

# Modélisation de l'éclairage

Nicolas Holzschuch

*i*MAGIS is a joint project of CNRS - INPG - INRIA - UJF

# Modélisation de l'éclairage

- Modélisation physique :
  - Équation de l'éclairage
- Résolution de l'équation :
  - Solution formelle
  - Solutions approchées
    - Lancer de rayons
    - Radiosité
  - Solutions exactes
    - Monte-Carlo

# Éclairage global, éclairage local

- Jusqu'ici, des propriétés locales :
  - Modèles de Gouraud, de Phong, lissage
  - Plaquage de texture,
- Algorithmes locaux :
  - Z-buffer, Peintre, Scan-line,
  - Très rapide (et même interactive).
- On s'intéresse au problème global

# Équation de l'éclairage

- Hypothèses :
  - Équilibre énergétique
  - Conservation de l'énergie
  - ... lumineuse
    - Pas d'échanges entre différentes formes d'énergie
- Énergie lumineuse en un point :
  - Énergie émise
  - Plus énergie réfléchie

# Équation de l'éclairage

- Équilibre énergétique :

$$L(x, \theta_0, \varphi_0) = L_e(x, \theta_0, \varphi_0) +$$

$$\rho_{bd}(x, \theta_0, \varphi_0, \theta, \varphi) L_i(x, \theta, \varphi) \cos \theta d\omega$$

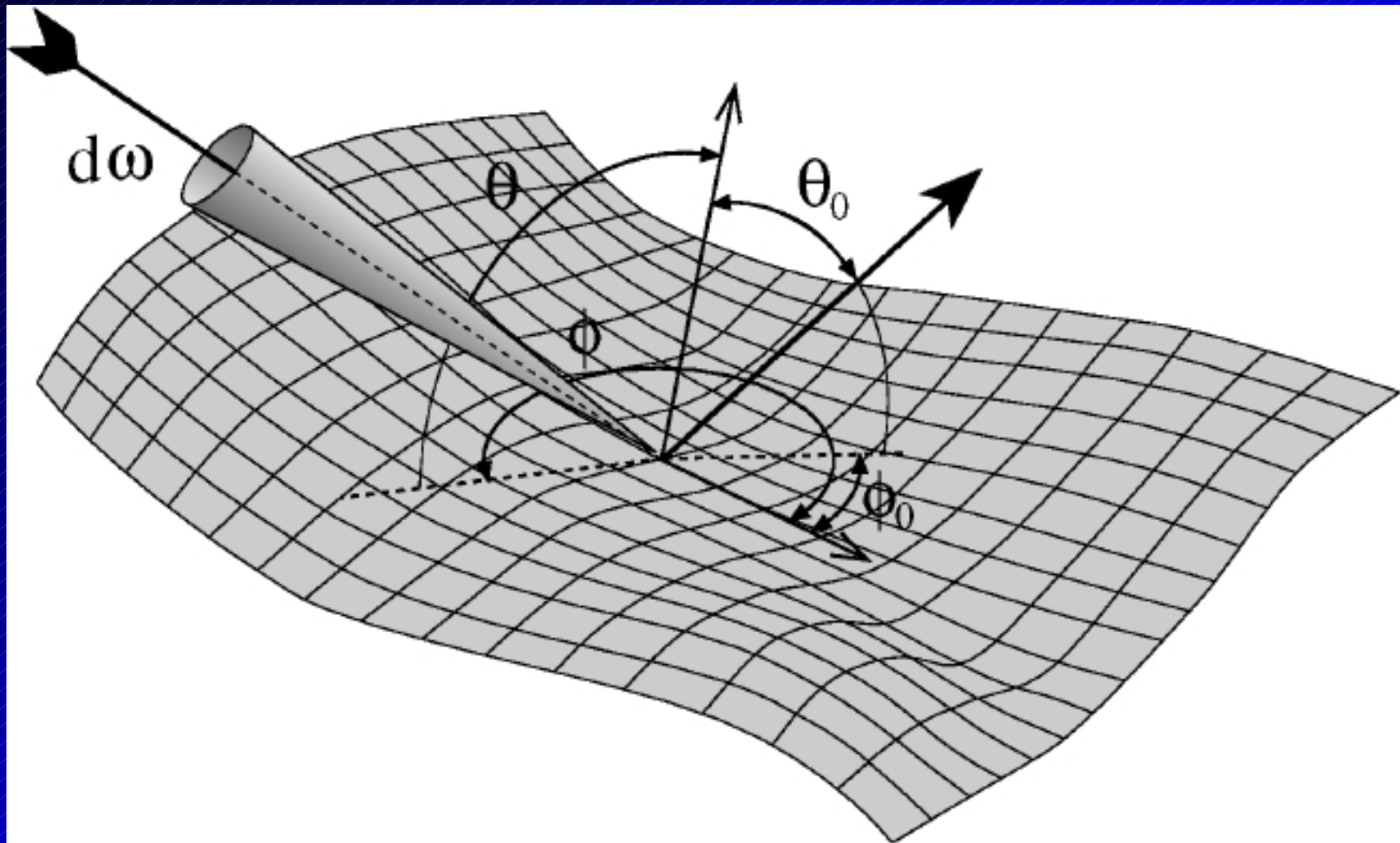
- Radiance totale = radiance émise + radiance réfléchie

# BRDF

- *Bi-directional Reflectance Distribution Function (BRDF)*
- Rapport entre la radiance dans la direction sortante et le flux de radiance dans la direction entrante :

$$\rho_{bd}(\theta_0, \varphi_0, \theta, \varphi) = \frac{L(x, \theta_0, \varphi_0)}{\int L_i(x, \theta, \varphi) \cos \theta d\omega}$$

# Notations



# La BRDF est une *distribution*

- Elle va de 0 à + (inclus)
- Unité  $\text{sr}^{-1}$
- Deux cas particuliers :
  - Réflecteur diffus idéal
  - Réflecteur spéculaire idéal



# Réflecteur diffus idéal

- BRDF uniforme :
  - Ne dépend pas des directions
  - La radiance réfléchie est identique dans toutes les directions

$$\rho_{bd}(\theta_0, \varphi_0, \theta, \varphi) = \rho$$

# Réflecteur spéculaire idéal

- La BRDF est une distribution de Dirac
  - Distribution de Dirac normalisée

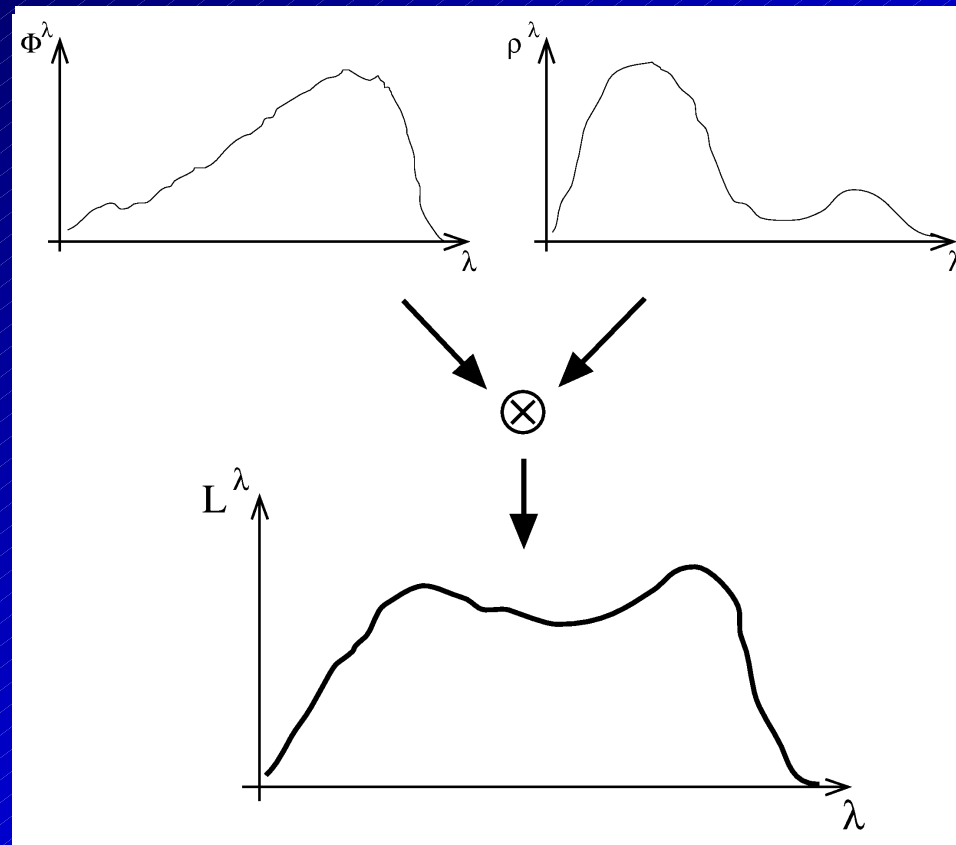
$$\rho_{bd}(\theta_0, \varphi_0, \theta, \varphi) =$$

$$\rho_s(\theta) 2\delta(\sin^2 \theta_0 - \sin^2 \theta) \delta(\varphi_0 - \varphi \pm \pi)$$

$\rho_s$  s'appelle la réflectance spéculaire

# Variation spectrale

- La BRDF dépend aussi de la longueur d'onde



# Résolution formelle de l'équation

- Opérateur de réflexion
  - Opérateur intégral  $R$
  - Opère sur la distribution de radiance
  - Défini par :

$$(RL)(x, \theta_0, \varphi_0) =$$

$$\rho_{bd}(x, \theta_0, \varphi_0, \theta, \varphi) L_i(x, \theta, \varphi) \cos \theta d\omega$$

# Solution formelle (2)

- L'équation devient :

$$L = L_e + RL$$

- Donc :

$$L = [I - R]^{-1} L_e$$

- En utilisant une série de Neumann :

$$L = \sum_{n=0} (R^n) L_e$$

# Interprétation physique

$$L = \sum_{n=0} (R^n) L_e$$

- Radiance émise ( $L_e$ )...
- plus radiance réfléchie une fois ( $RL_e$ )...
- plus radiance réfléchie deux fois ( $R^2L_e$ )...
- plus radiance réfléchie trois fois...

# Concrètement, cette équation ?

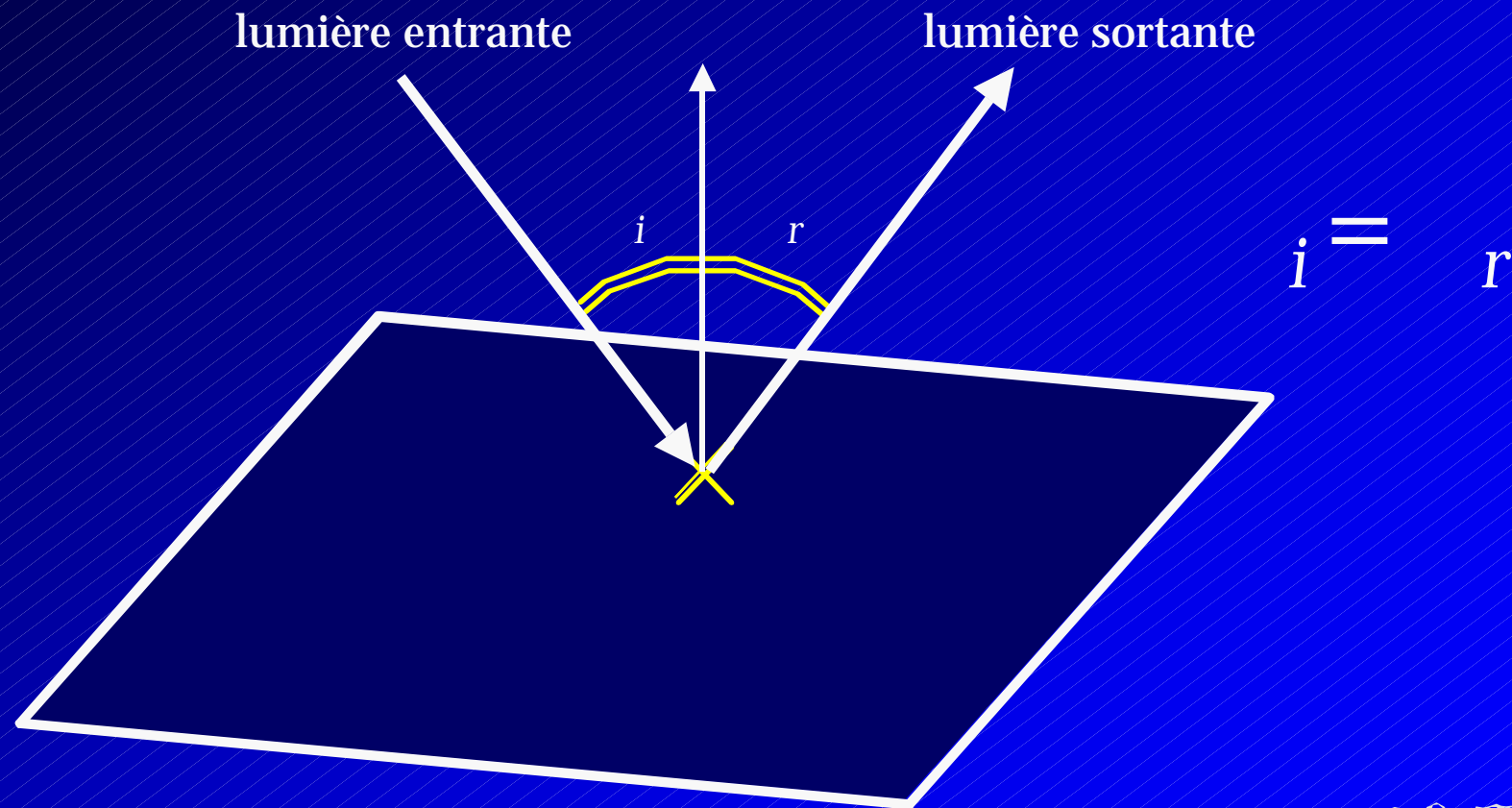
- Très joli... et complètement inutile
- Pas de solution analytique
- L'éclairage a des besoins spécifiques
  - Frontières d'ombre, reflets...
  - La résolution doit prendre en compte ces spécificités
- Hypothèses de simplification
  - ou résolution complète

# Lancer de rayons

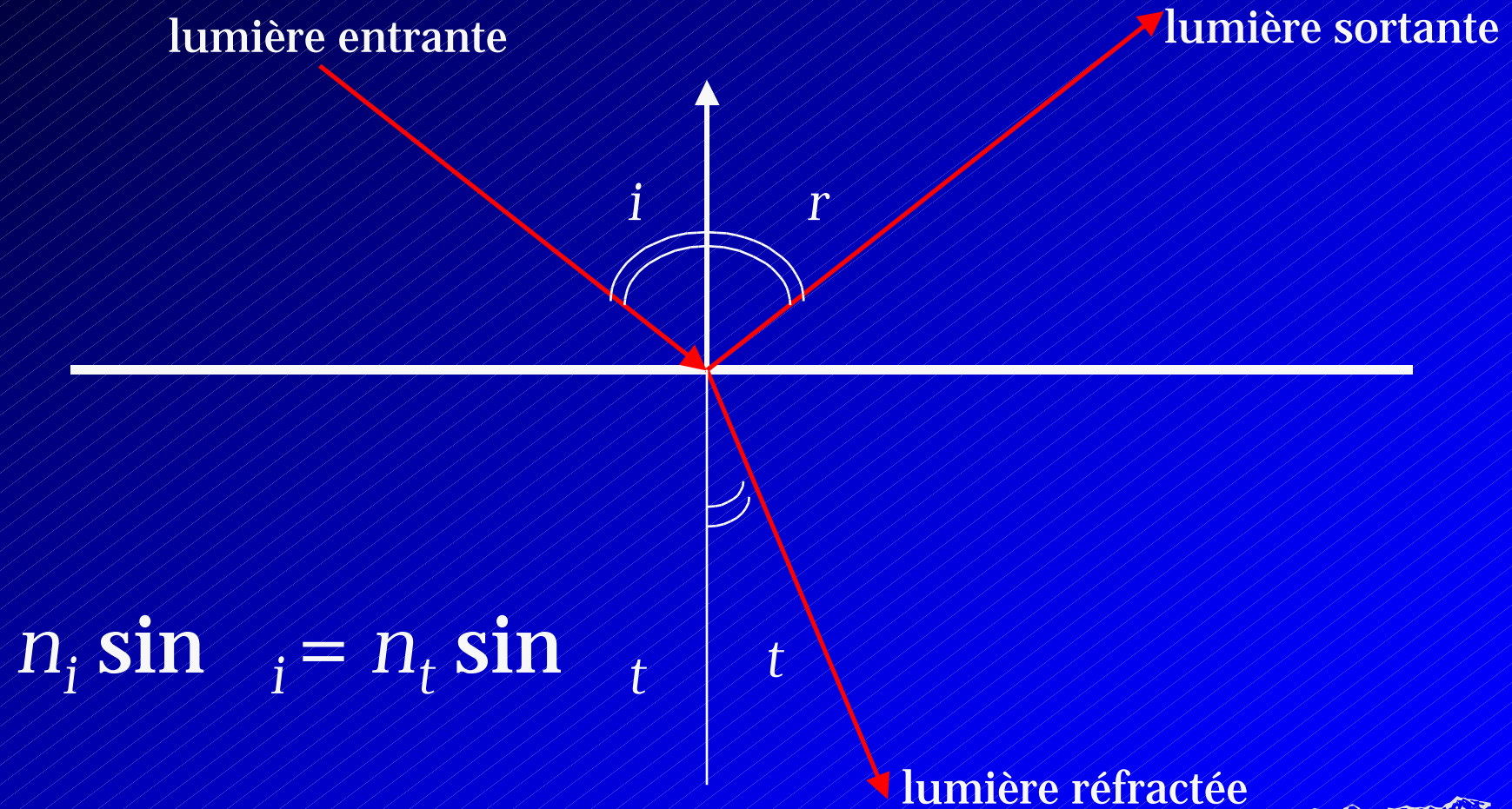
- Toute la complexité vient de l'aspect global
- Différencier local/global :
  - Local : modèle de surface quelconque
  - Global : purement spéculaire



# Réflexion spéculaire



# Loi de Snell-Descartes



# Lancer de rayon

- Un rayon touchant une surface engendre au plus deux rayons
  - Complexité contrôlable
- Première méthode :
  - Lancer de rayons en partant des sources
  - Suivre les rayons qui quittent les sources lumineuses
  - Ils rebondissent jusqu'à ce qu'ils touchent la caméra
  - Couleur du pixel = couleur du rayon

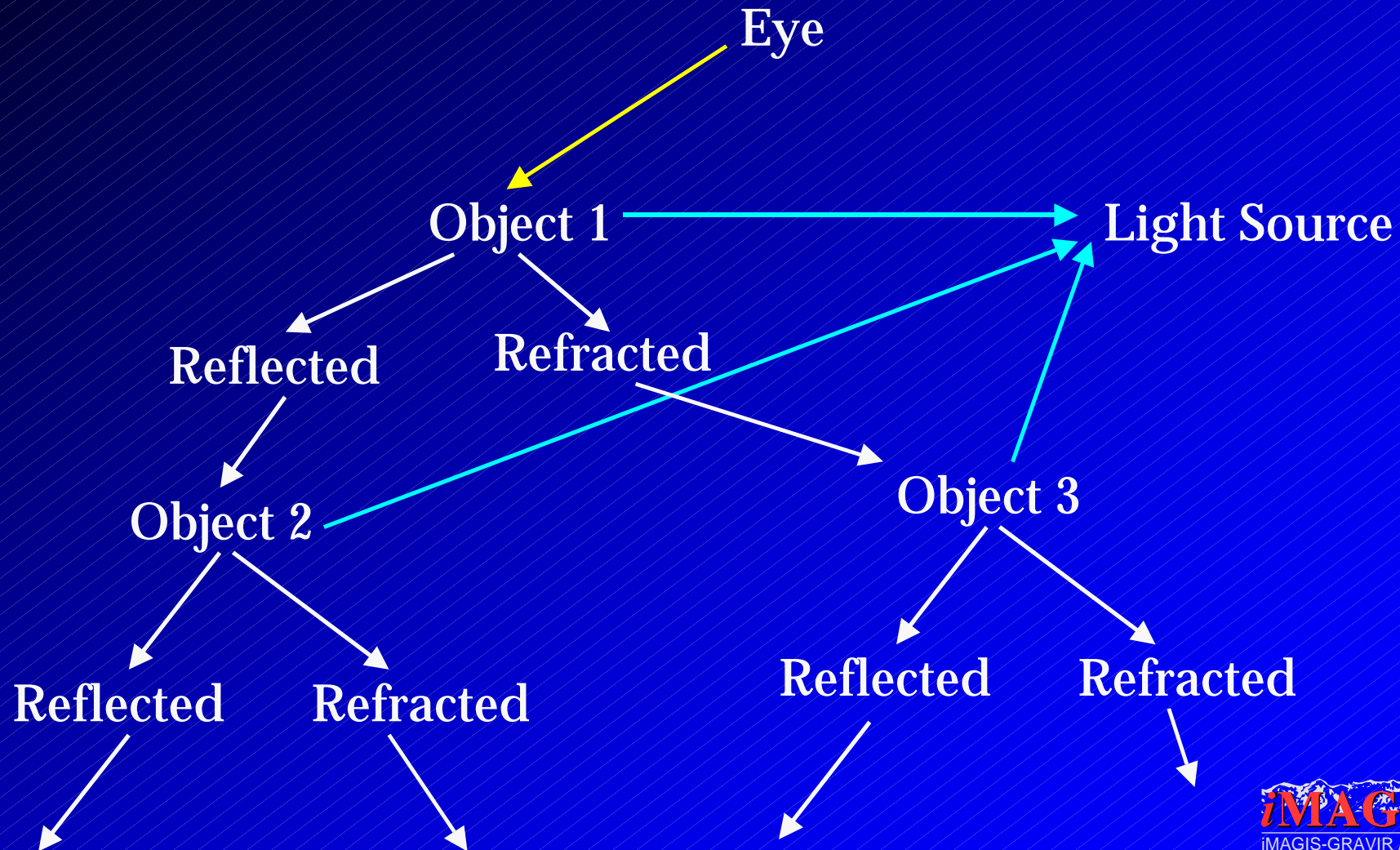
# Le retour inverse

- On ne sait pas à l'avance quels rayons vont atteindre l'œil :
  - Des calculs inutiles
- Loi du retour inverse :
  - Si la lumière peut aller de  $a$  à  $b$ , alors elle peut aussi aller de  $b$  à  $a$ .
  - Il suffit de suivre les rayons qui quittent l'œil

# Lancer de rayons en partant de l'œil

- Modèle de caméra standard
  - Pour chaque pixel,
    - Créer le rayon qui va de l'œil à ce pixel
    - Trouver le premier objet touché,
    - Point d'intersection visible d'une source lumineuse ?
      - Oui : on utilise le modèle de matériau complet ;
    - Créer le rayon réfléchi, trouver le prochain objet touché
- Les objets dans l'ombre sont invisibles
  - On utilise la couleur ambiante.

# Arbre de rayons



# Intersection rayon-objet

- Rayon-sphère : distance point-ligne.
- Rayon-cylindre : distance ligne-ligne.
- Rayon-plan : intersection ligne-plan
- Rayon-polygone :
  - Intersection avec le plan support.
  - Savoir si le point d'intersection est dans le polygones :
    - On projette sur le plan  $xy$ , vérif. sur un polygone 2D.

# Accélération

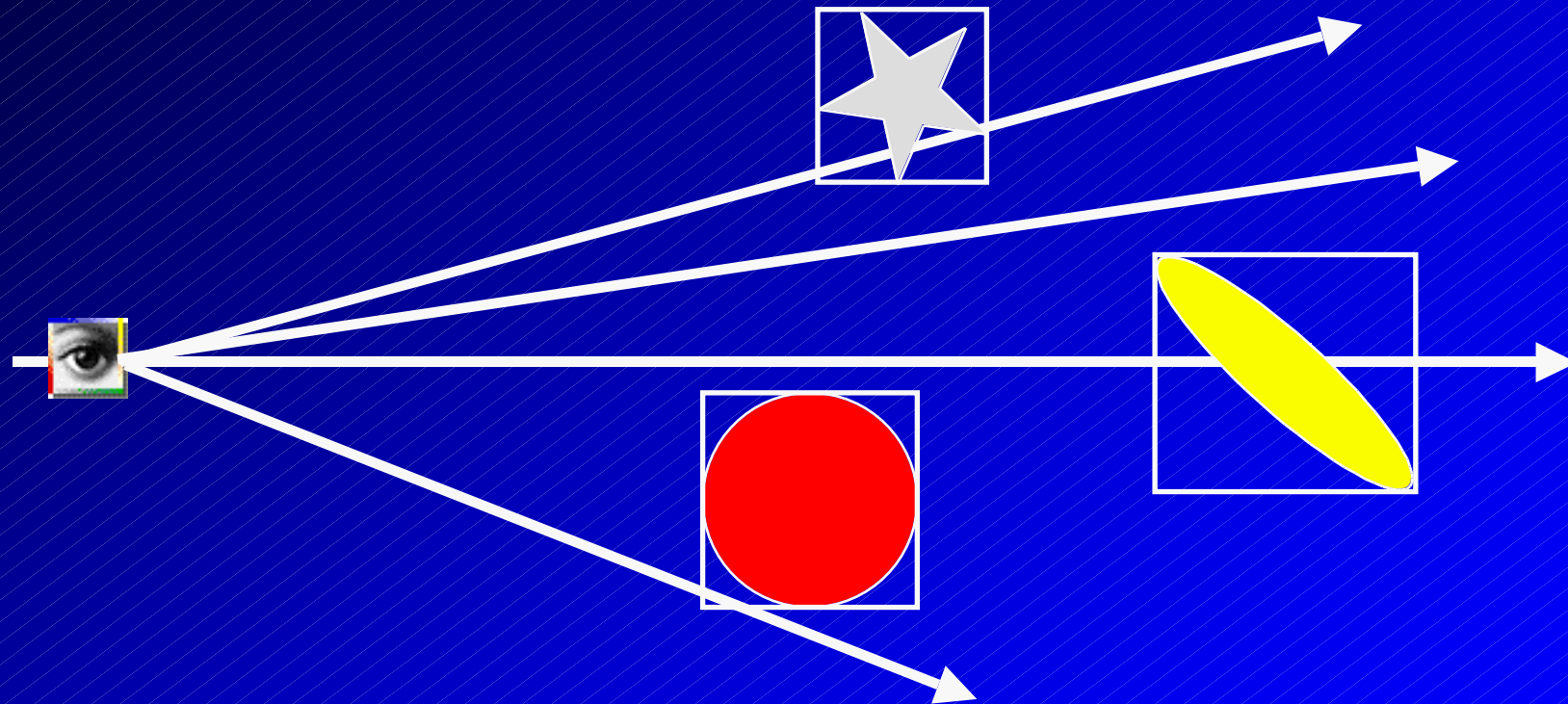
- Le cœur de l'algorithme, ce sont les intersections rayon-objet
  - Le travail sur chaque intersection est presque négligeable
  - On peut utiliser un modèle de réflexion complexe
- Accélérer la recherche d'intersections
  - Volumes englobants,
  - Grilles uniformes (voxels),
  - Octrees,
  - BSP-trees,
  - Méthodes spécifiques



# Volumes englobants

- Boite englobante :
  - $X_{\min}$ ,  $X_{\max}$ ,
  - $Y_{\min}$ ,  $Y_{\max}$
  - $Z_{\min}$ ,  $Z_{\max}$ .
- On teste la BE avant l'objet
  - Un test rapide pour économiser un test plus long
- On peut aussi utiliser d'autres volumes :
  - Sphères englobantes.

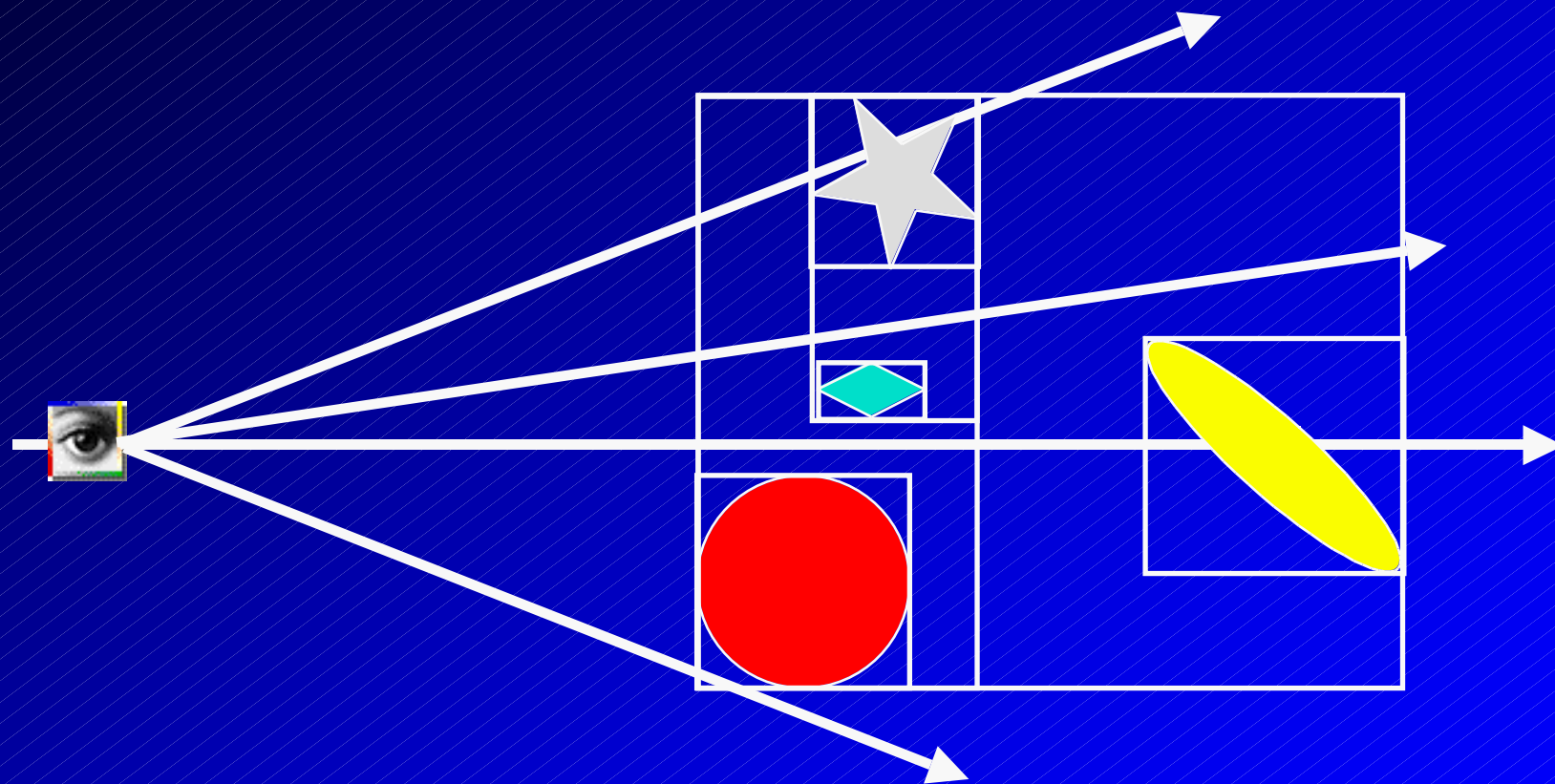
# Boites englobantes



# Hiérarchie de boites englobantes

- Pré-traitement :
  - Construire les boites englobantes,
  - Construire une hiérarchie de boites englobantes,
    - En utilisant le modèle de la scène
    - De façon automatique
- Pour chaque rayon :
  - On teste la BE
  - S'il y a intersection : on teste les enfants

# Hiérarchie de boîtes englobantes



# Grille uniforme

- On construit une grille 3D
- On place les objets dans les cellules qu'ils touchent
- Le rayon traverse la grille :
  - Si une cellule est non-vide :
    - Intersection du rayon avec chaque objet de la cellule
    - Vérifier que le point d'intersection est dans la cellule
- Résolution de la grille ?

# Octrees

- Construire un *octree* pour la scène :
- Pour chaque rayon :
  - Tester s'il y a intersection avec les enfants
  - Si intersection : on descend dans la hiérarchie
  - Si la cellule est une feuille : on teste l'intersection avec les objets dans la cellule
  - Vérifier que le point d'intersection est dans la cellule

# BSP Trees

- Structure générique :
  - Couper l'espace en deux sous-espaces ;
- Ne fonctionne qu'avec des polyèdres
- Récursivement :
  - Intersection du rayon avec les objets devant,
  - Intersection du rayon avec le support du nœud
  - Intersection du rayon avec les objets derrière

# Comparatif

- Hiérarchie de BE :
  - Construction assez longue, requêtes rapides
- Grille uniforme :
  - Construction rapide, requêtes rapides... à la bonne résolution
- Octrees, BSP trees:
  - Très simple, construction rapide, requêtes plus longues

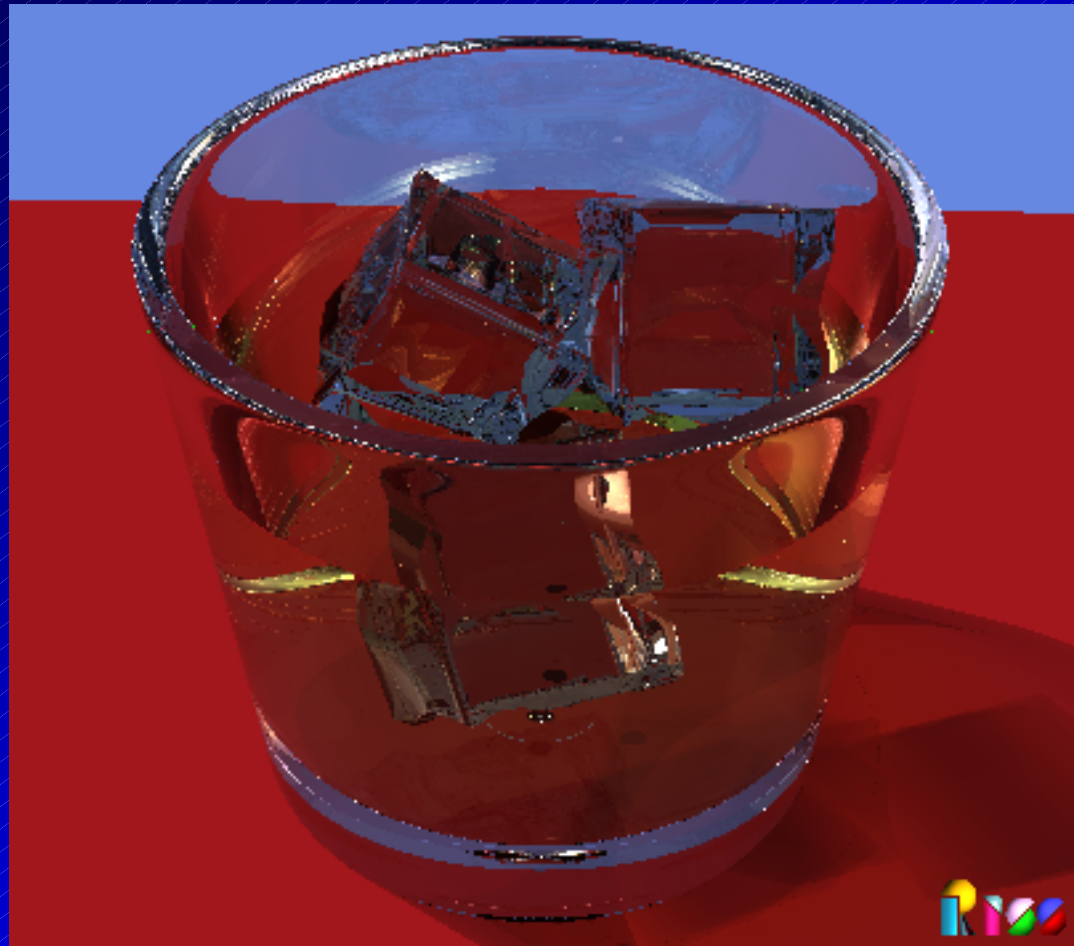
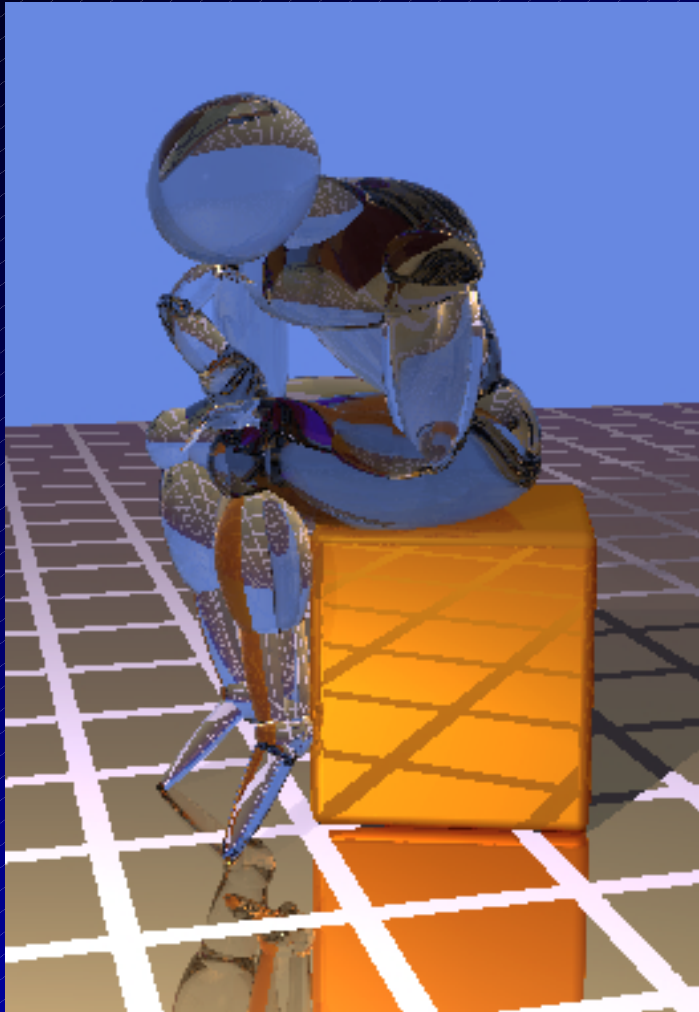


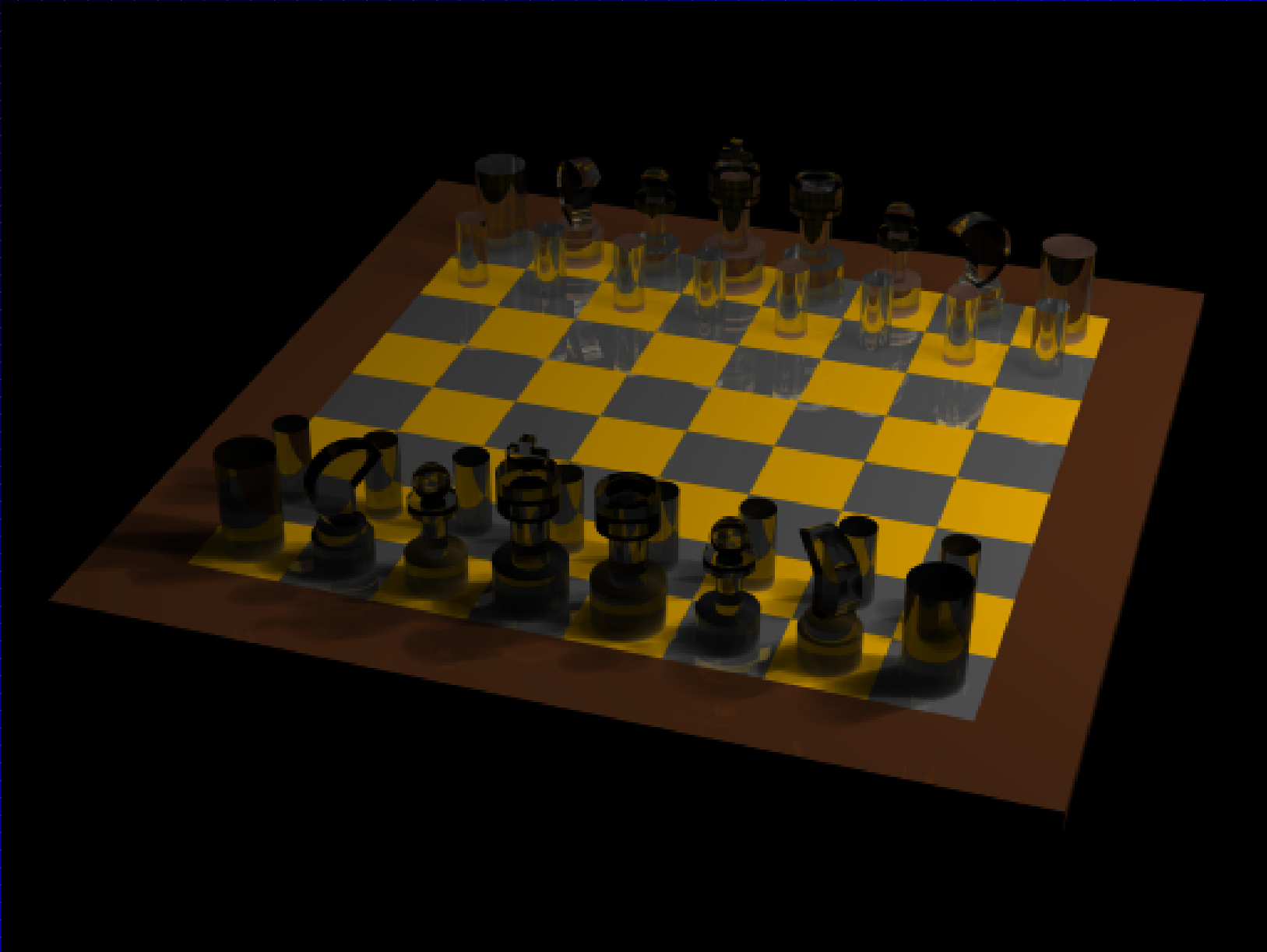
# Lancer de rayons : résumé

- Lent, mais fait aussi :
  - L'élimination des parties cachées,
  - Les ombres,
  - La transparence,
  - La CSG,
  - Le plaquage de textures (même procédurales).
- Réflexions entre les objets
- N'importe quelle primitive graphique

# Lancer de rayons : limitations

- Limité à un certain modèle de matériaux :
  - Toutes les réflexions sont métalliques
  - (même si les objets peuvent être diffus)
- Profondeur de l'arbre :
  - Déterminée à l'avance ?
  - Test d'arrêt ?
  - Boucle infinie ?





# Radiosité

- Hypothèse de simplification :
  - Toutes les surfaces sont diffuses
- Radiance, BRDF, ... indépendantes de la direction
- Simplification de l'équation de rendu
- Discrétisation de l'équation simplifiée
- Résolution de l'équation discrétisée

# Radiosité

- Forme simplifiée de l'équation de rendu :

$$B(x) = E(x) + \rho_d(x) \int_y S B(y) V(x, y) \frac{\cos\theta \cos\theta}{\pi r^2} dy$$

- Discrétisation de cette équation :

$$B_i = E_i + \rho_i \sum_{j=1}^n F_{ij} B_j$$

# Facteur de forme

- $F_{ij}$  : facteur de forme
  - Purement géométrique
  - Proportion d'énergie transmise de l'un à l'autre
  - Méthodes de calcul :
    - Quadratures
    - Approximation point-surface
    - Approximations discrètes (hémicube)

# L'équation discrétisée

- On regroupe tous les éléments :

$$\begin{matrix} B_0 \\ \\ B_n \end{matrix} = \begin{matrix} E_0 \\ \\ E_n \end{matrix} + \begin{matrix} B_0 \\ \\ B_n \end{matrix} \left[ \rho_i F_{ij} \right] :$$

- Ou, en utilisant des vecteurs :

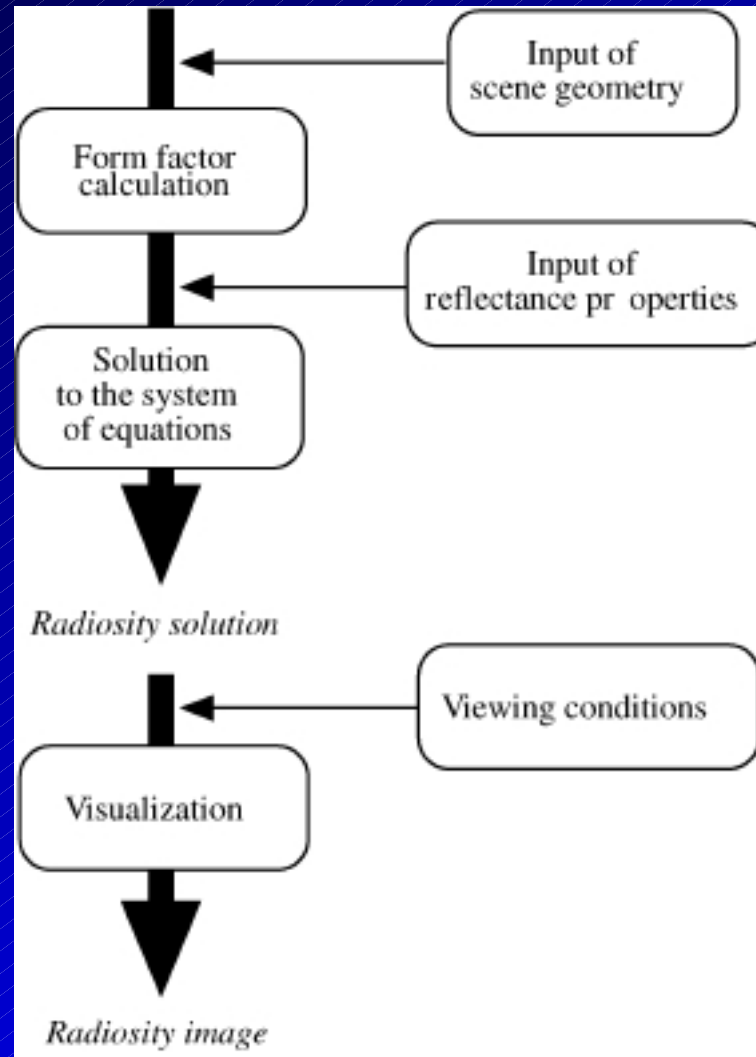
$$\mathbf{B} = \mathbf{E} + \mathbf{MB}$$



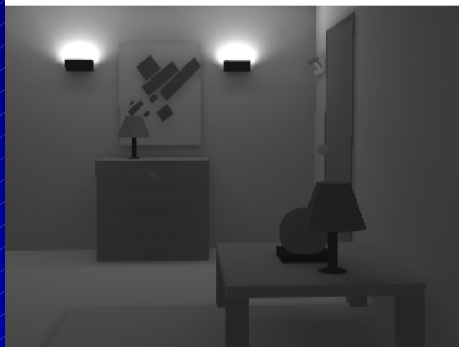
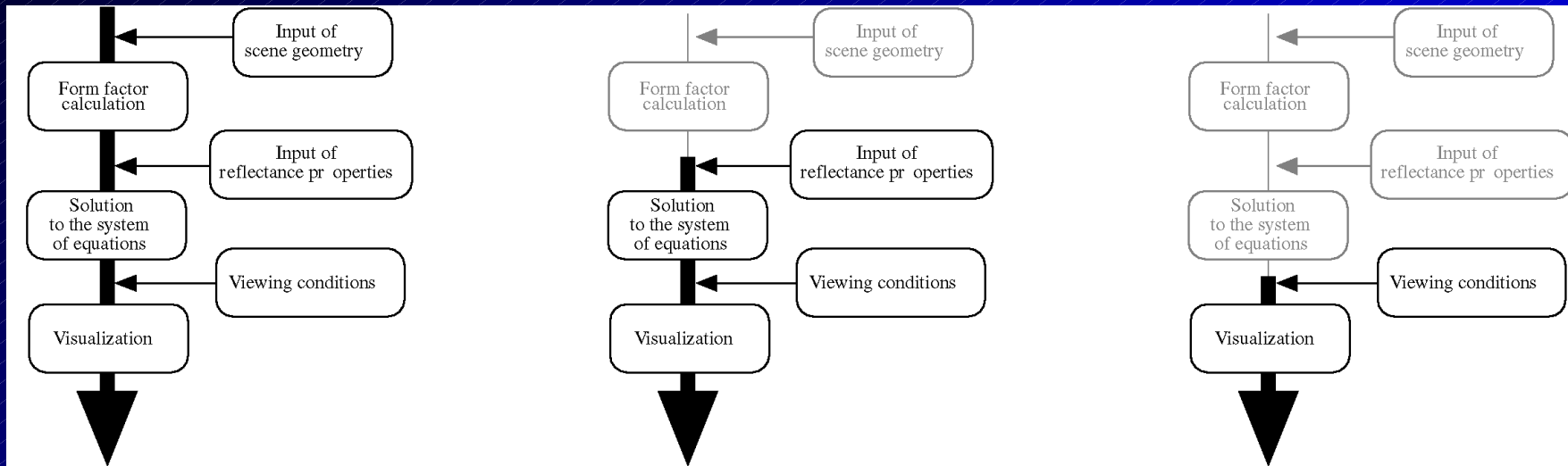
# Radiosité : avantages

- Toutes les surfaces sont diffuses
- Affichage en temps réel
  - en utilisant le Z-buffer
- Fonction continue :
  - calculer des valeurs aux sommets
  - interpolation entre les sommets
    - par Gouraud-shading (en hardware)

# Du modèle à la solution



# Considérations de coût

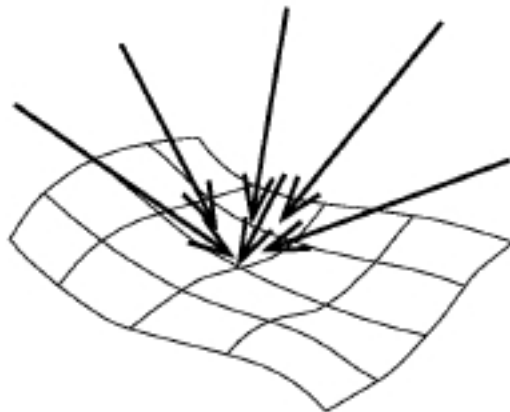


# Résolution du système

- Equation Matricielle :
- Méthodes de relaxation :  $\mathbf{B} = \mathbf{E} + \mathbf{MB}$ 
  - Gathering
    - relaxation de Jacobi
    - relaxation de Gauss-Seidel
  - Shooting
    - relaxation de Southwell

# Gathering vs. Shooting

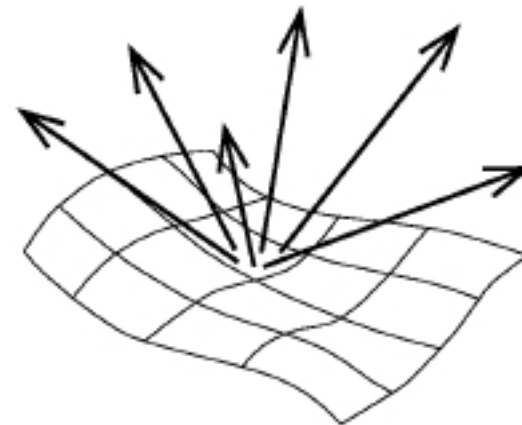
Gathering



$$\begin{bmatrix} X \\ X \\ X \\ X \\ X \\ X \end{bmatrix} = \begin{bmatrix} X \\ X \\ X \\ X \\ X \\ X \end{bmatrix} + \begin{bmatrix} X & X & X & X & X \\ X & X & X & X & X \\ X & X & X & X & X \\ X & X & X & X & X \\ X & X & X & X & X \\ X & X & X & X & X \end{bmatrix} \begin{bmatrix} X \\ X \\ X \\ X \\ X \\ X \end{bmatrix}$$

For a selected  $i$ ,  $B_i^{(k+1)} = E_i + \sum_{j=1}^N (\rho_i F_{ij}) B_j^{(k)}$

Shooting



$$\begin{bmatrix} X \\ X \\ X \\ X \\ X \\ X \end{bmatrix} = \begin{bmatrix} X \\ X \\ X \\ X \\ X \\ X \end{bmatrix} + \begin{bmatrix} X \\ X \\ X \\ X \\ X \\ X \end{bmatrix} \begin{bmatrix} X \\ X \\ X \\ X \\ X \\ X \end{bmatrix}$$

For all  $j$ ,  $\beta_j^{(k+1)} = \beta_j^{(k)} + (\rho_j F_{ij}) r_i^{(k)}$

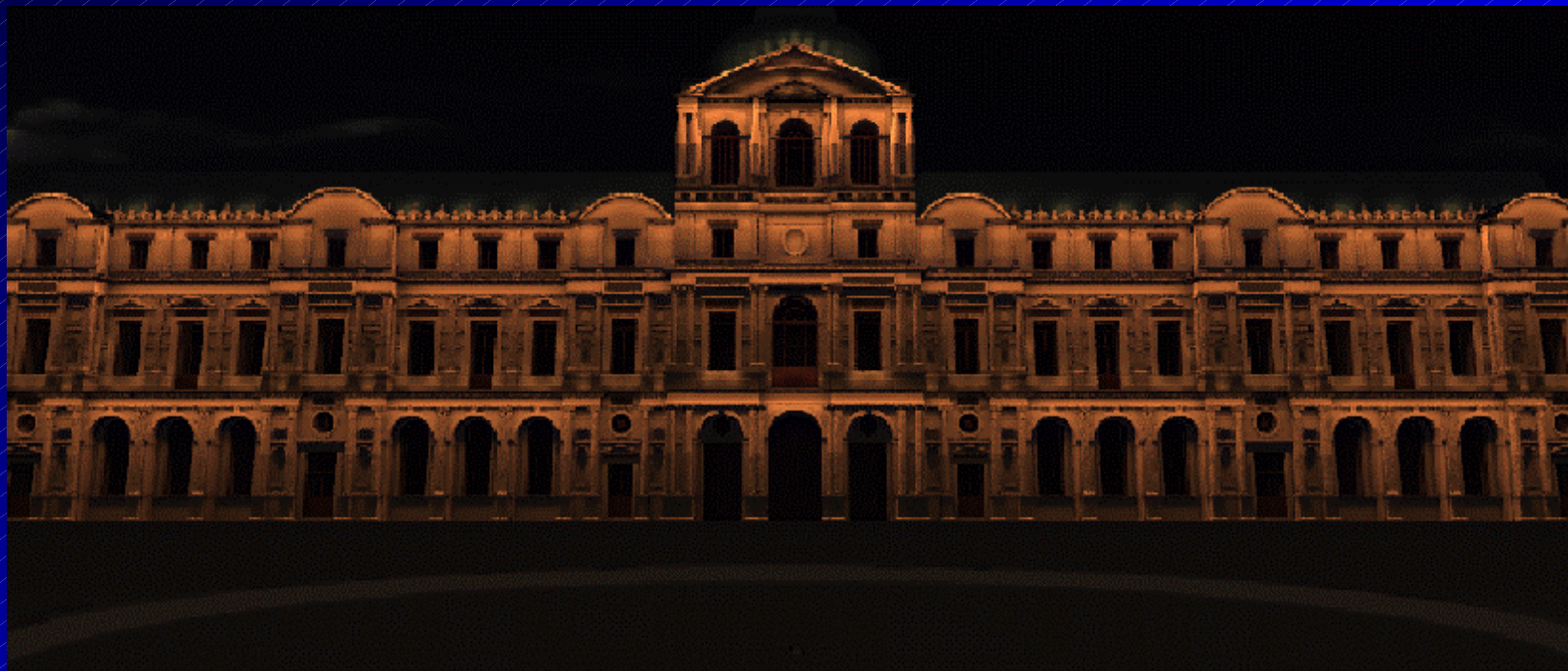
# Gathering/shooting

- Shooting :
  - Premières images plus vite
  - Éclairage direct dans les premières itérations
  - Besoin de stocker l'énergie pas encore renvoyée
- Gathering :
  - Plus lent pour les premiers résultats
- Pour la convergence totale de la scène
  - même temps de calcul

# Radiosité : exemple



# Radiosité : exemple (2)





# Radiosité : résumé

- Calculs très long
  - Accélération possibles,
    - Radiosité hiérarchique
    - Clustering
    - Instanciation
- Affichage rapide
  - Sauf si  $10^7$  polygones...

# Lancer de rayons de Monte-Carlo

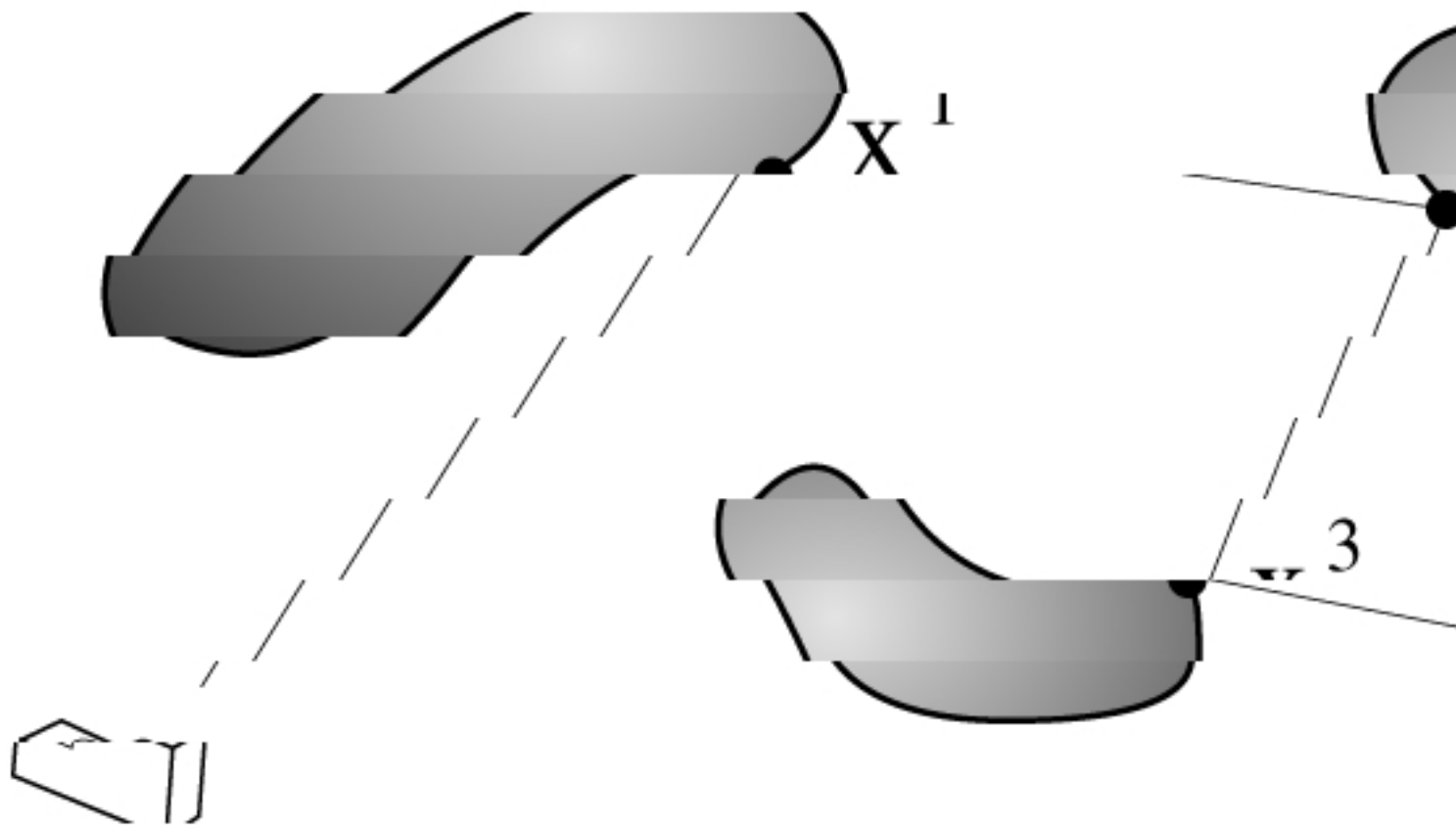
- On repart de l'équation de rendu
- Utilisation de méthodes mathématiques d'approximation des intégrales

# Intégration de Monte-Carlo

- Chaînes de Markov
- Méthode générique de calcul d'intégrales multi-dimensionnelles
- Principe (en gros) :
  - Échantillonner au hasard la fonction à intégrer
  - Additionner les échantillons multipliés par leur probabilité
  - Le résultat, c'est la valeur de la fonction

# Chemin

- Point de départ : un pixel de l'image,  $x$
- Premier état : la radiance de la surface visible de ce pixel, au point  $x'$
- État suivant : la radiance d'une surface visible de  $x'$ ,  $x''$
- Transition : BRDF au point  $x'$ , venant de  $x''$ , dans la direction de  $x$
- État suivant : la radiance d'une surface visible de  $x''$ ,  $x'''$ ...



# Choix du chemin

- Pixel de départ, fixé.
- Pour chaque point d'intersection :
  - Tirer la direction du rayon réfléchi au hasard
  - En tenant compte de la BRDF
- Recommencer pour chaque pixel
  - 10, 100, 1000 échantillons par pixel
- L'image converge lentement :
  - Neige au début, puis image bruité, puis image

# Terminaison du chemin

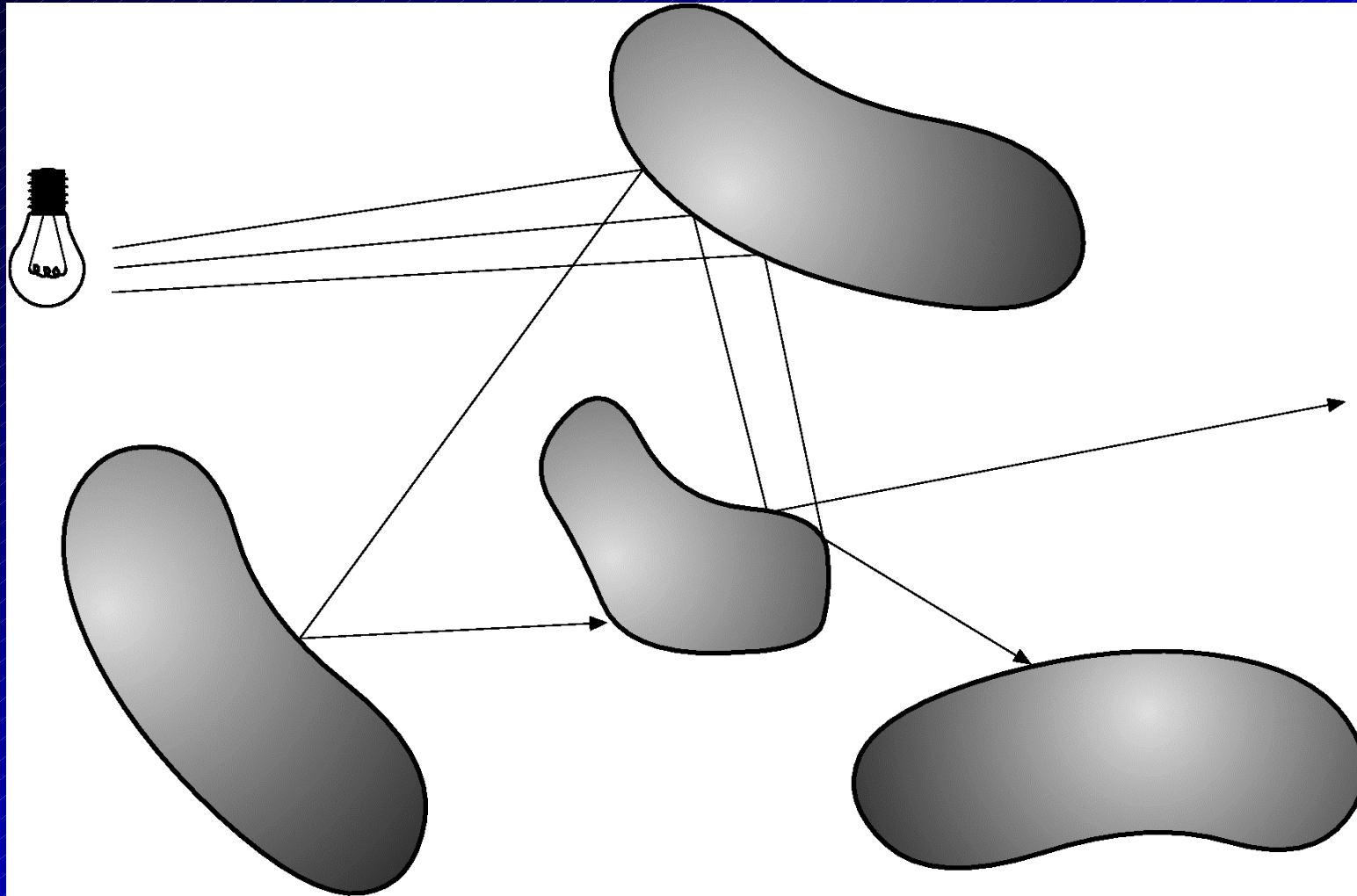
- Nombre aléatoire  $t$  entre 0 et 1
- Comparer avec la réflectance
  - $t > R$  : absorption
  - $t < R$  : réflexion
- Autres techniques:
  - Poids  $w$  de la particule influencé par la réflectance
  - Terminaison si  $w$  en dessous d'un certain seuil
  - Roulette russe : on augmente le poids des chemins survivants

# (Distributed) Light Ray-Tracing

- Pareil, mais les rayons partent des sources lumineuses
  - Sources lumineuses potentiellement surfaciques
  - Échantillonnage spatial de la source
- 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



# Light ray-tracing

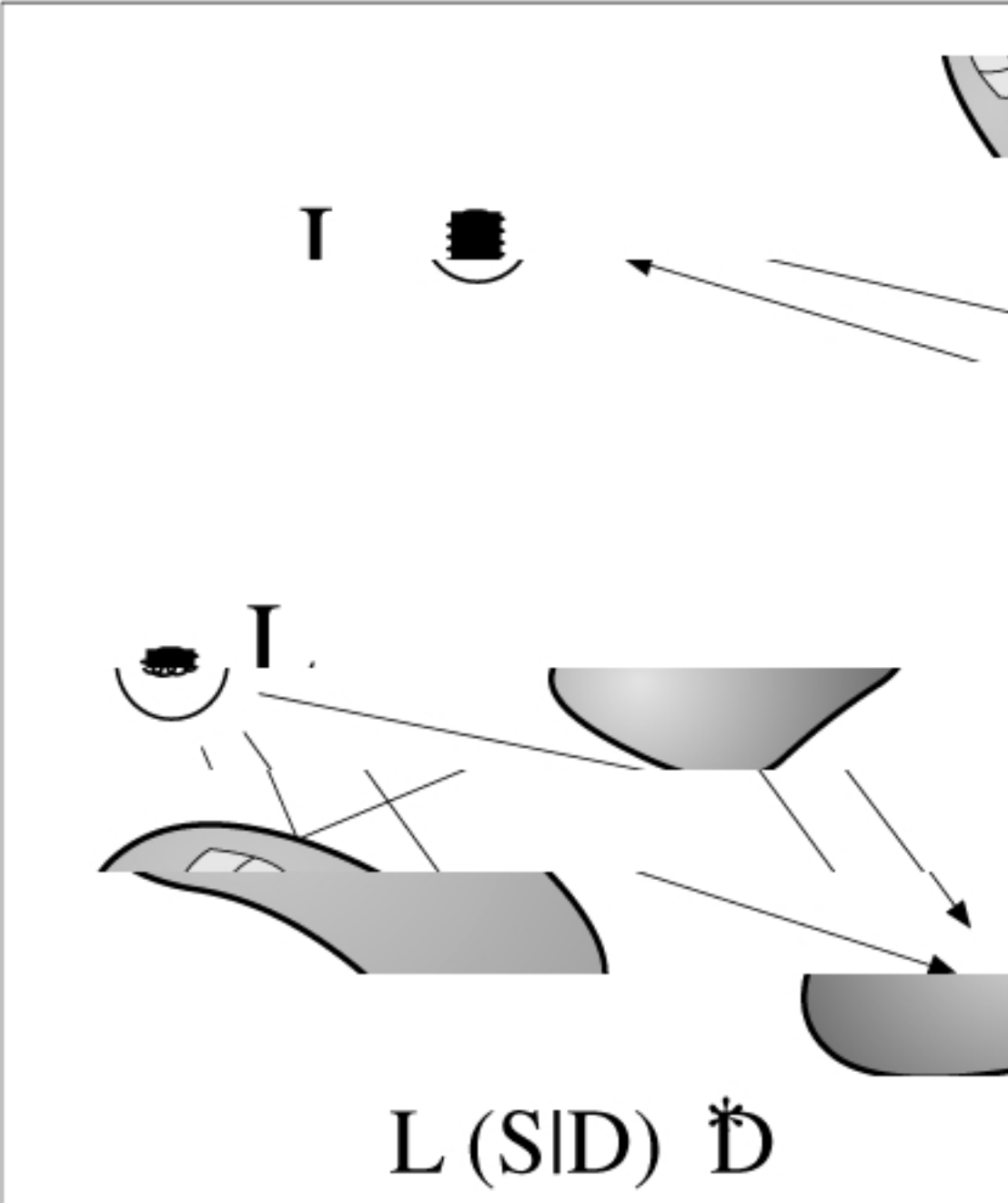


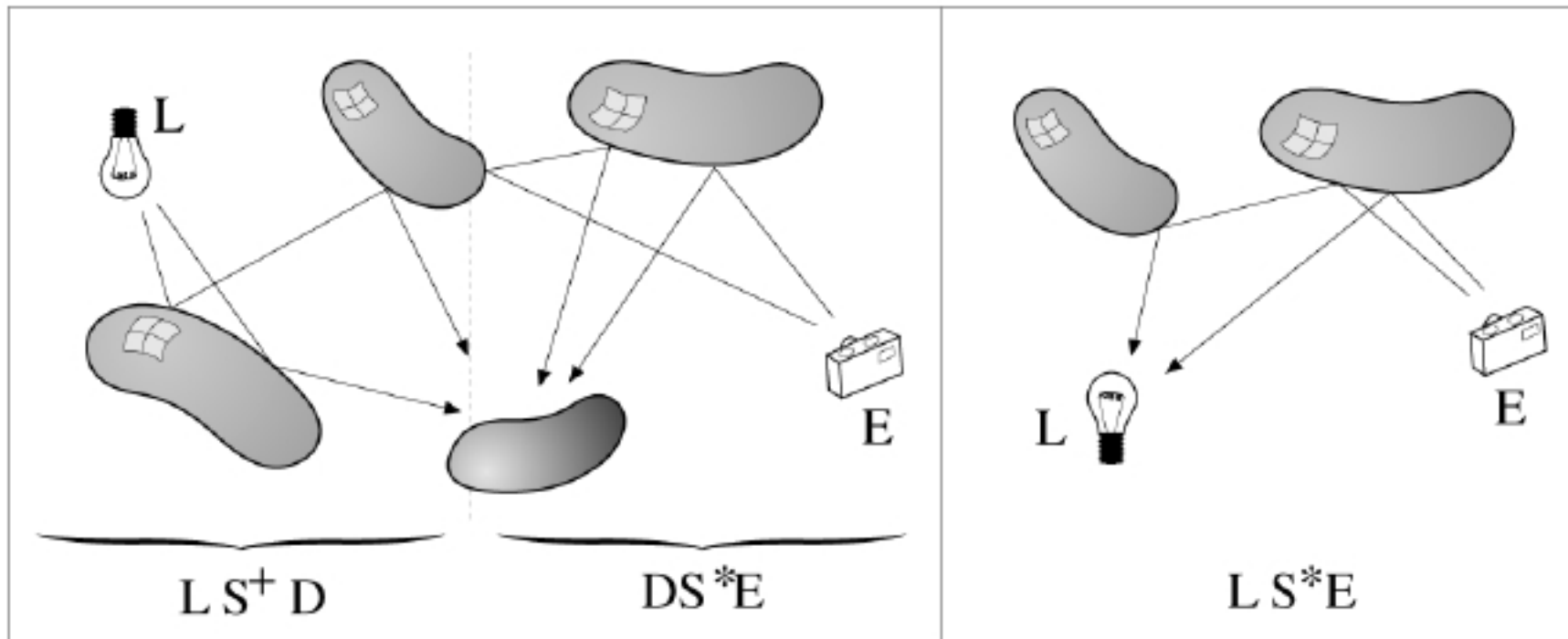
# Méthodes multi-passes

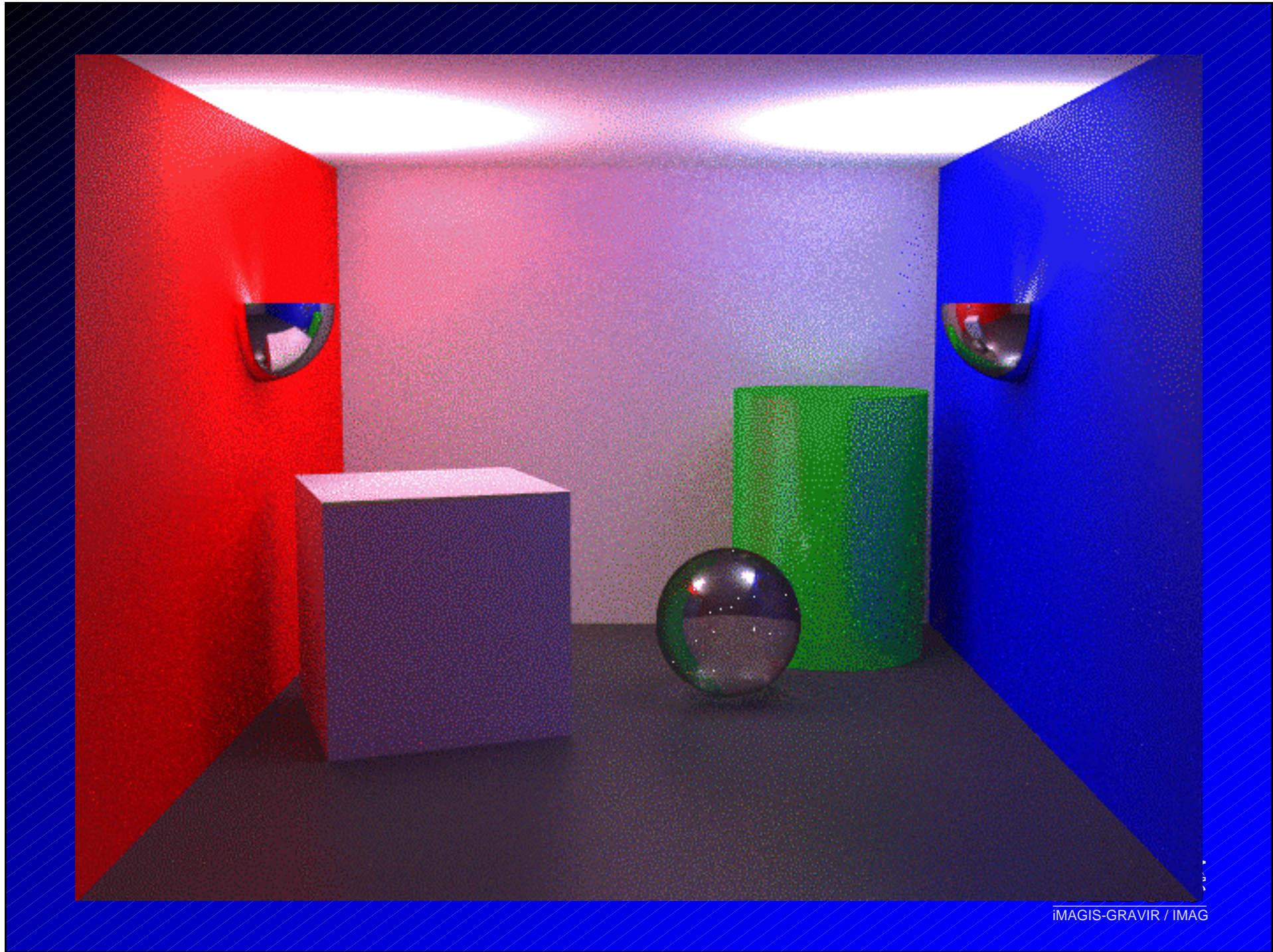
- MCRT :
  - moins il y a de rayons, plus ça converge vite
  - Cas idéal : toutes les surfaces sont presque spéculaires
  - Cas le pire : toutes les surfaces sont diffuses
- Idée :
  - Traiter chaque chemin par la méthode adaptée
    - Les surfaces diffuses par la radiosit 
    - L' clairage direct par Eye Ray-Tracing
    - Les caustiques par Light Ray-Tracing
    - Les choses compliqu es par MCRT

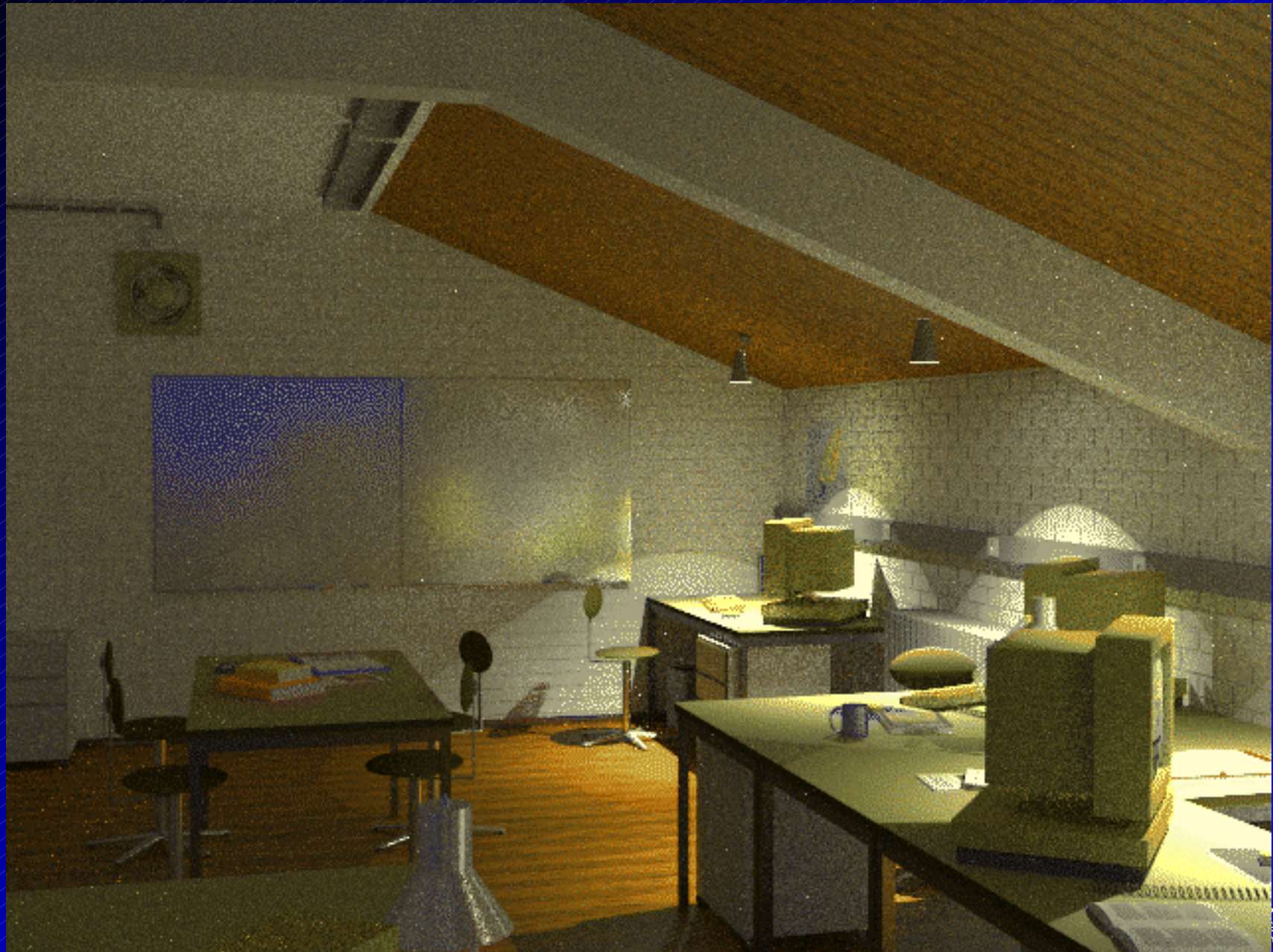
# Combinaison des méthodes

- Problèmes :
  - Ordre des méthodes
  - Représentation commune (LRT/Radiosité)
- Ordre :
  - Light Ray-Tracing
  - Passe de radiosité
  - Passe de MCRT
  - Passe de Eye Ray-Tracing















Bi-directional Path Tracing, 40 échantillons par pixel



# Méthodes de Monte-Carlo

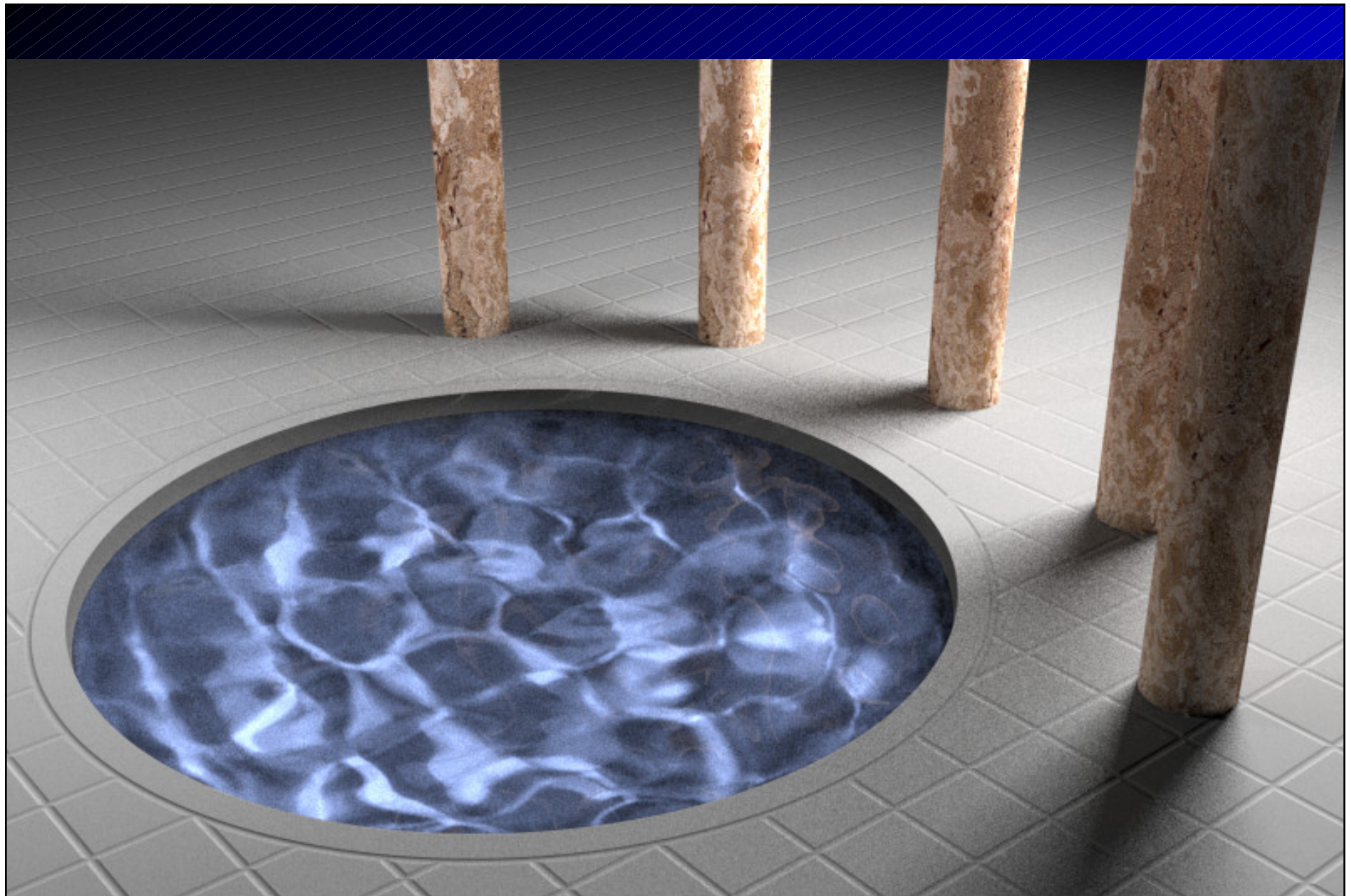
- Tous les types de matériaux, tous les objets, toutes les réflexions
- Facile à implémenter
  - Difficile à implémenter de façon efficace
- Solution de référence
- Convergence très lente
- Images très bruitées au début

# Méthodes de Monte-Carlo

- Accélération possibles :
  - Bi-Directional Path Tracing
  - Photon Maps
  - Metropolis light transport



Metropolis LT, 250 mutations par pixel (même temps de calcul)



Metropolis LT, 100 mutations par pixel (même temps de calcul)





HENRIK WANN JENSEN 1996

MAGIS-GRAYIR/IMAG

Rendu complet, 5 mn+51 mn



Figure 5: Dusty room illuminated by sunlight through a stained glass window.



Figure 6: Caustics in a swimming pool seen through a wavy water surface.