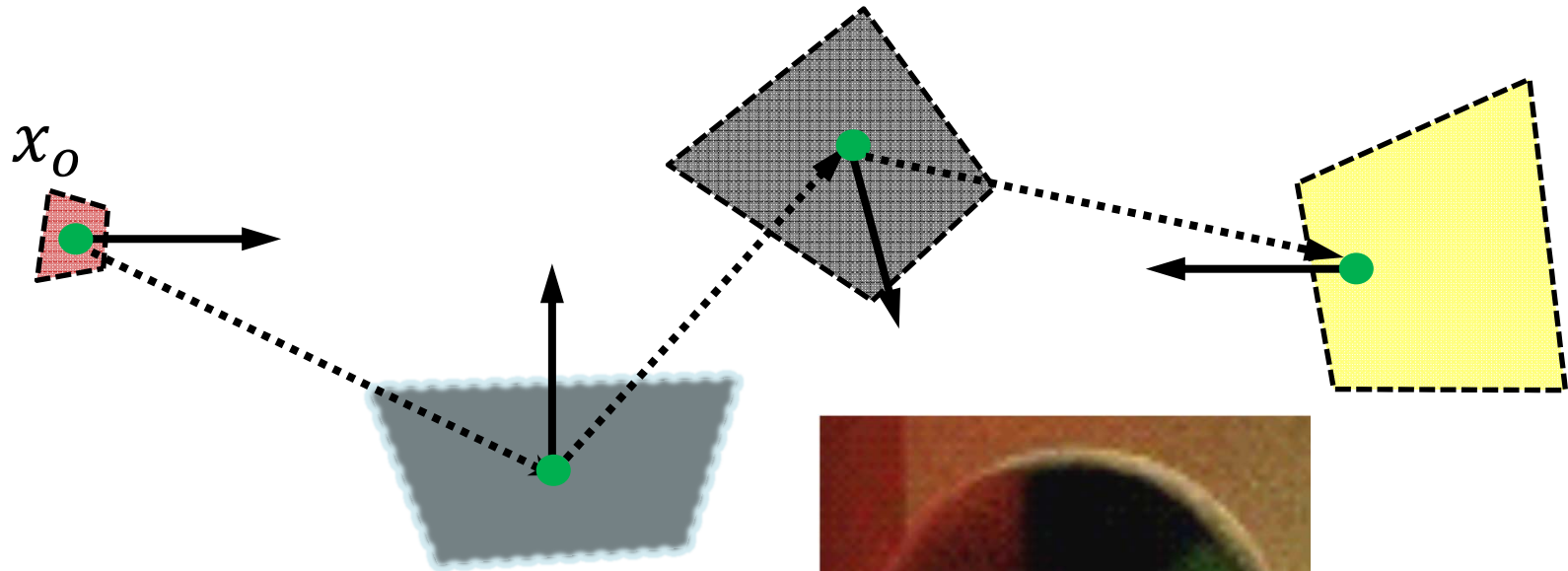
Quelques exemples de chemin particulier

- Illumination indirecte *1-bounce* diffus
 - Chemin de taille de 4



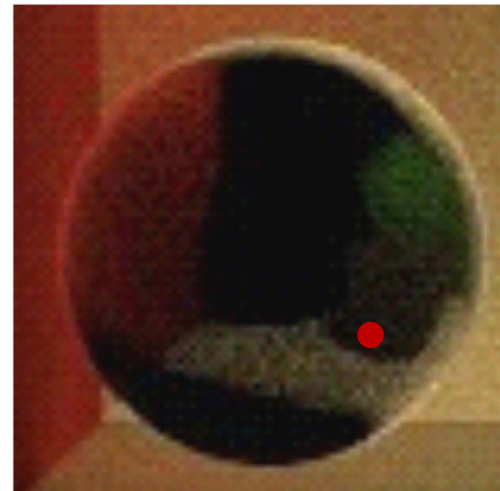
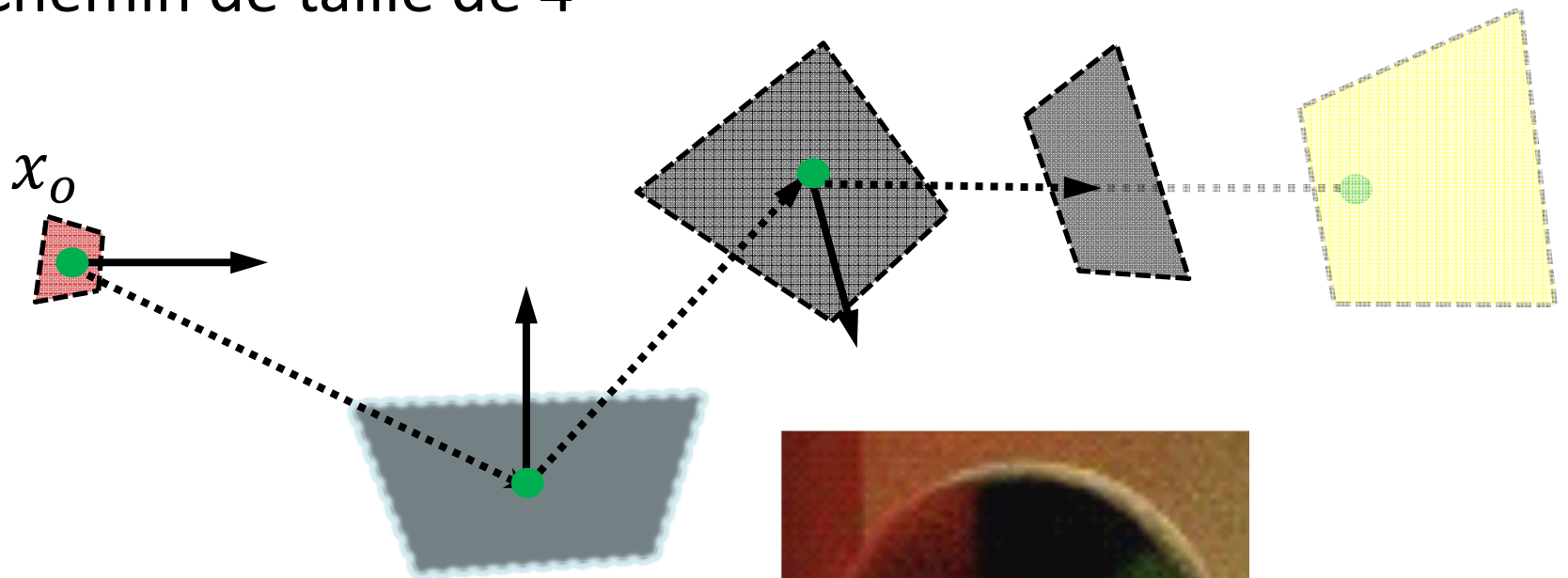
Quelques exemples de chemin particulier

- Illumination indirecte *1-bounce* spéculaire
 - Chemin de taille de 4



Quelques exemples de chemin particulier

- Illumination indirecte *1-bounce* spéculaire
 - Chemin de taille de 4



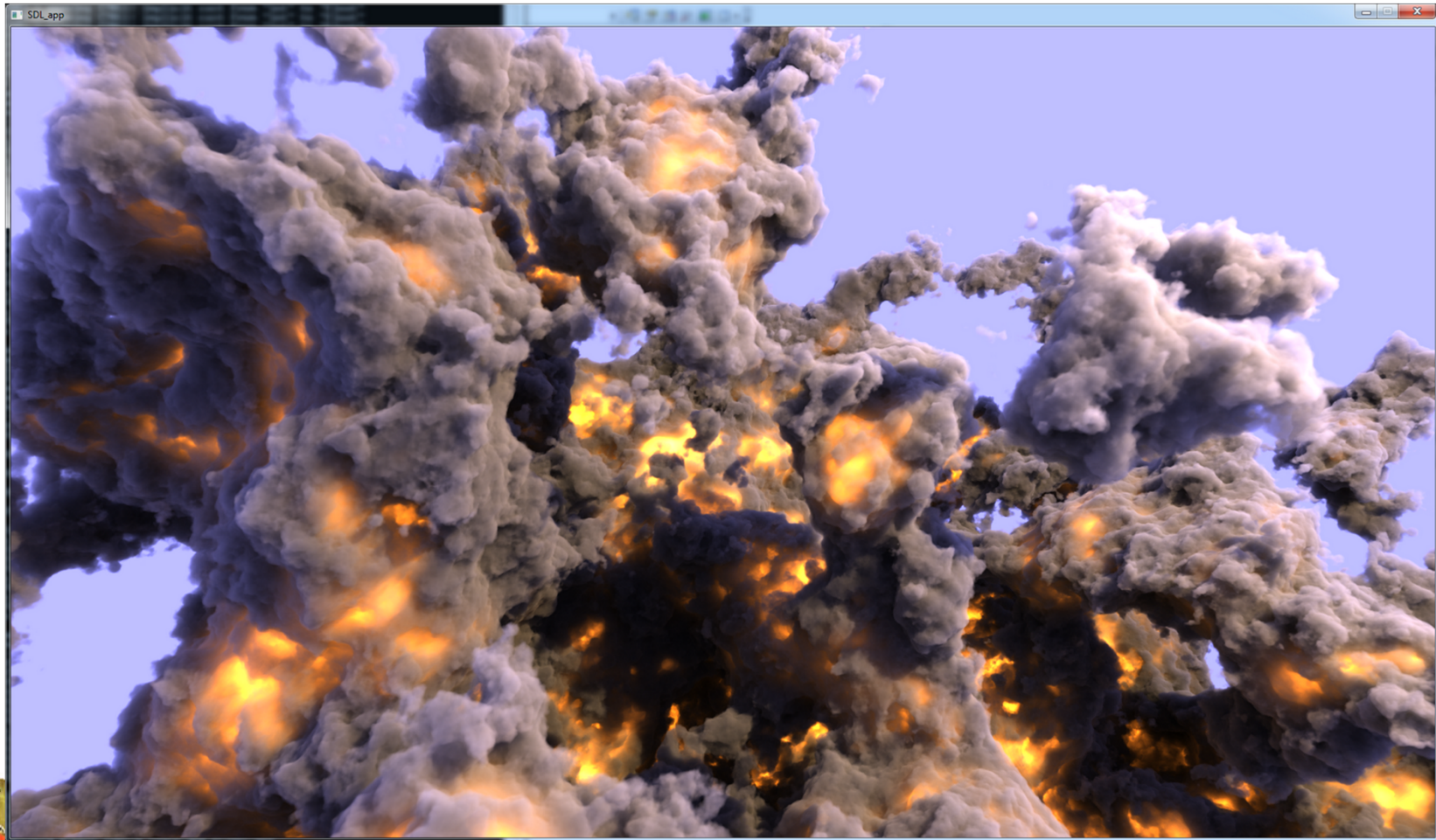
Problèmes avec *path-tracing*

- Avantages de *path-tracing*
 - Facile à implémenter
 - Converge à la bonne réponse



Problèmes avec *path-tracing*

– Convergence à la bonne réponse



Problèmes avec *path-tracing*

- Convergence à la bonne réponse



Problèmes avec *path-tracing*

- Désavantages de *path-tracing*
 - Convergence L-E-N-T → bruit



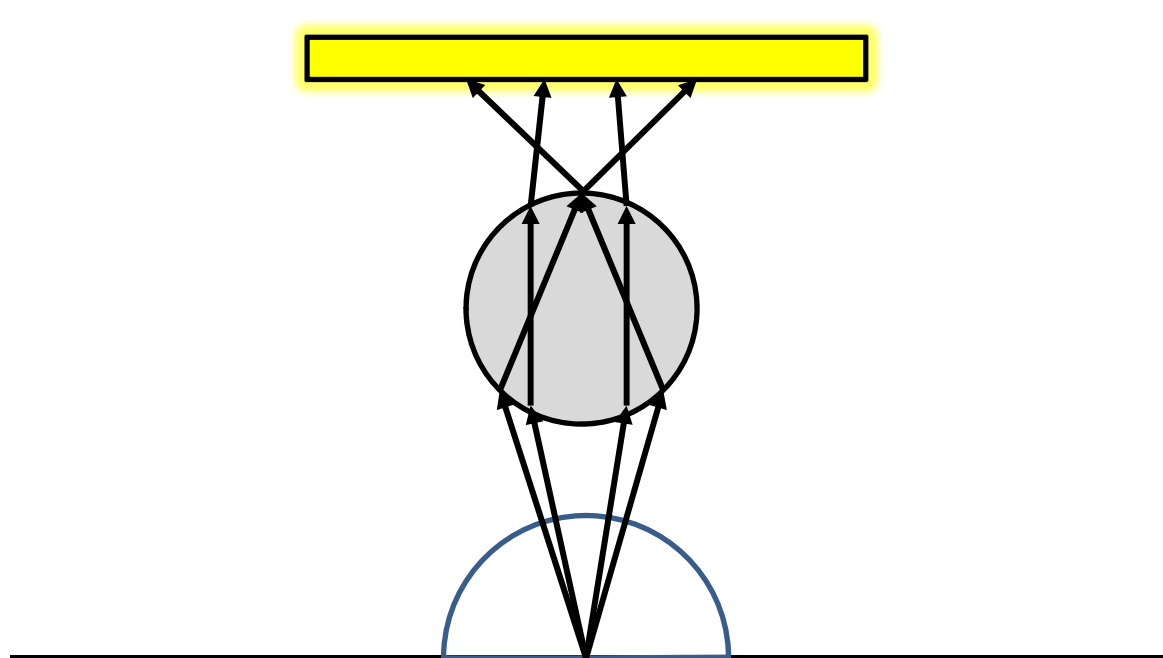
Problèmes avec *path-tracing*

- Convergence L-E-N-T \rightarrow bruit



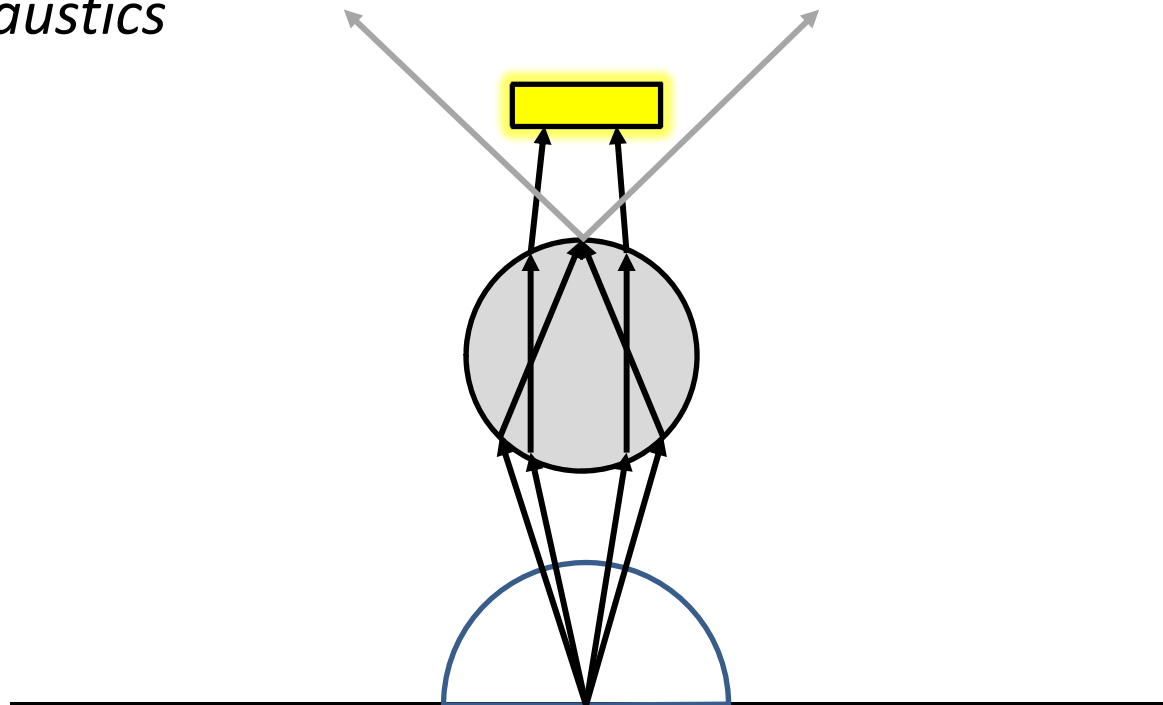
Problèmes avec *path-tracing*

- Désavantages de *path-tracing*
 - Difficultés avec les chemins d'illumination complexes:
 - *caustics*



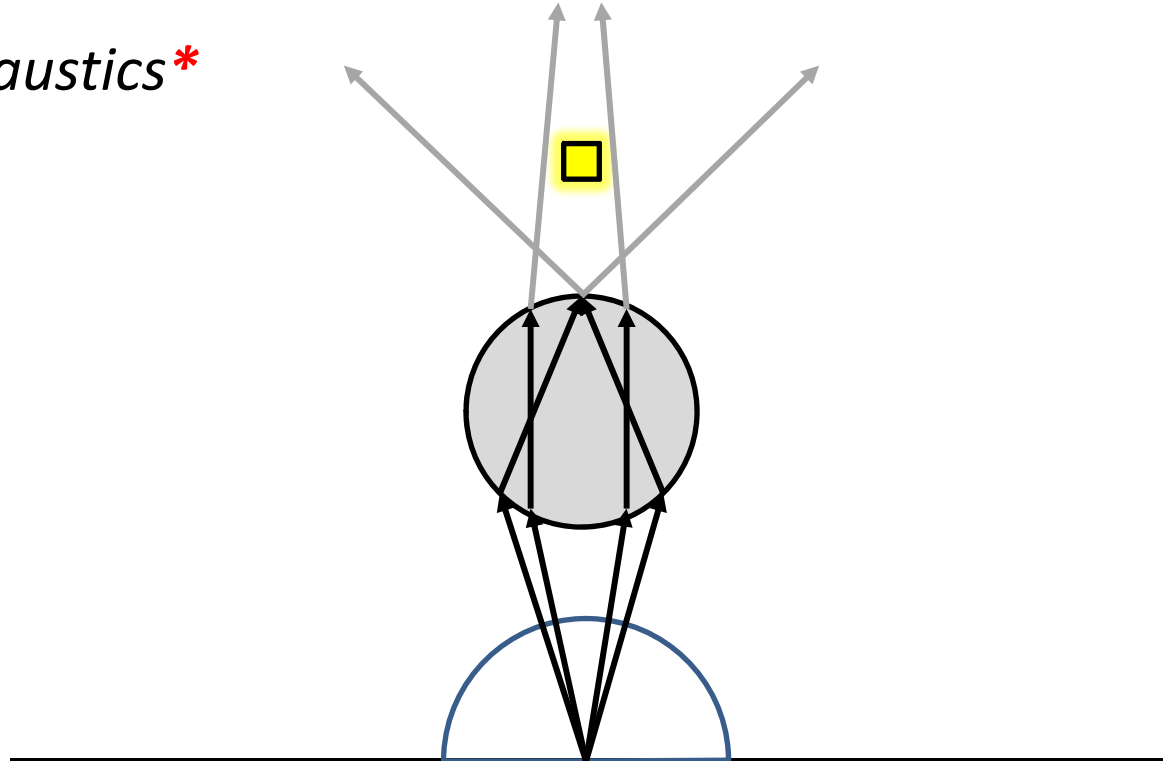
Problèmes avec *path-tracing*

- Désavantages de *path-tracing*
 - Difficultés avec les chemins d'illumination complexes:
 - *caustics*



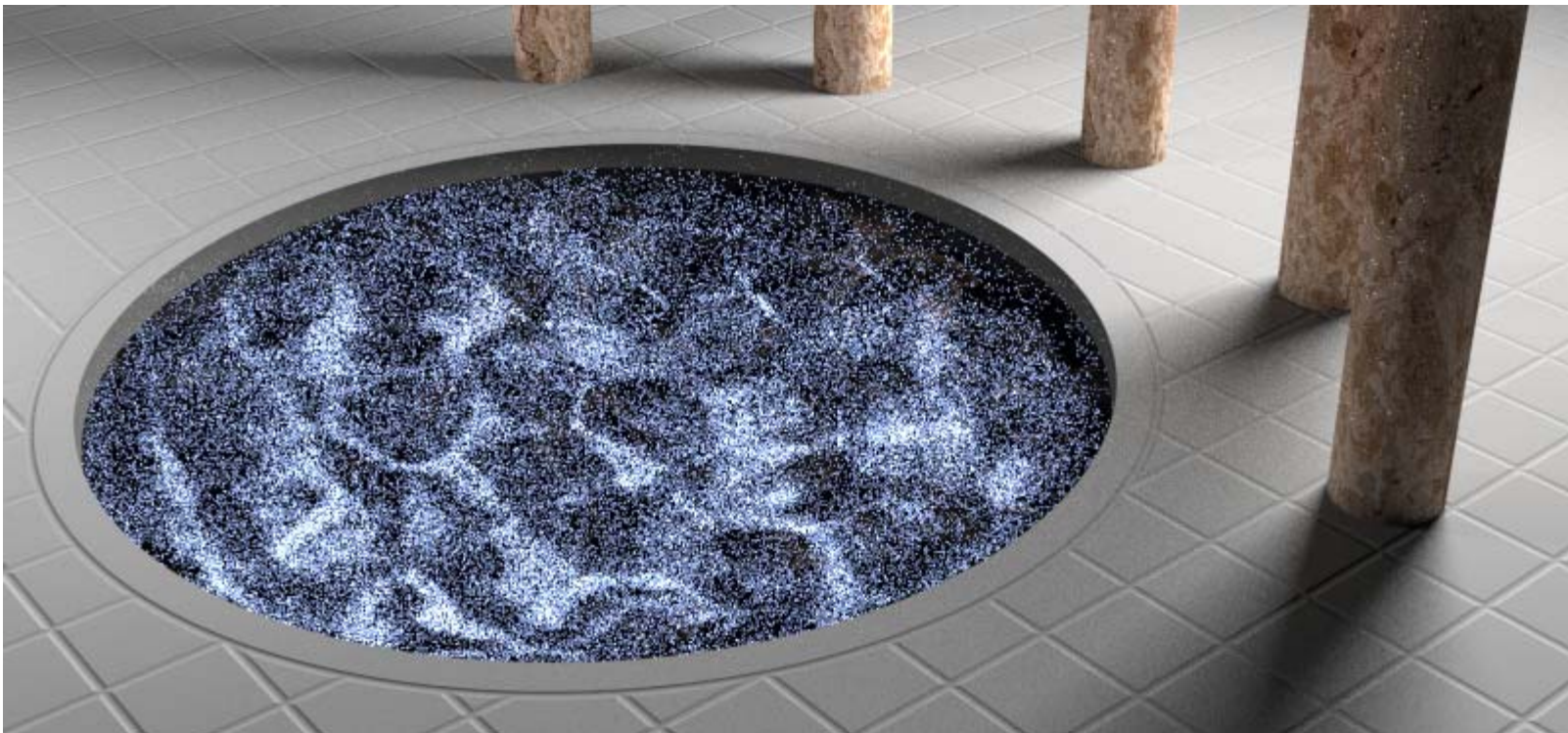
Problèmes avec *path-tracing*

- Désavantages de *path-tracing*
 - Difficultés avec les chemins d'illumination complexes:
 - *caustics**



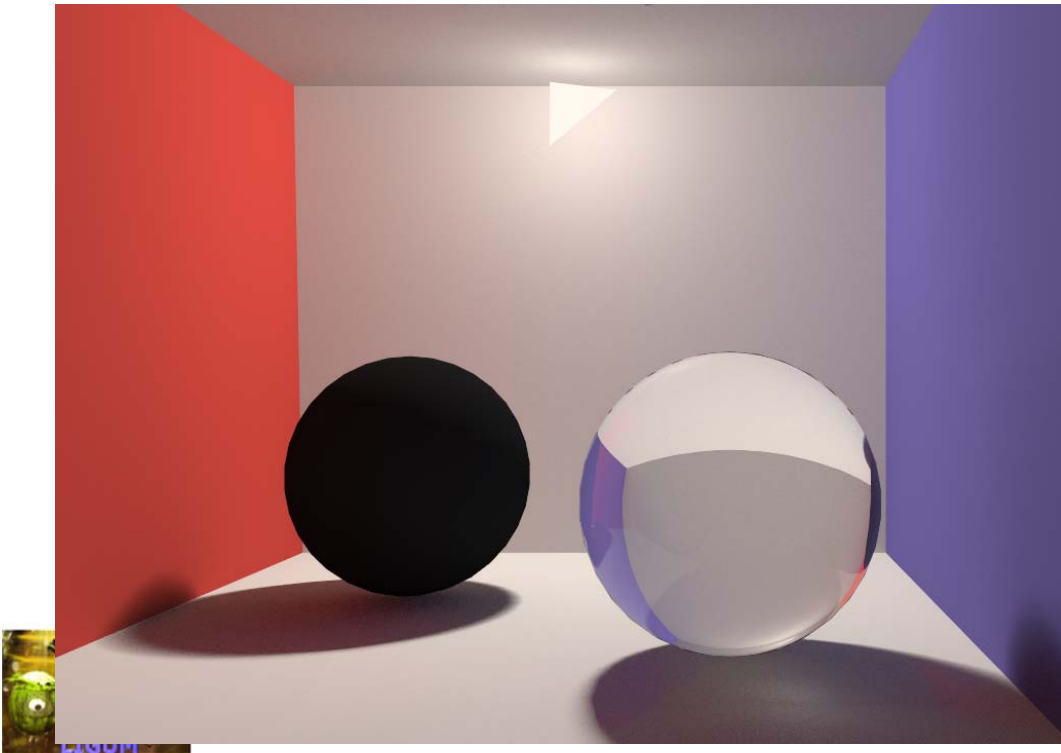
Problèmes avec *path-tracing*

- Désavantages de *path-tracing*
 - Difficultés avec les chemins d'illumination complexes:
 - *caustics*



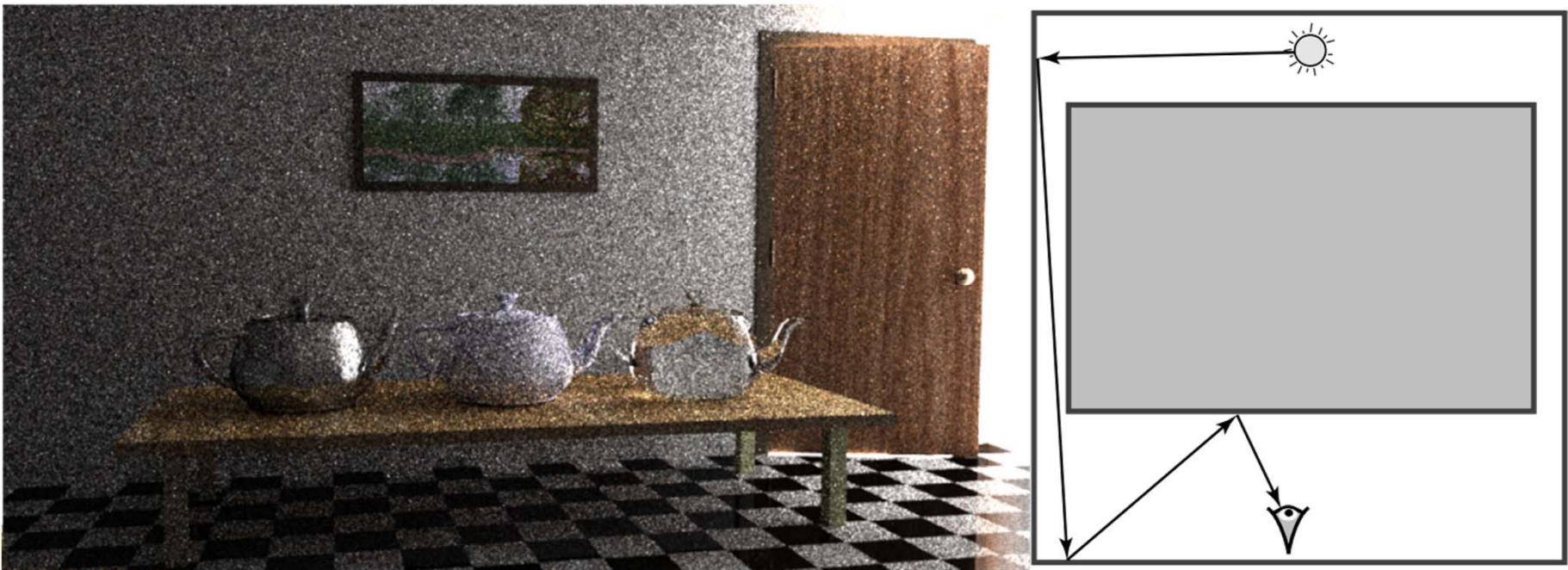
Problèmes avec *path-tracing*

- Désavantages de *path-tracing*
 - Difficultés avec les chemins d'illumination complexes:
 - *caustics*



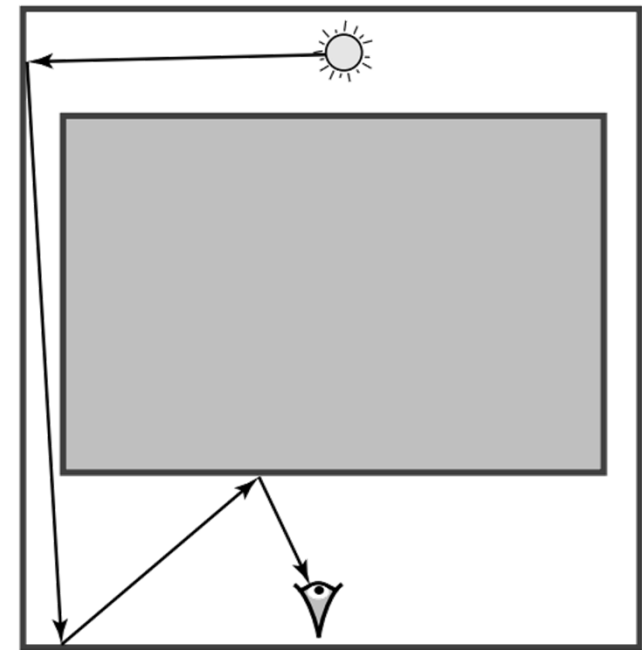
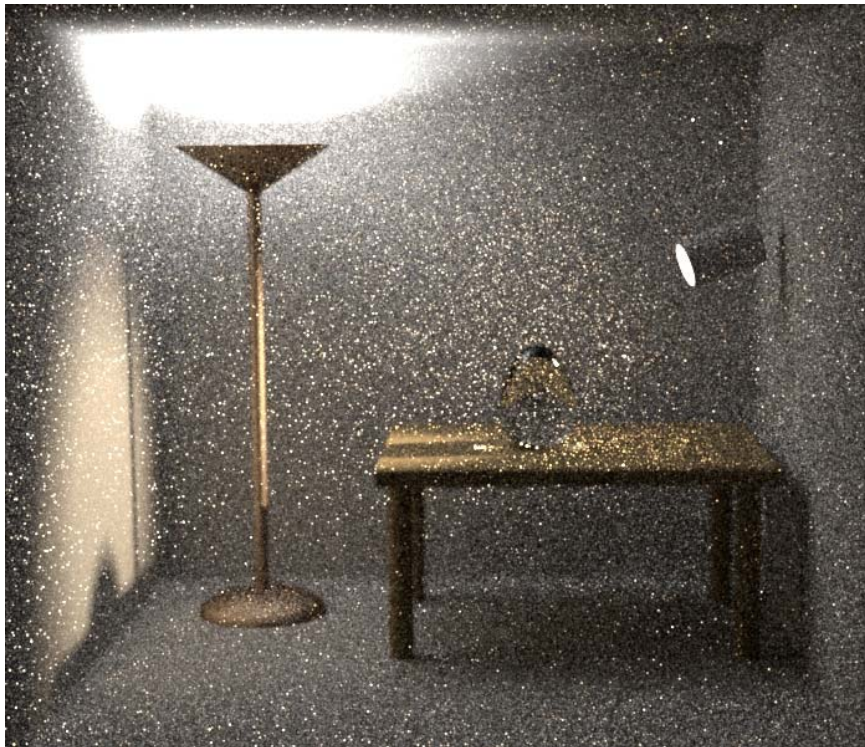
Problèmes avec *path-tracing*

- Désavantages de *path-tracing*
 - Difficultés avec les chemins d'illumination complexes:
 - contribution venant principalement des chemins indirectes



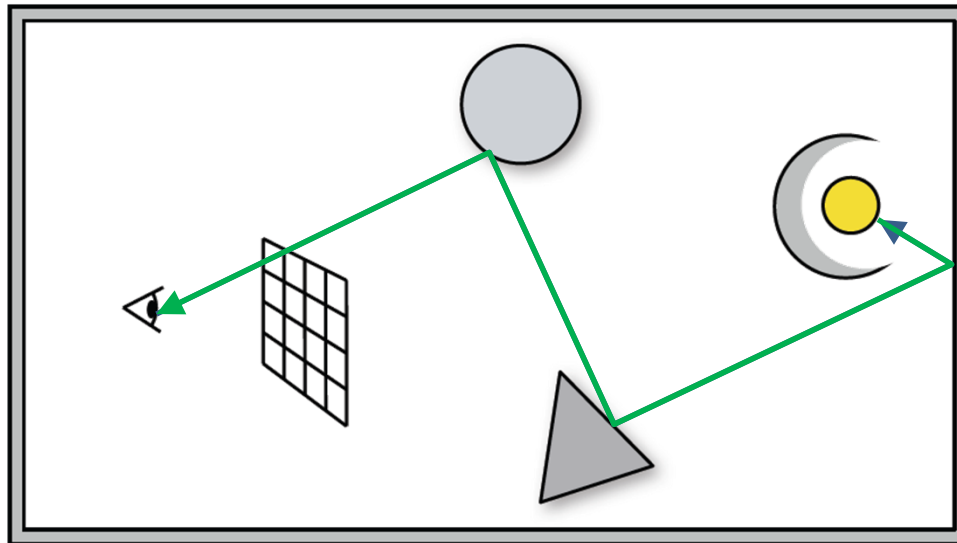
Problèmes avec *path-tracing*

- Désavantages de *path-tracing*
 - Difficultés avec les chemins d'illumination complexes:
 - contribution venant principalement des chemins indirectes



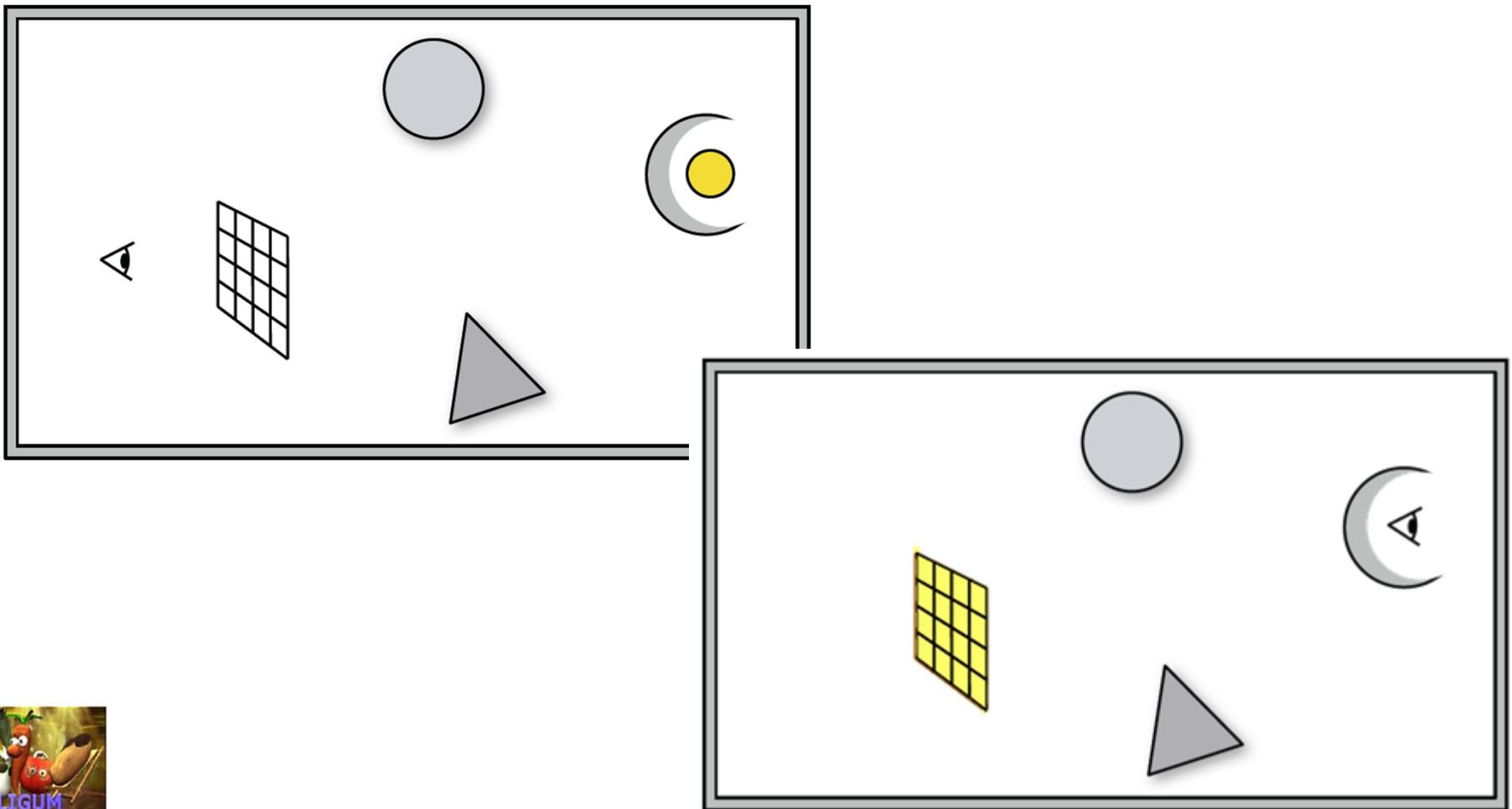
Light-tracing

- Pourquoi pas commencer de l'autre bout?
 - Les caustics sont beaucoup plus facile de simuler commençant de la lumière



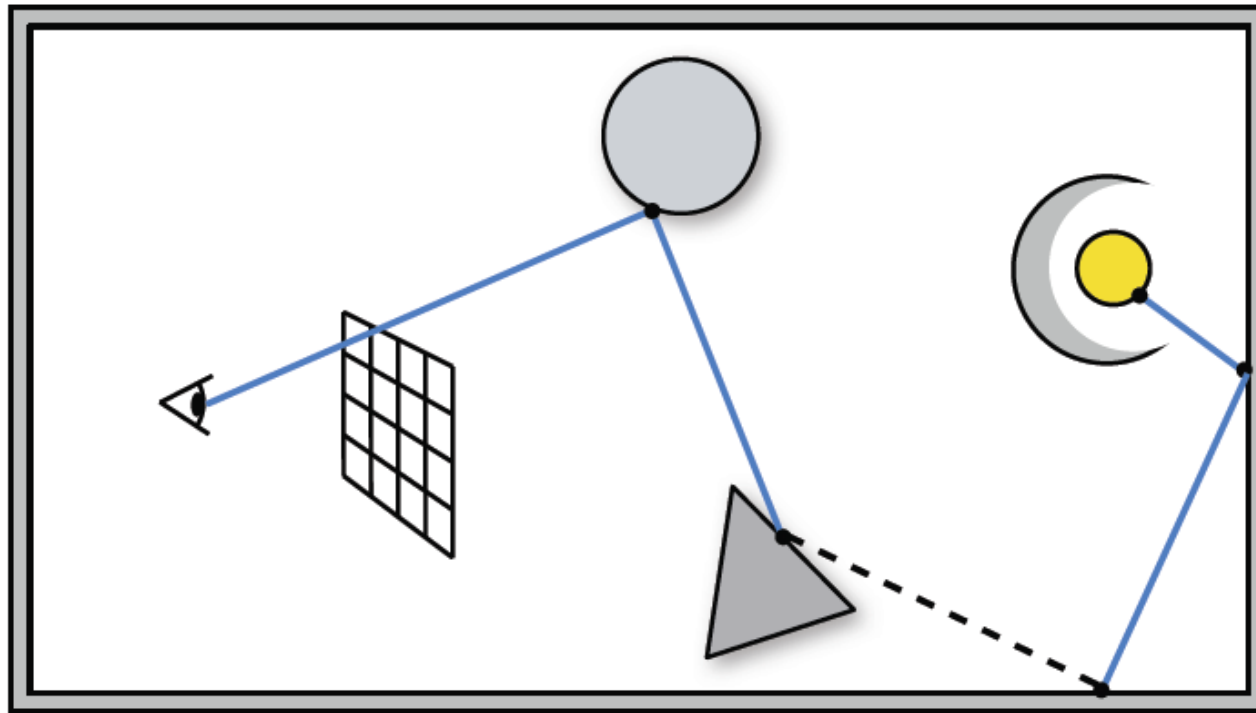
Light-tracing vs. path-tracing

- Dualité caméra-lumière



Bi-directional path-tracing

- Meilleur des deux mondes
 - On peut générer des *sous-chemins* de l'oeil et de la lumière, et puis on va connecter ces sous-chemins pour former nos chemins complets



Bi-directional path-tracing



Path tracing with 56
samples per pixel

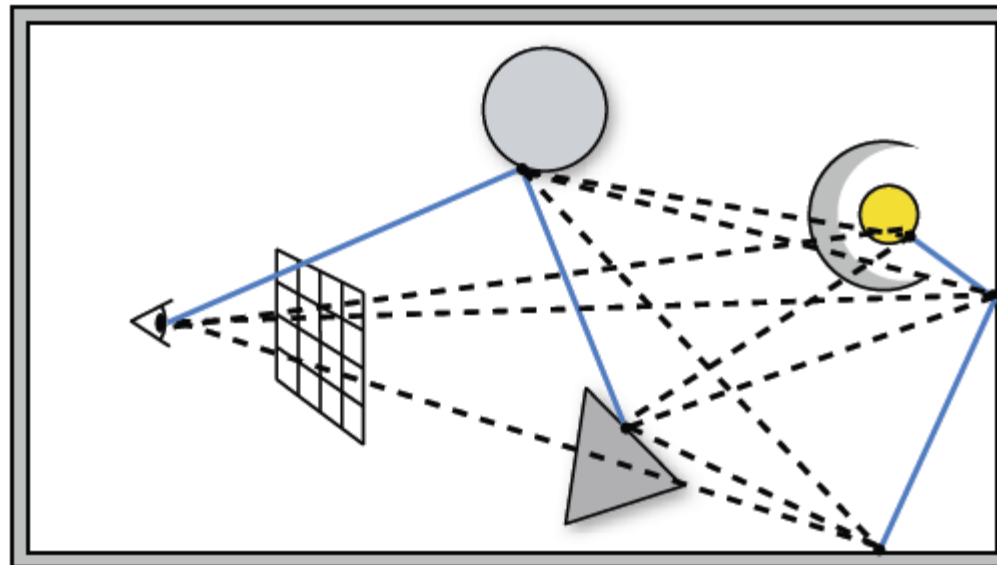


Bidirectional path tracing with
25 samples per pixel



Bi-directional path-tracing

- Étant donné deux sous-chemins (de l'oeil et de la lumière) de taille i et j , il y a $N-1$ manières de bâtir un chemin de taille N :
 - Ex: pour $N=4$, on a $i=1$ et $j=3$, ou $i=2$ et $j=2$, ou ...
- On peut aussi re-utiliser des sous-chemins:



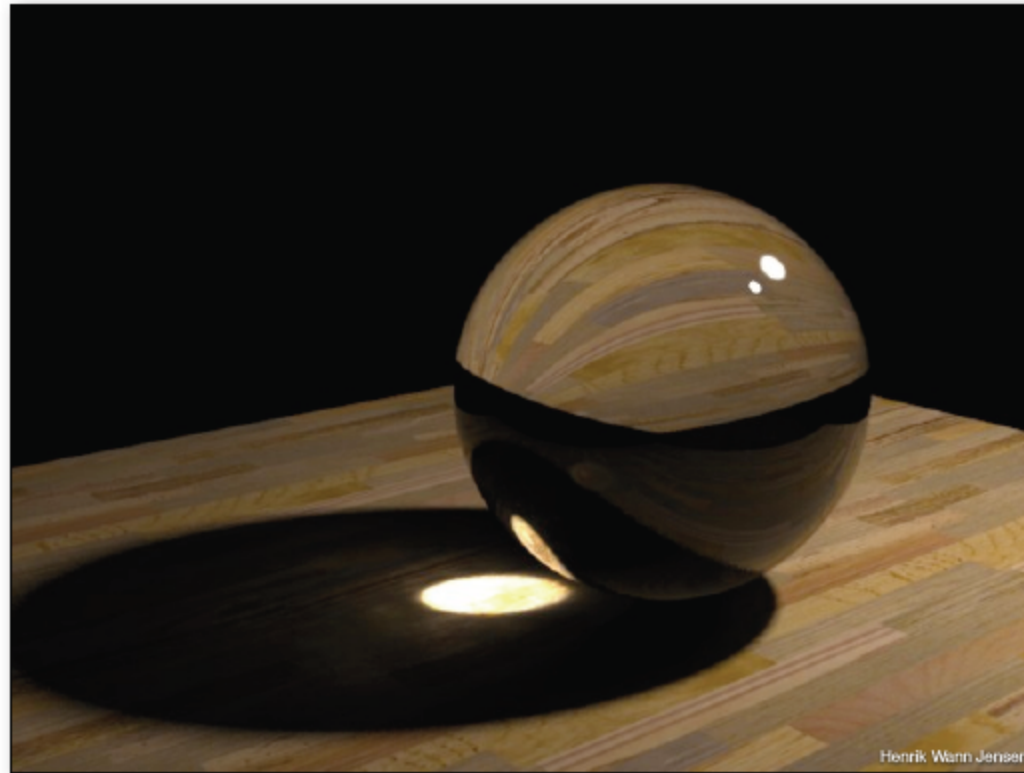
Bi-directional path-tracing



Bi-directional path-tracing

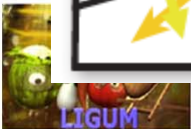
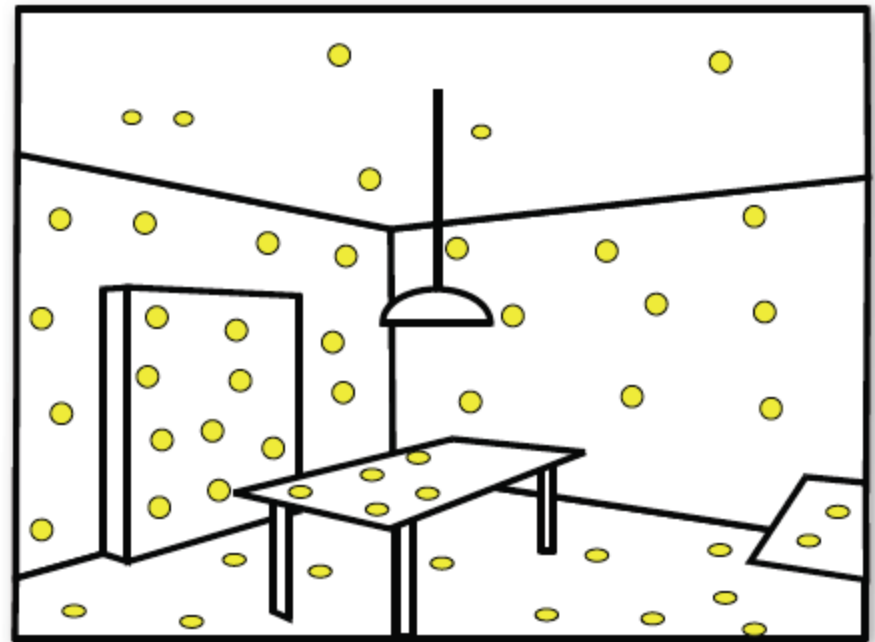
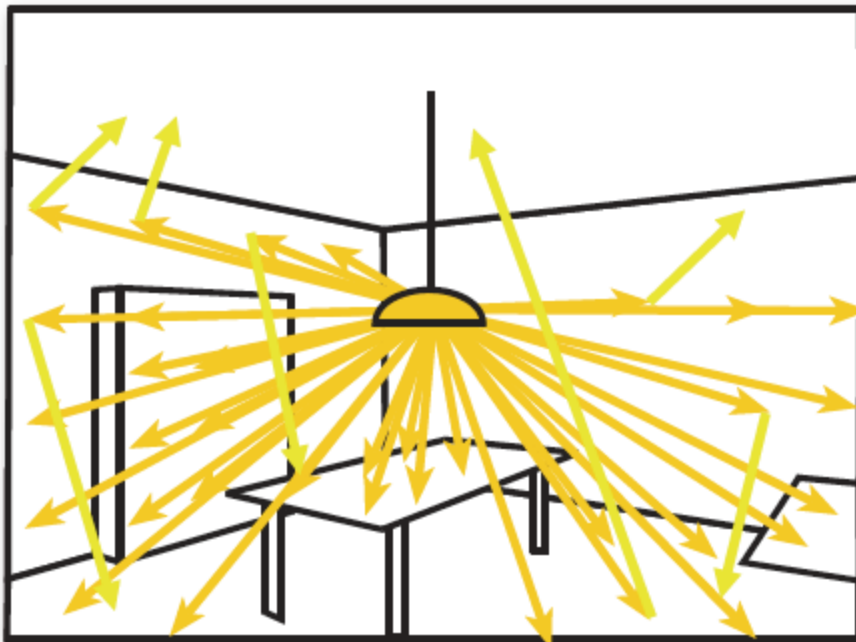


Photon Mapping



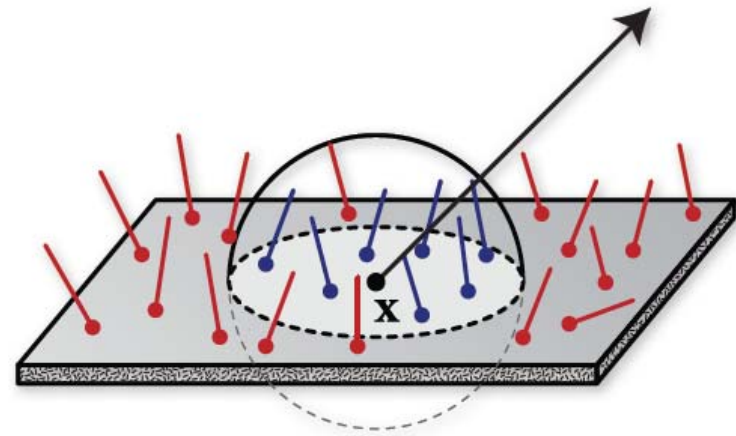
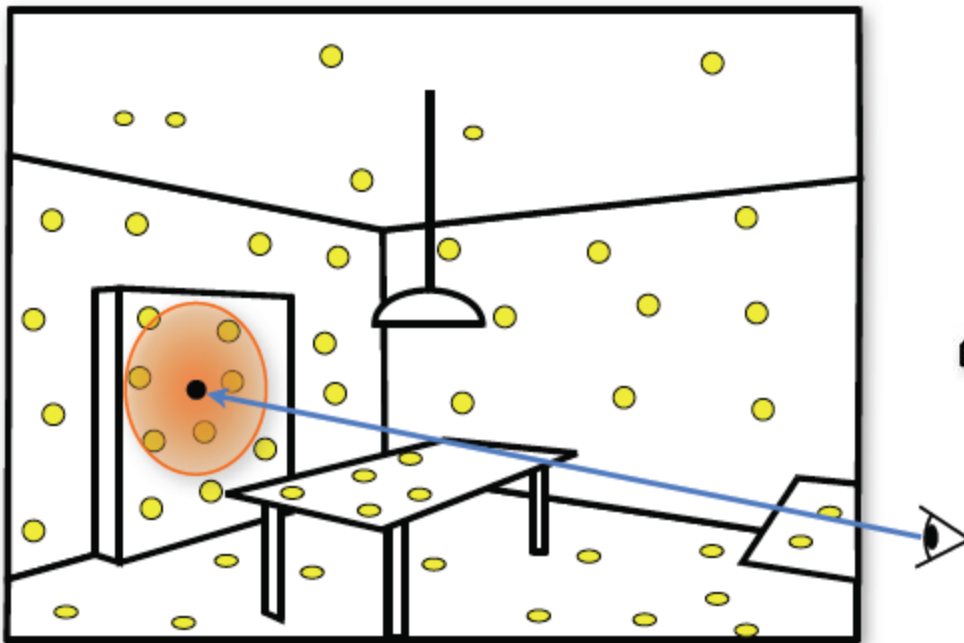
Photon Mapping

- Algorithme avec 2 étapes:
 1. Tracer de particule d'illumination (*photons*)
 2. Estimation de densité pour générer l'image finale



Photon Mapping

- Algorithme avec 2 étapes:
 1. Tracer de particule d'illumination (*photons*)
 2. Estimation de densité pour générer l'image finale

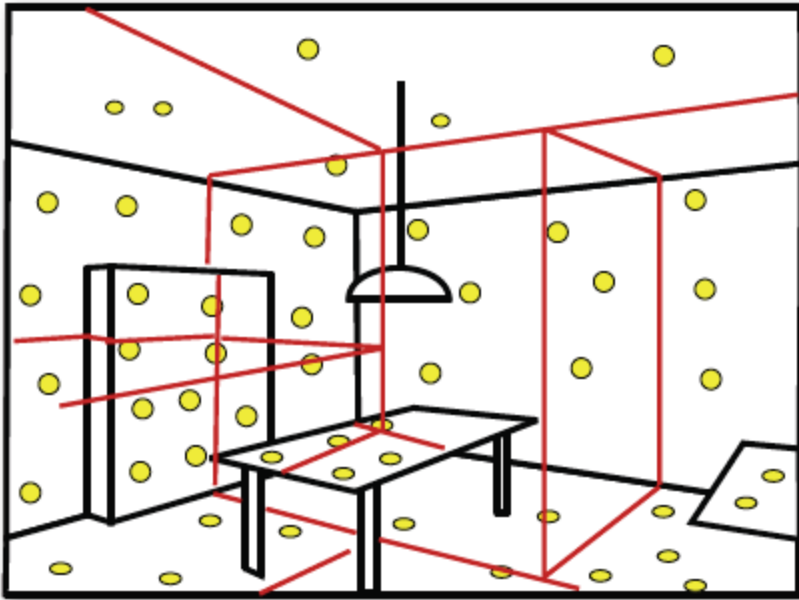


$$L_r(\mathbf{x}, \vec{\omega}) \approx \sum_{p=1}^n f_r(\mathbf{x}, \vec{\omega}_p, \vec{\omega}) \frac{\Delta\Phi_p(\mathbf{x}, \vec{\omega}_p)}{\Delta A}$$



Photon Mapping

- Typiquement on stocke les photons dans un *kd-tree* afin d'accélérer la recherche nécessaire pour l'étape d'estimation de densité



Photon Mapping

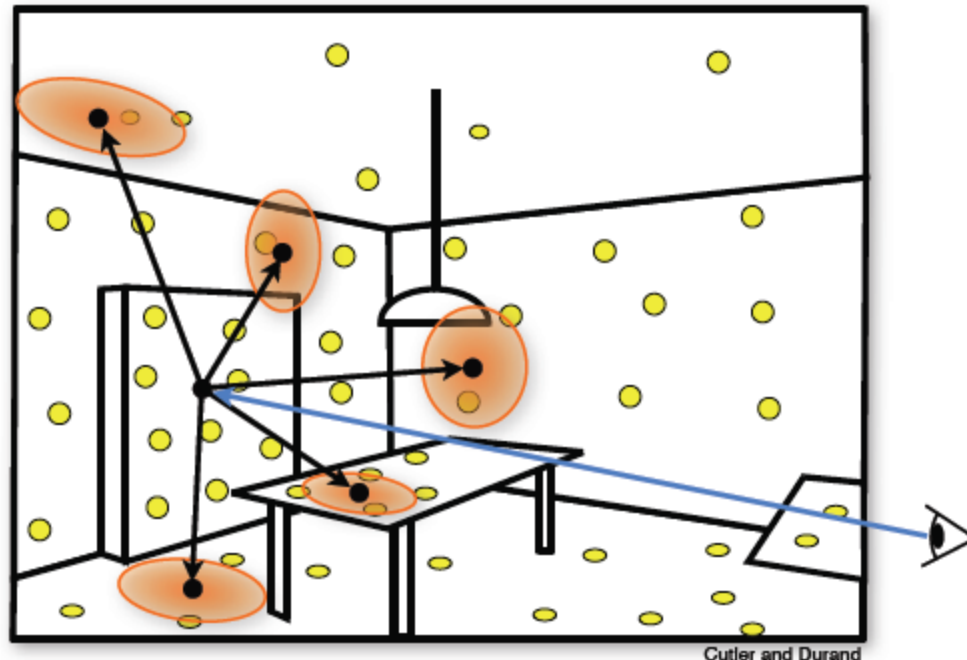


500000 photons / 500 photons in radiance estimate



Photon Mapping

- L'estimation de densité est souvent bruité
 - Qualité dépend sur la densité de photons, ainsi que la taille de recherche
- *Final gathering* nous permet de réduire le bruit



Photon Mapping

