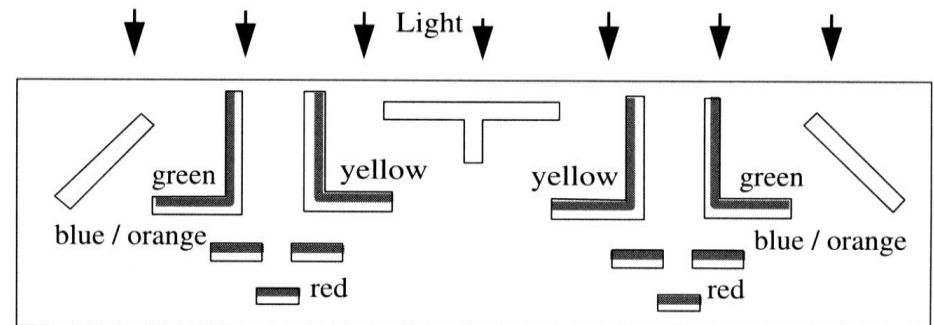


Radiosité

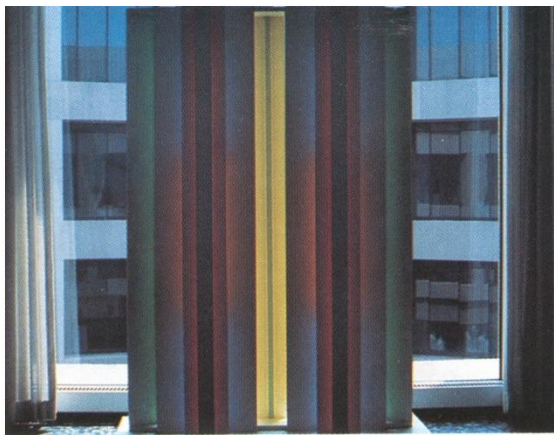
Radiosité

- ▶ Prendre en compte toutes les inter-réflexions

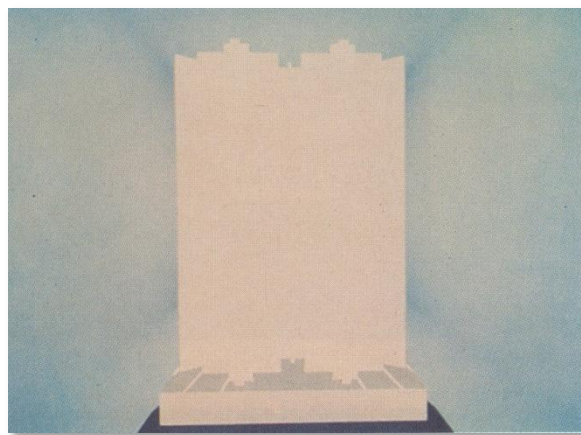
Sculpture de John Ferren



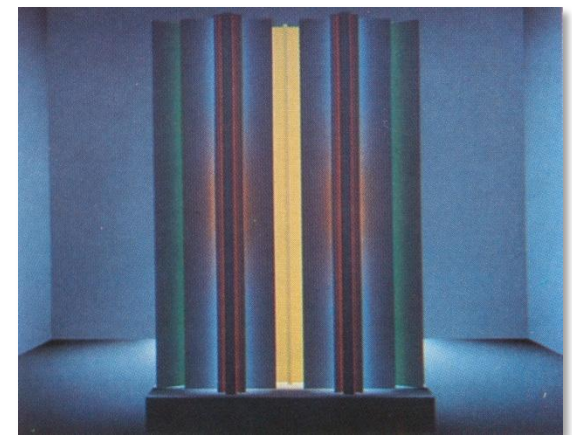
All visible surfaces, white.



Photo



Lancer de rayon



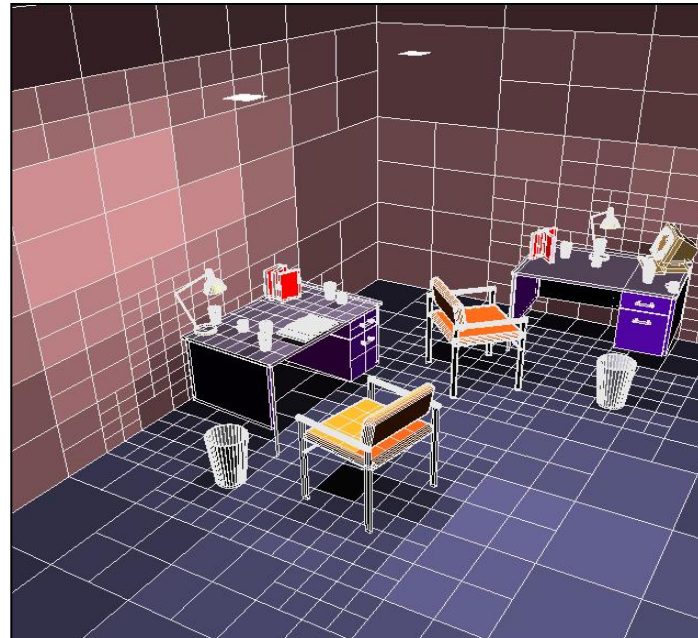
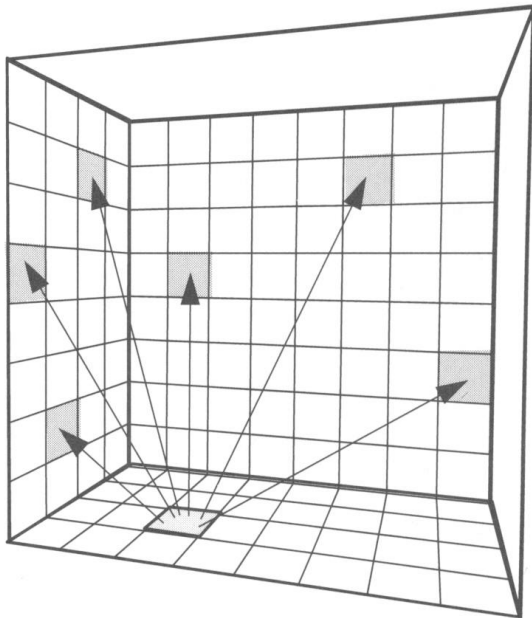
Radiosité

Les méthodes de radiosit  [1984]

- ▶ Hypoth se : mat riaux diffus
- ▶ Radiance, BRDF... ind pendantes de la direction
- ⇒ Simplification de l' quation de l' clairage
- ▶ M thode de radiosit  :
 - Discr tisation de cette  quation en espace objet ind pendamment du point de vue
 - R solution de l' quation discr tis e
 - G n ration finale des images en fonction du point de vue

L'équation de radiosité

- ▶ Environnement échantillonné sous la forme de patches discrets, de taille finie, émettant et réfléchissant la lumière uniformément sur leurs surface (choix d'une base)



Simplification et discrétisation

$$L(\mathbf{x}, \mathbf{d}) = E(\mathbf{x}, \mathbf{d}) + \int_S \rho(\mathbf{x}, \mathbf{d}, \mathbf{d}') v(\mathbf{x}, \mathbf{x}') \cos(\theta) dL(\mathbf{x}, \mathbf{d}')$$

► Forme simplifiée :

$$B(\mathbf{x}) = E(\mathbf{x}) + \int_S B(\mathbf{x}') v(\mathbf{x}, \mathbf{x}') \cos(\theta) \cos(\theta') / \pi r^2 d\mathbf{x}'$$

► Discrétisation :

$$B_i = E_i + \rho_i \sum_j F_{ji} B_j A_j / A_i$$

- B_i, B_j sont les radiosités des patchs i et j (en W/m^2)
- E_i est le taux d'émission du patch i
- F_{ji} est le facteur de forme qui caractérise la proportion d'énergie quittant le patch j qui arrive sur le patch i
- A_i et A_j sont les surfaces des patchs i et j

Notation matricielle

- ▶ En regroupant tous les éléments :

$$\begin{pmatrix} \mathbf{B}_0 \\ \mathbf{B}_n \end{pmatrix} = \begin{pmatrix} \mathbf{E}_0 \\ \mathbf{E}_n \end{pmatrix} + \begin{pmatrix} \rho_i \mathbf{F}_{ji} \end{pmatrix} \begin{pmatrix} \mathbf{B}_0 \\ \mathbf{B}_n \end{pmatrix} \Leftrightarrow \mathbf{B} = \mathbf{E} + \mathbf{M}\mathbf{B}$$

- ▶ Équation matricielle à résoudre itérativement pour chaque longueur d'onde traitée
 - Méthodes de relaxations (*gathering* / *shooting*)

Facteur de forme

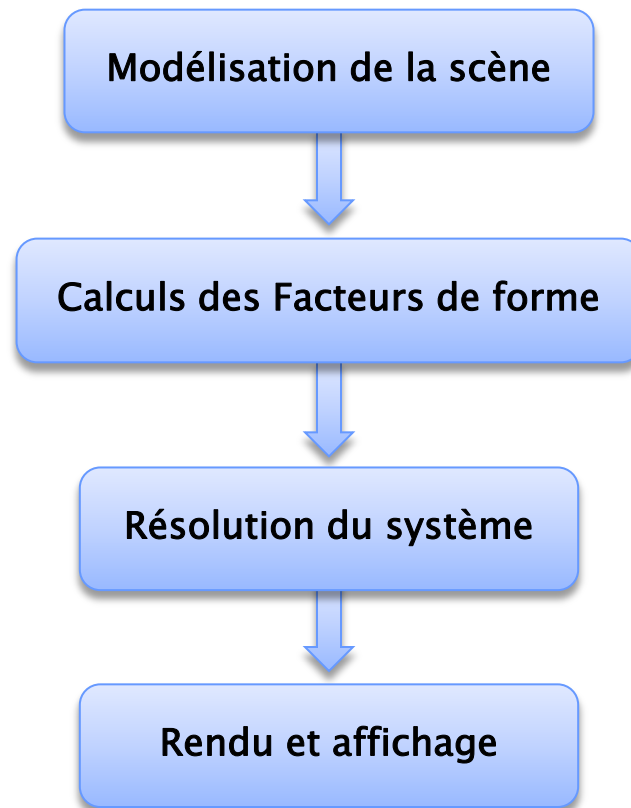
- ▶ Facteur de forme F_{ij} d'une surface A_i vers une surface A_j :

$$F_{ij} = \int_{A_i} \int_{A_j} v(\mathbf{x}, \mathbf{x}') \frac{\cos(\theta)\cos(\theta')}{\pi r^2} d\mathbf{x}d\mathbf{x}'$$

- ▶ **Problème** : réaliser cette intégration (intégrale quadruple) car pas de méthode analytique
 - ⇒ Solutions approximées : projection sur une hémisphère ou un hémicube.

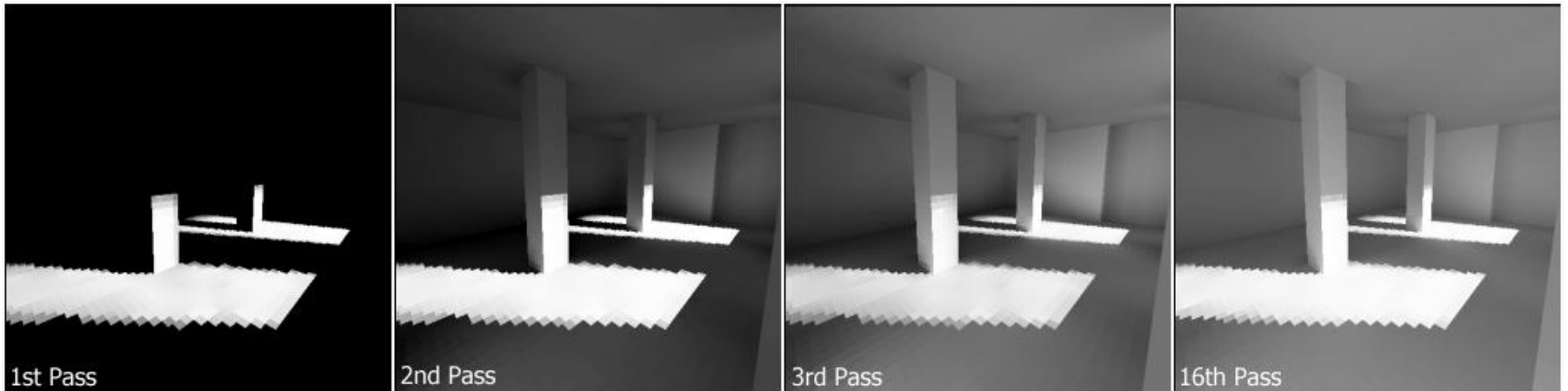
Résolution de la radiosité

- ▶ Calcul de l'illumination selon le pipeline :



Résolution de la radiosité

- ▶ Résolution itérative du système :

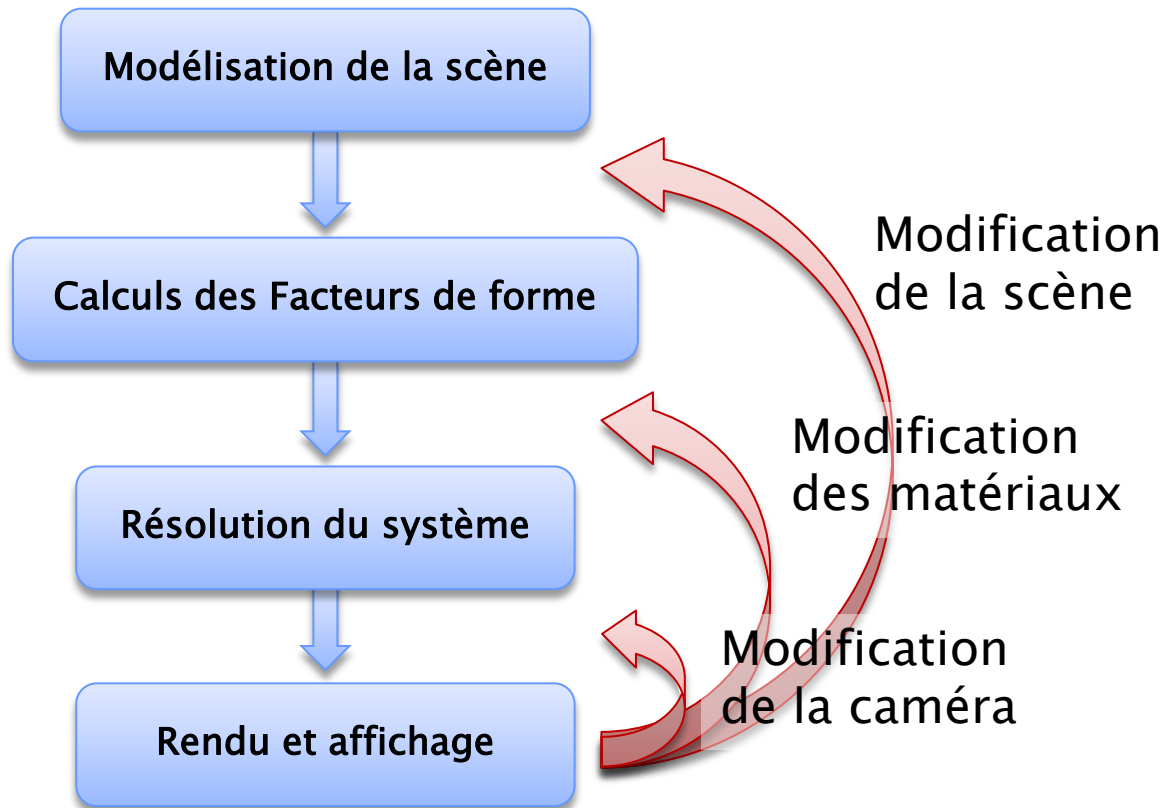


Question – 1 mn

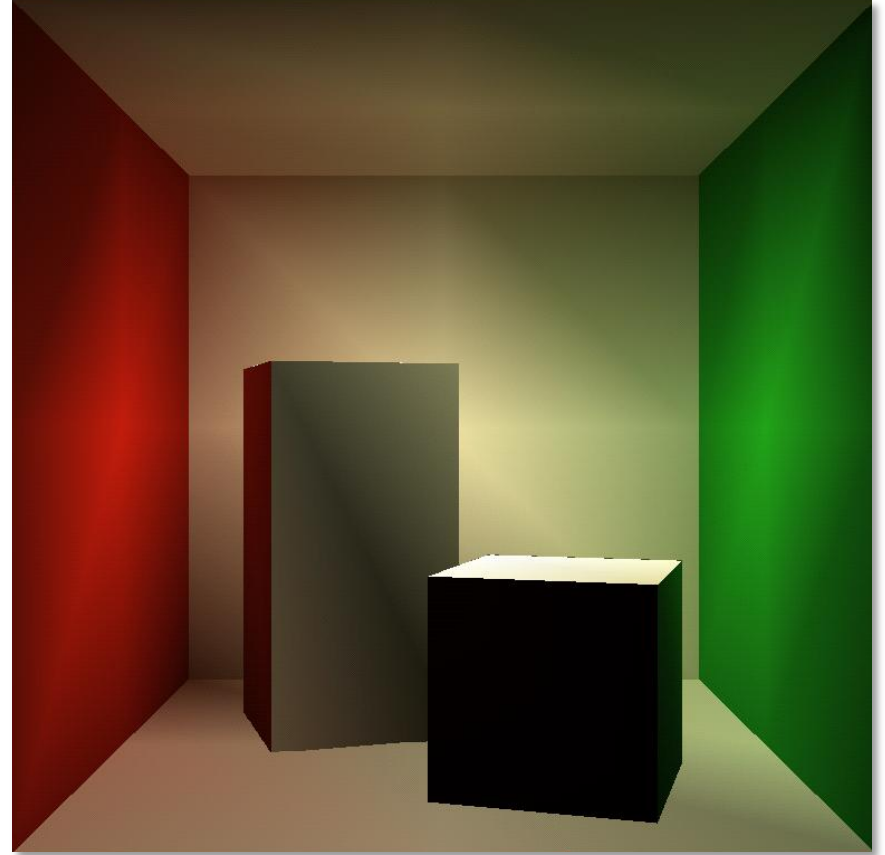
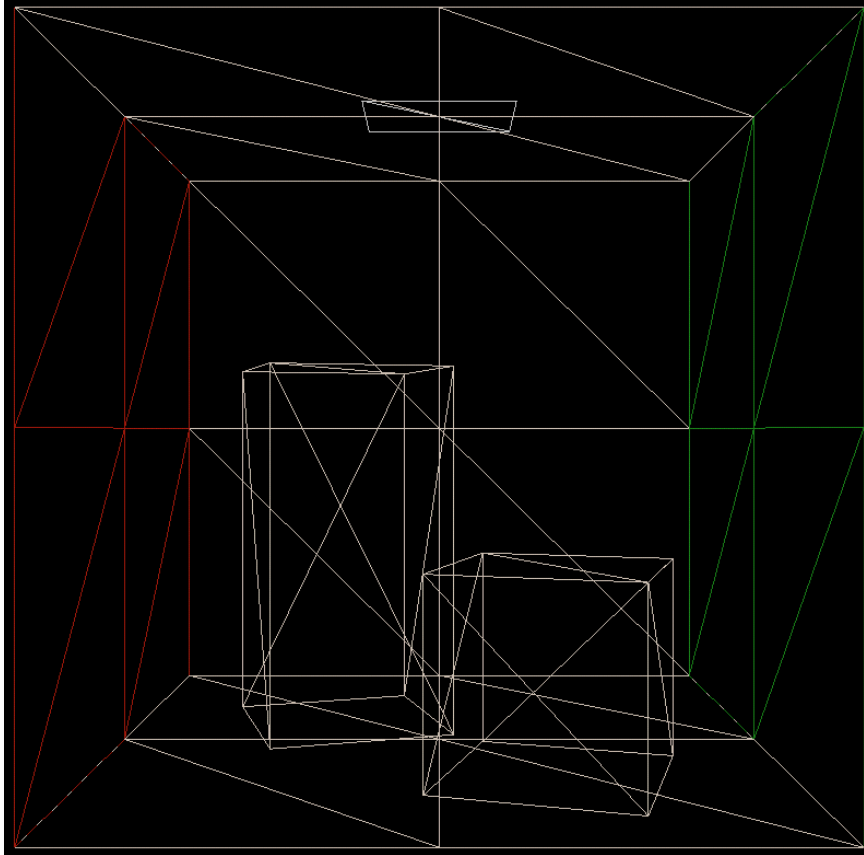
- ▶ Que faut-il recalculer quand quelque chose change dans la scène ?
 - Géométrie
 - Matériaux
 - Point de vue



Résolution de la radiosité



Solution de Radiosité



Solution de Radiosité



Museum simulation. Cornell University. 50,000 patches.

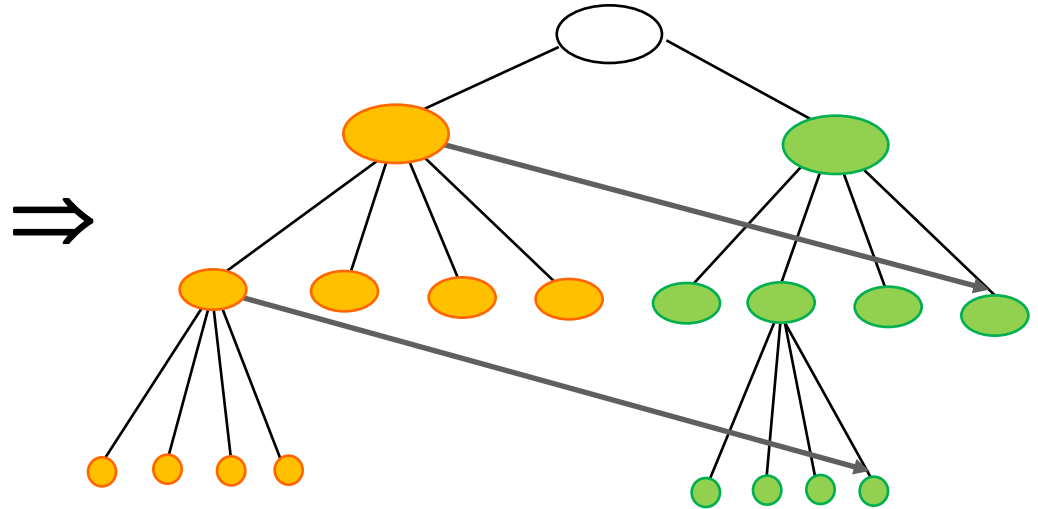
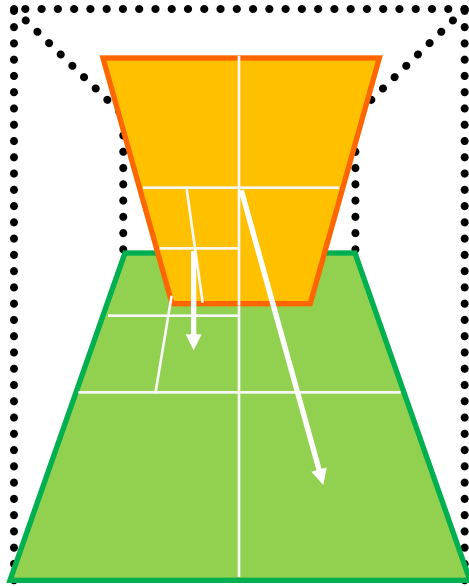
Radiosité : avantages

- ▶ Calcul indépendant du point de vue
- ▶ Adapté aux scènes complexes
- ▶ Partionnement des échanges lumineux
 - Mise à jour interactive

Radiosité : inconvénients

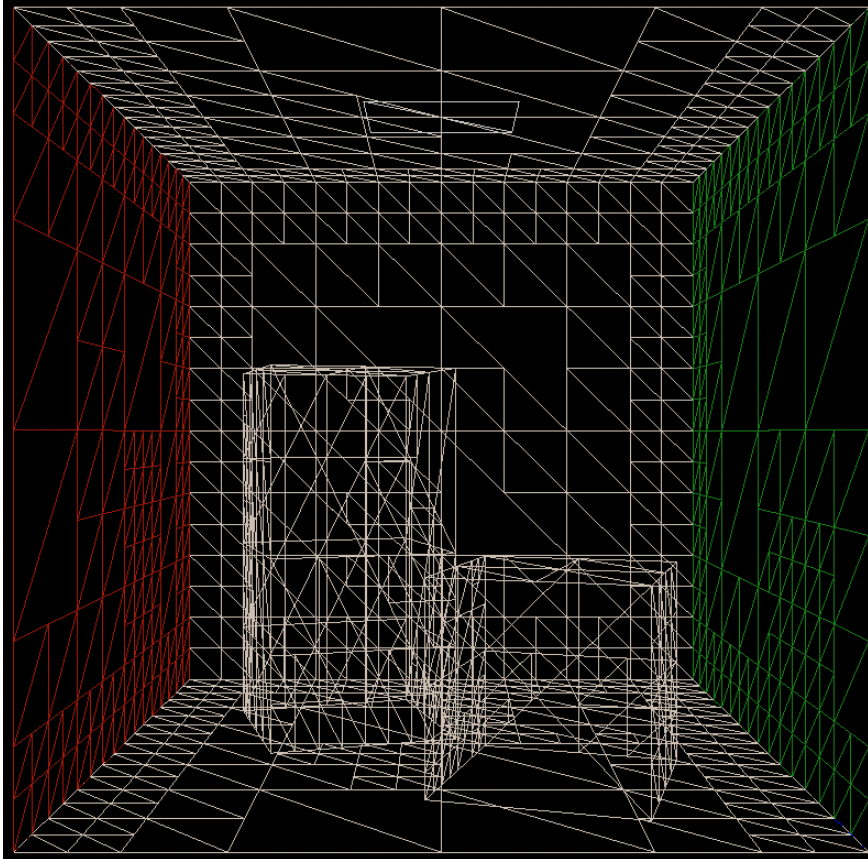
- ▶ Coût mémoire
- ▶ Diffus pur
 - « final gather » par Ray-Tracing
- ▶ Maillage
 - Maillage de discontinuité
- ▶ Pré-calculs très longs
 - Accélération possibles : radiosité hiérarchique

Radiosité hiérarchique [Hanrahan91]



- ▶ Calcul à différents niveau hiérarchique
- ▶ Regroupement

Radiosité hiérarchique



Radiosité hiérarchique



Maillage des discontinuités



Maillage des discontinuités



Maillage des discontinuités



Résultats et comparaison



Rendu Scan-line.
(3DS MAX)

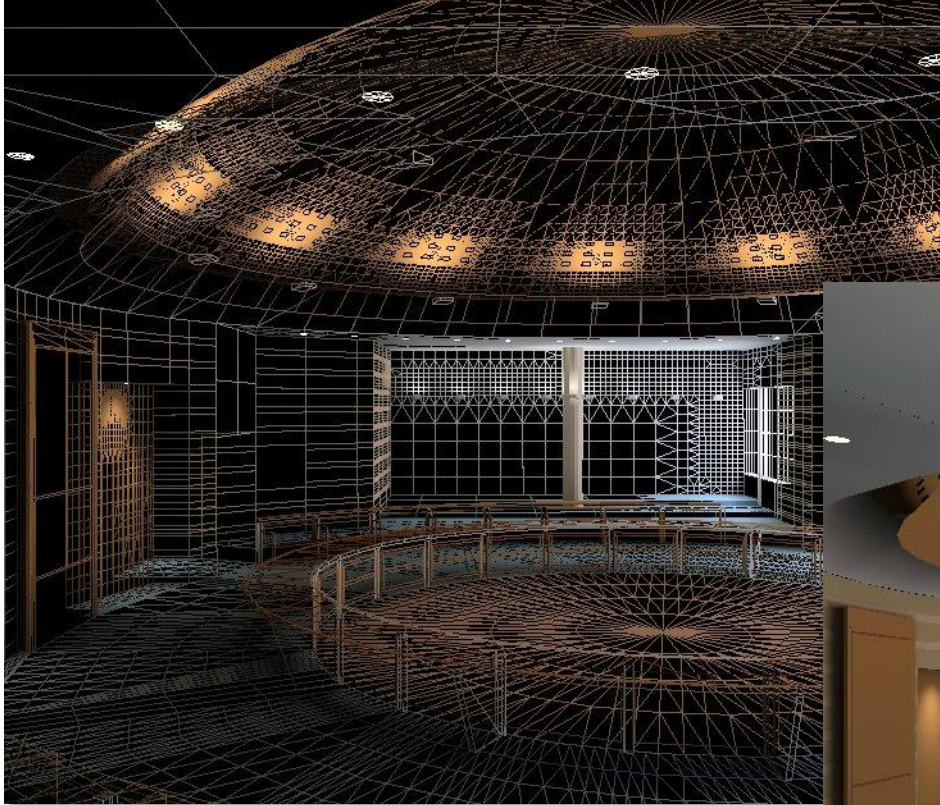
Rendu en radiosité et
lancer de rayons.



Résultats et comparaison



Résultats et comparaison



Résultats et comparaison



Résultats et comparaison



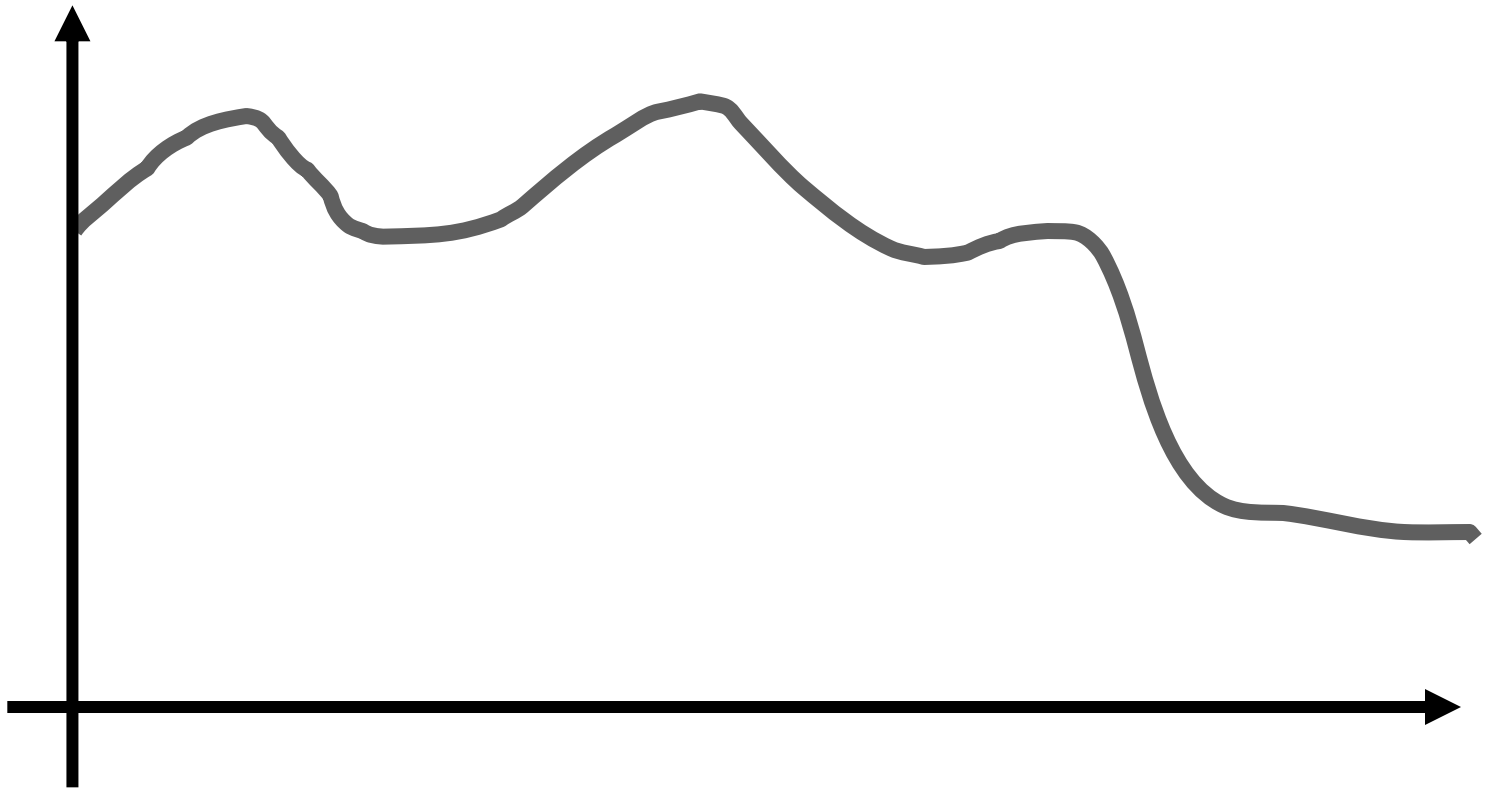
Monte Carlo

Intégration de Monte Carlo

- ▶ Méthode générique de calcul d'intégrales multi-dimensionnelles
- ▶ Intégration = problèmes récurrent en graphique :
 - Pixel : anti-aliasing
 - Sources de lumière : ombres douces
 - Lentille : profondeur de champ
 - Temps : flou cinétique
 - BRDF : réflexions glossy
 - Hémisphère : éclairage indirect

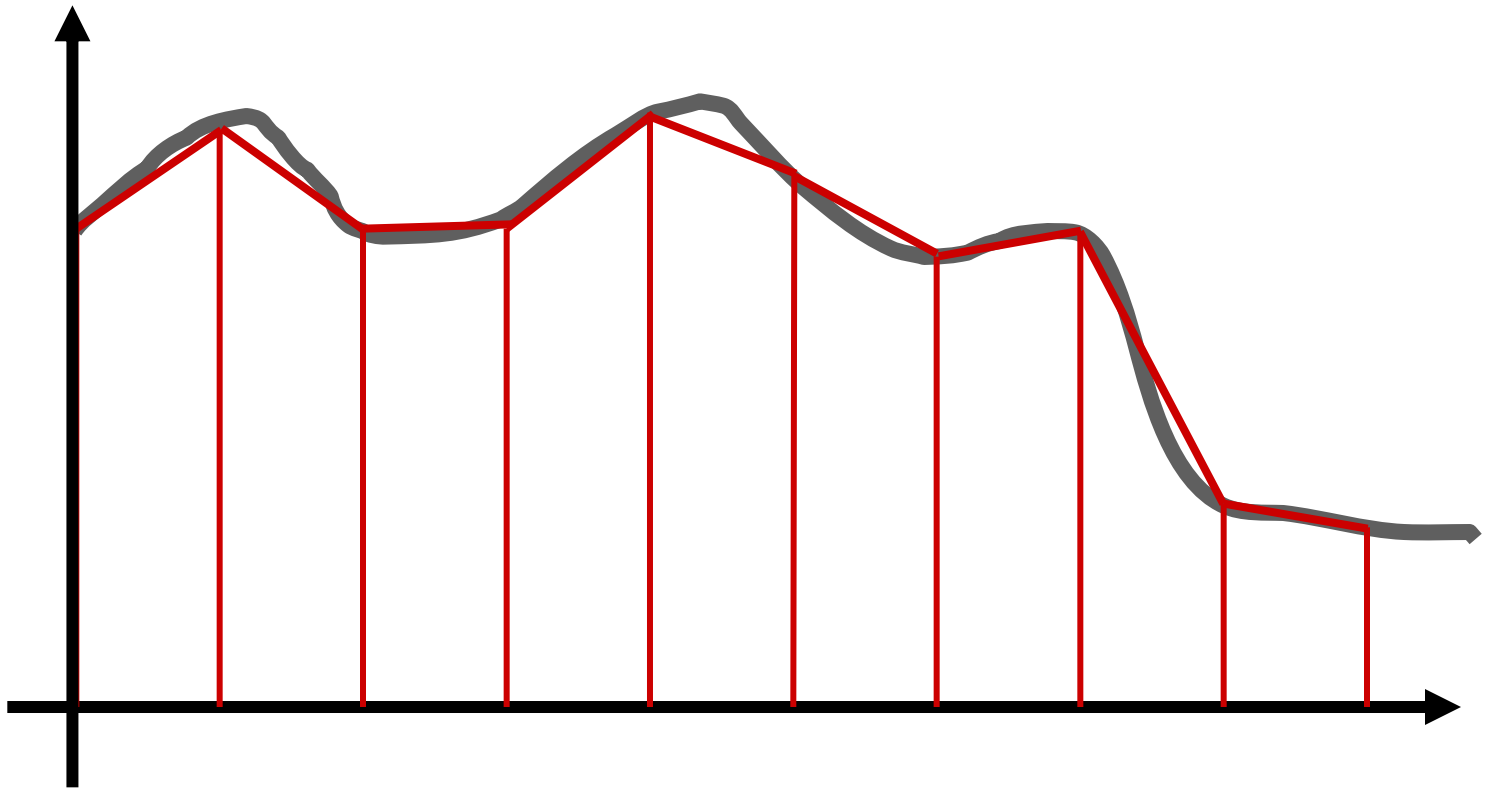
Integration en 1D

- ▶ Intégrale d'une fonction quelconque
- ▶ Problème continu \Rightarrow discrétisation



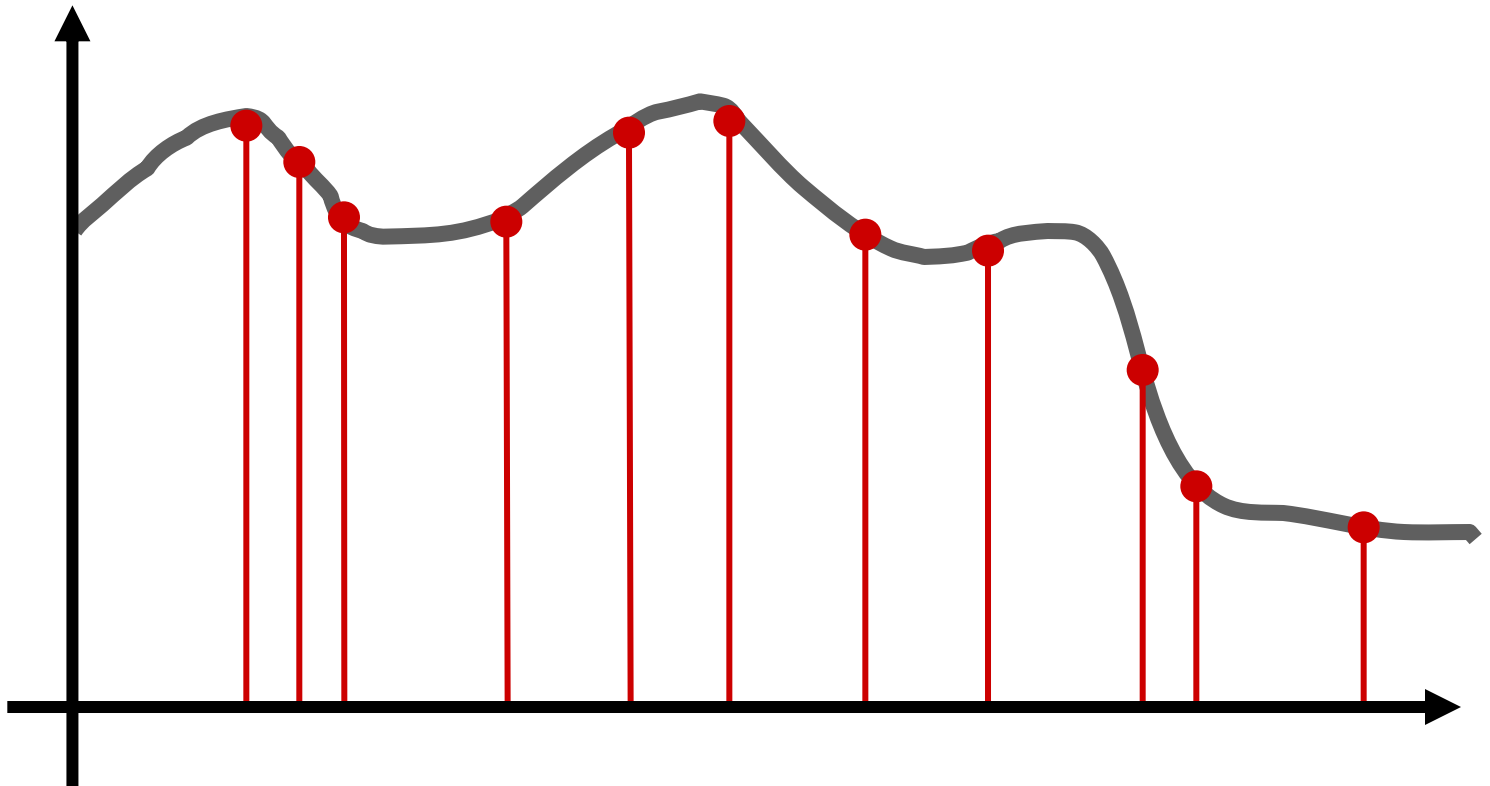
Integration en 1D

- ▶ Approximation par des trapèzes :
méthode de Simpson



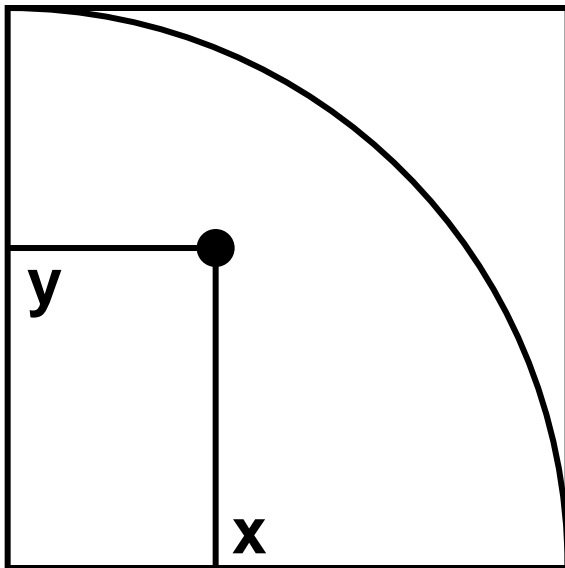
Integration 1D

- ▶ Monte Carlo : échantillonnage aléatoire
 - Inutile de mémoriser l'écart entre les n échantillons
 - Mais on espère qu'en moyenne il est de $1/n$



Monte Carlo : calcul de π

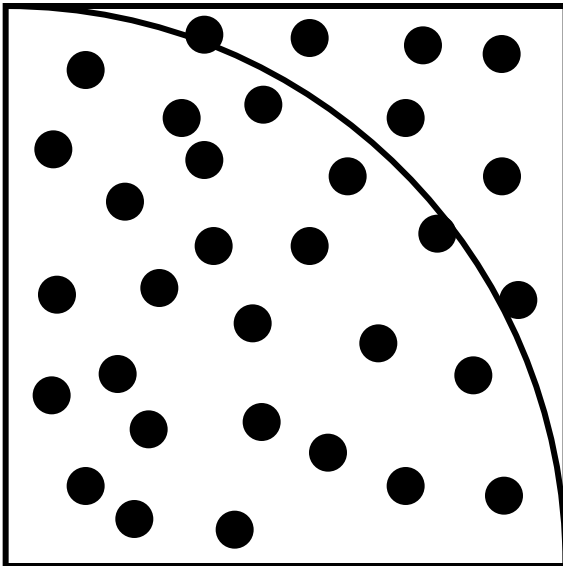
- ▶ Prenons un carré
- ▶ Prenons un point aléatoire (x,y) dans ce carré
- ▶ Testons s'il est dans le $\frac{1}{4}$ de disque ($x^2+y^2 < 1$)
- ▶ La probabilité est de $\pi / 4$



C'est l'intégral de la fonction valant 1 dans le cercle et 0 en dehors

Monte-Carlo computation of π

- ▶ La probabilité est $\pi / 4$
- ▶ Soit $n = \# \text{ points dedans} / \# \text{ points total}$
- ▶ $\pi \approx n * 4$
- ▶ L'erreur dépend du nombre d'essais



Pourquoi ne pas utiliser Simpson ?

- ▶ Pour calculer π , Monte Carlo n'est pas très efficace
- ▶ Mais la convergence est indépendante de la dimension
- ⇒ Intégration de Monte Carlo très efficace en grande dimensionnalité

Variables aléatoires continues

- ▶ Variable aléatoire réelle x
- ▶ Densité de probabilité : $p(x)$
 - probabilité que cette variable soit entre x et $x+dx$ est $p(x) dx$

Espérance

$$E[x] = \int_{-\infty}^{\infty} xp(x)dx$$

$$E[f(x)] = \int_{-\infty}^{\infty} f(x)p(x)dx$$

► L'espérance est linéaire :

$$E[f_1(x) + a f_2(x)] = E[f_1(x)] + a E[f_2(x)]$$

Intégration de Monte Carlo

- ▶ Soit la fonction $f(x)$ avec x dans $[a, b]$
- ▶ Calculons :
$$I = \int_a^b f(x) dx$$
- ▶ Considérons une variable aléatoire x
- ▶ Si x a une distribution uniforme, $I = E[f(x)]$
 - par définition de l'espérance

Somme de variables aléatoires

- ▶ Soit N variables indépendantes identiquement distribuées (IID) x_i (N échantillons)
 - de même probabilité (ici uniforme)
- ▶ Définissons : $F_N = \frac{1}{N} \sum_{j=1}^n f(x_j)$ **estimateur de Monte Carlo**
- ▶ Par linéarité de l'espérance :
 $E[F_N] = E[f(x)]$

Variance

$$\sigma^2 = E[(x - E[x])^2] = \int_{-\infty}^{\infty} (x - E[x])^2 p(x) dx$$

- ▶ Mesure de l'écart à l'espérance
- ▶ Déviation standard σ : racine de la variance
- ▶ Propriétés :
 - $\sigma^2[x+y] = \sigma^2[x] + \sigma^2[y] + 2 \text{Cov}[x,y]$
 - $\sigma^2[ax] = a^2 \sigma^2[x]$

Étude de la variance

$$\sigma^2[F_N] = \sigma^2 \left[\sum_{j=1}^n \frac{f(x_j)}{N} \right]$$

- ▶ Variables indépendantes $\Leftrightarrow \text{Cov}[x_i, x_j]=0$ si $i \neq j$

$$\sigma^2[F_N] = \frac{\sigma^2[f(x)]}{N}$$

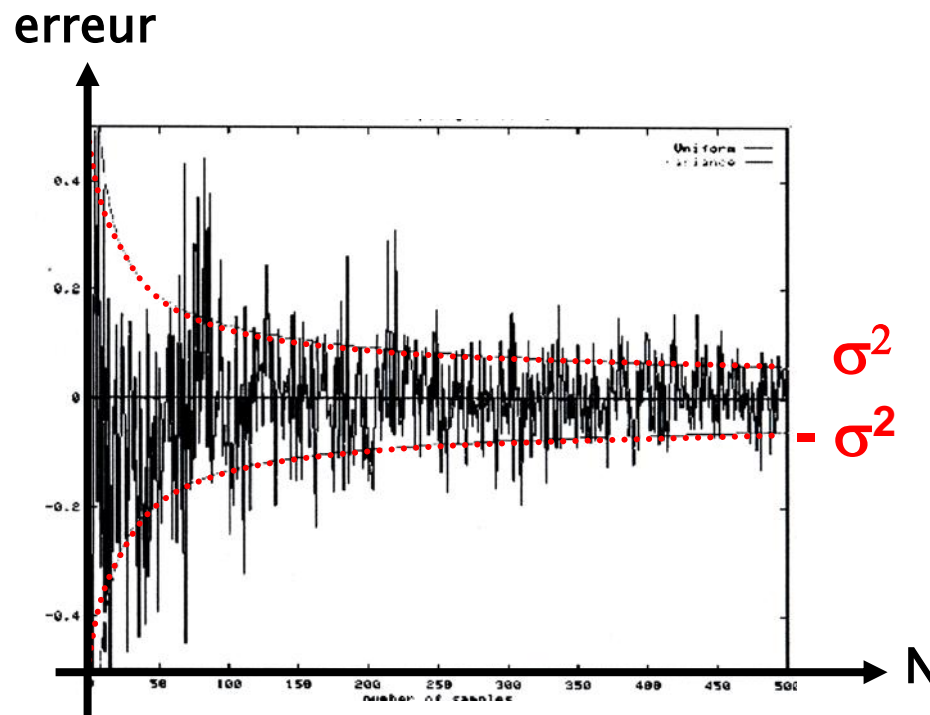
- ▶ Donc σ (erreur) diminue en
- \Rightarrow **convergence lente**

$$\sqrt{N}$$

Exemple

$$I = \int_0^1 5x^4 dx$$

- ▶ En théorie, $I=1.0$
- ▶ En pratique, avec une distribution uniforme



Intégration de MC : avantages

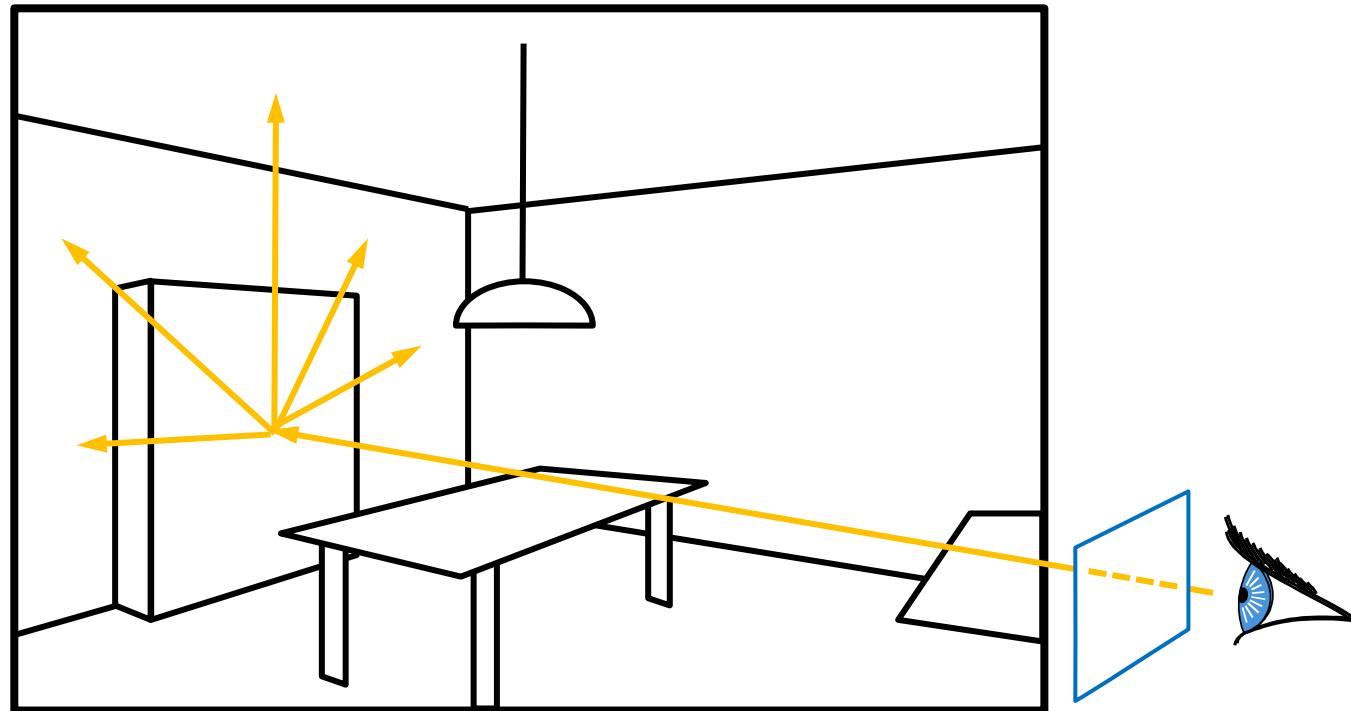
- ▶ Peu de restriction sur la fonction à intégrer
 - Pas de problème de continuité, régularité...
 - Nécessite seulement une évaluation ponctuelle
- ▶ Même convergence en dimension supérieure
- ▶ Conceptuellement simple

Intégration de MC : inconvénients

- ▶ Bruité
- ▶ Convergence lente
- ▶ Implémentation efficace difficile

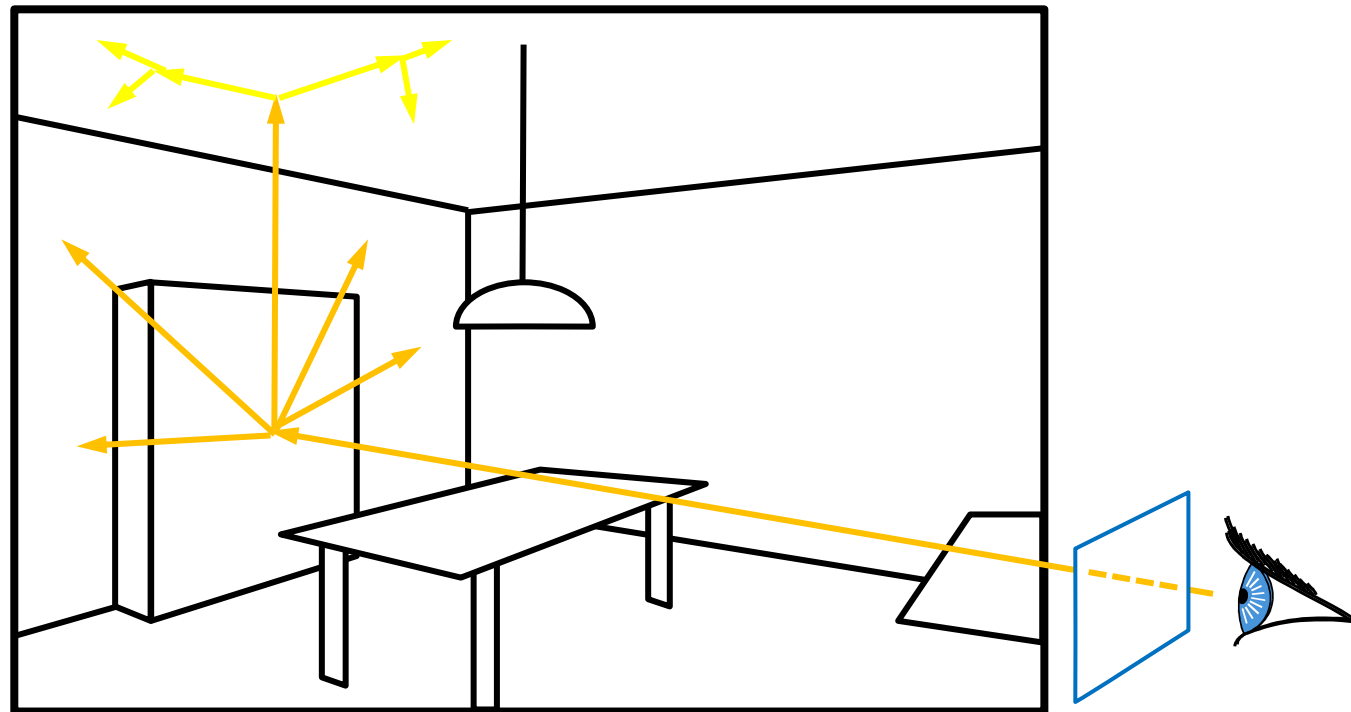
Méthodes de Monte Carlo

- ▶ Un rayon par pixel
- ▶ En chaque point visible : tirage aléatoire de rayons pour accumuler la radiance



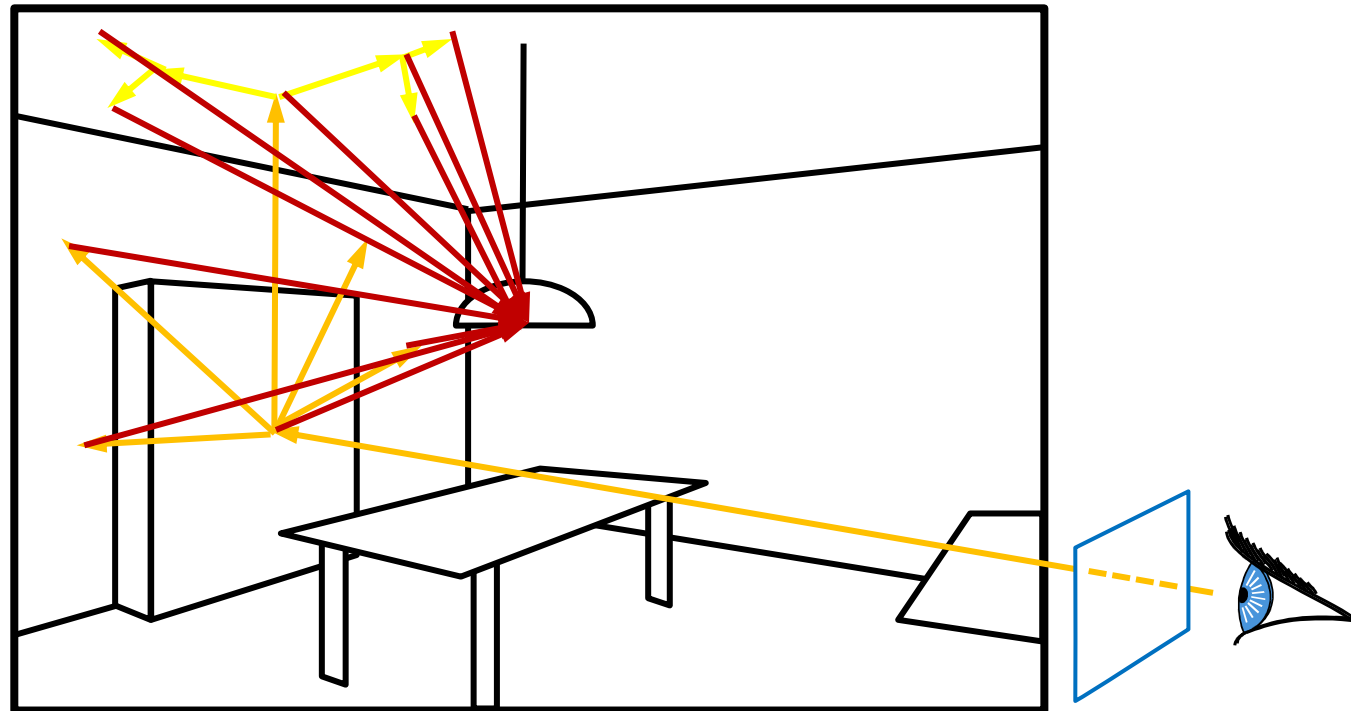
Méthodes de Monte Carlo

- ▶ Un rayon par pixel
- ▶ En chaque point visible : tirage aléatoire de rayons pour accumuler la radiance
- ▶ Continuer récursivement

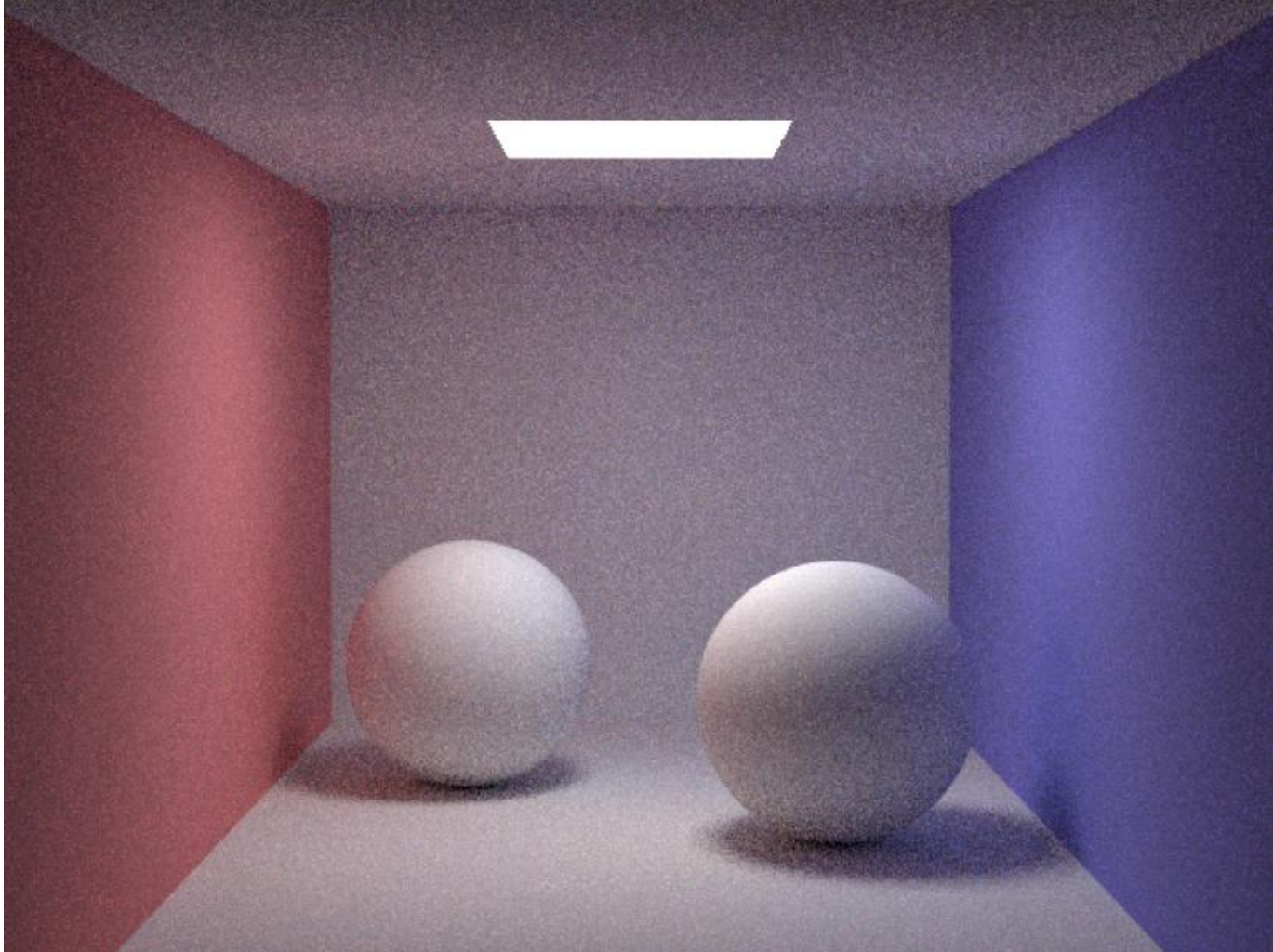


Méthodes de Monte Carlo

- ▶ Un rayon par pixel
- ▶ En chaque point visible : tirage aléatoire de rayons pour accumuler la radiance
- ▶ Continuer récursivement
- ▶ Échantillonner la lampe systématiquement

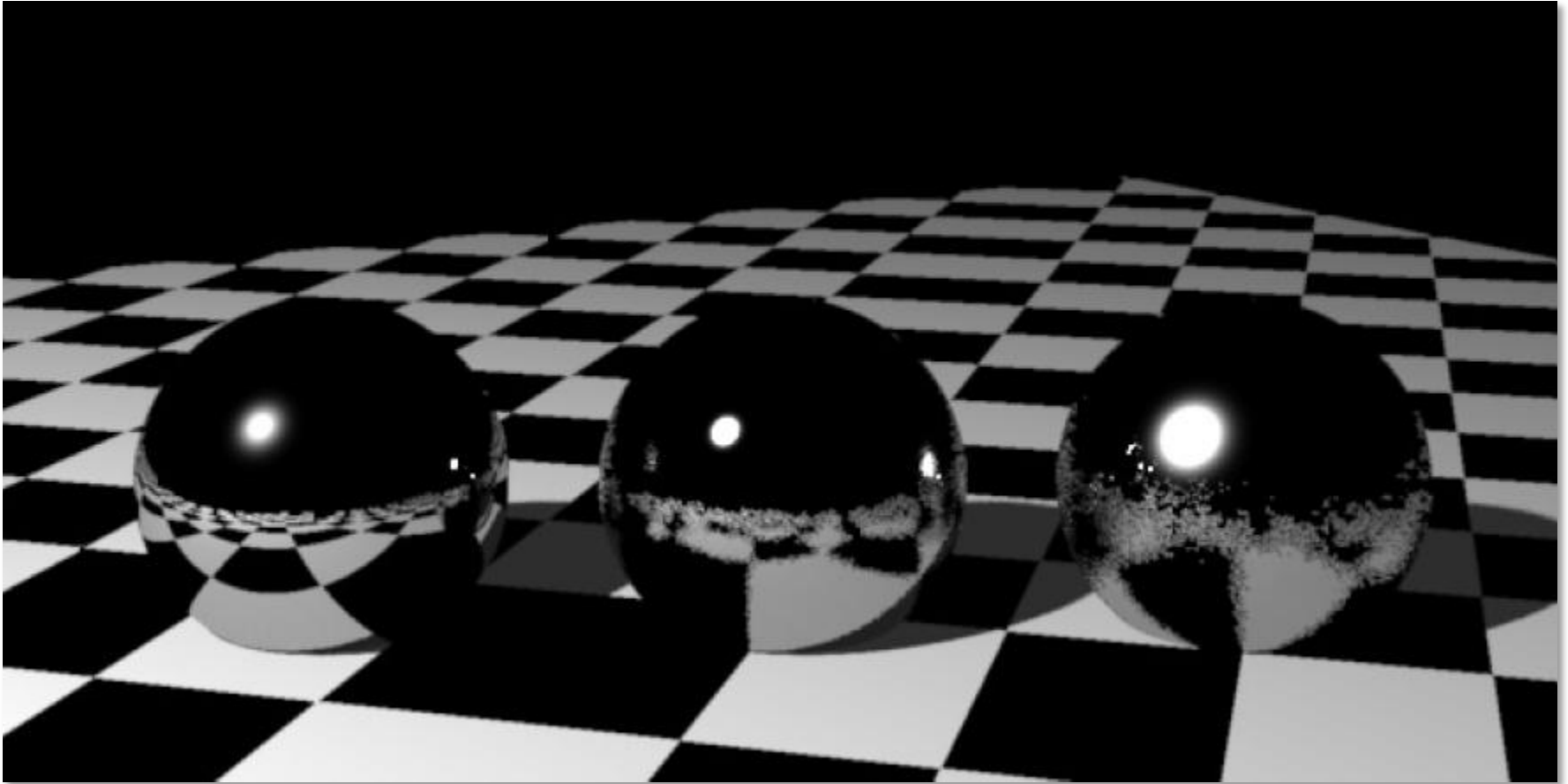


Résultats



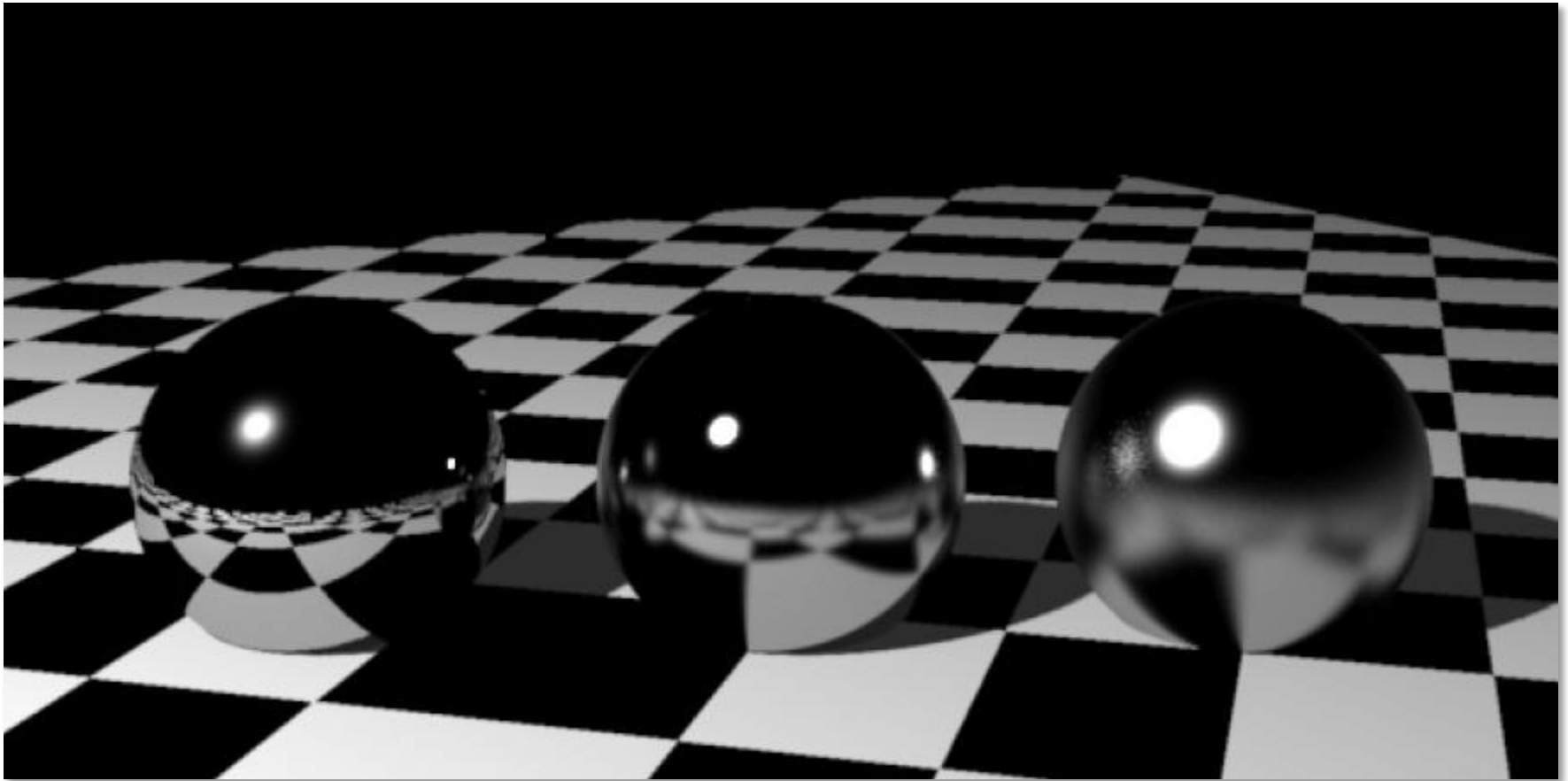
Résultats

- ▶ 1 échantillon par pixel



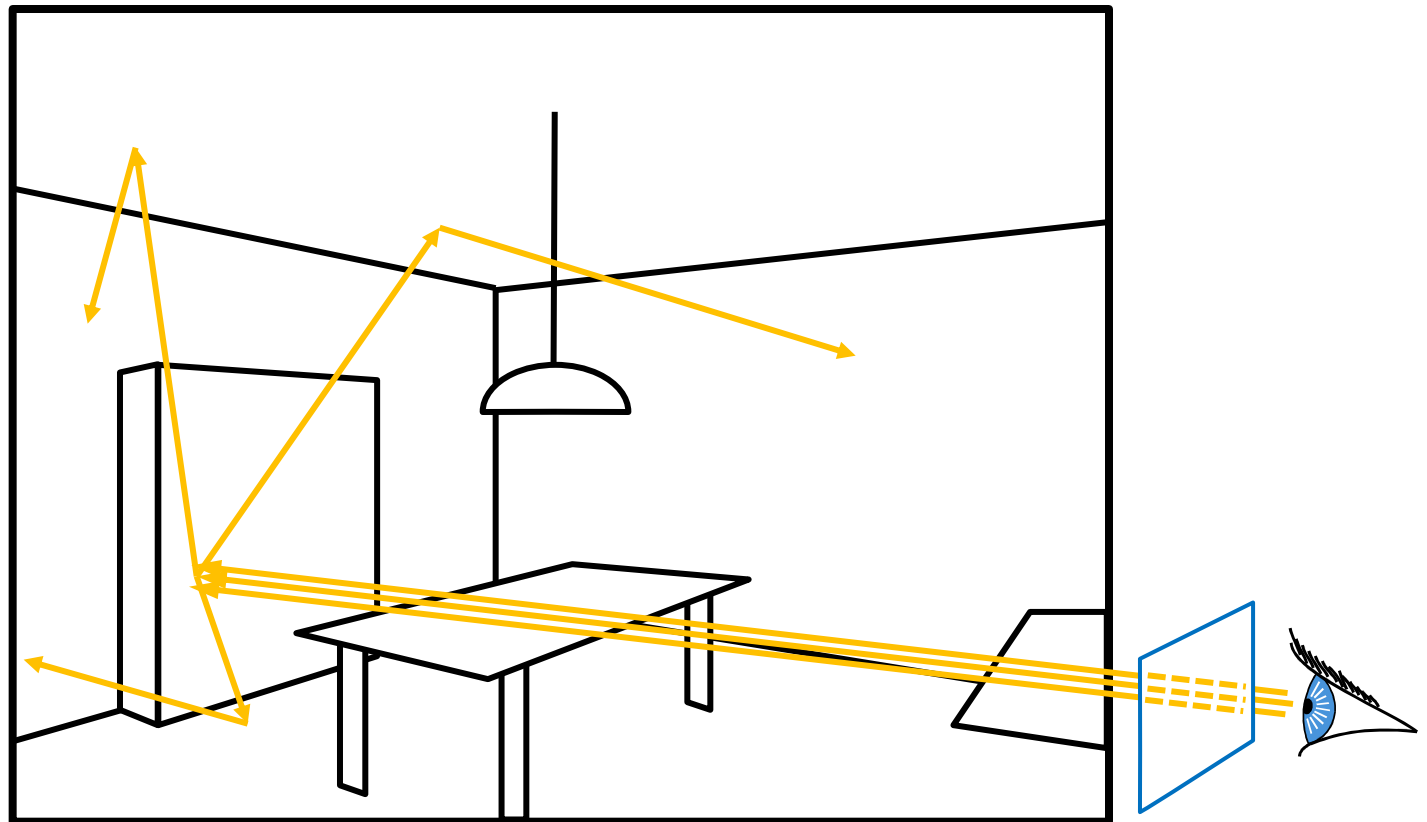
Résultats

- ▶ 256 échantillons par pixel



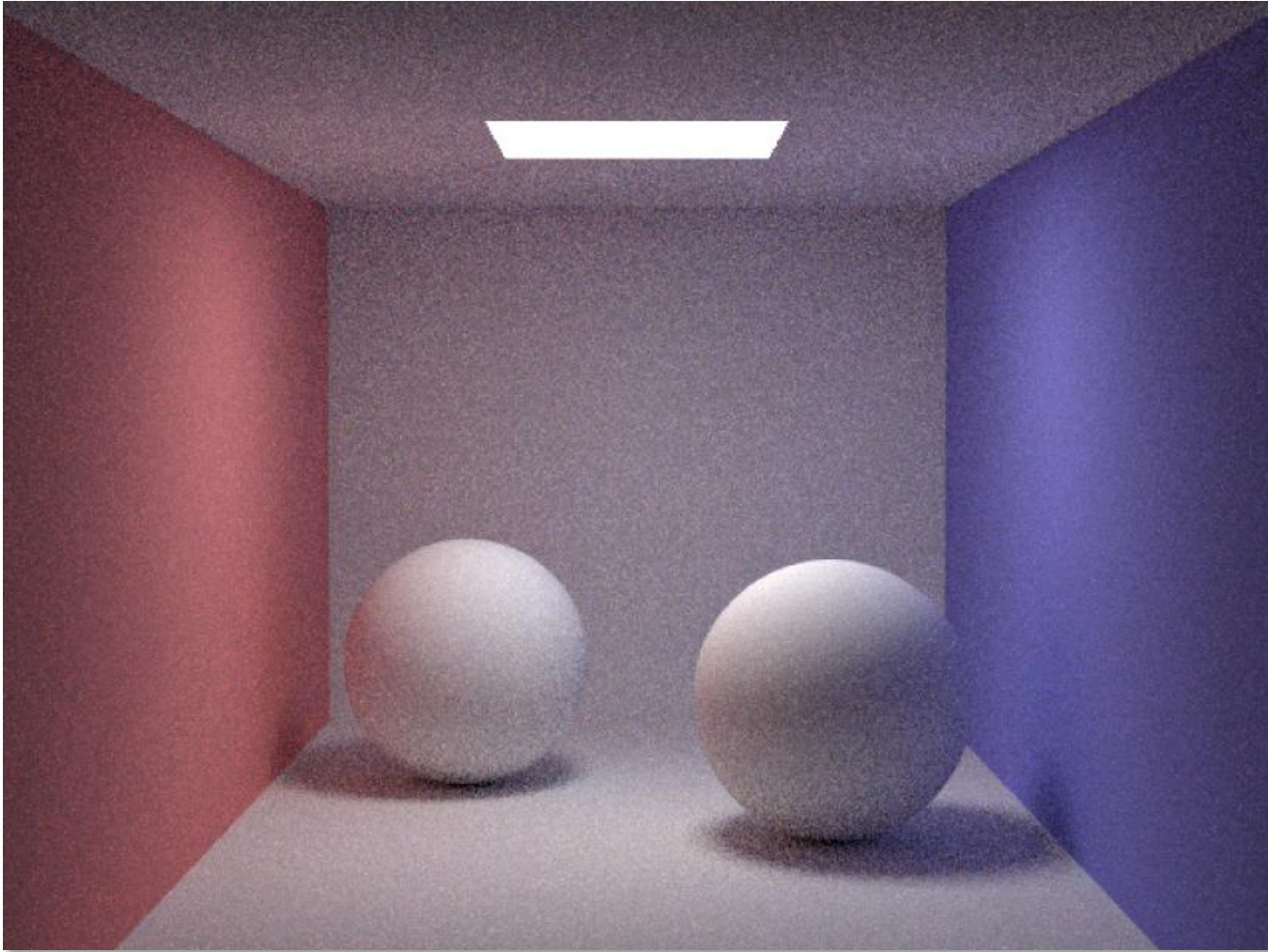
Monte Carlo Path Tracing

- ▶ Tracer seulement un rayon par récursion
- ▶ Mais lancer plusieurs rayons primaires par pixel



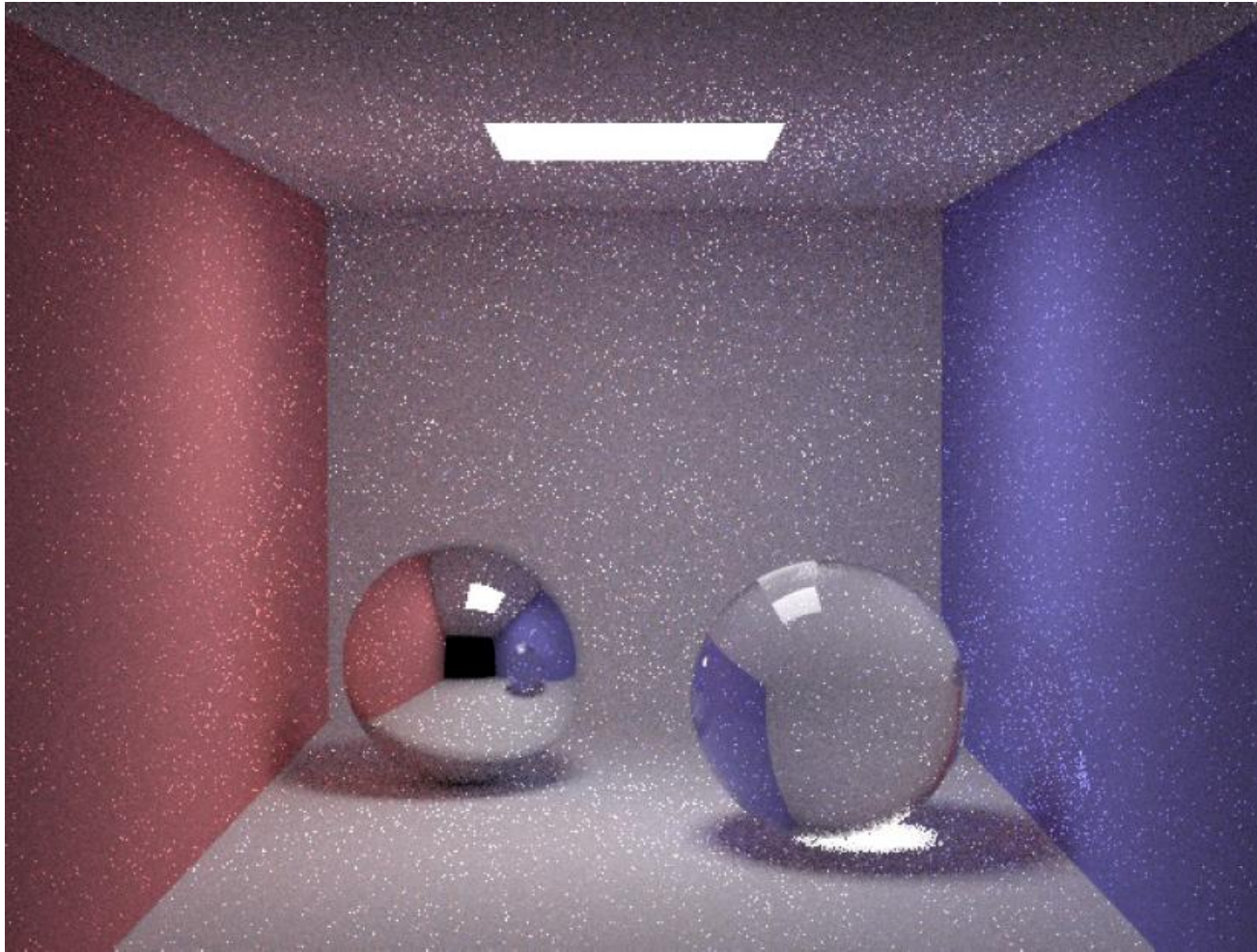
Résultats

- ▶ 10 paths/pixel



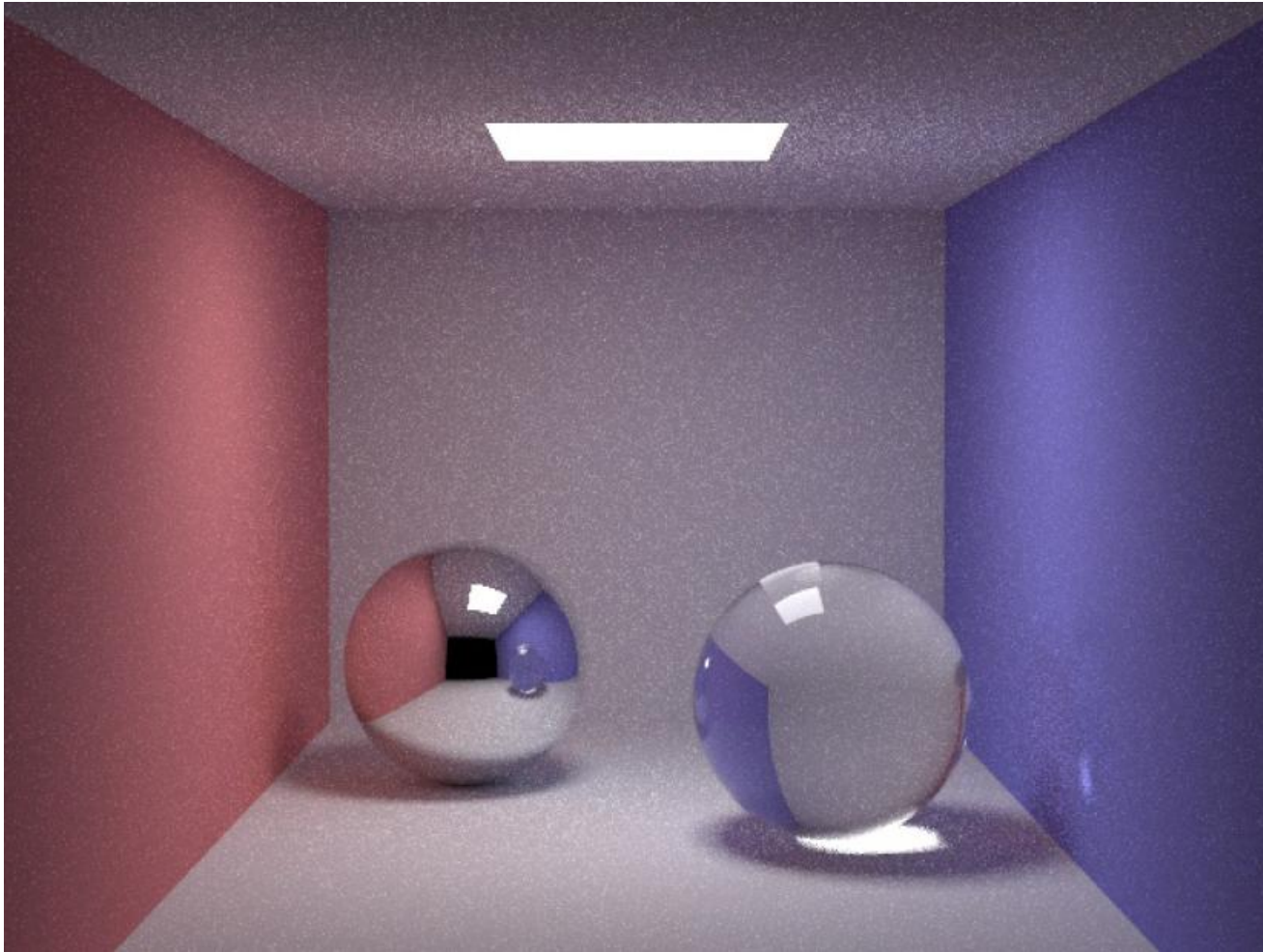
Résultats : matériaux brillant

- ▶ 10 paths/pixel



Résultats : matériaux brillant

- ▶ 100 paths/pixel



Pourquoi l'aléatoire ?

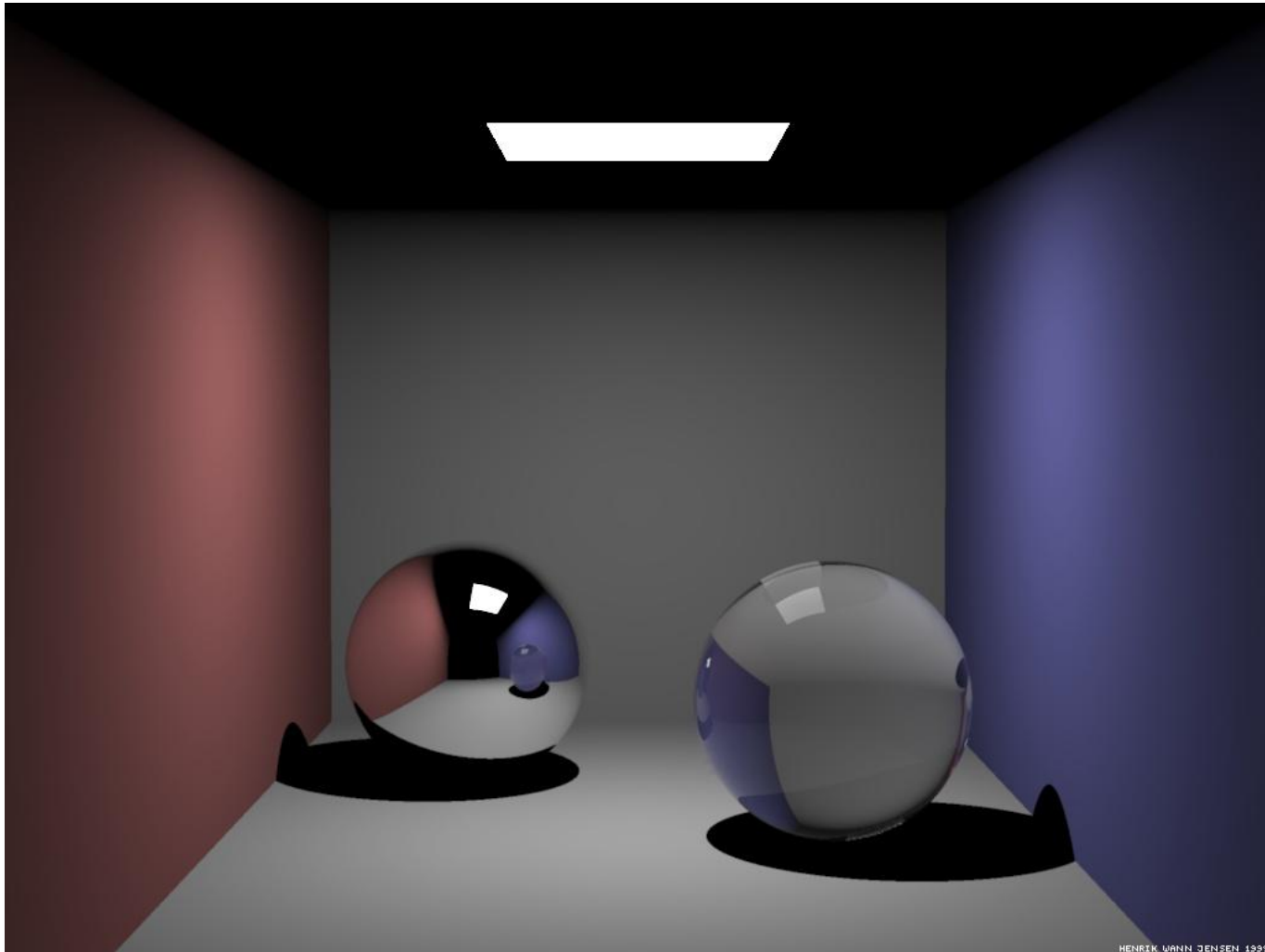
- ▶ Séquence aléatoire fixe
- ▶ La structure apparait dans l'erreur



Résumé

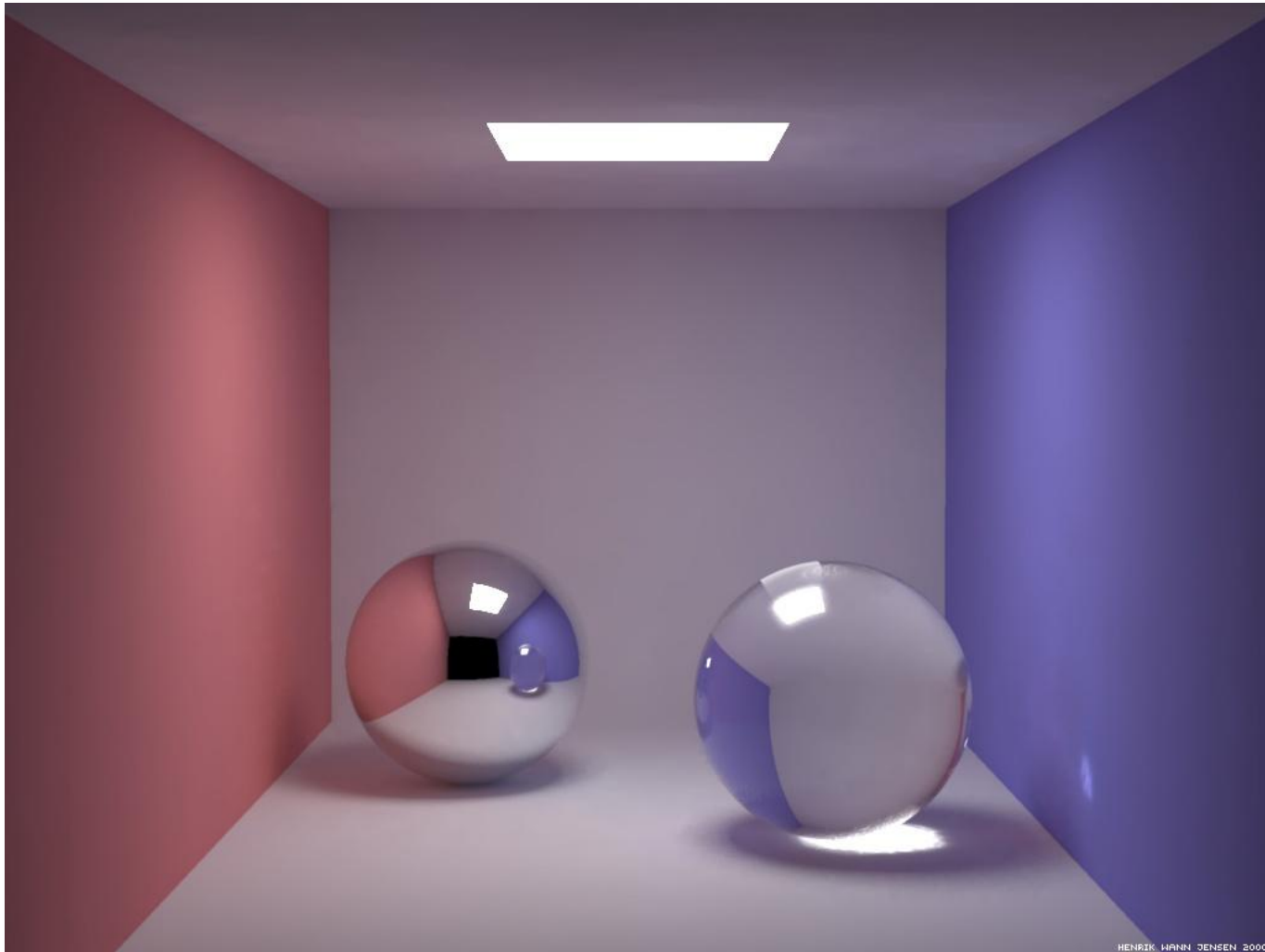
- ▶ Envoyer des rayons aléatoires
- ▶ Échantillonner l'équation de rendu
- ▶ Pas de maillage nécessaire ni de stockage supplémentaire
- ▶ Pas de limitation
 - Ni sur la réflectance
 - Ni sur la géométrie
- ▶ Extrêmement flexible
- ▶ Peut être bruité et/ou très lent
 - Accélération possible : « irradiance caching »

Illumination directe

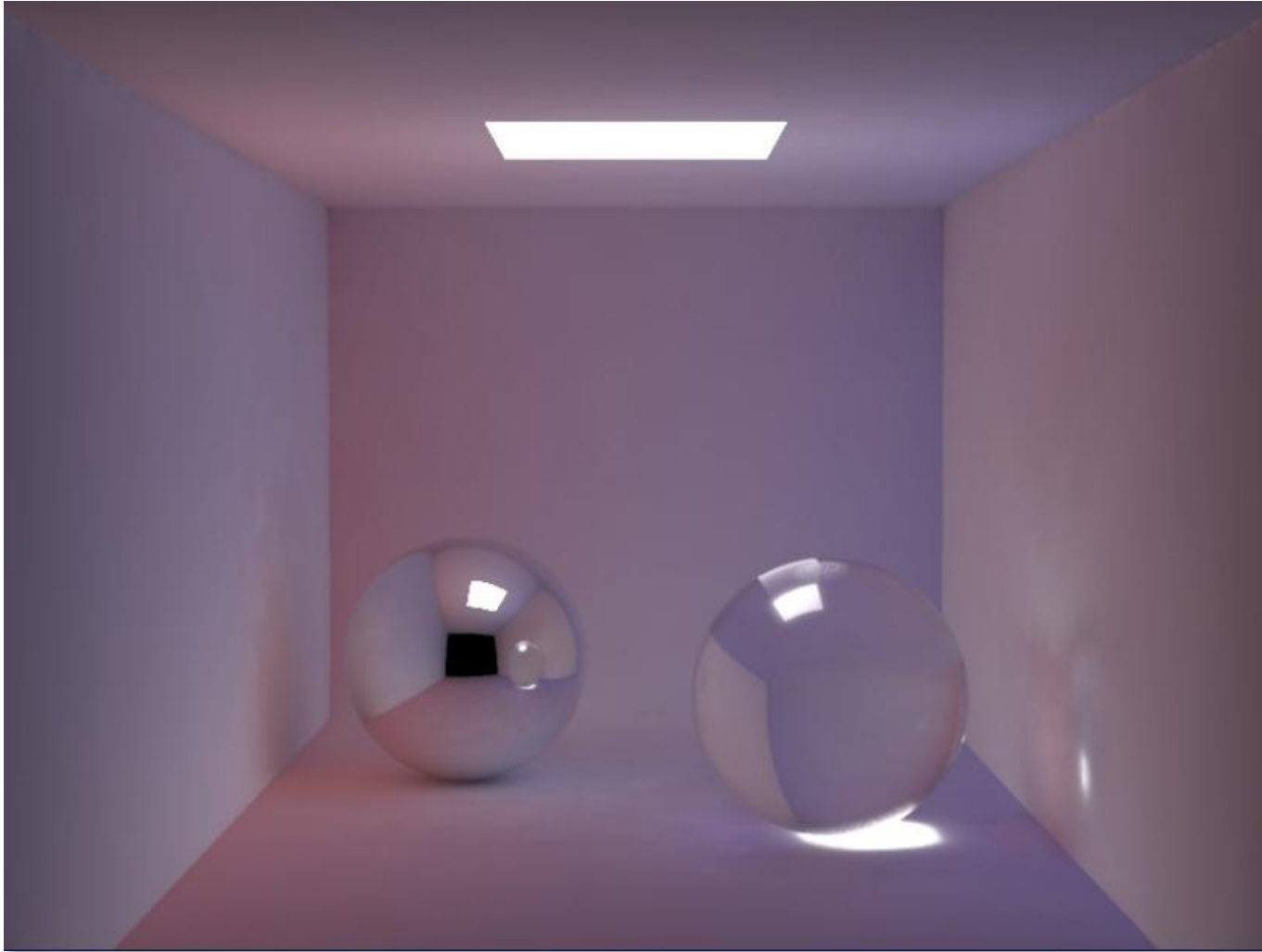


HENRIK WANN JENSEN 1999

Illumination globale

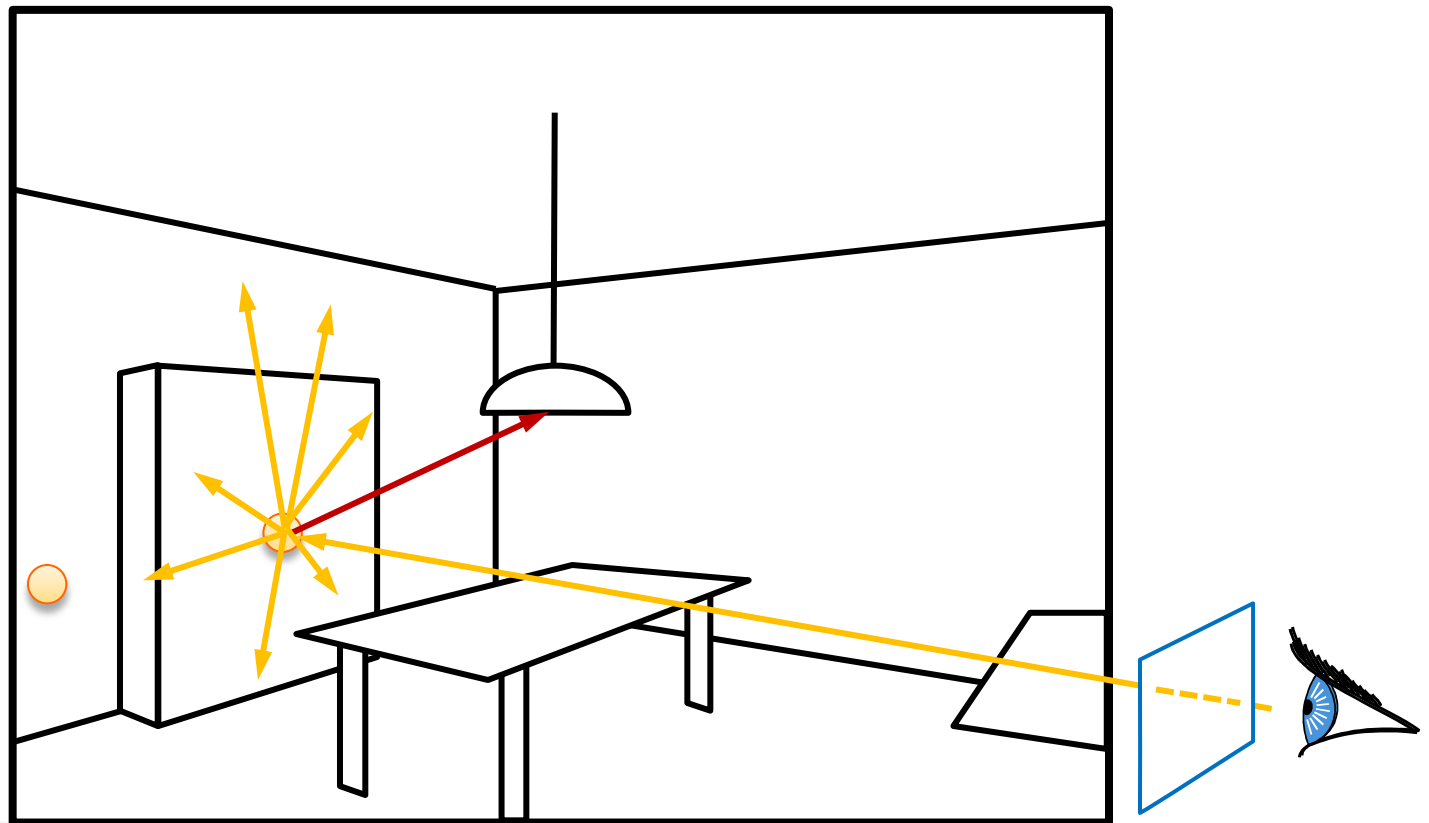


Illumination indirecte



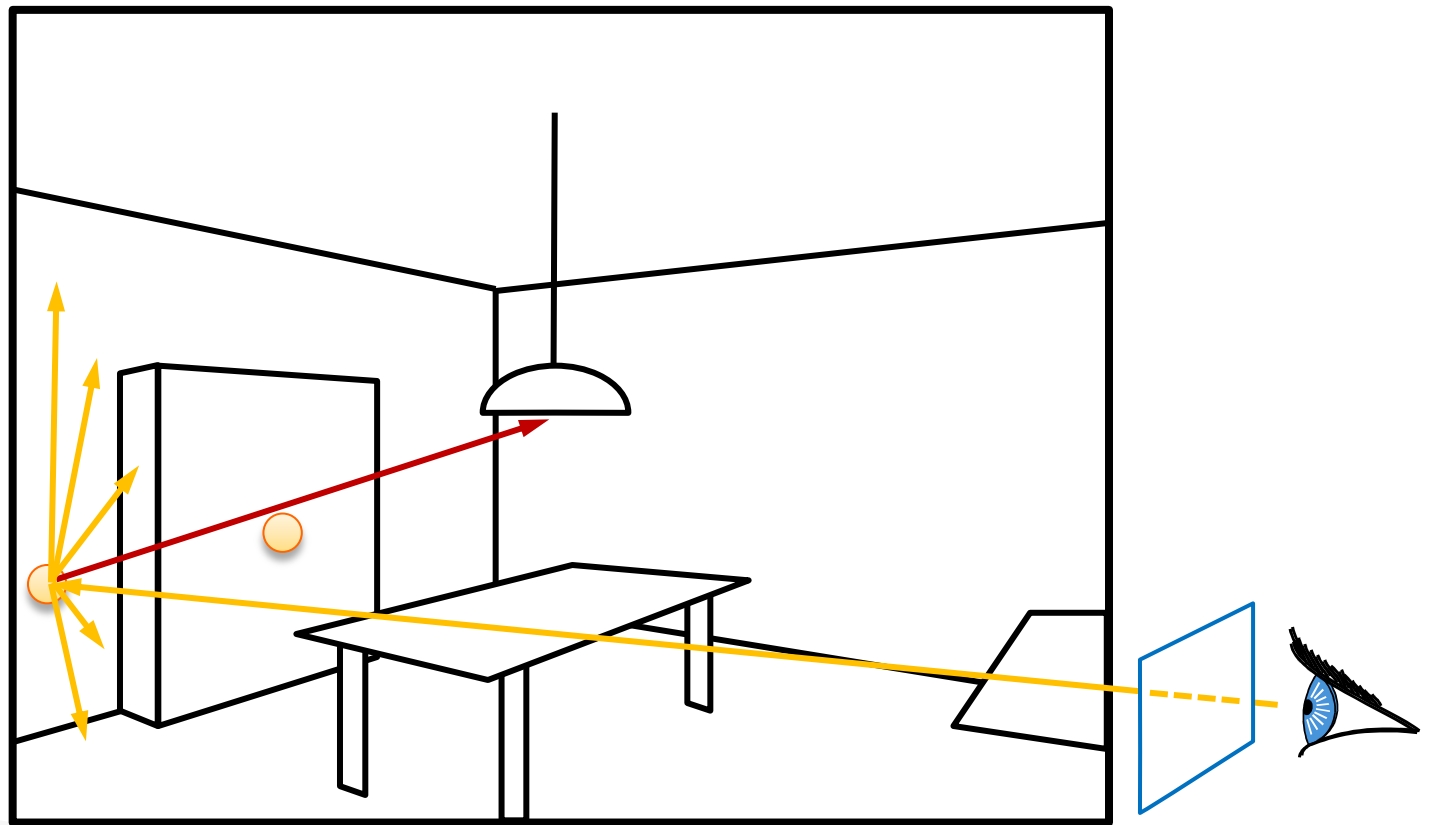
Irradiance cache

- ▶ L'illumination indirecte varie spatialement lentement



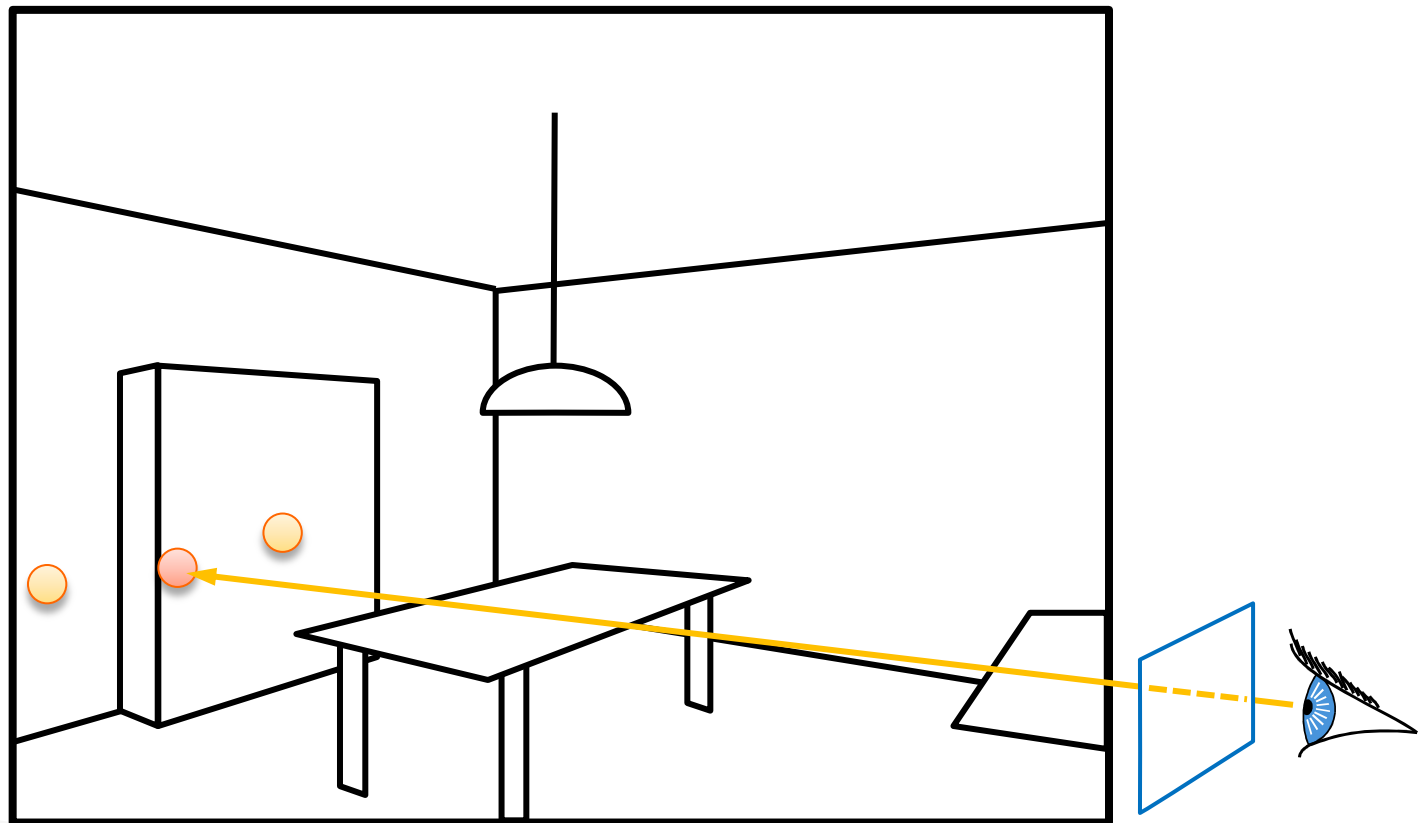
Irradiance cache

- ▶ L'illumination indirecte varie spatialement lentement



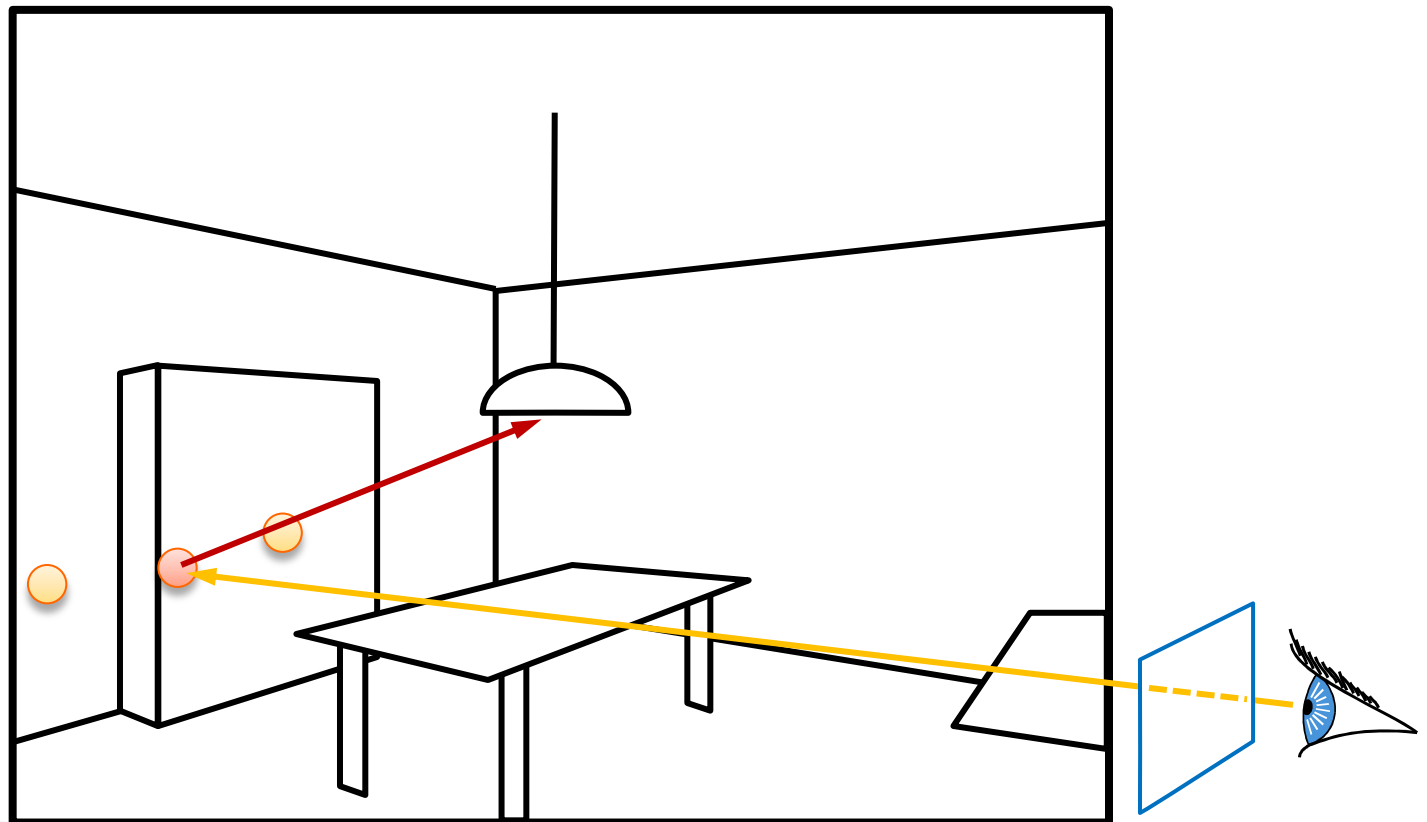
Irradiance cache

- ▶ L'illumination indirecte varie lentement
- ▶ Interpoler entre des valeurs proches



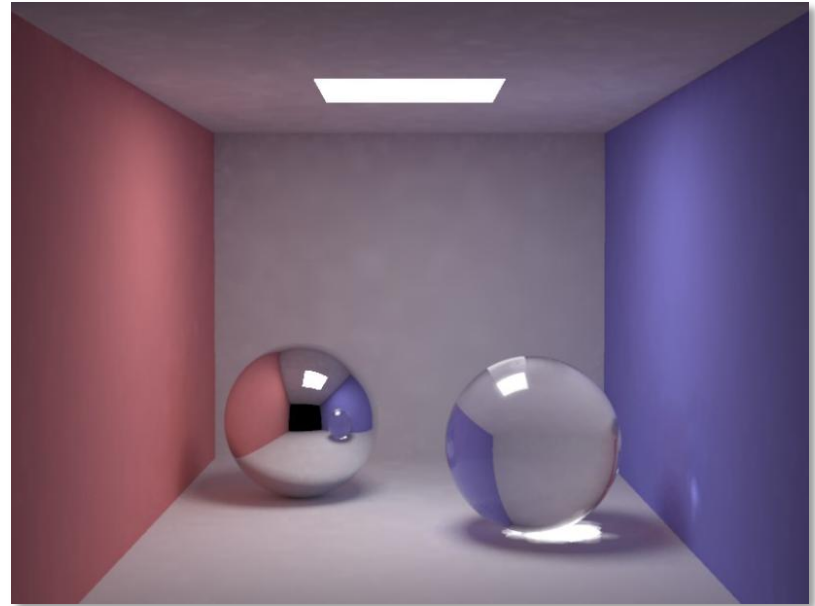
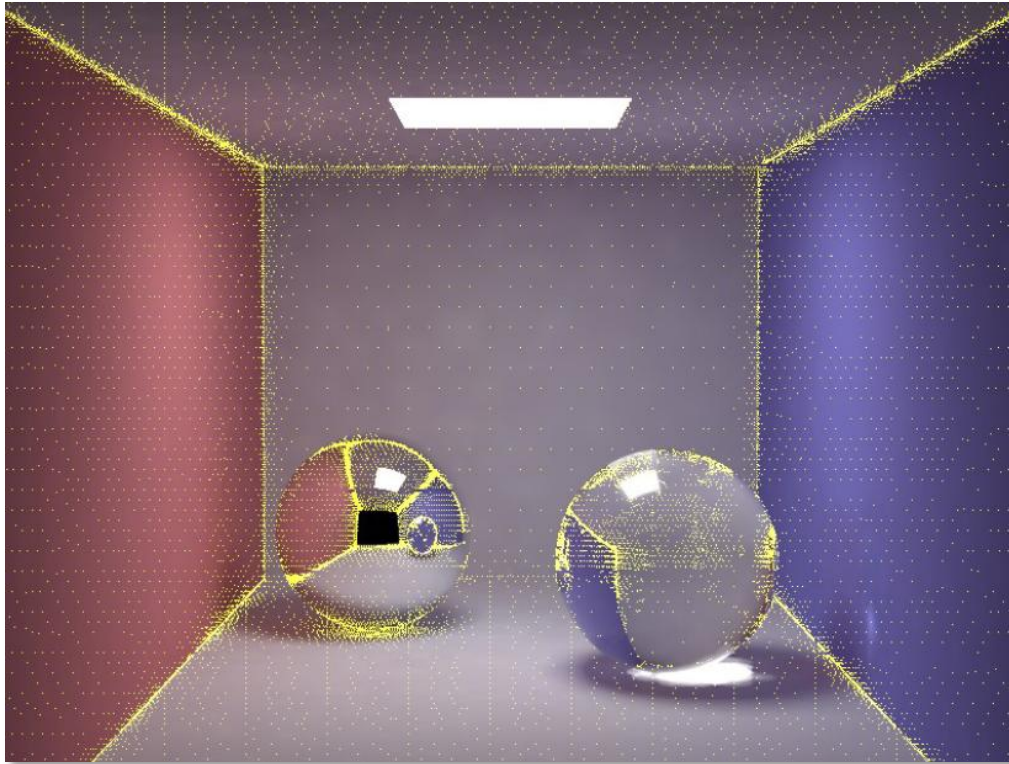
Irradiance cache

- ▶ L'illumination indirecte varie lentement
- ▶ Interpoler entre des valeurs du cache
- ▶ Mais calculer l'éclairage directe complètement



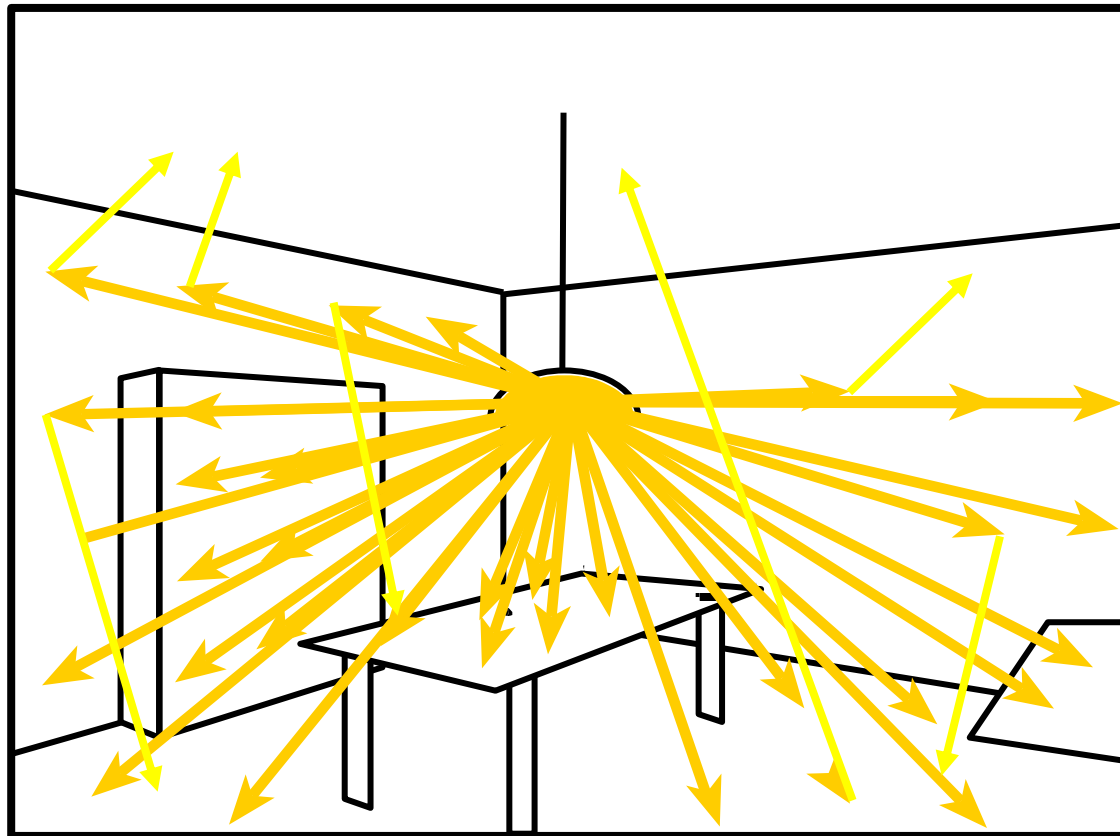
Irradiance caching

- ▶ Points jaunes : contribution de l'illumination indirecte



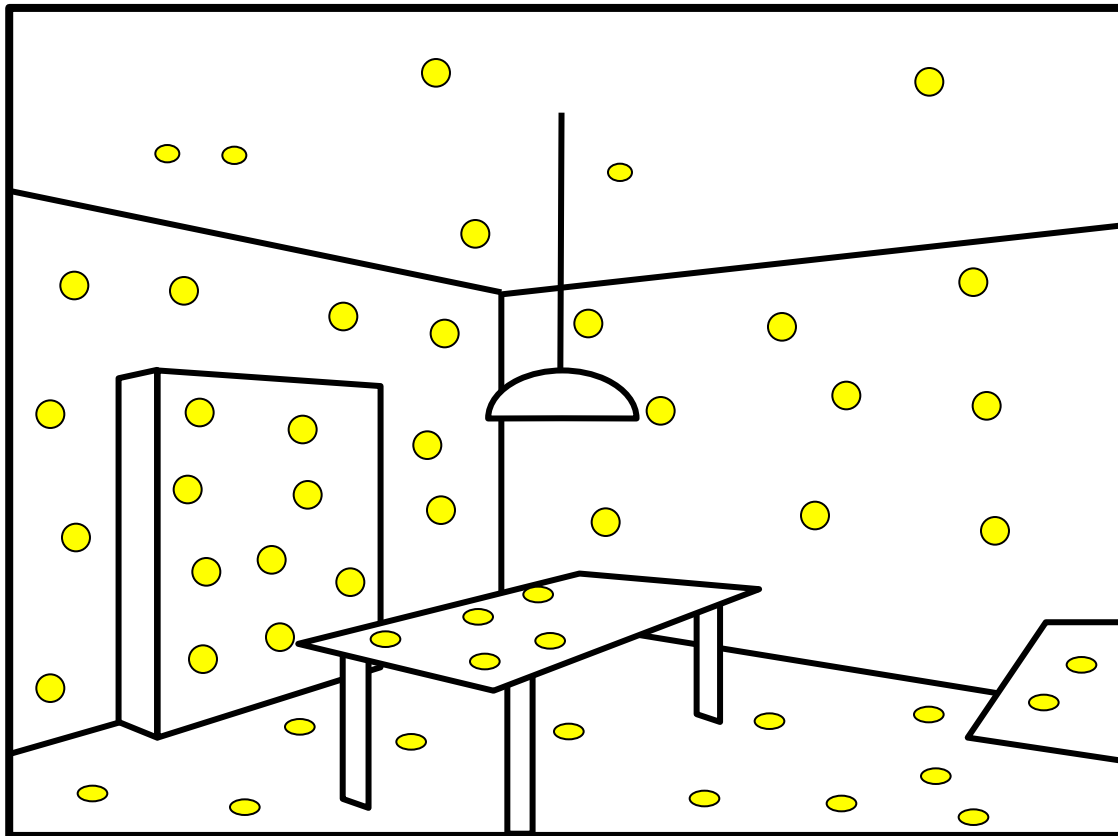
Photon mapping

- ▶ Pré-calcul : lancer des rayons depuis les lampes



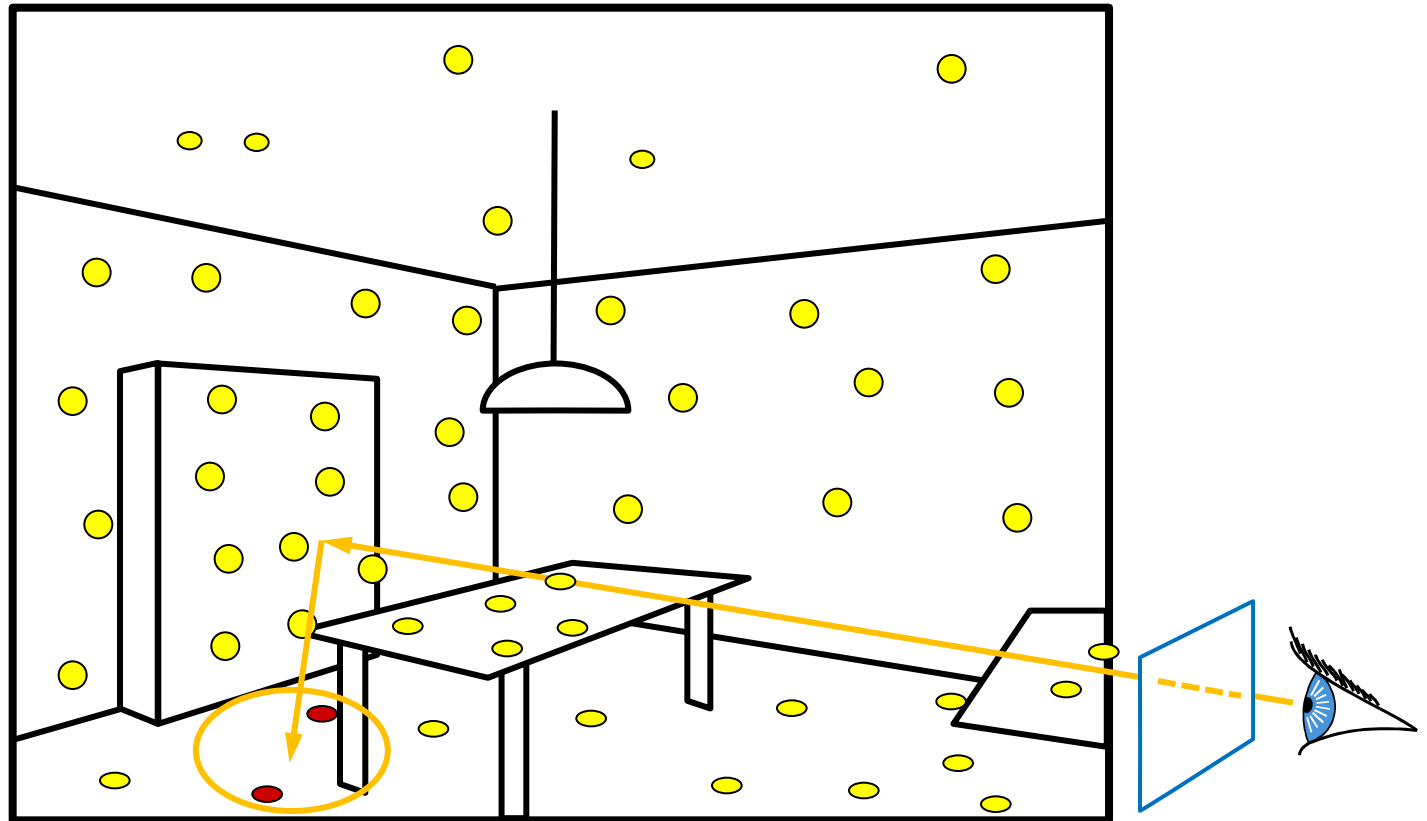
Photon mapping

- ▶ Stocker les photons (position + intensité + direction) sur la géométrie ou dans une structure accélératrice

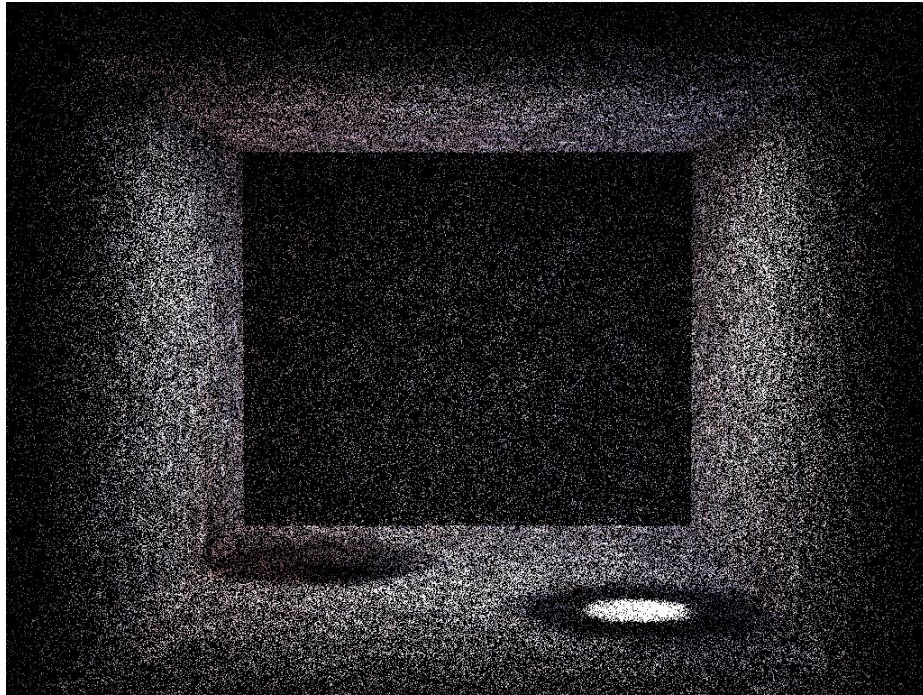


Photon mapping – rendu

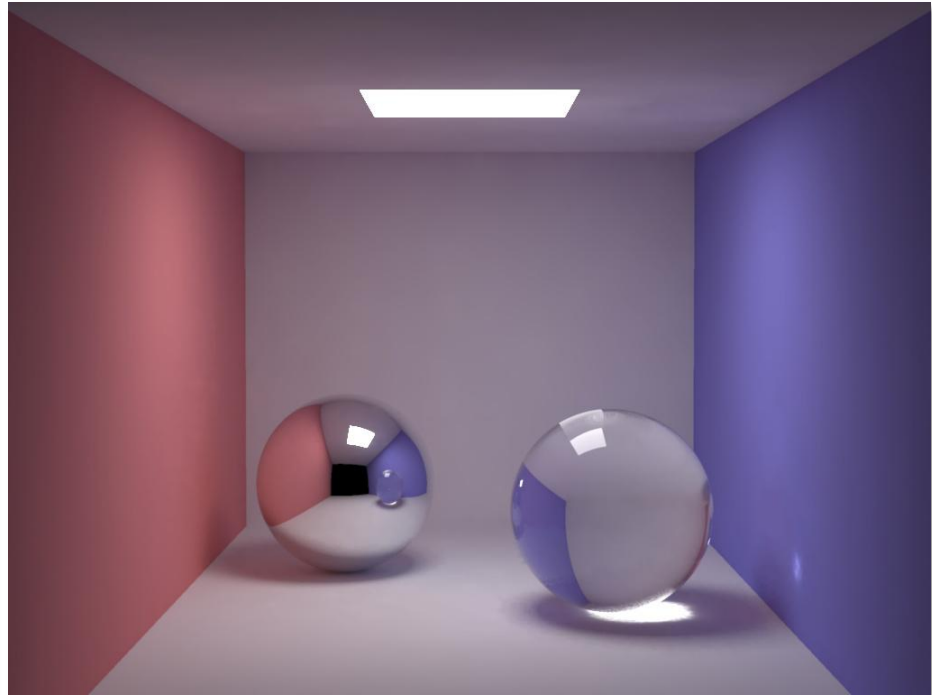
- ▶ Lancer les rayons primaires
- ▶ Reconstruire la radiance des rayons secondaire en regardant les photons voisins



Résultats



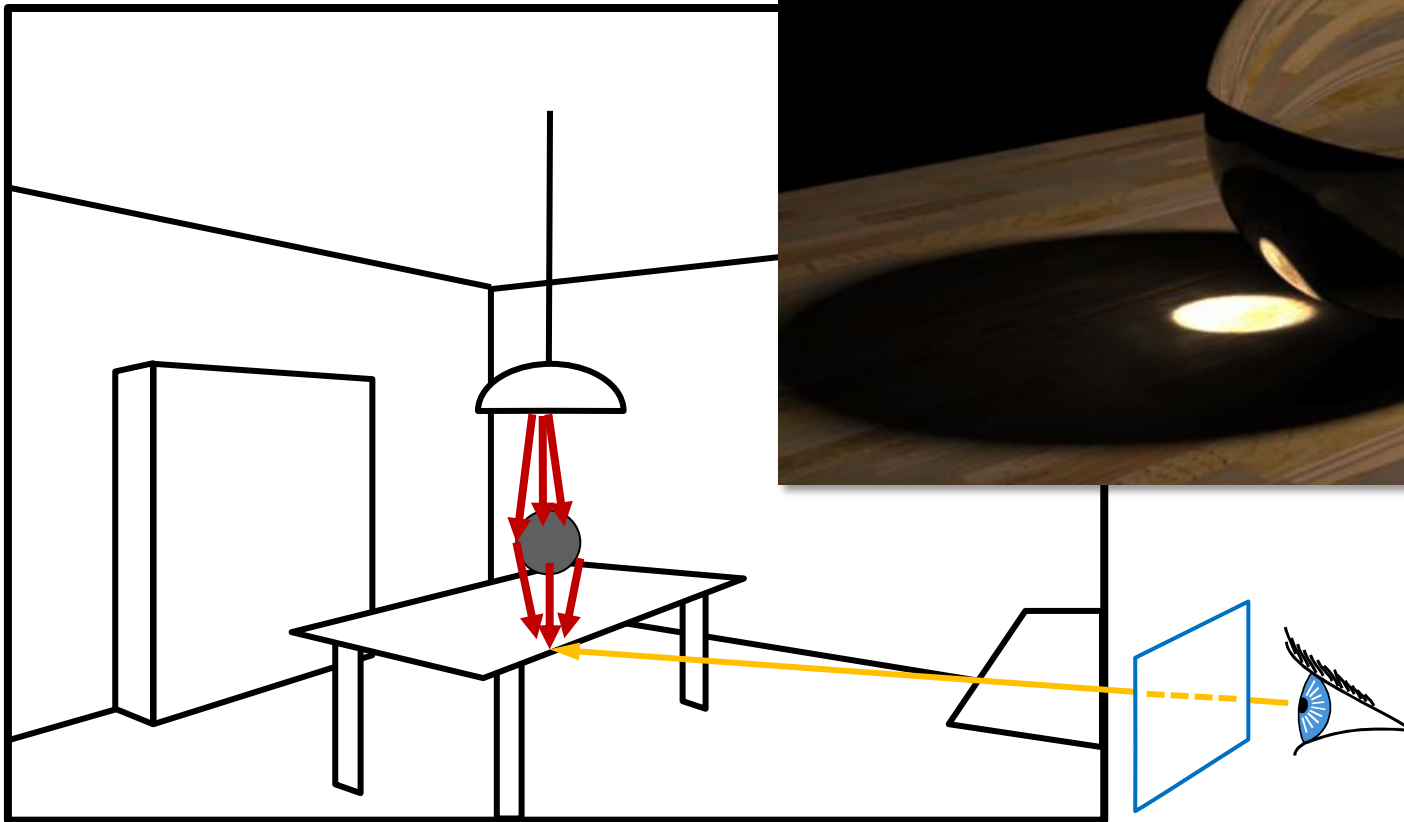
Photon map



Rendu final

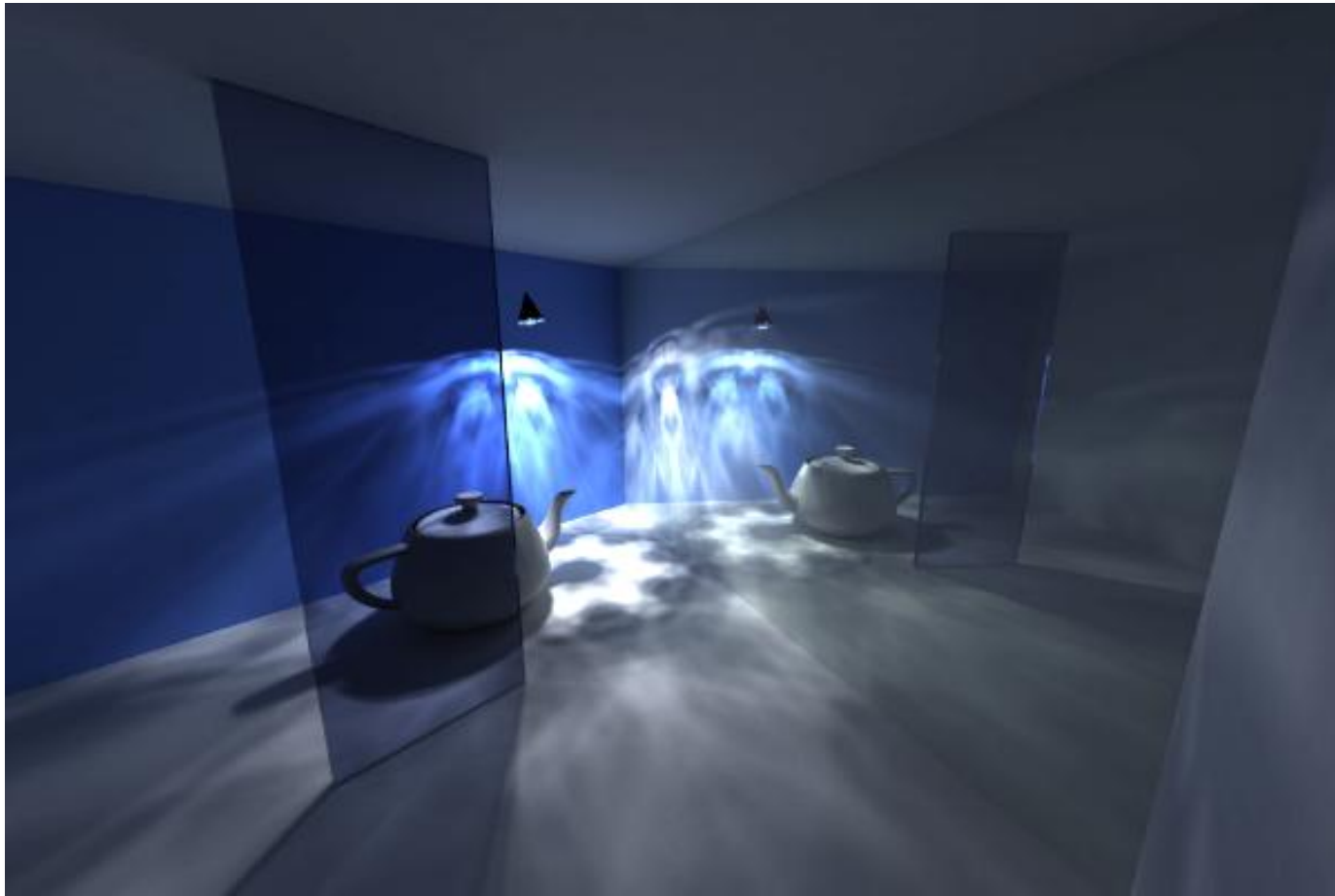
Caustiques

- ▶ Photon map spéciale réfraction



Résultats

V-Ray 1.5 for 3ds Max



Résultats



5 millions de photons depuis une seule lampe

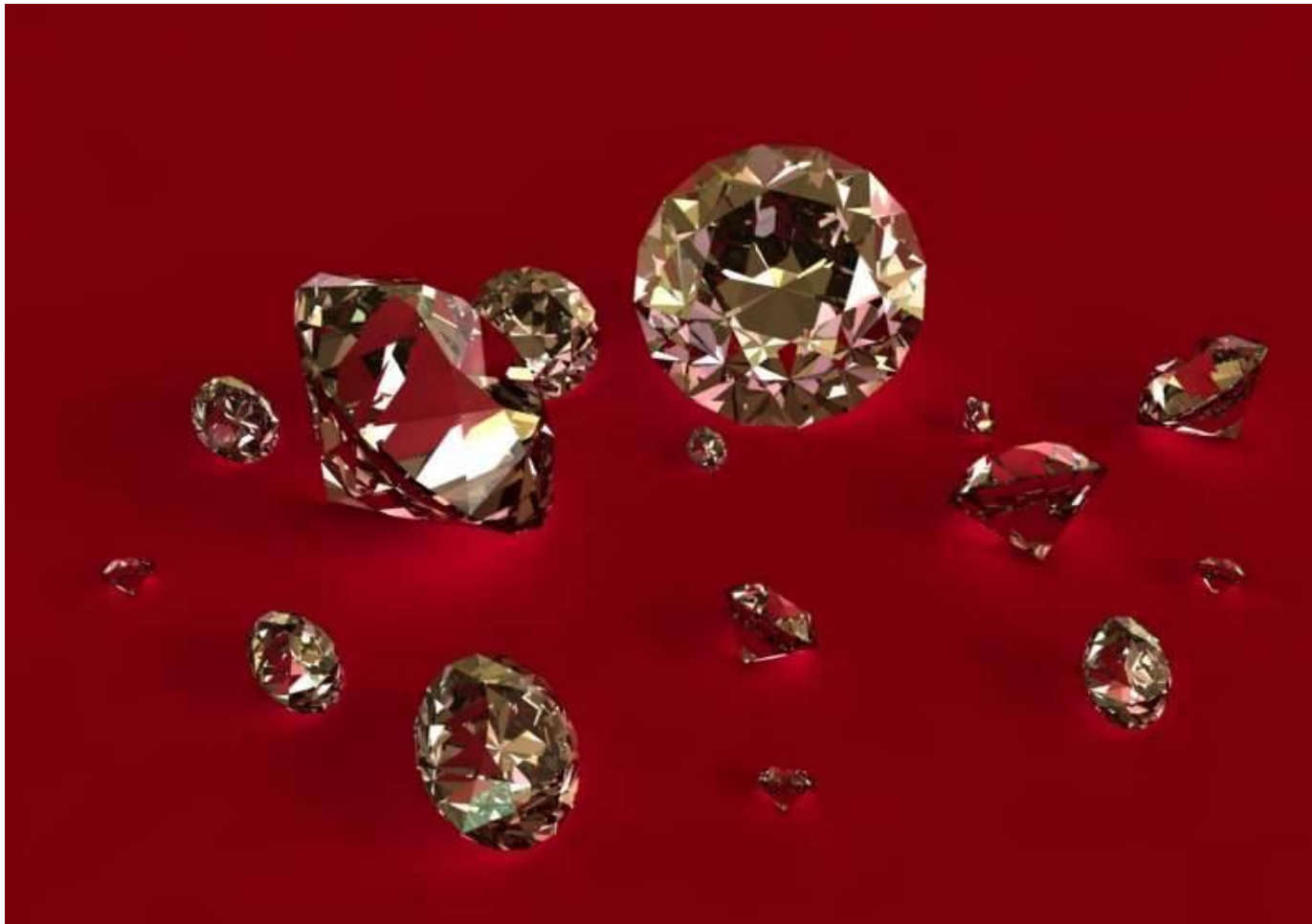
Résultats



CyberMotion 3D-Designer



Résultats



Yafray : ray tracer open source avec Photon Mapping,
intégré dans Blender.

Résultats



<http://en.wikipedia.org/wiki/POV-Ray>

Résumé Photon Mapping

- ▶ Solution indépendante du point de vue
 - Stockage sur les surfaces
- ▶ Bonne représentation des caustiques
- ▶ Bruité : phase de lissage des échantillons
 - Reconstruction de la fonction de radiance
- ▶ Se code en deux passes à partir d'un lancer de rayon dans les deux directions

Qui va gagner ?

- ▶ On ne sait pas...
 - Dépend de l'évolution du hardware
 - Des méthodes de résolution d'équations (intégrales, systèmes linéaires...)
 - Des méthodes d'accélération
 - De la quantité de mémoire (part du pré-calcul, stockage de structure de donnée complexe...)
- ▶ Tout le monde attend l'illumination en temps réel
 - Pré-calcul du transfert de radiance
 - Ambient occlusion en espace image
 - Illumination indirecte approximée en espace image