
Synthèse d'images

Xavier Granier <xavier.granier@institutoptique.fr>

Bibliographie pour ce cours

Livres en Français

Informatique graphique et rendu

Bernard Péroche, Dominique Bechmann (éditeurs)Hermes Publication

Livres en Anglais

Advanced Global Illumination

Philip Dutré, Kavita Bala, Philip Bekaer - A.K. Peters

Physically Based Rendering

Matt Pharr and Greg Humphreys - Morgan Kaufmann

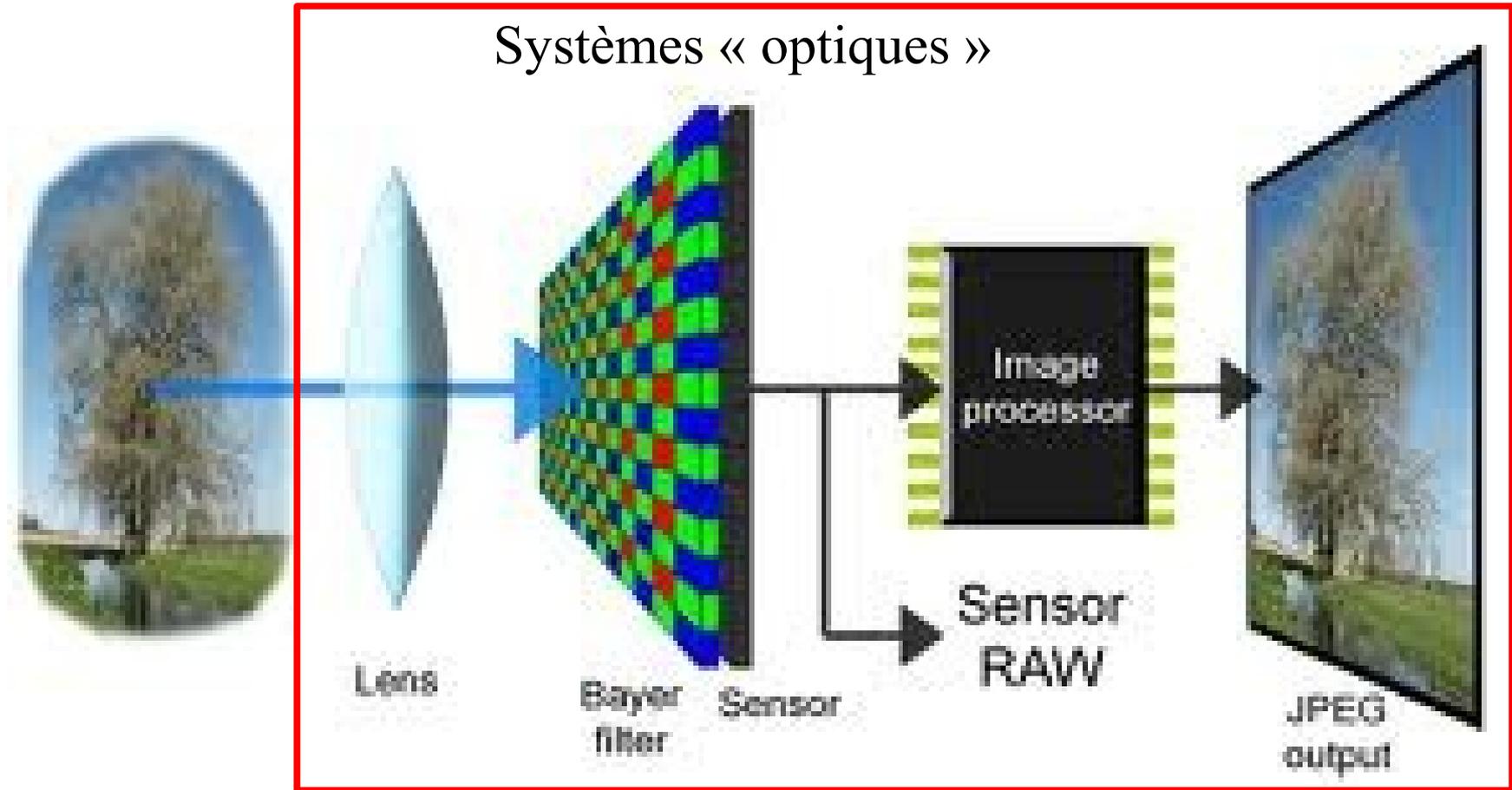
Realistic image synthesis using photon mapping

Henrik Wann Jensen – A.K. Peters

Qu'est-ce que la synthèse d'image?

- Selon vous?

Génération des images



trustedreviews.com

Abberations – Lens Flare



Polynomial Optics: A Construction Kit for Efficient Ray-Tracing of Lens Systems

Matthias B. Hullin, Johannes Hanika and Wolfgang Heidrich

In: Computer Graphics Forum / Proceedings of EUROGRAPHICS Symposium on Rendering (EGSR) 2012

Profondeur de champ



Fourier Depth of Field

C. Soler, K. Subr, F. Durand, N. Holzschuch, F. X. Sillion

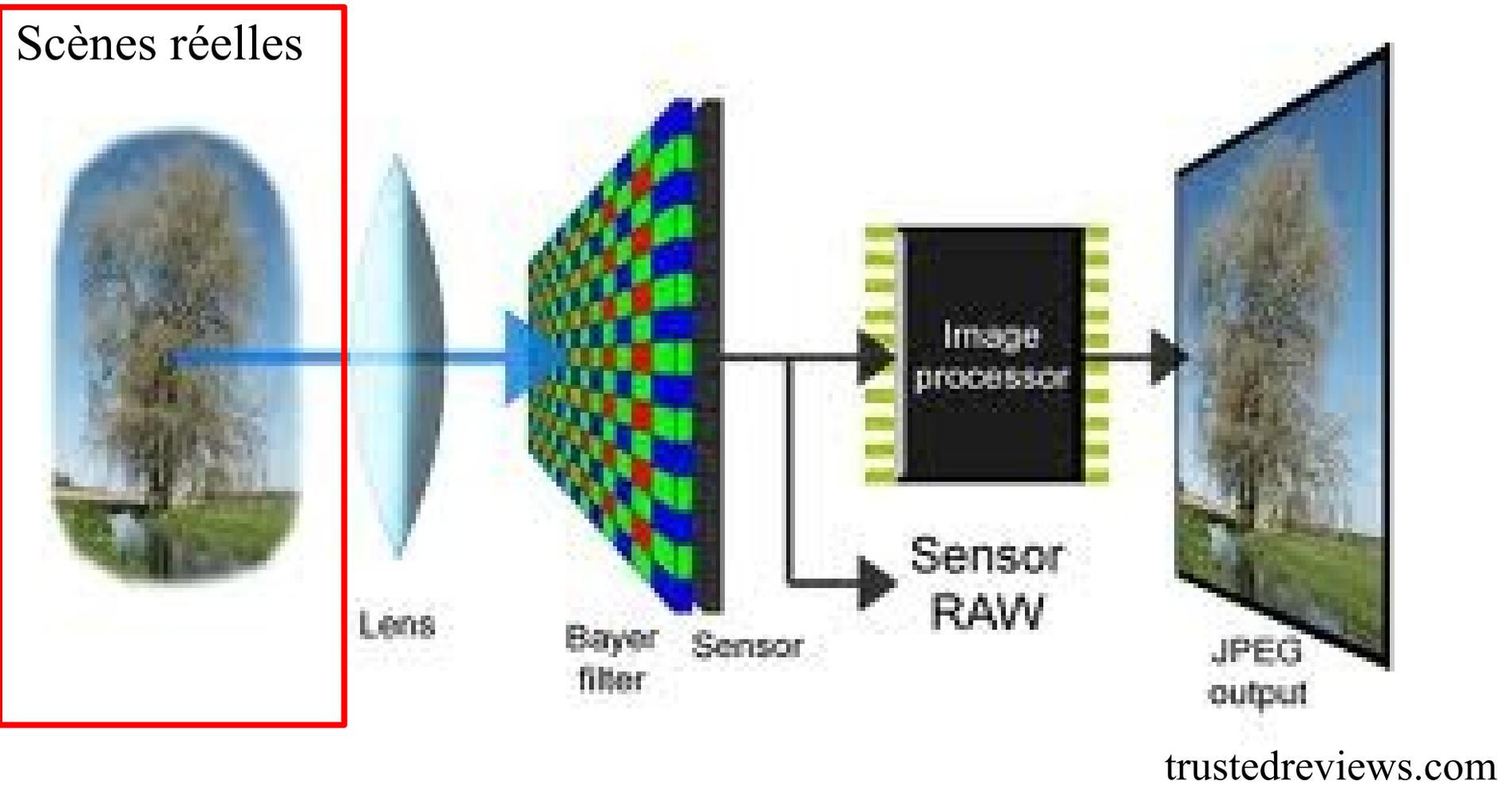
ACM Transactions on Graphics 28, 2 (2009) 18:1--18:12

Flou de mouvement



Crytek

Génération des images

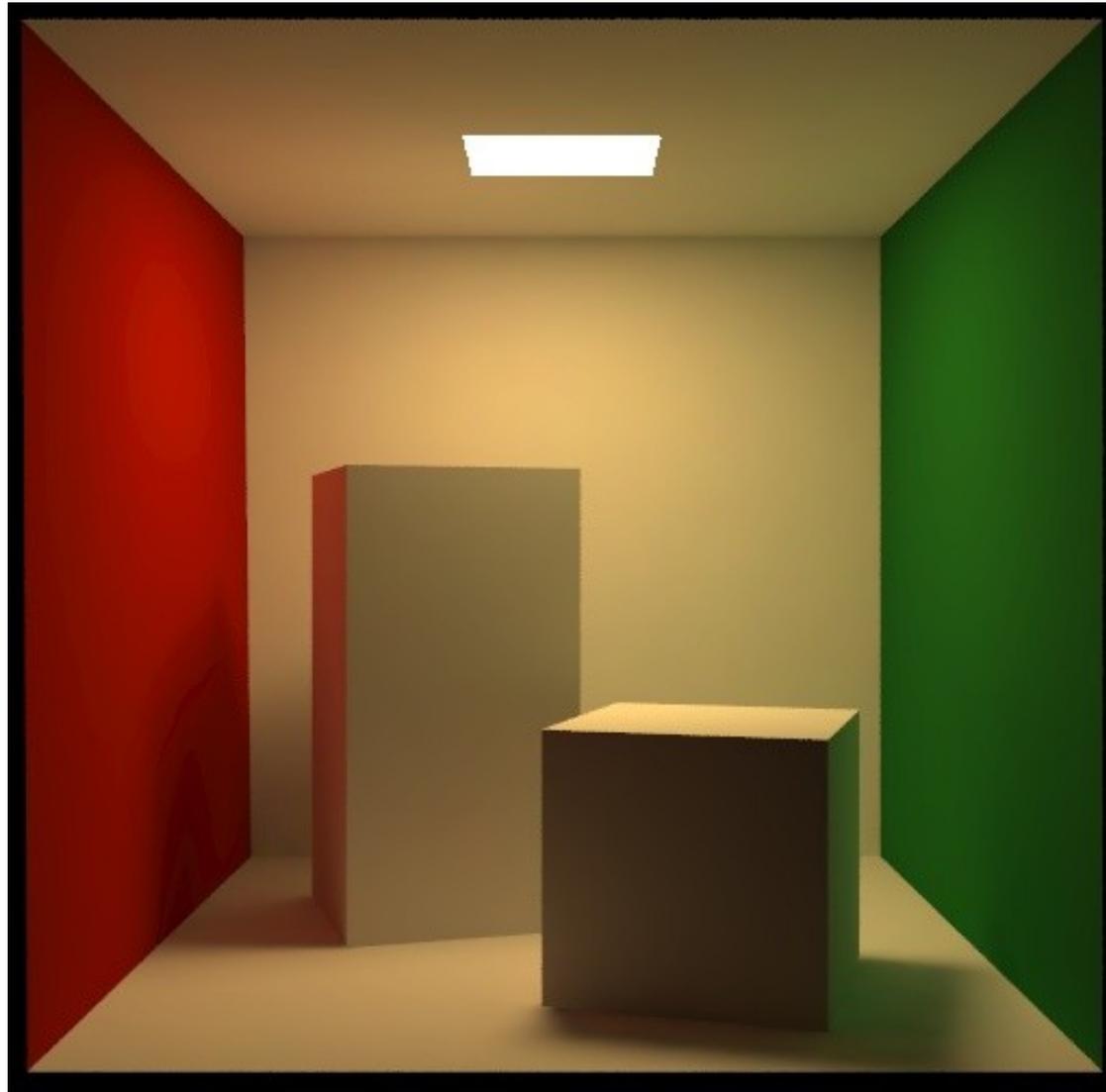


Caustiques



Realistic Image Synthesis Using Photon Mapping
Henrik Wann Jensen

Éclairage indirect



Éclairage indirect



Ombres douces directes et indirectes



Herf

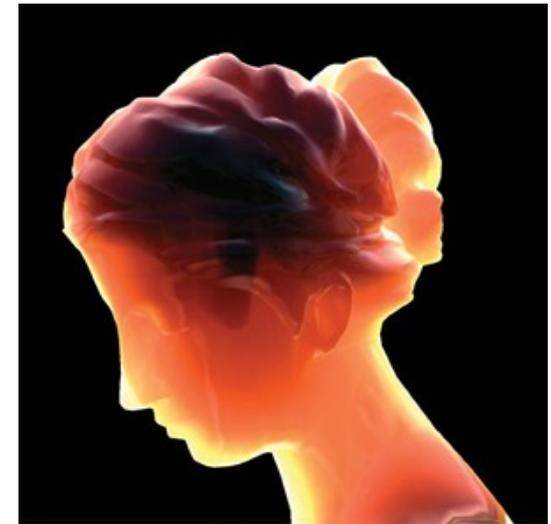
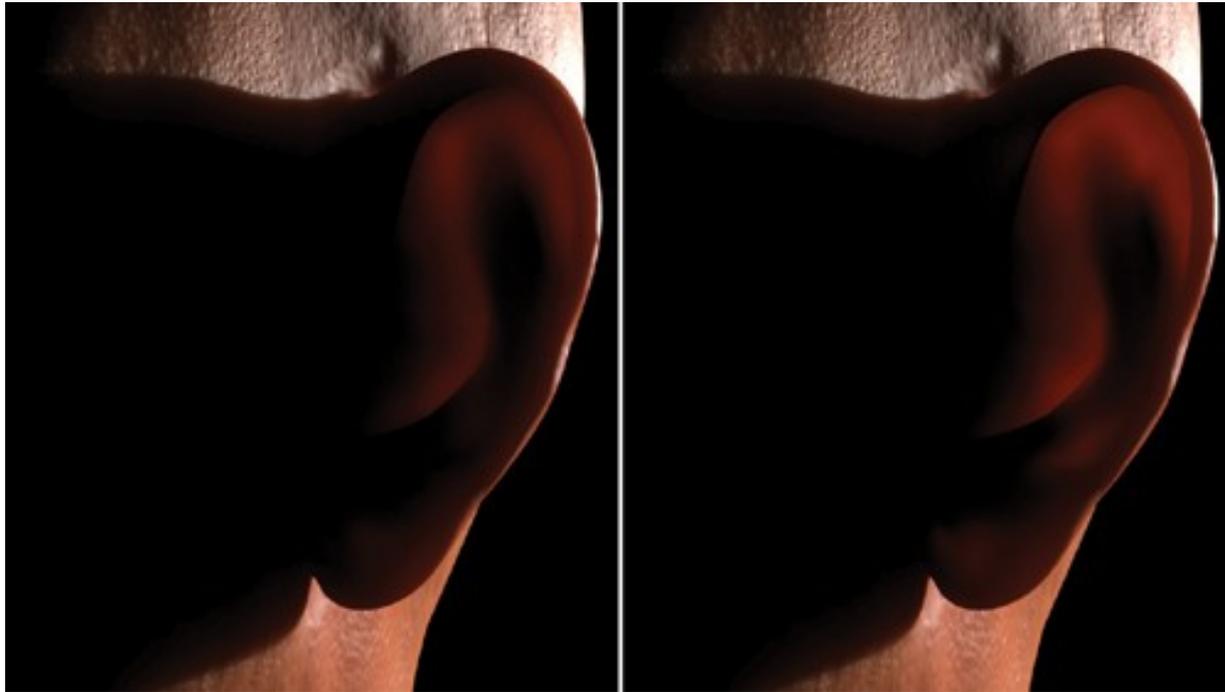
Phénomènes complexes

- Propriété d'absorption/re-émission de la lumière



Matériau diffusant

- Matériaux transparents, translucides
 - Diffusion dans le volume (ex : peau humaine)



La synthèse d'image

- Modèles
 - Sources de lumières
 - Réflexions, réfractions, diffusion, ...
 - Géométries / Volumes
 - Systèmes optiques
- Propagation de la lumière
 - Principes de radiométries
 - Optique géométrique
 - (Optique ondulatoire de temps en temps)
- Méthodes numériques de simulation

Méthodes de simulation

- Éléments finis
 - Objet mathématique : la maille
 - Intégration déterministe
 - Grandeur principale : éclairement
- Suivi de chemin lumineux
 - Objet mathématique : le rayon
 - Intégration stochastique
 - Grandeur principale : luminance
- Densité de particules
 - Objet mathématique : le photon, la particule
 - Estimation de densité
 - Grandeur principale : le flux

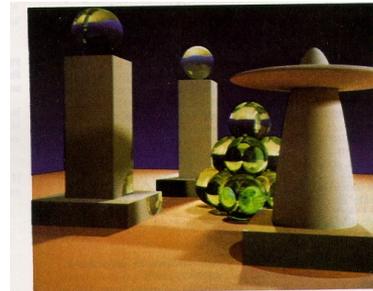
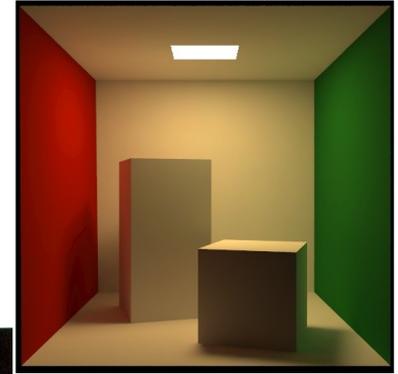
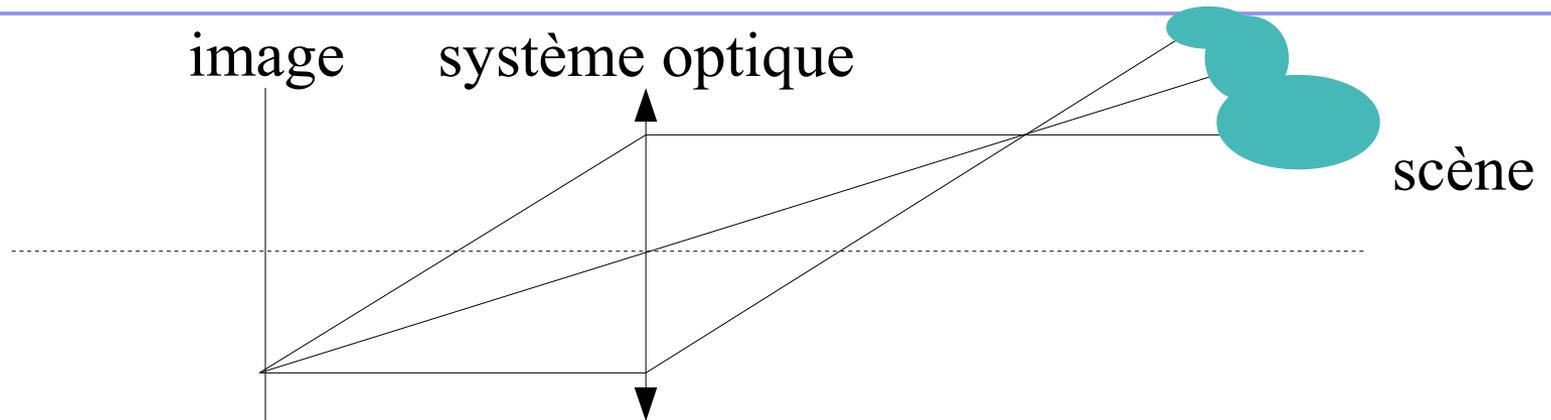


Figure 6. A sample image. All objects are neutral grey is due to caustics from the green glass balls and color t polygon.



Lancer de rayon

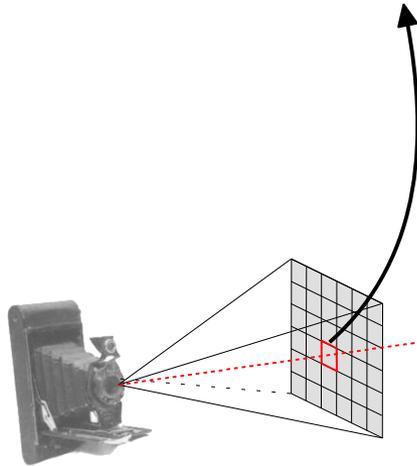


- Le principe
 - Partir de l'imageur
 - Rechercher l'ensemble des chemins lumineux
 - Calculer la luminance pour chacun d'un
- Approche
 - Lancer des rayons

En synthèse d'images

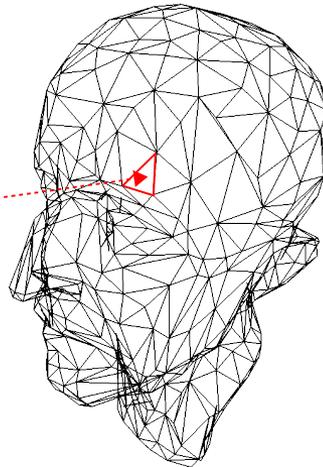
problématique :

- quelle est la couleur de chacun des pixels ?



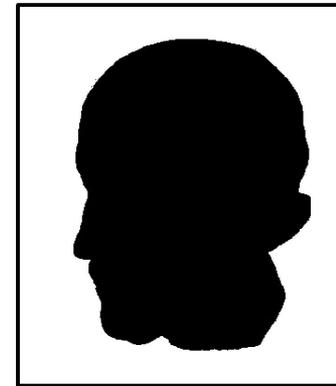
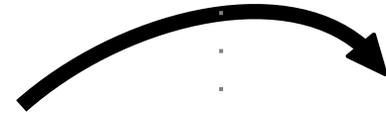
caméra virtuelle

- tableau de pixels RGB
- position, orientation
- résolution
- etc.



modèles 3D

Cours sur la géométrie des surfaces

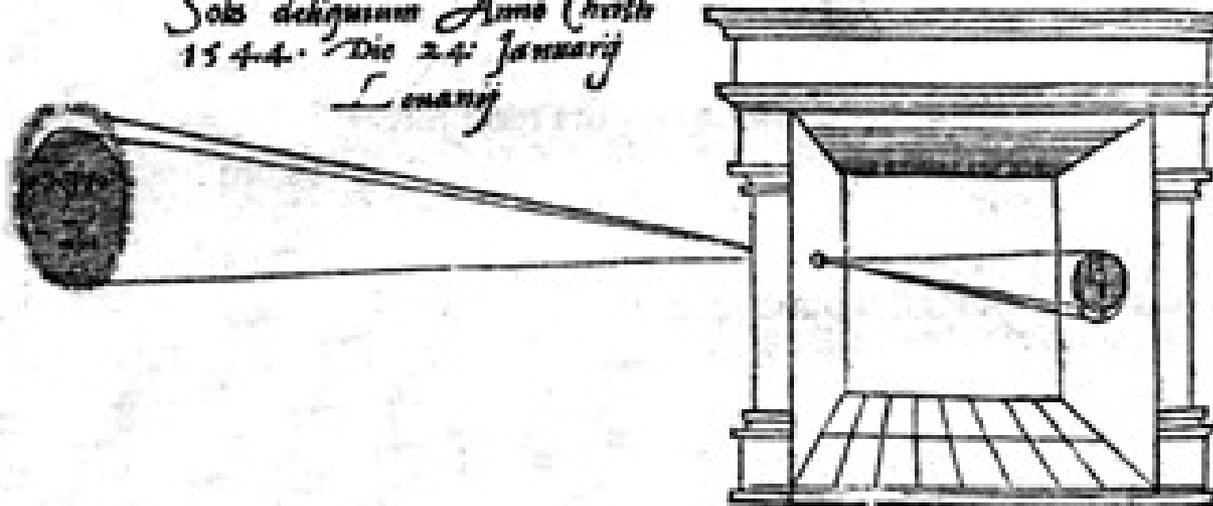


2D image

Rappel : Modèle de camera – Sténopé

illum in tabula per radios Solis, quàm in cœlo contin-
git: hoc est, si in cœlo superior pars deliquiū patiat, in
radiis apparebit inferior deficere, vt ratio exigit optica.

*Soli deliquium Anno Christi
1544. Die 24. Januarij
Louanij*



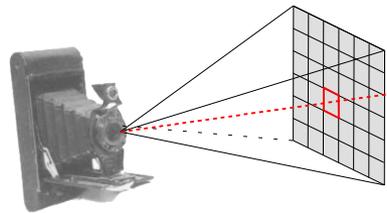
Sic nos exactè Anno .1544. Louanii eclipsim Solis
obseruauimus, inuenimusq; deficere paulò plus q̄ dex-

En synthèse d'images

sources de lumière

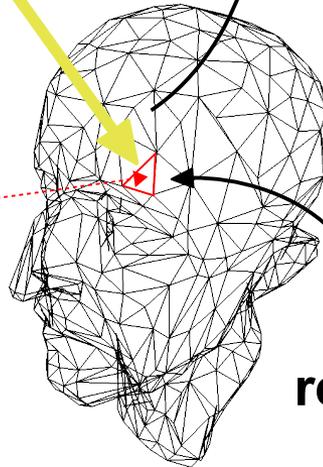


matériaux



caméra virtuelle

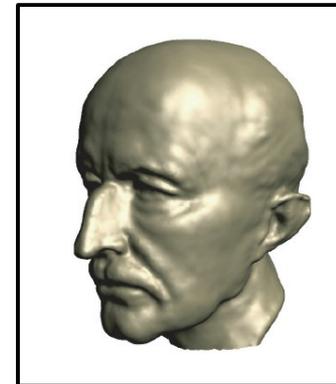
- tableau de pixels RGB
- position, orientation
- résolution
- etc.



réflexions

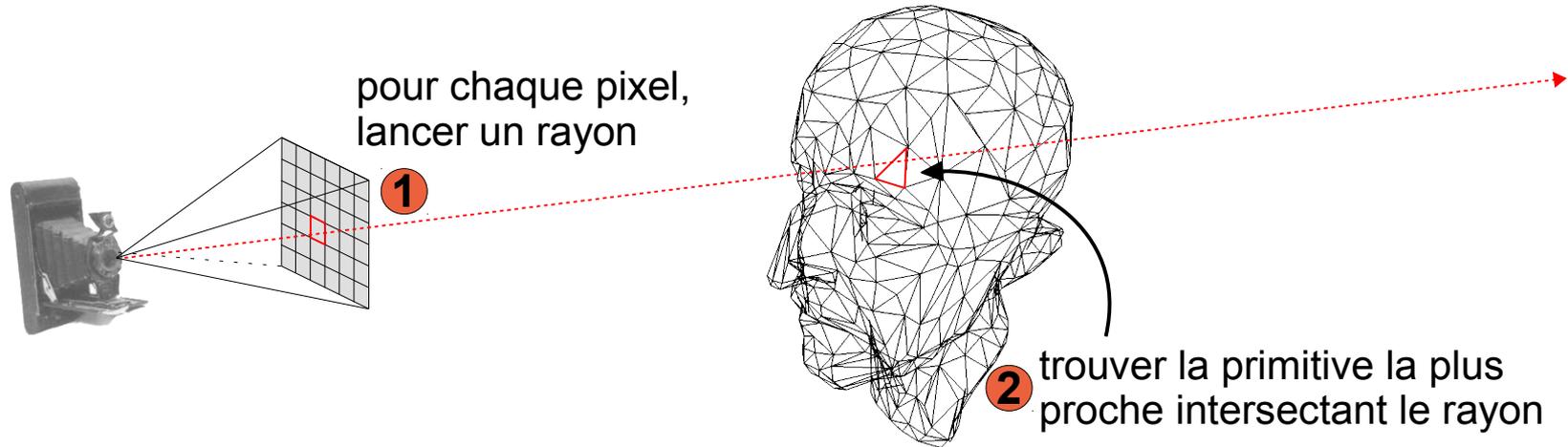
modèles 3D

Cours sur la géométrie
des surfaces



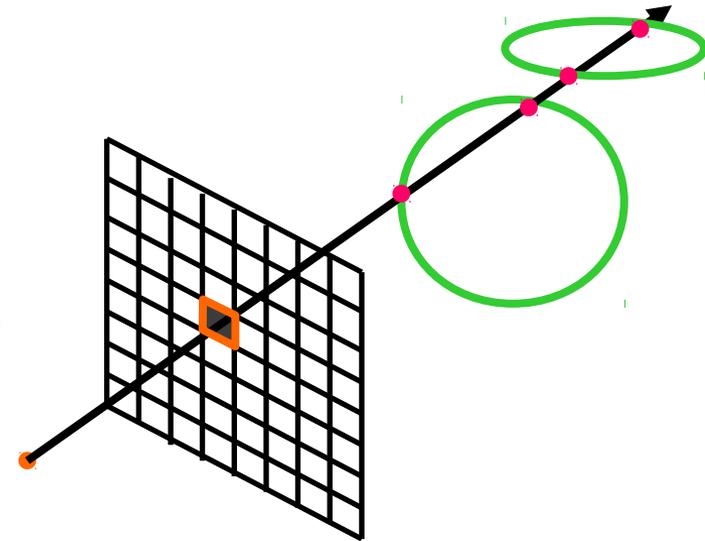
2D image

Lancer de rayons : principe

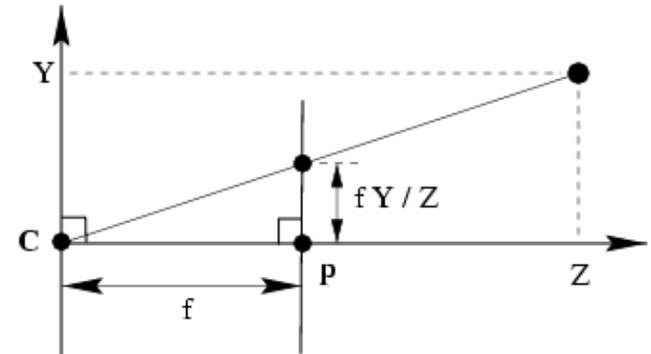
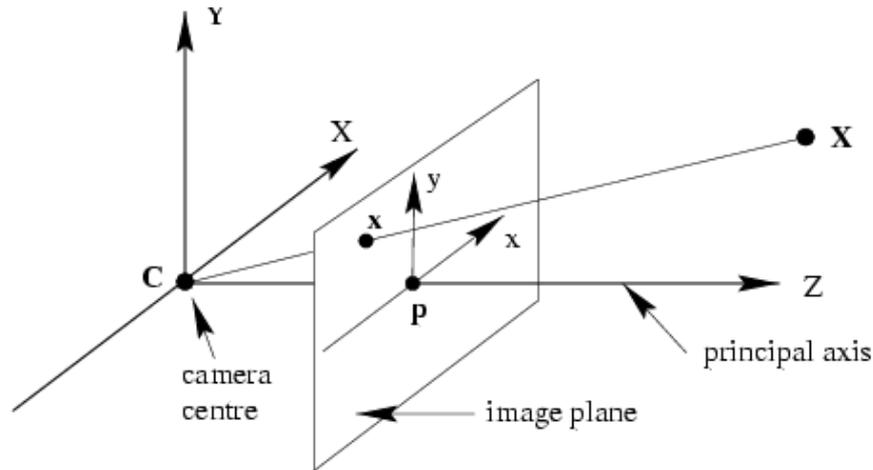


- **Lancer de rayons (Ray-tracing – Ray casting)**

- Pour chaque rayon, trouver la primitive la plus proche intersectant le rayon
- Propagation d'un rayon a partir des pixels
- scène = ensemble de primitives « intersectables »



Rappel : projection perspective



$$[x, y, z]^T \rightarrow [f x/z, f y/z]^T$$

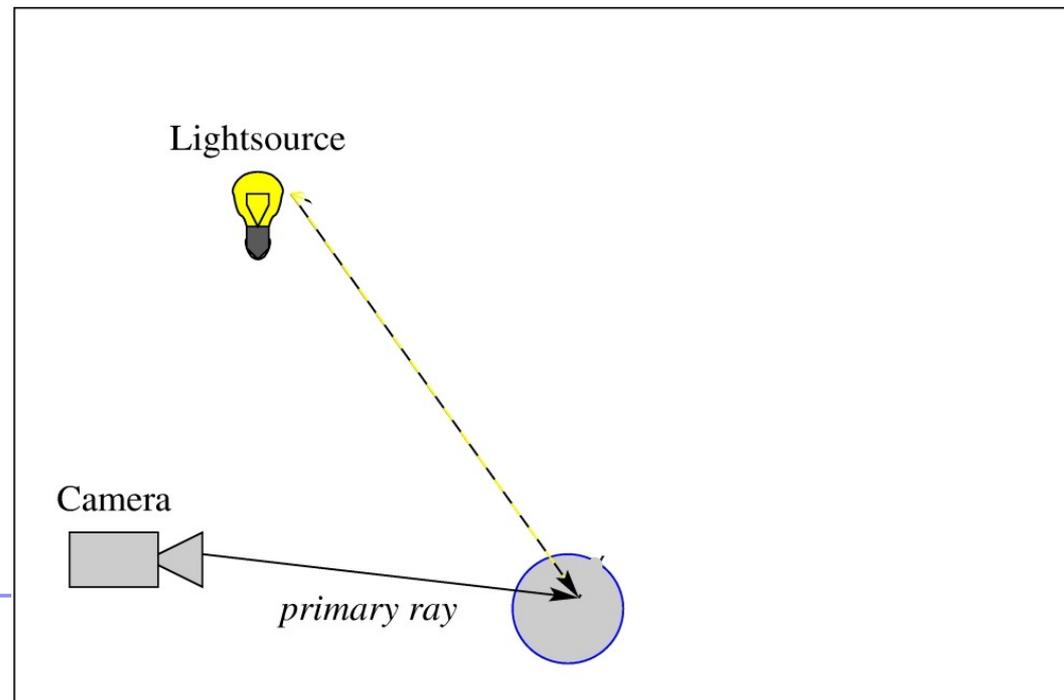
$$\begin{bmatrix} f x \\ f y \\ z \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Comparaison avec la rastérisation

- Rastérisation:
 - Pour chaque primitive P_i , trouver les rayons intersectant P_i qui passent par un pixel
 - Rendu en deux étapes
 - projection des primitives sur l'écran (*forward projection*)
 - discrétisation (conversion des primitives en pixels)
- Lancer de Rayon (Ray-tracing – Ray casting)
 - Pour chaque rayon, trouver la primitive la plus proche intersectant le rayon
 - Propagation d'un rayon a partir des pixels

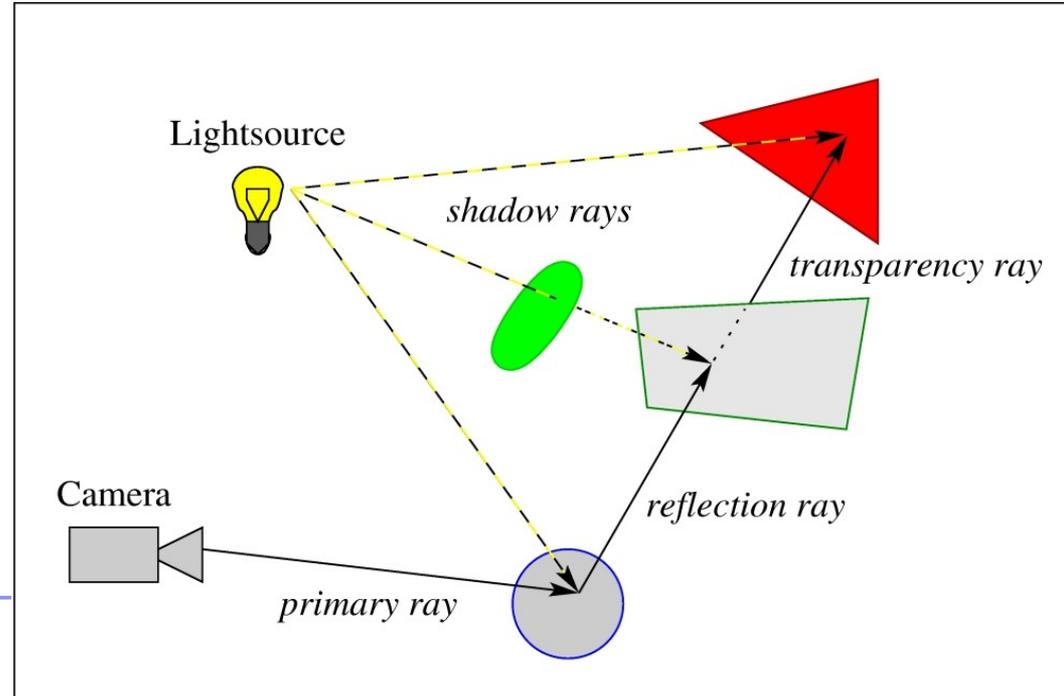
Lancer de rayon: Principe

- Un rayon =
 - Une origine + ne direction
- Génération des rayons
 - Rayons primaires: rayons émis à partir d'un point de vue, au travers d'une image, vers la scène
- Lancer de rayon
 - Trouver la plus proche intersection avec la scène 3D
- Calcul de l'apparence
 - Éclairement direct à partir des sources de lumières
 - Éclairement indirect par propagation



Lancer de rayon: Principe

- Un rayon =
 - Une origine + ne direction
- Génération des rayons
 - Rayons primaires: rayons émis à partir d'un point de vue, au travers d'une image, vers la scène
- Lancer de rayon
 - Trouver la plus proche intersection avec la scène 3D
- Calcul de l'apparence
 - Éclairement direct à partir des sources de lumières
 - Éclairement indirect par propagation



Composantes/Difficultés majeurs

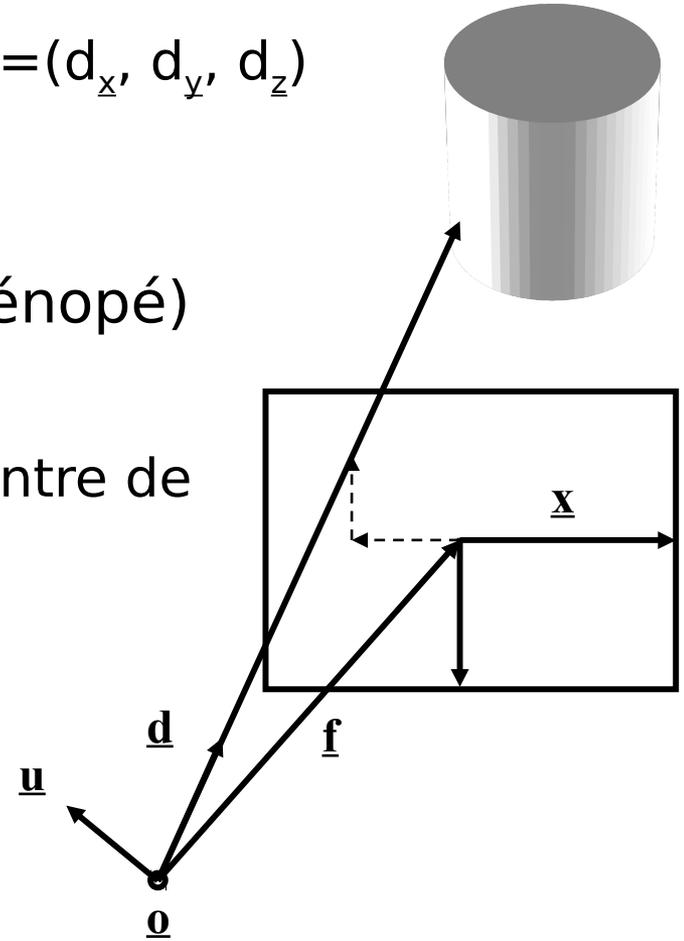
- Génération des rayons primaires
- Tracé de rayon
 - intersections rayon/objet
 - trouver l'intersection la plus proche le plus rapidement possible !
- Éclairement
 - ombres, réflexions, réfractions, etc.

Composantes/Difficultés majeurs

- Génération des rayons primaires
- Tracé de rayon
 - intersections rayon/objet
 - trouver l'intersection la plus proche le plus rapidement possible !
- Éclairement
 - ombres, réflexions, réfractions, etc.

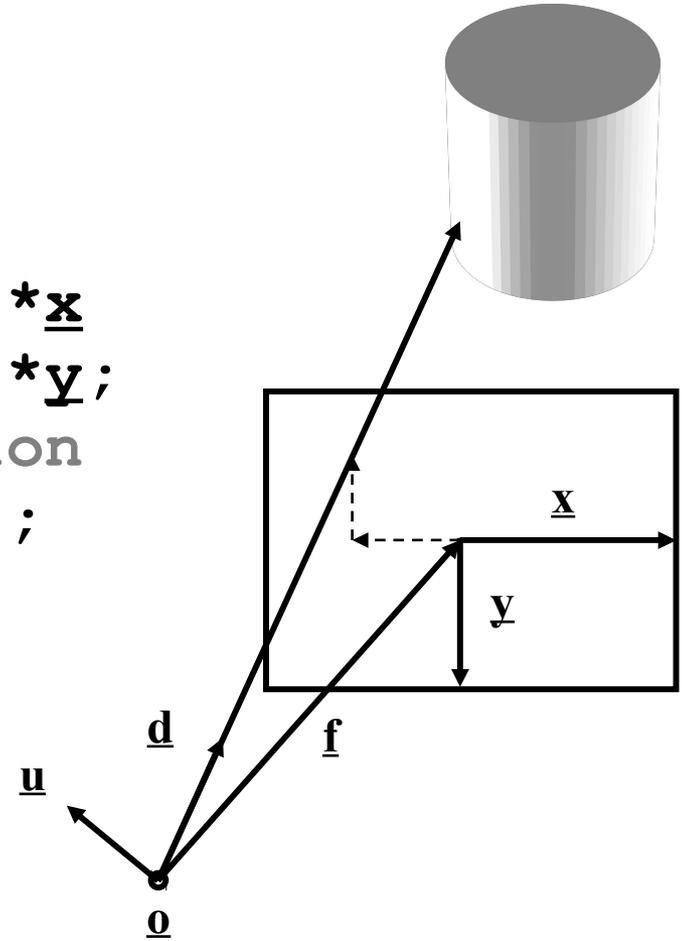
Génération des rayons primaires

- Un rayon: $\underline{r}(t) = \underline{o} + t \underline{d}$
 - origine $\underline{o} = (o_x, o_y, o_z)$, direction $\underline{d} = (d_x, d_y, d_z)$ avec $|\underline{d}| = 1$
- Hypothèse : “Pinhole camera” (sténopé)
 - \underline{o} : origine (point de vue)
 - \underline{f} : axe optique (vecteur vers le centre de l'image)
 - $\underline{x}, \underline{y}$: axes sur l'image
 - résolution de l'image
- Direction pour le pixel (i,j) ?



Génération des rayons primaires

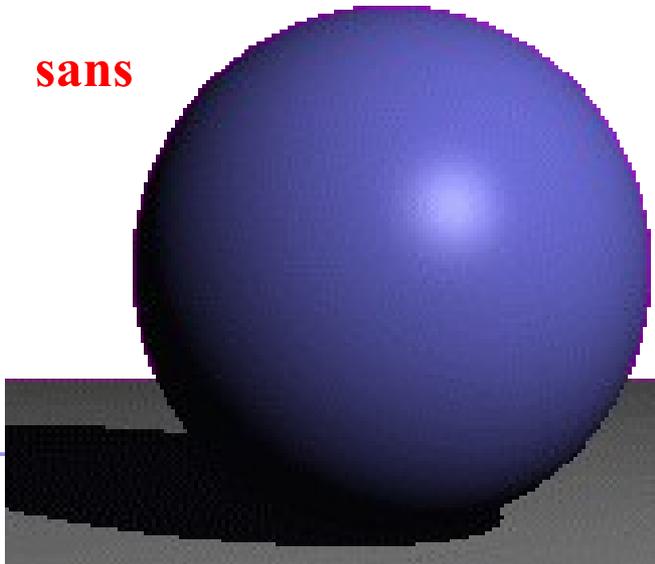
```
//pour chacun des pixels
for(i=0; i<width; i++)
  for(j=0; j<height; j++)
  {
    d = f + 2(i/width - 0.5)*x
        + 2(j/height - 0.5)*y;
    d = d/|d|; // normalisation
    color = ray_tracing(o, d);
    write_pixel(i, j, color);
  }
```



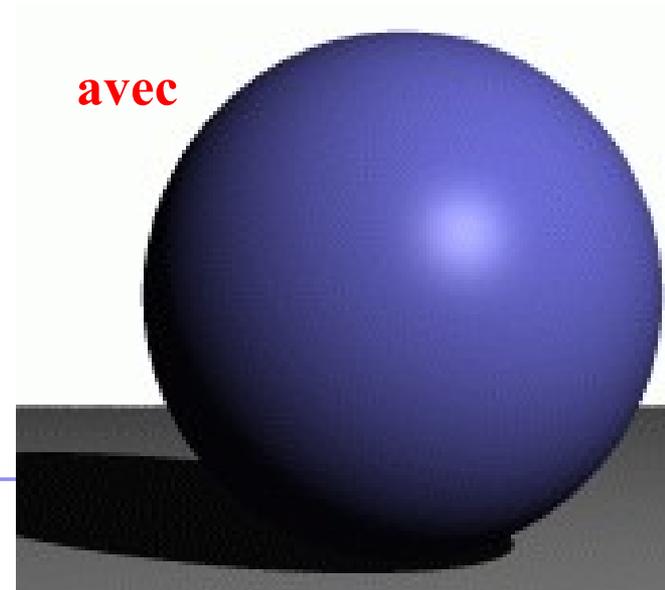
Antialiasing → sur-échantillonnage

- 1 pixel \neq 1 point → 1 pixel = petit élément de surface
 - 1 rayon par pixel : quel point choisir ? le centre ?? → aliasing
 - idéalement : intégrer sur toute la surface du pixel
- Générer plusieurs rayons par pixels (et moyenner)
 - optimisations possibles
 - sur-échantillonnage adaptatif → détecter les bords au préalable !
 - choisir les échantillons « intelligemment »

sans

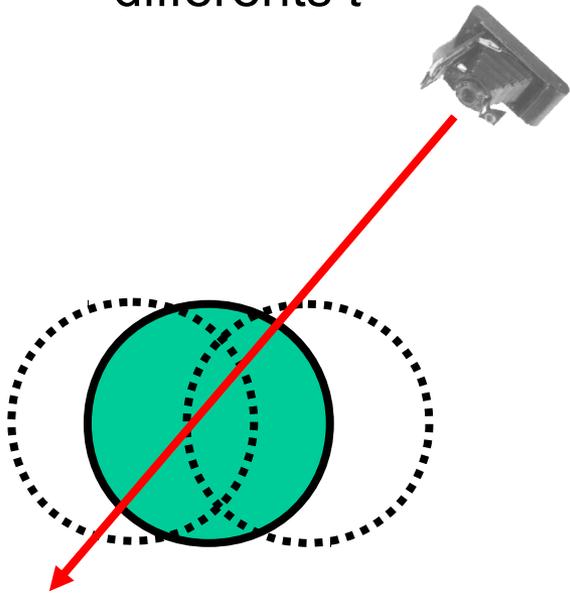


avec



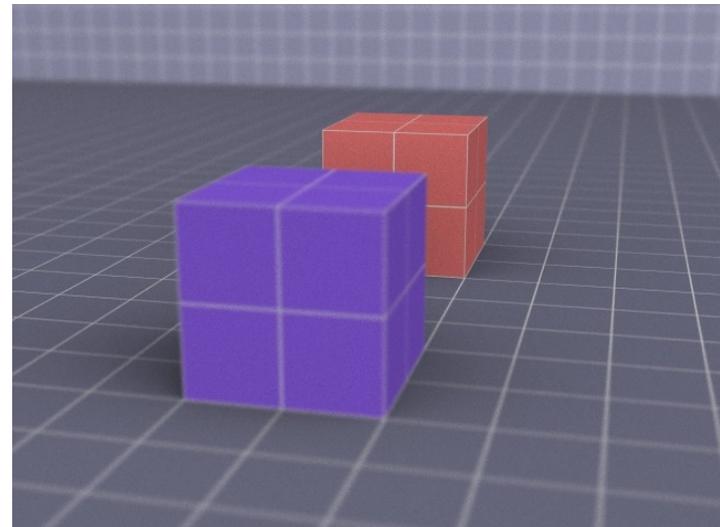
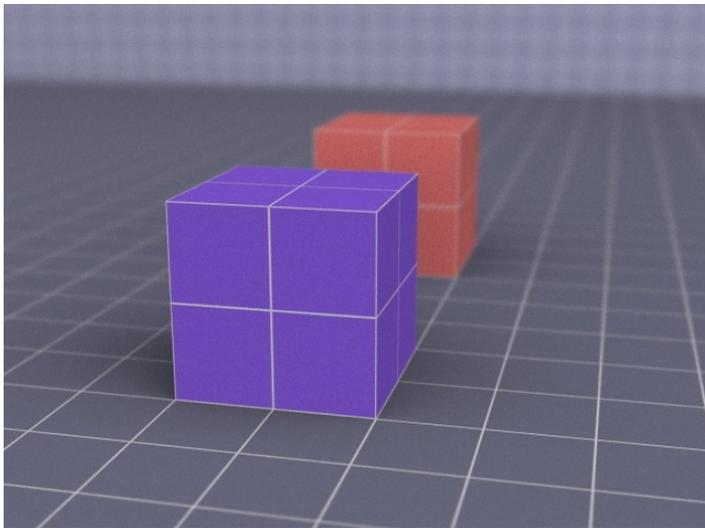
Flou de mouvement (Motion Blur)

- Caméra réaliste : durée d'ouverture > 0
→ intégrer sur un interval de temps
- En pratique : sur-échantillonnage temporel
 - moyenner plusieurs images calculer à différents t



Profondeur de champ (Depth of Field)

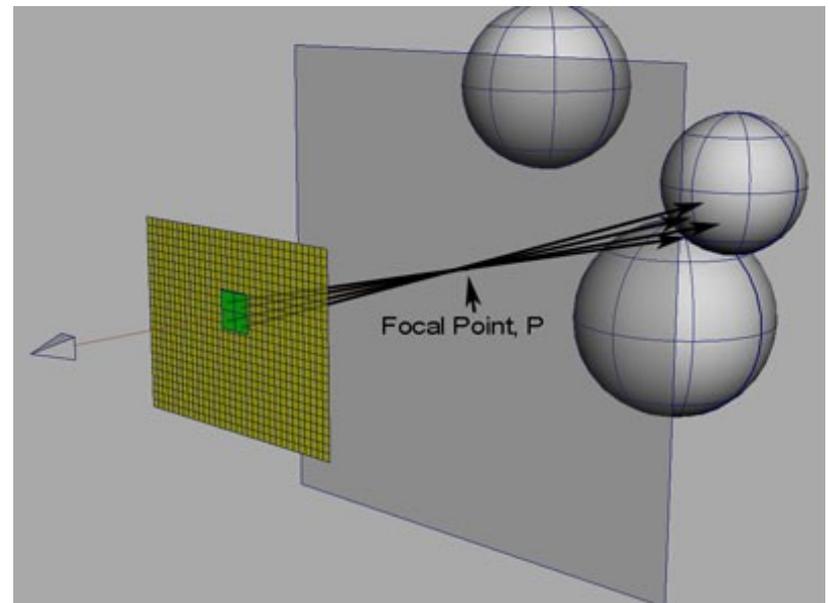
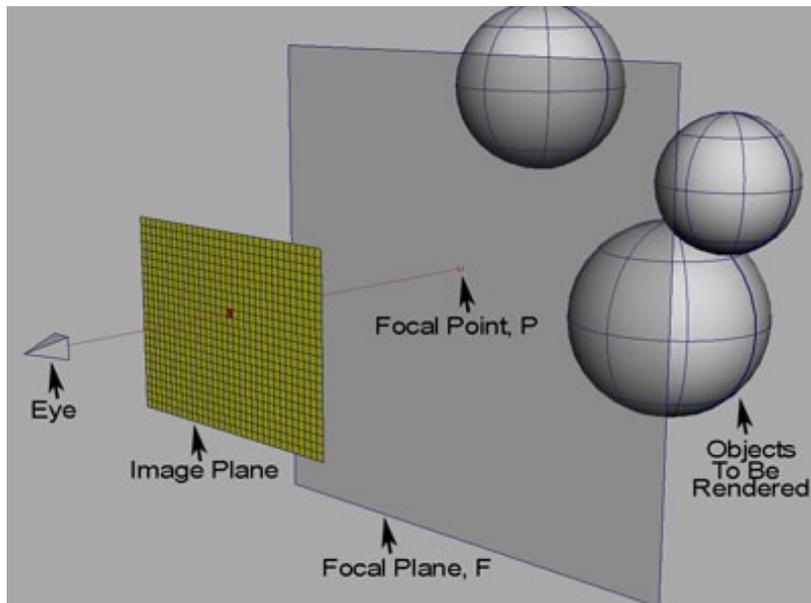
- Modèle réaliste
 - sténopé → système optique (lentilles)
- En pratique : tracer plusieurs rayons / pixels
 - changements de direction du à l'objectif.



Justin Legakis

Simuler l'ouverture

- Garder le principe du rayon
 - 1 point sur sur le plan image + 1 direction
- Introduire un plan focale
 - P = intersection rayon – plan
- Pour chaque rayon : une ouverture virtuelle
 - Plusieurs rayons par P



Composantes/Difficultés majeurs

- Génération des rayons primaires
- Tracé de rayon
 - intersections rayon/objet
 - trouver l'intersection la plus proche le plus rapidement possible !
- Éclairement
 - ombres, réflexions, réfractions, etc.

Exercices : Intersections rayons / objets

- Un rayon: $\underline{r}(t) = \underline{o} + t \underline{d}$
 - $\underline{o} = (o_x, o_y, o_z)$, $\underline{d} = (d_x, d_y, d_z)$
- Géométrie de la scène
 - Plan : $(\underline{p} - \underline{a}) \cdot \underline{n} = 0$
 - Définition implicite (\underline{n} : normale à la surface, \underline{a} : un point sur le plan)
 - Sphère : $(\underline{p} - \underline{c})^2 - r^2 = 0$
 - \underline{c} : centre de la sphère, r : rayon de la sphère
 - Triangle : Partie d'un plan

Intersection rayon/plan

- Équation du plan : $\underline{x} \cdot \underline{n} - D = 0$, $|\underline{n}| = 1$
 - Normale : \underline{n}
 - Distance du plan au centre (0, 0, 0) : $D = \underline{a} \cdot \underline{n}$
 - Remplacer \underline{x} par l'équation du rayon $\underline{o} + t\underline{d}$
 - $(\underline{o} + t\underline{d}) \cdot \underline{n} - D = 0$
 - La solution devient $t = \frac{D - \underline{o} \cdot \underline{n}}{\underline{d} \cdot \underline{n}}$

Intersection avec un triangle (1)

- Coordonnées barycentriques

- Pour un triangle non-dégénéré ABC

$$\underline{X} = \lambda_1 \underline{A} + \lambda_2 \underline{B} + \lambda_3 \underline{C}$$

- $\lambda_1 + \lambda_2 + \lambda_3 = 1$

- $\lambda_3 = A(\underline{A}\underline{X}\underline{B}) / A(\underline{A}\underline{C}\underline{B})$

- Aire relative signée

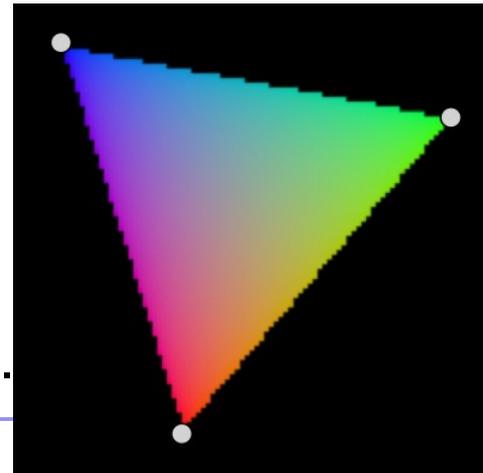
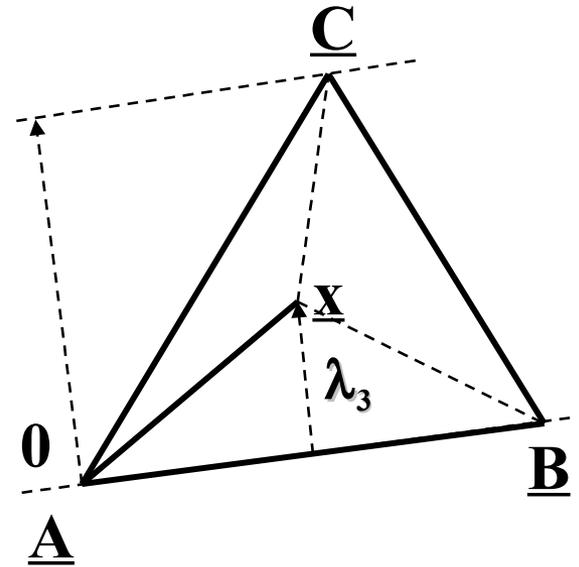
$$A(\underline{A}\underline{X}\underline{B}) = \frac{1}{2} (\underline{B} - \underline{A}) \wedge (\underline{X} - \underline{A})$$

- Test d'appartenance au polygone

- A l'intérieur si tous les λ_i sont plus grand ou égal à zéro

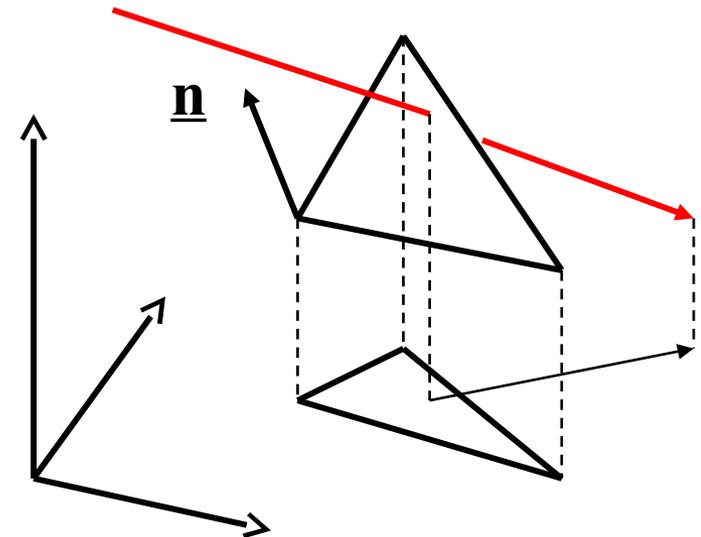
- Interpolation des attributs aux sommets

- normales, couleurs, coordonnées de texture, etc.



Intersection avec un triangle (1)

- Calculer l'intersection avec le plan
 - Attention aux cas tangents !
- Test d'appartenance au polygone en 2D
 - Projection sur un des plans du repère global (choix en fonction de la normale)
 - Test avec les coordonnées barycentriques



Intersection avec un triangle (2)

- Test d'intersection rapide sans normale

- t : distance à l'origine

- u, v : coordonnées dans le triangle (dans $[0, 1]$)

$$\underline{o} + t \underline{d} = u \underline{A} + v \underline{B} + (1 - u - v) \underline{C}$$

- système de 3 équations à 3 inconnues, forme matricielle :

$$\begin{bmatrix} -\underline{d} & \underline{A} - \underline{C} & \underline{B} - \underline{C} \end{bmatrix} \begin{bmatrix} t \\ u \\ v \end{bmatrix} = \underline{o} - \underline{C}$$

- optimisation : stop dès que u ou v négatif

- Méthode de référence !

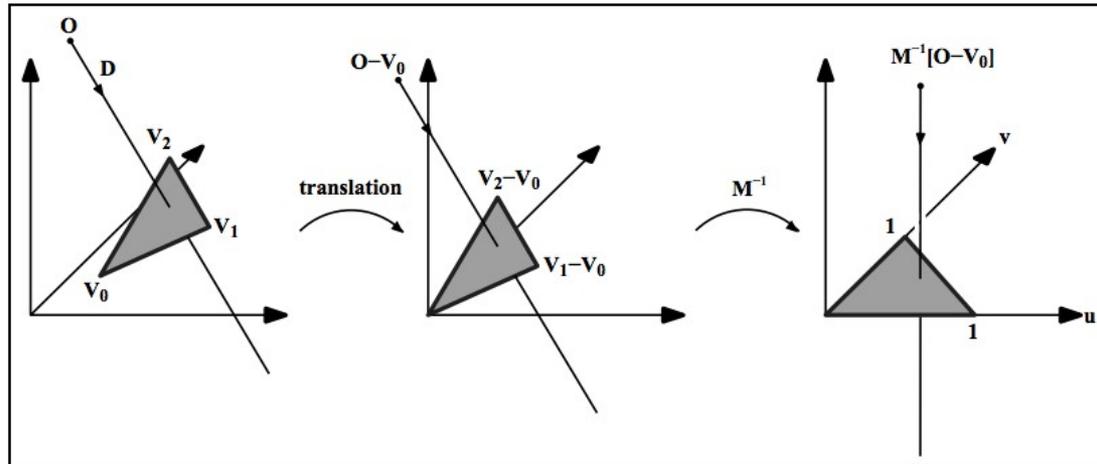
« Fast, minimum storage ray-triangle intersection »

Tomas Möller and Ben Trumbore, Journal of Graphics Tools

Intersection avec un triangle (2)

- Interprétation géométrique

$$\begin{bmatrix} -d & A-C & B-C \end{bmatrix} \begin{bmatrix} t \\ u \\ v \end{bmatrix} = \underline{o} - \underline{C}$$



- Cramer's rule :

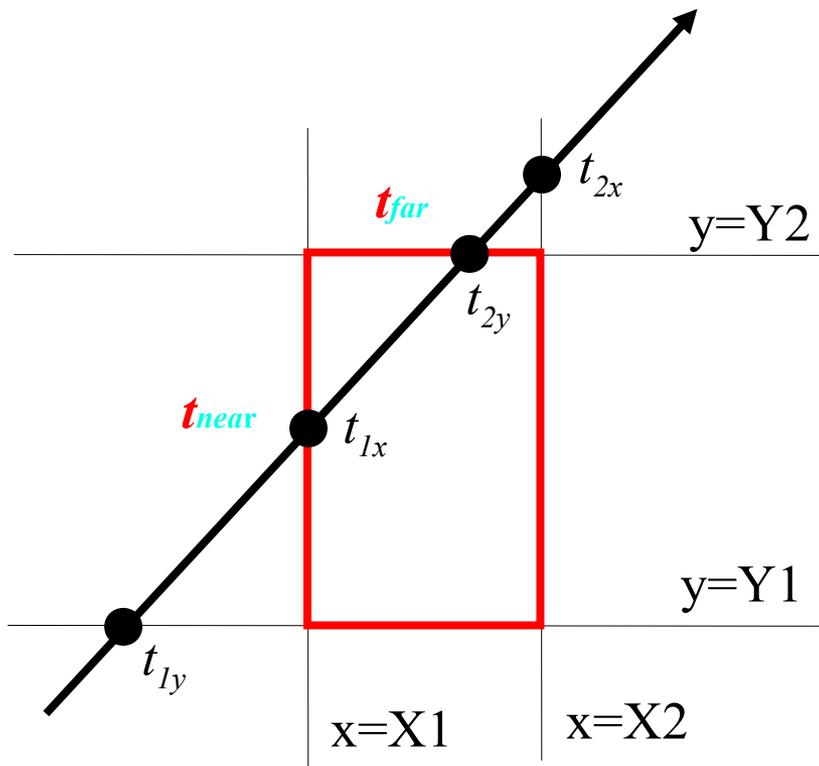
$$\begin{bmatrix} t \\ u \\ v \end{bmatrix} = \frac{1}{\begin{vmatrix} -D & E_1 & E_2 \end{vmatrix}} \begin{bmatrix} \begin{vmatrix} T & E_1 & E_2 \end{vmatrix} \\ \begin{vmatrix} -D & T & E_2 \end{vmatrix} \\ \begin{vmatrix} -D & E_1 & T \end{vmatrix} \end{bmatrix}$$

- Déterminant : $|A, B, C| = -(A \times C) \cdot B = -(C \times B) \cdot A$.

- D'où :

$$\begin{bmatrix} t \\ u \\ v \end{bmatrix} = \frac{1}{(D \times E_2) \cdot E_1} \begin{bmatrix} (T \times E_1) \cdot E_2 \\ (D \times E_2) \cdot T \\ (T \times E_1) \cdot D \end{bmatrix} = \frac{1}{P \cdot E_1} \begin{bmatrix} Q \cdot E_2 \\ P \cdot T \\ Q \cdot D \end{bmatrix}$$

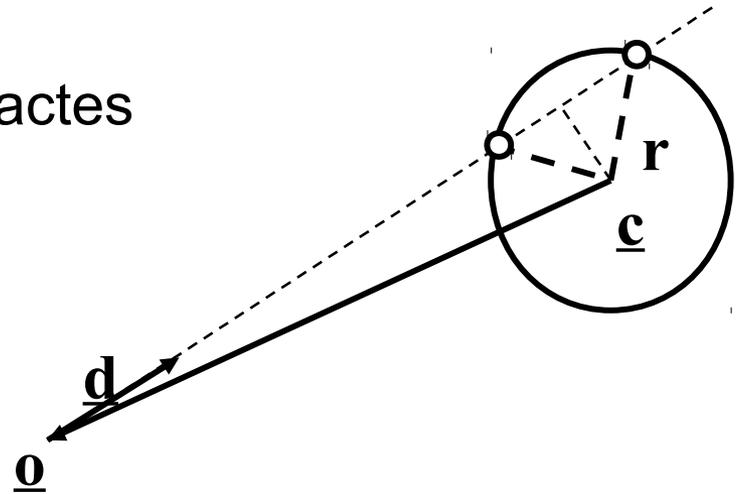
Boîte englobante alignée sur les axes



- Calculer les distances d'intersection t_1 et t_2 pour chaque axe
 - $t_{near} = \max(t_{1x}, t_{1y}, t_{1z})$
 - $t_{far} = \min(t_{2x}, t_{2y}, t_{2z})$
- Si $t_{near} > t_{far}$, pas d'intersection
- Si $t_{far} < t_{min}$, la boîte est derrière
- Sinon, l'intersection se fait à la distance t_{near}

Intersection rayon/sphère

- Étant donnée l'équation de la sphère $|\underline{x}-\underline{c}|^2-r^2=0$
 - \underline{c} : centre de la sphère, r : rayon de la sphère
- L'introduction de l'équation du rayon à la place de \underline{x} donne
$$t^2 \underline{d}.\underline{d} + 2t \underline{d}.\underline{o}-\underline{c} + |\underline{o}-\underline{c}|^2 - r^2 = 0$$
 - Équation du second degré
- Permet d'obtenir des sphères exactes

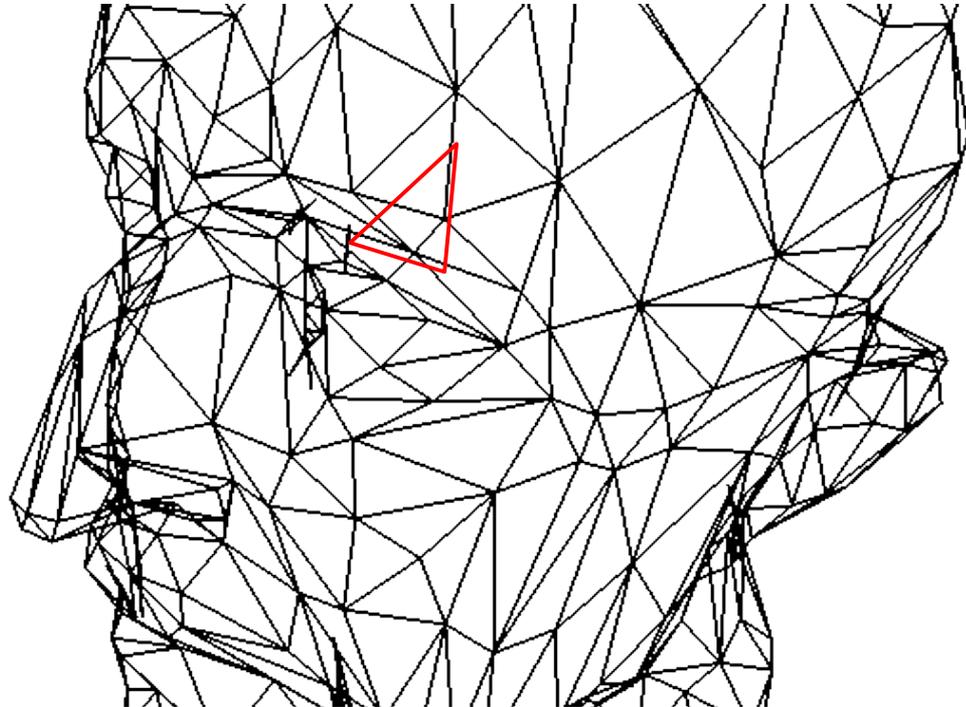


Autres intersections

- Polygones : [Appel '68]
- Quadriques, CSG : [Goldstein & Nagel '71]
- Torres : [Roth '82]
- Patches bi-cubiques : [Whitted '80, Kajiya '82, Benthin '04]
- Surfaces algébriques : [Hanrahan '82]
- Swept surfaces : [Kajiya '83, van Wijk '84]
- Fractals : [Kajiya '83]
- NURBS : [Stürzlinger '98]
- Surfaces de subdivision : [Kobbelt et al '98, Benthin '04]
- Points : [Schaufler et al.'00, Wald '05]
-

Intersection avec un maillage

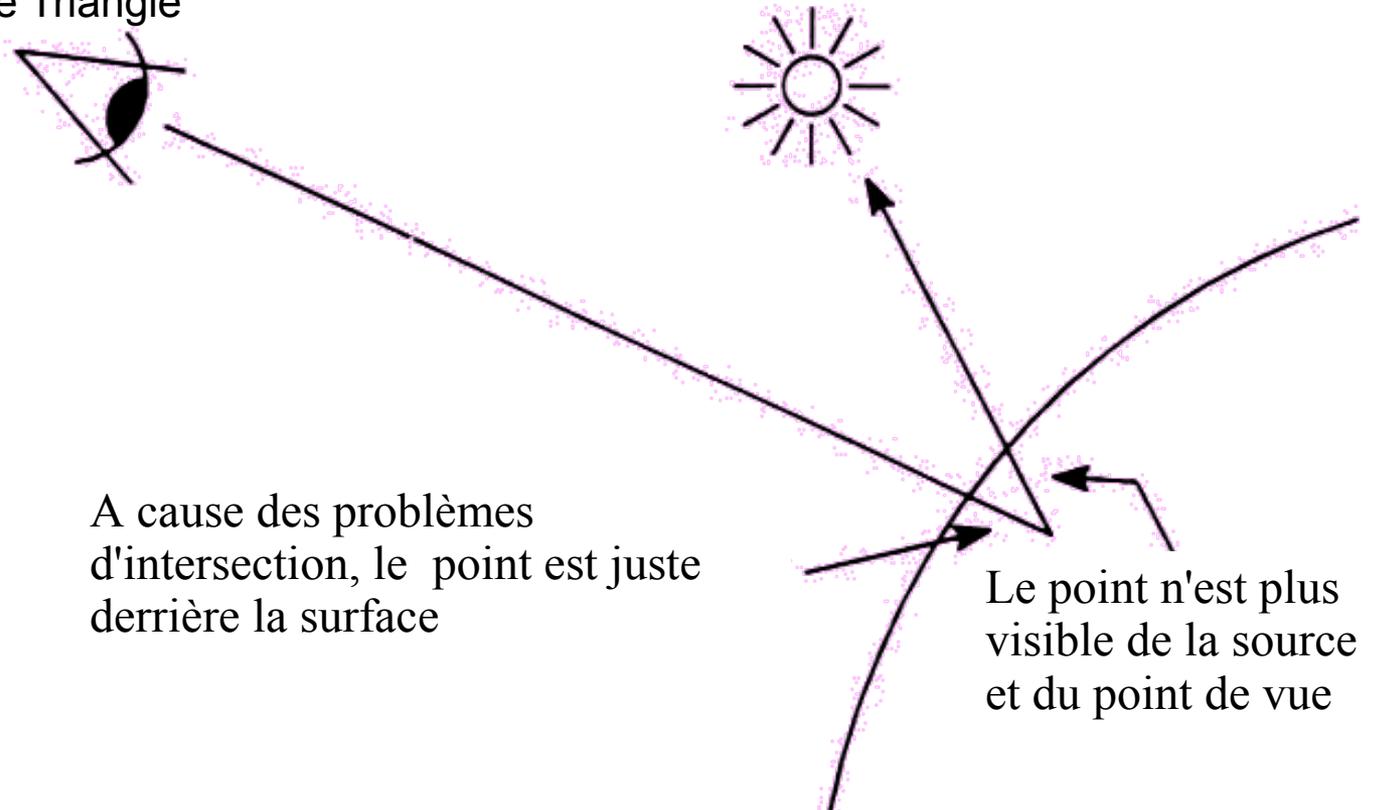
- Tester les triangles un à un, conserver l'intersection la plus proche



- Attention : coordonnées exprimées dans des bases différentes !

Problèmes de précisions

- Précision numérique
 - Un point n'est jamais exactement
 - Sur le Plan
 - Dans le Triangle



Analyse de la complexité

- Comparaison Tracé de Rayons vs Rastérisation/z-buffer
 - complexité algorithme de base
 - comment réduire la complexité ?

- Critères d'arrêts ?

Composantes/Difficultés majeurs

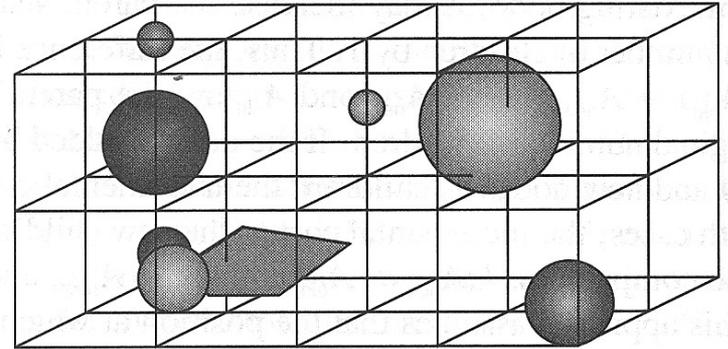
- Génération des rayons primaires
- Tracé de rayon
 - intersections rayon/objet
 - trouver l'intersection la plus proche le plus rapidement possible !
- Éclairement
 - ombres, réflexions, réfractions, etc.

Structures d'accélération

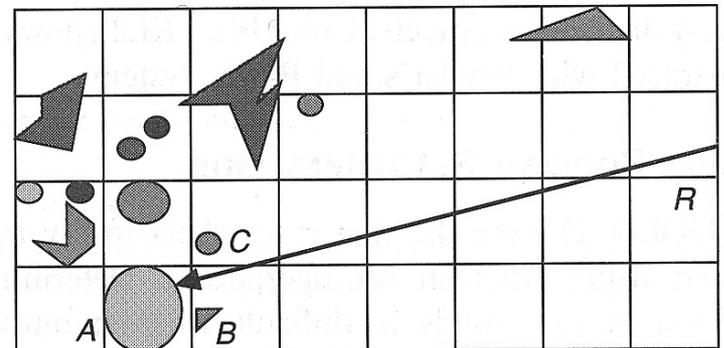
- Trouver l'intersection la plus proche
 - **Intersecter** le rayon avec **tous** les objets
 - Trop coûteux !
- Objectif
 - Faire en sorte que la 1ère intersection calculée soit la bonne !
 - en pratique : compromis
- Solution
 - Partitionnement de l'espace (souvent hiérarchique)
 - Grille, octree, BSP or kd-tree, Hiérarchie de Boîtes Englobantes (BVH)

Grille Régulière

- Construction
 - Subdivision de la boîte englobante
 - Résolution: souvent $\sim \sqrt[3]{n}$
 - Une cellule :
 - liste des objets intersectant

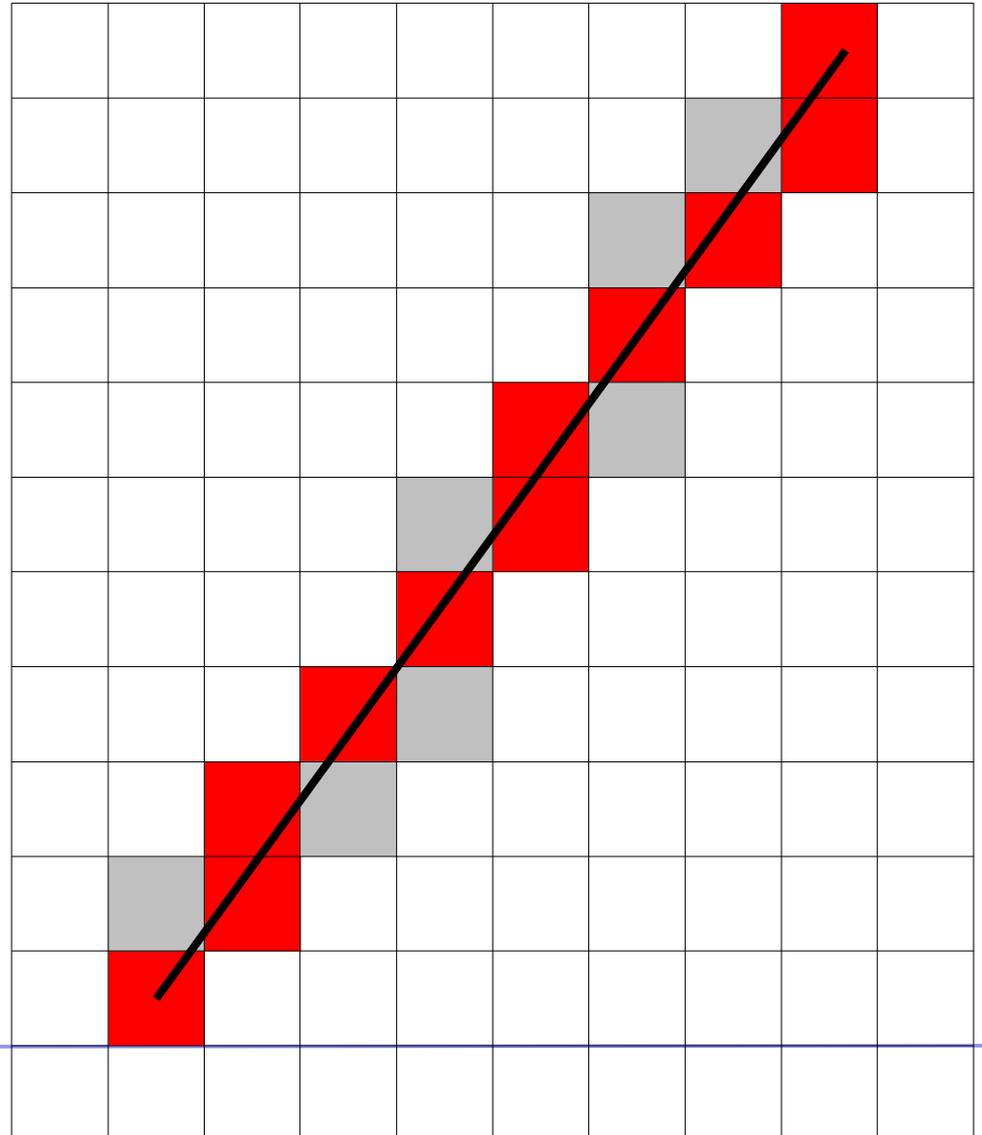


- Parcours de la grille
 - De proche en proche
 - De l'origine vers l'arrière
 - Arrêt si une intersection est trouvée



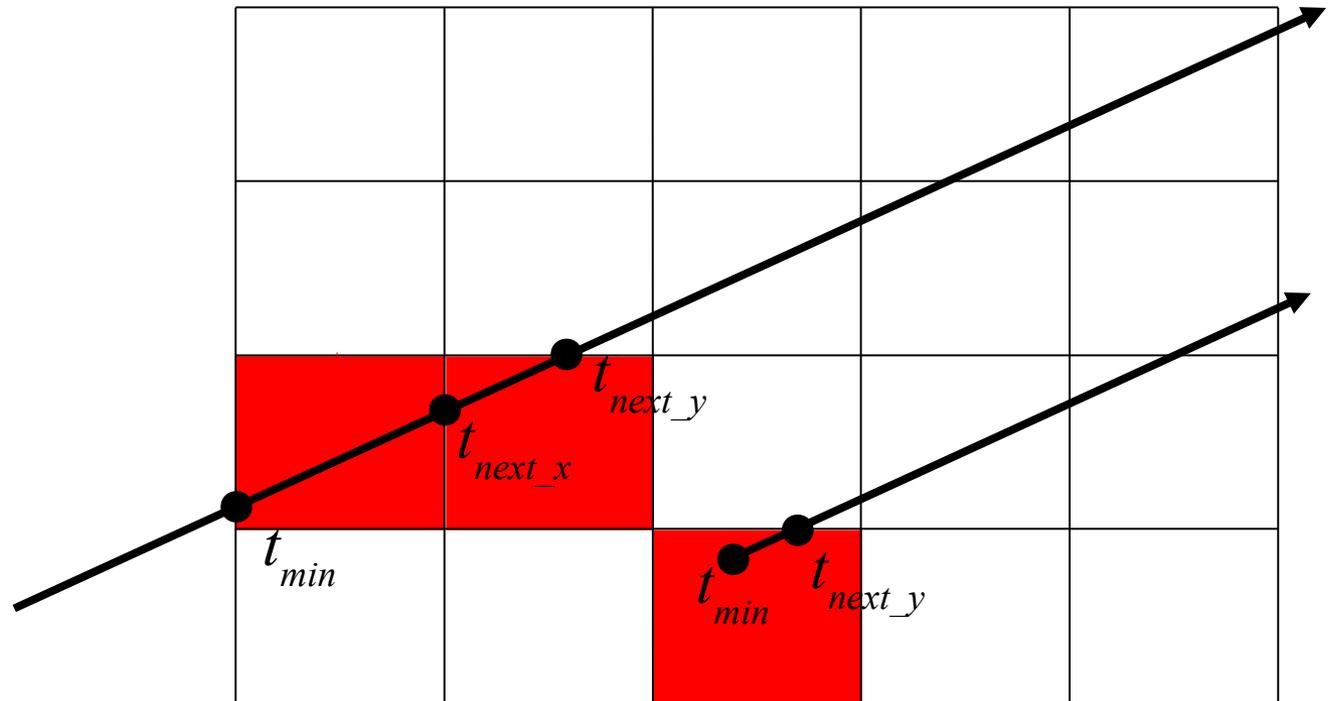
Parcours dans la grille

- 3DDDA – Three Dimensional Digital Difference Analyzer
- Similaire à la rasterisation
 - on veut **toutes** les cellules intersectant le rayon !



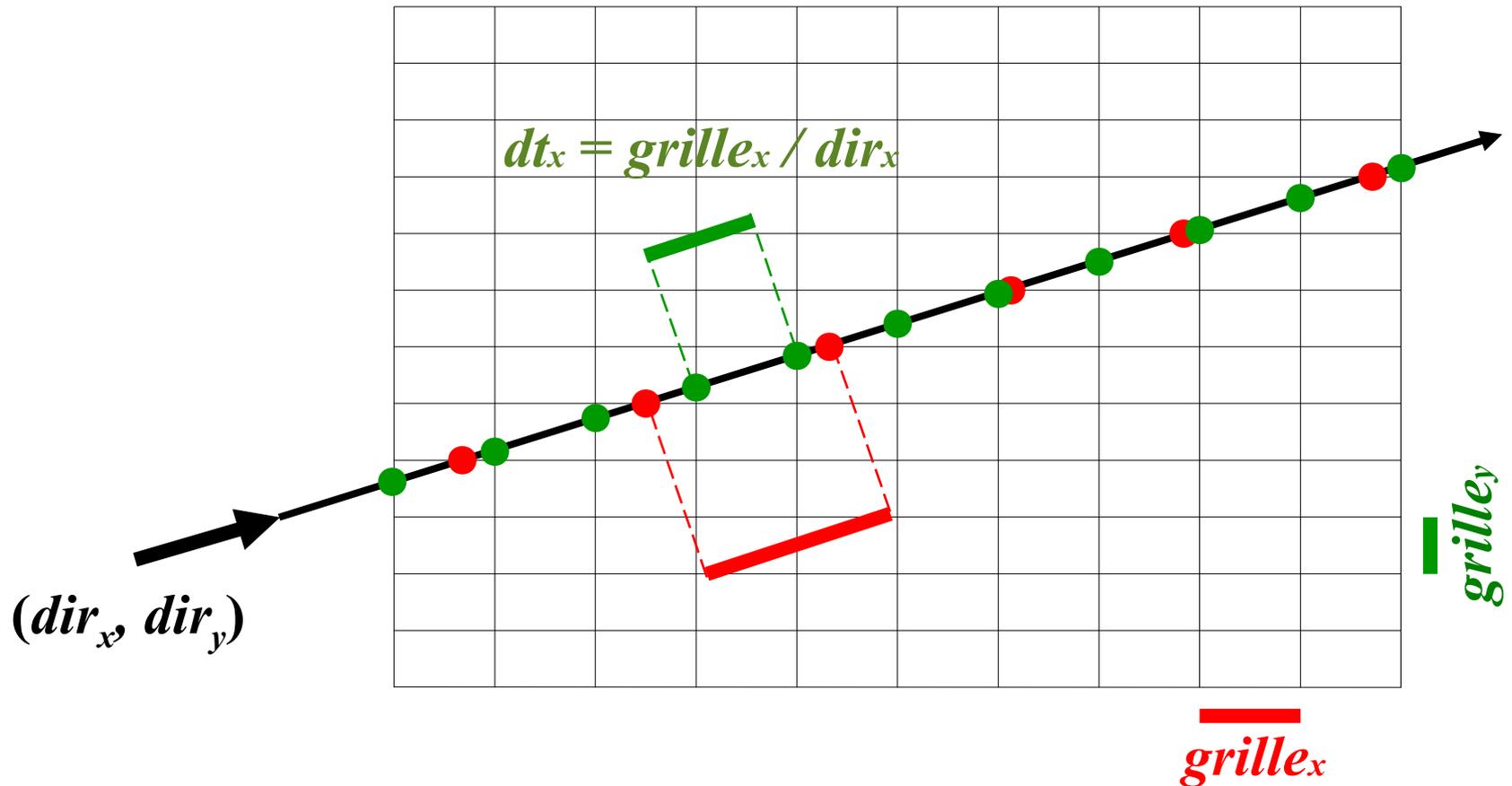
Initialisation

- Calculer l'intersection avec la boîte englobante
- Attention, l'origine du rayon peut être dans la boîte



L'intersection est répétitive

- Les intersections sur les axes sont équi-distantes

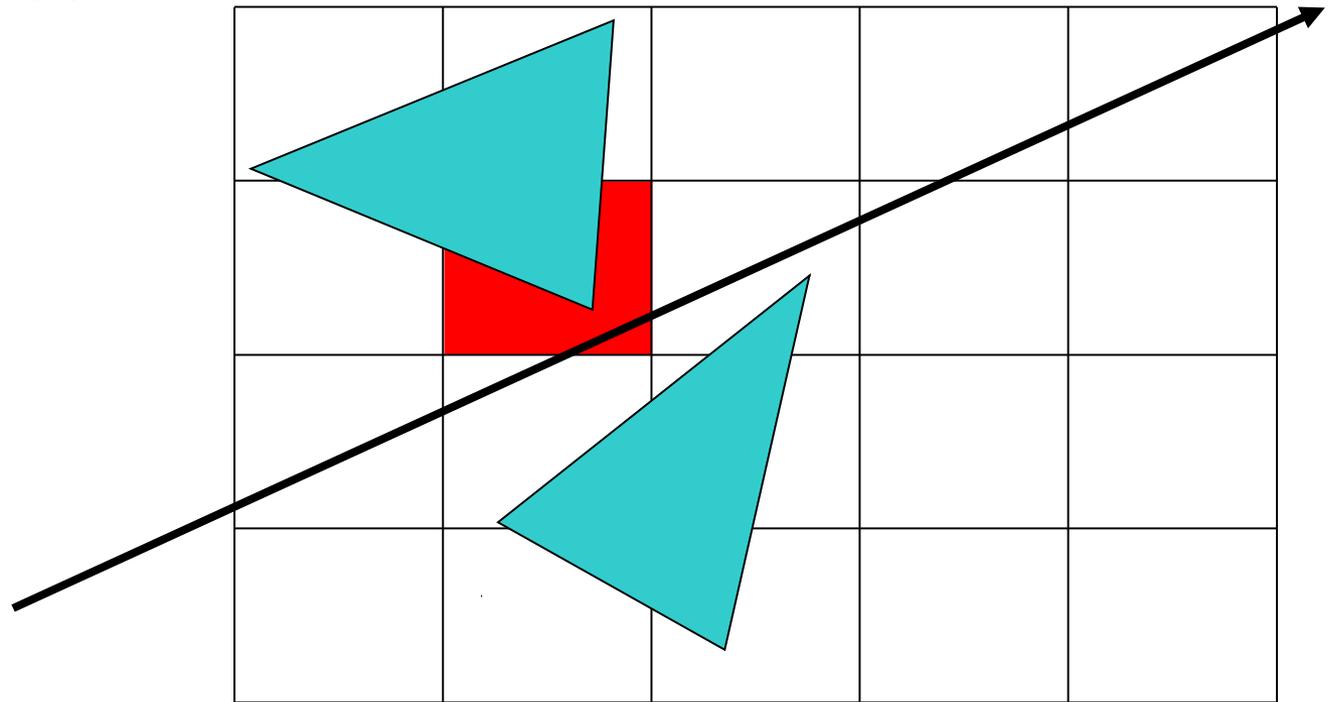


Pseudo-code

```
create grid
insert primitives into grid
for each ray  $r$ 
    find initial cell  $c(i, j)$ ,  $t_{\min}$ ,  $t_{\text{next}_x}$  &
     $t_{\text{next}_y}$ 
    compute  $dt_x$ ,  $dt_y$ ,  $\text{sign}_x$  and  $\text{sign}_y$ 
    while  $c \neq \text{NULL}$ 
        for each primitive  $p$  in  $c$ 
            intersect  $r$  with  $p$ 
            if intersection in range found
                return
     $c = \text{find next cell}$ 
```

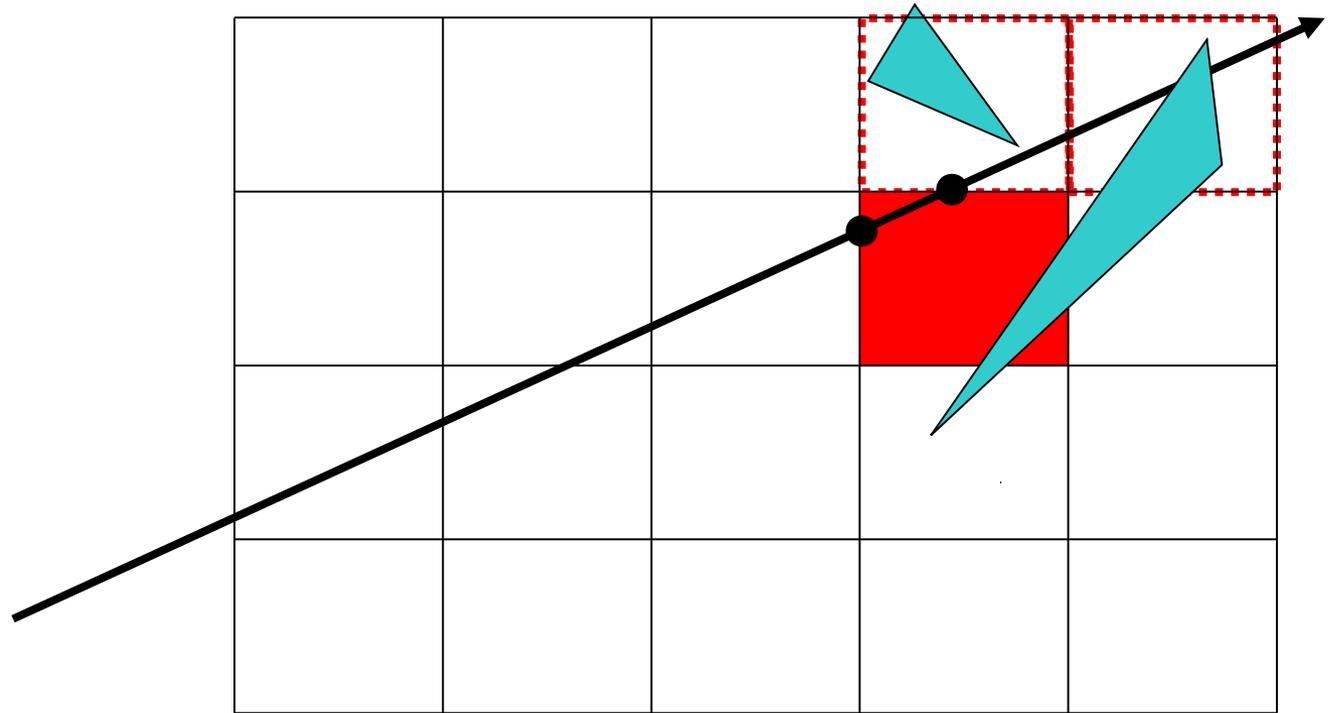
Test à faire sur chaque cellule

- Intersection(s) dans la cellule ?
- Oui : retourner la plus proche
- Non: continuer



Optimisation : marquer les objets déjà testés

Test à faire sur chaque cellule



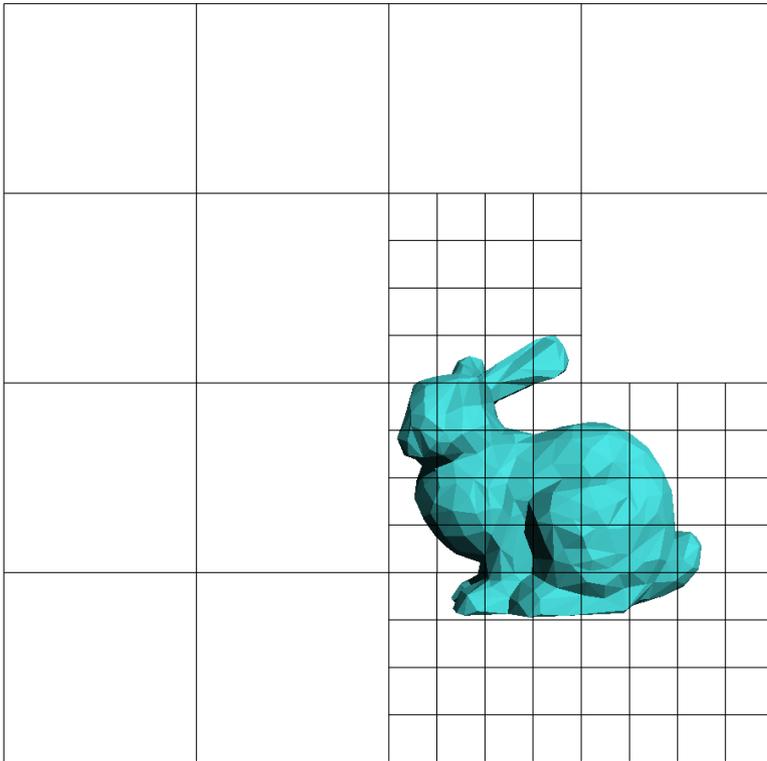
Optimisation : tester sur un segment

Grille régulière : Bilan

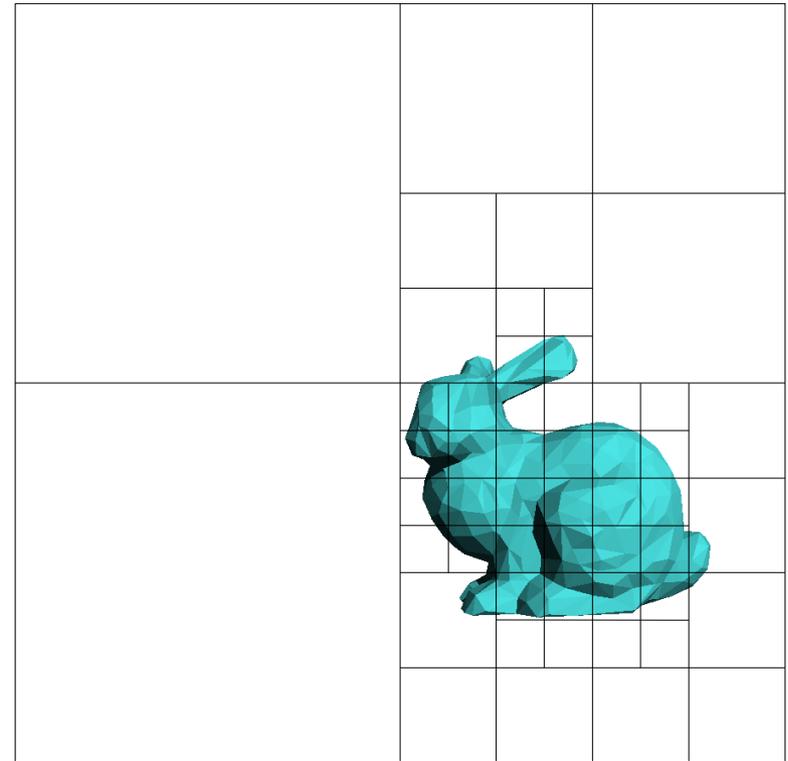
- Avantages?
 - Construction facile
 - Théorème des axes séparateurs: Voir: “Fast 3D Triangle-Box Overlap Testing »
 - Parcours simple et facilement portable sur du matériel adapté
- Désavantages?
 - Nombreuses cellules vides
 - Des cellules avec beaucoup d'objets
 - Non-adapté à certaines géométries
 - Problème du choix de la résolution
- Traitement des objets dans plusieurs cellules
 - Ne stocker que des références
 - Garder en mémoire les intersections déjà faites
 - Couples (rayon - objet)
 - Simple, mais peut casser la cohérence du cache CPU

Grilles hiérarchiques / Adaptatives

- Subdiviser jusqu'à qu'aucune cellule ne contienne plus de n élément ou avec une profondeur maximale d



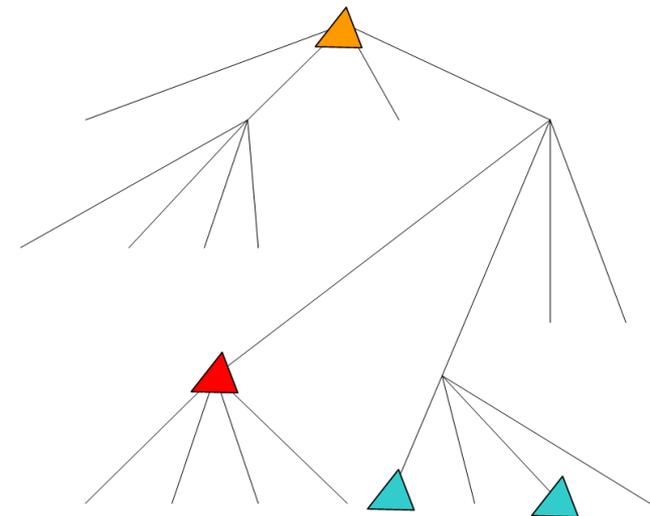
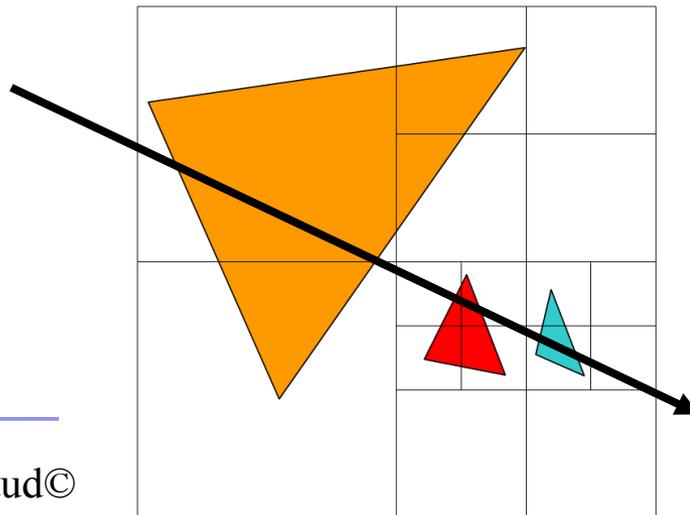
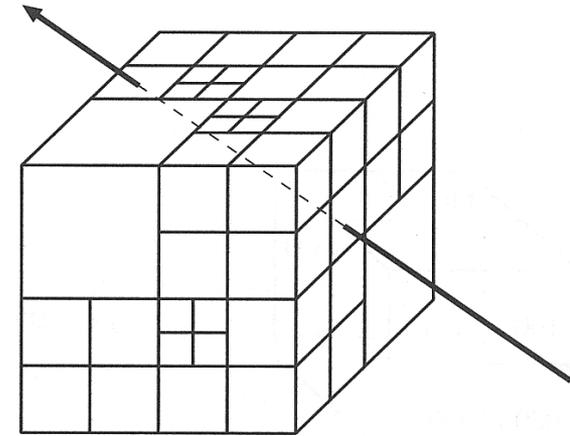
Grilles hiérarchiques
(*multigrid*)



Octree/(Quadtrees)

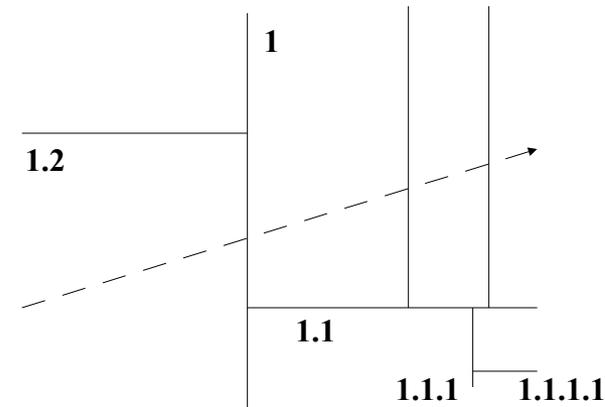
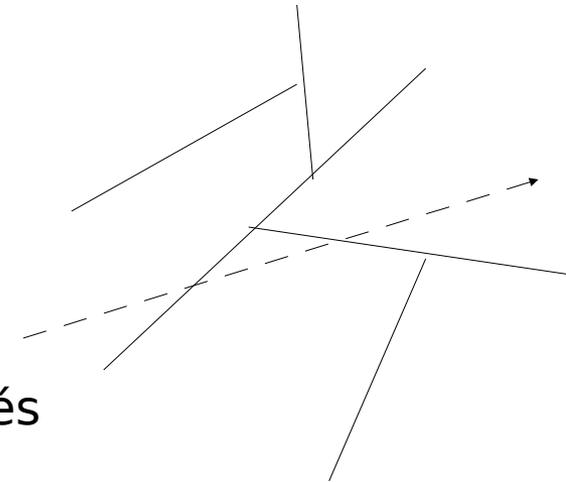
Octree / Quadtree

- Partitionnement hiérarchique de l'espace
 - Subdivise adaptative ment chaque voxel en 8 sous-voxels de manière récursive
 - Différents critères possibles:
 - nombre de primitives / cellules
 - ratio de "vide"
- Limitations
 - Parcours récursif complexe et donc coûteux
 - Convergence lente pour les zones complexes



BSP- and Kd-Trees

- Arbres binaires
 - Nœud = plan de subdivision de l'espace
- Binary Space Partition (BSP):
 - Plans quelconques « judicieusement » placés
 - Coûteux à construire, stocker et utiliser
- Kd-Tree
 - Plans alignés avec les axes
 - Simple, léger et très efficace



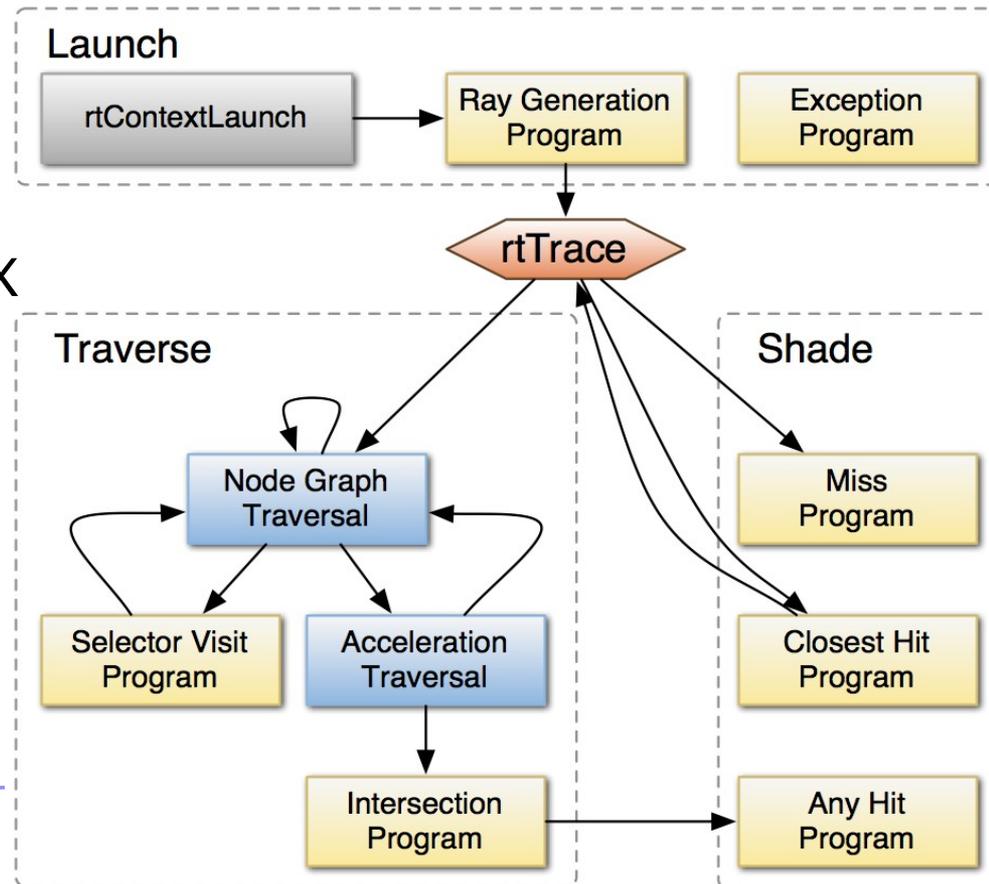
Lancer de rayon temps réel

- Exploitation des GPU
 - Structures de données sophistiquées
 - (peu adapté au GPU des générations précédentes)
 - OK pour les GPU les plus récents, et futures

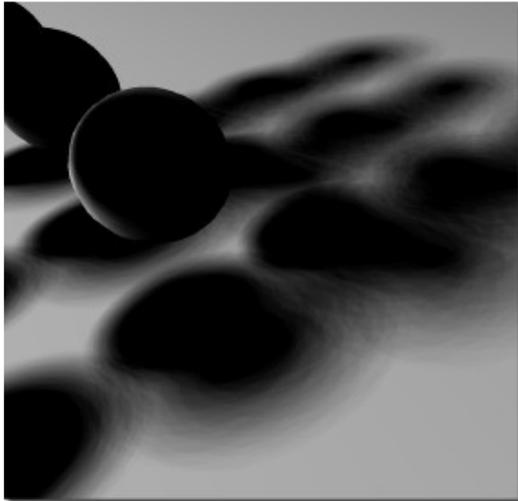


Lancer de rayon sur GPU

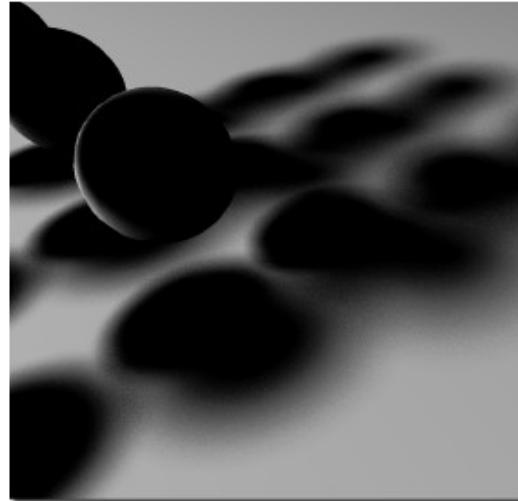
- Nvidia OptiX
 - voir vidéo « siggraph-2011-gpu-ray-tracing »
 - bibliothèque bas niveau destiné aux développeurs
 - structure de données : *BVH
 - génération des rayons entièrement configurable
- Autres
 - Lightworks : basé sur OptiX
 - Arion : GPU(CUDA)/CPU
 - OctaneRender (CUDA)
 - V-Ray RT (OpenCL)
 - ...



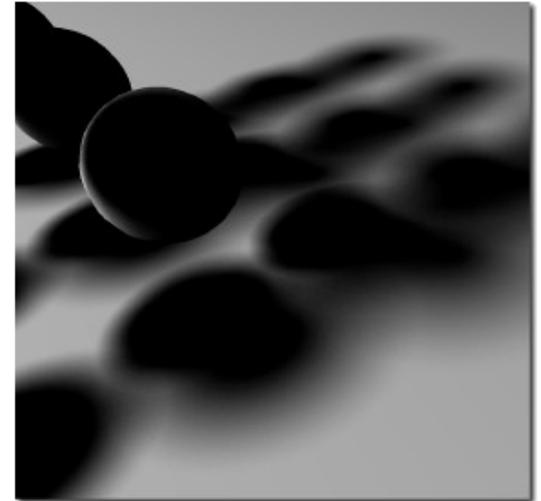
Sampling Strategies



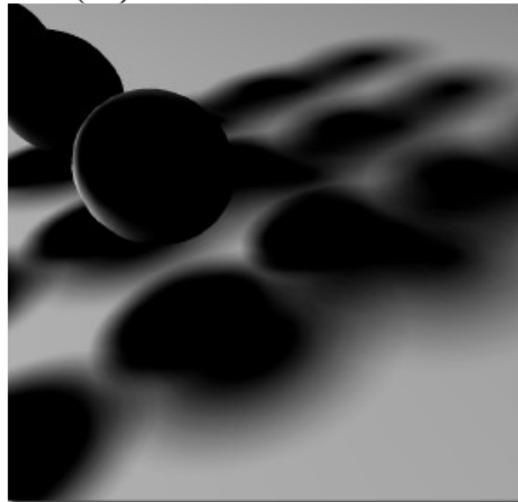
(a) 64 correlated



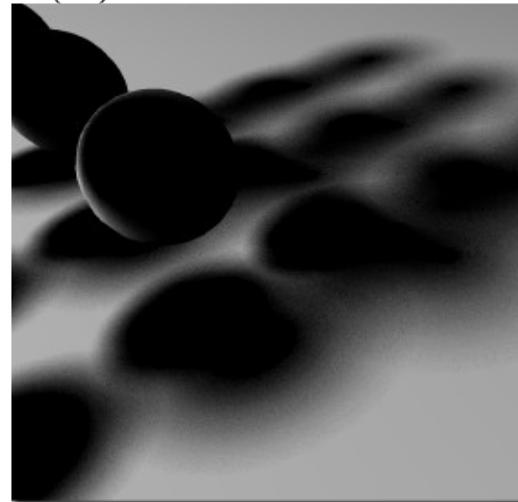
(b) 64 decorrelated



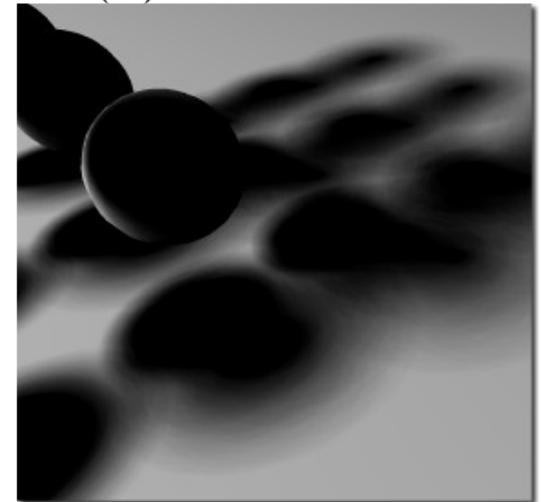
(c) 64 stratified



(d) 64 interleaved

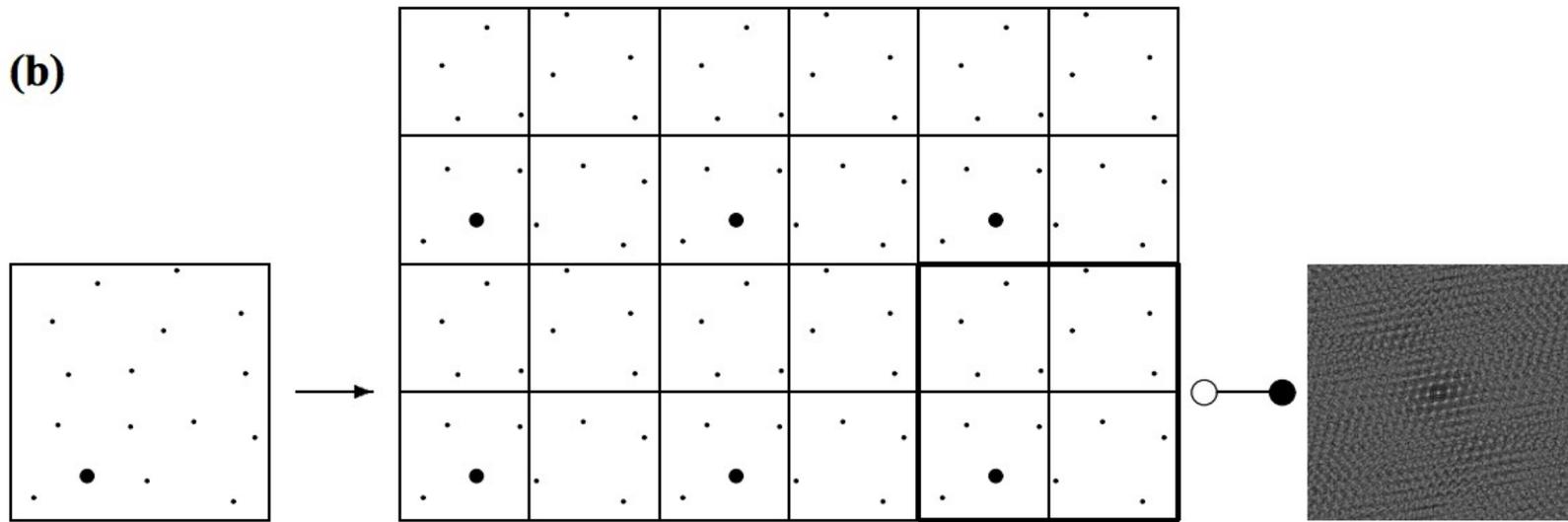
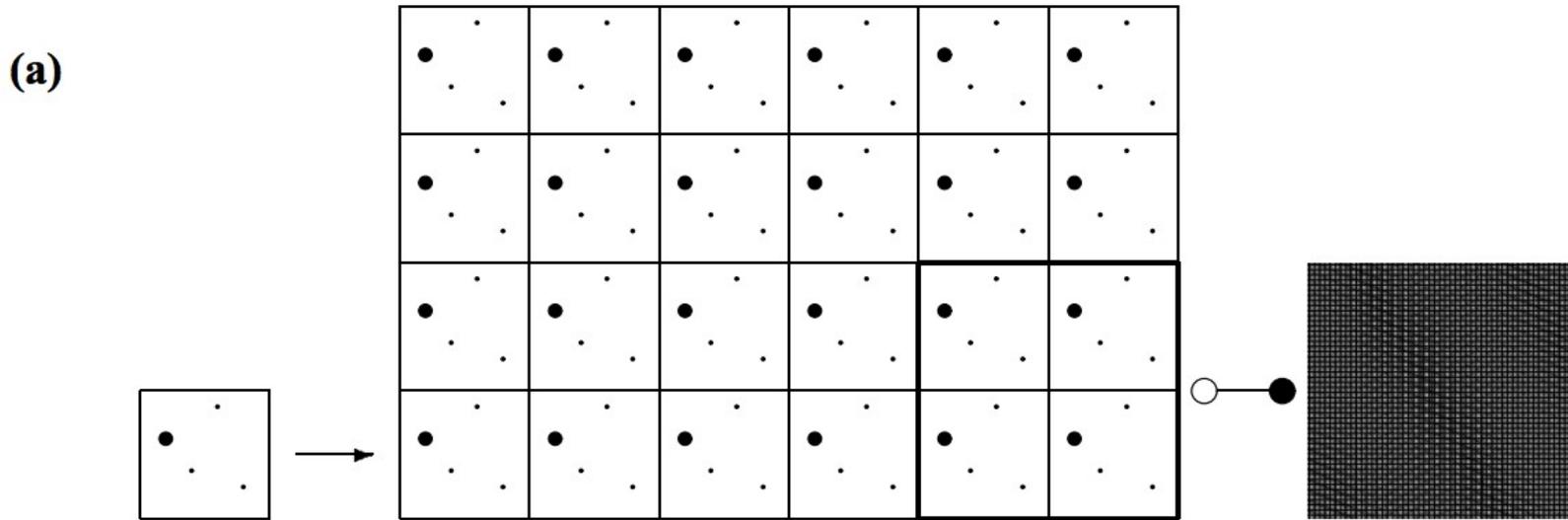


(e) 16 stratified



(f) 16 interleaved

Interleaved Sampling



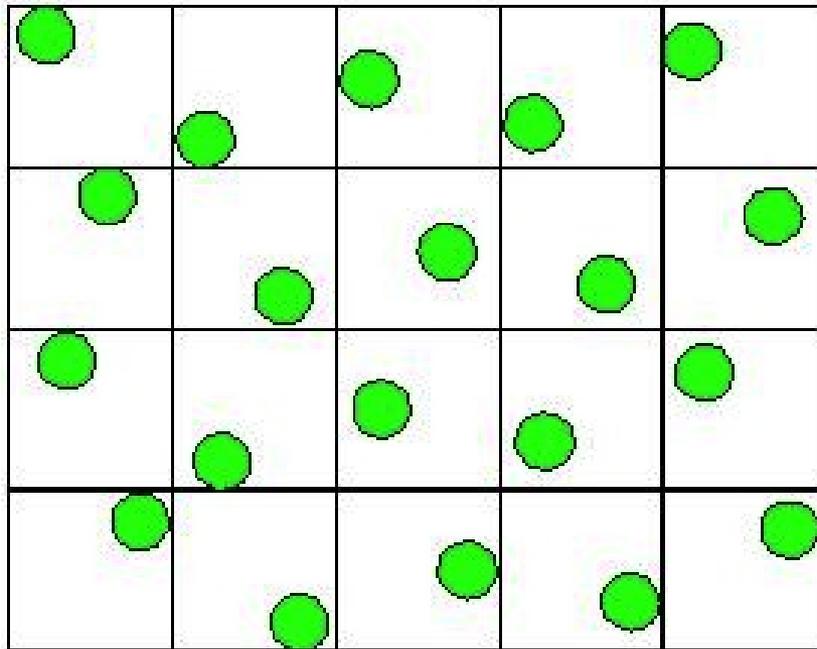
irregular basis pattern

regularly replicated basis pattern

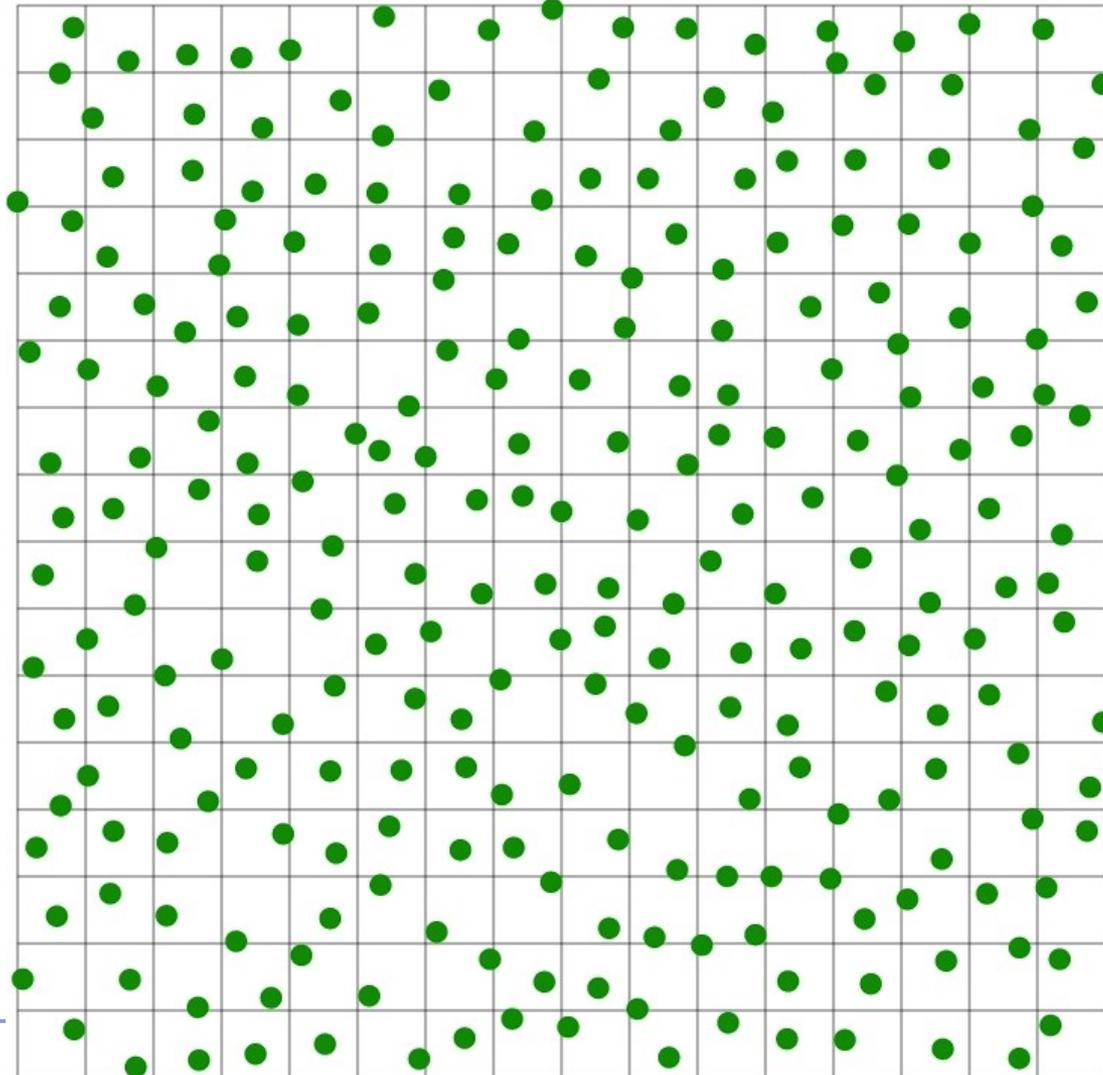
Fourier transform 86

Stratified Sampling

Unaligned Grid



Poisson disk sampling



Composantes/Difficultés majeurs

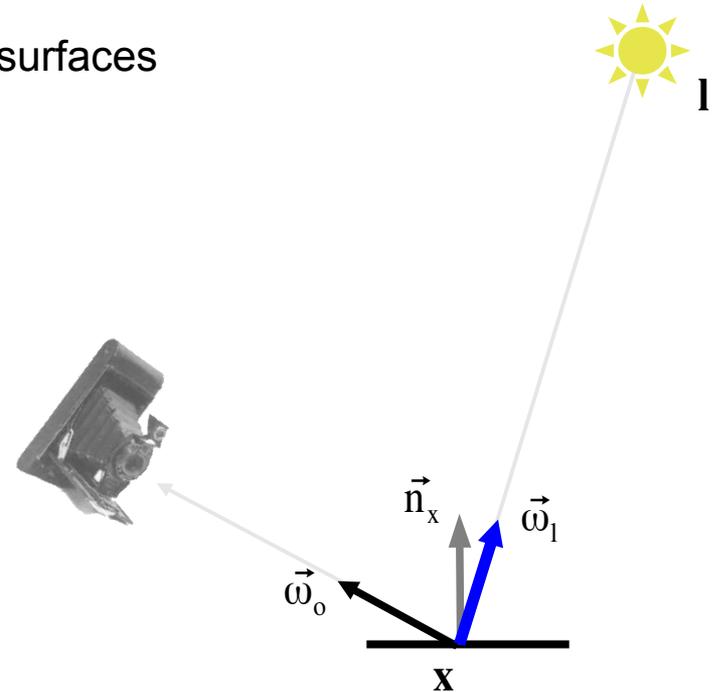
- Génération des rayons primaires
- Tracé de rayon
 - intersections rayon/objet
 - trouver l'intersection la plus proche le plus rapidement possible !
- Éclairement
 - ombres, réflexions, réfractions, etc.

Éclairage direct

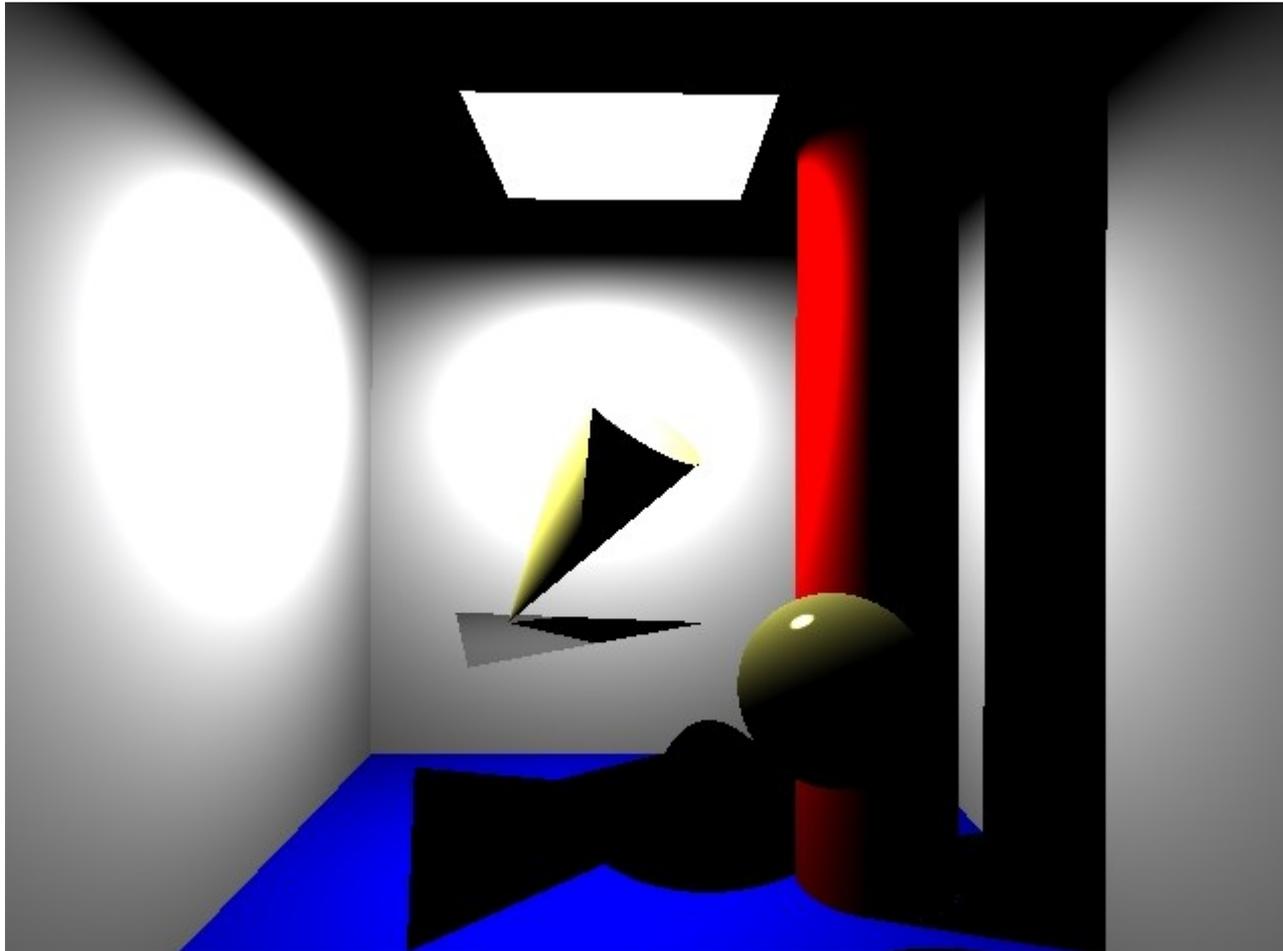
$$\mathbf{L}(\mathbf{x} \rightarrow \vec{\omega}_o) = \mathbf{L}_p(\mathbf{x} \rightarrow \vec{\omega}_o)$$

$$+ \sum_1 \rho(\vec{\omega}_1 \rightarrow \mathbf{x} \rightarrow \vec{\omega}_o) (\vec{n}_x \cdot \vec{\omega}_1) V(\mathbf{x}, \mathbf{l}) L_1$$

- Lumière propre (L_p)
 - Source de lumière par exemple
- Lumière ambiante (approximation par $\rho_a L_a$)
 - Moyenne de la lumière émise par toutes les surfaces
 - puis réfléchi par la surface
- Autres sources de lumières ($\sum_1 \dots$)
 - Chaque source est réfléchi
 - Propriété de réflexion = BRDF (ρ)
 - Grandeur : sr-1
 - Positive
 - Symétrique (réciprocité d'Helmoltz)
 - Conservation de l'énergie



Approche par échantillonnage



1 source

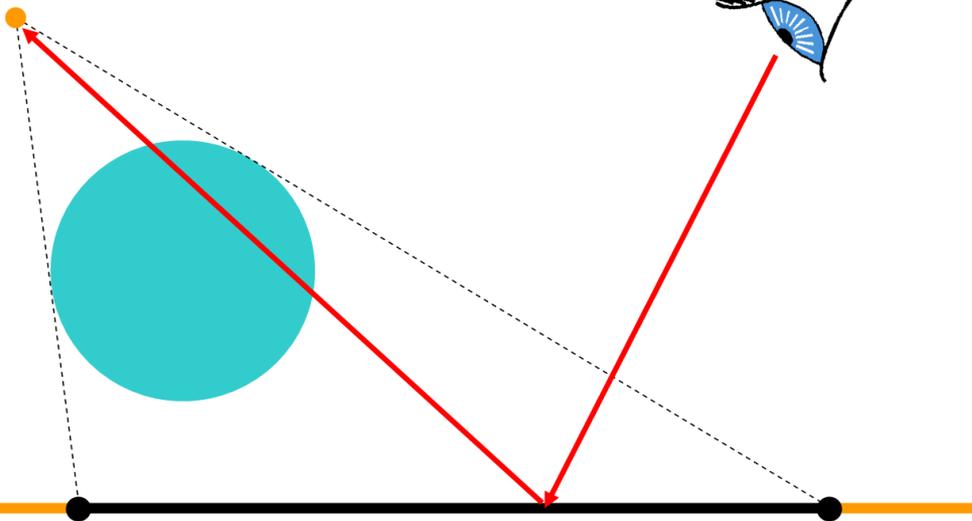
Test de visibilité

- Le rayon d'ombre
 - Une position initiale
 - Une position finale
- Le test
 - Existe-t-il une surface qui intersecte le segment ?
 - Même test mais sur une distance finie

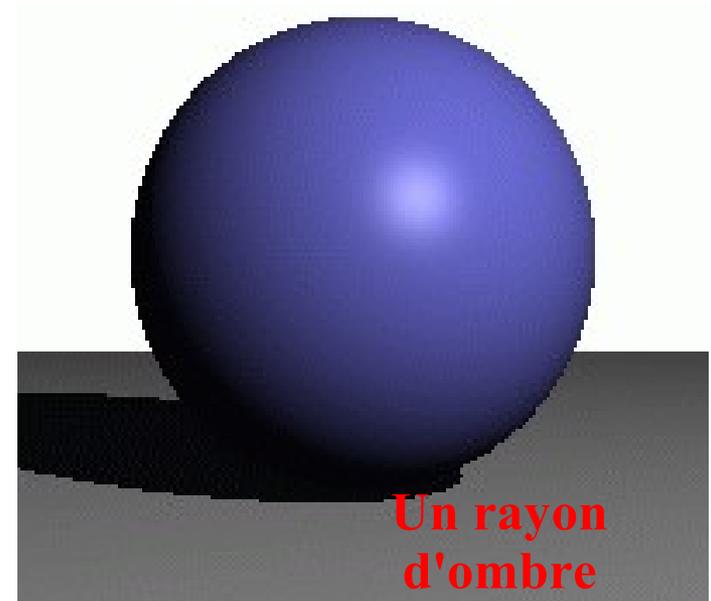
Rayons d'ombre

- Un rayon par source de lumières
 - Rayon = deux positions
- Exo : sources étendues ??

point light source



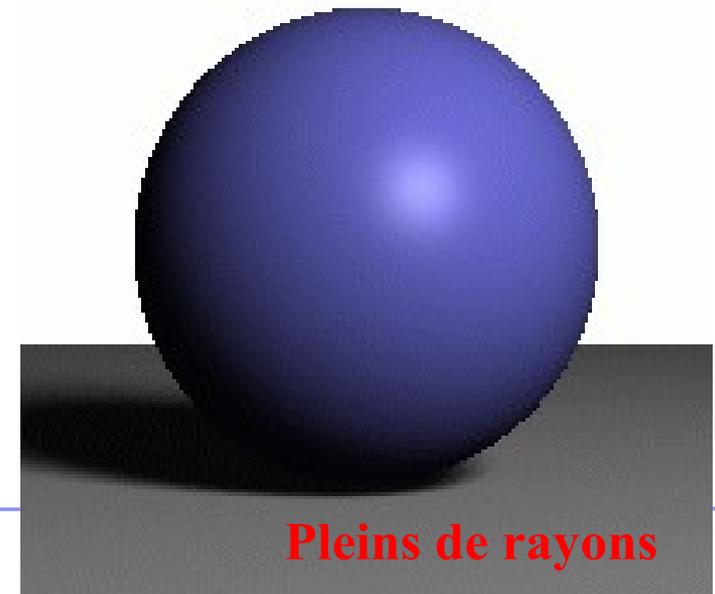
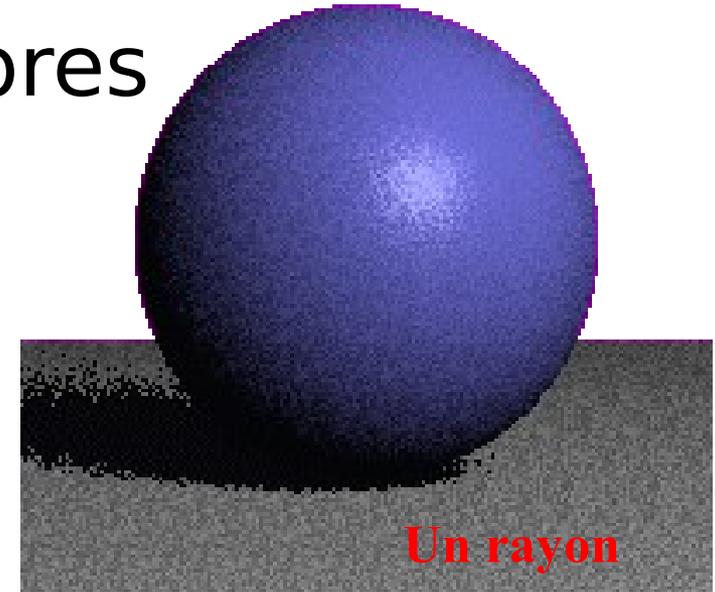
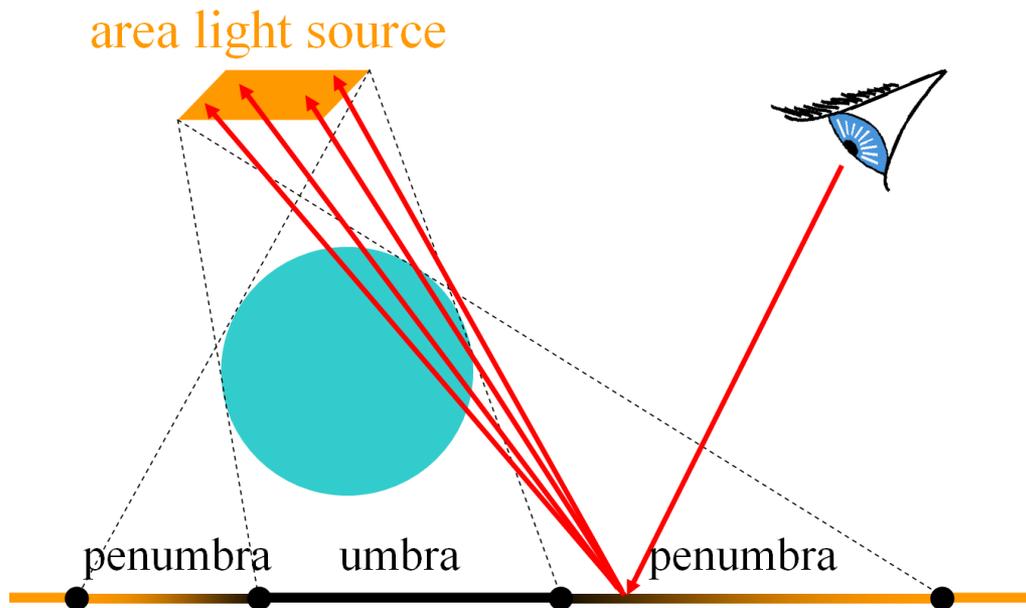
Sans ombre



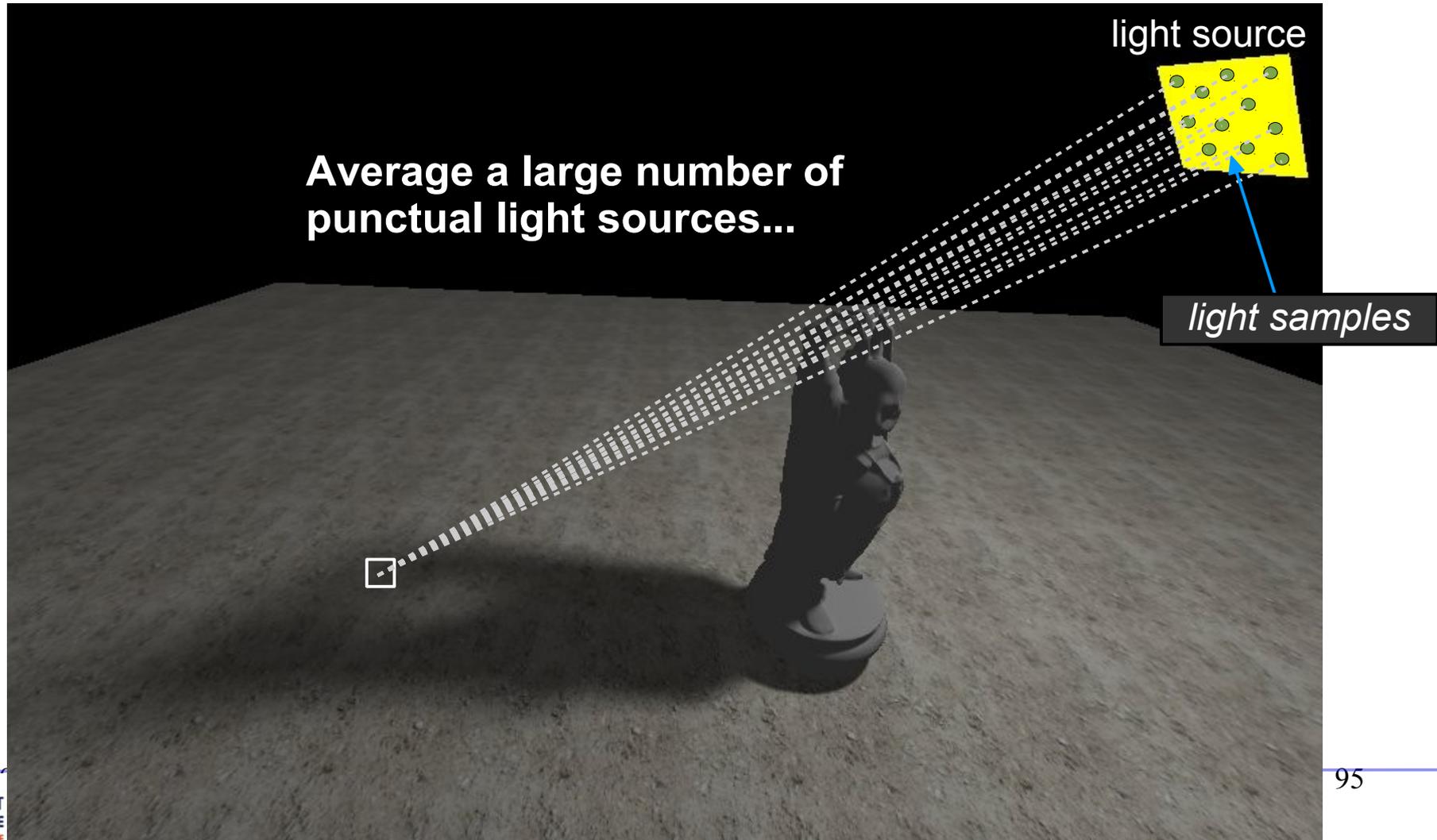
Un rayon
d'ombre

Soft Shadows

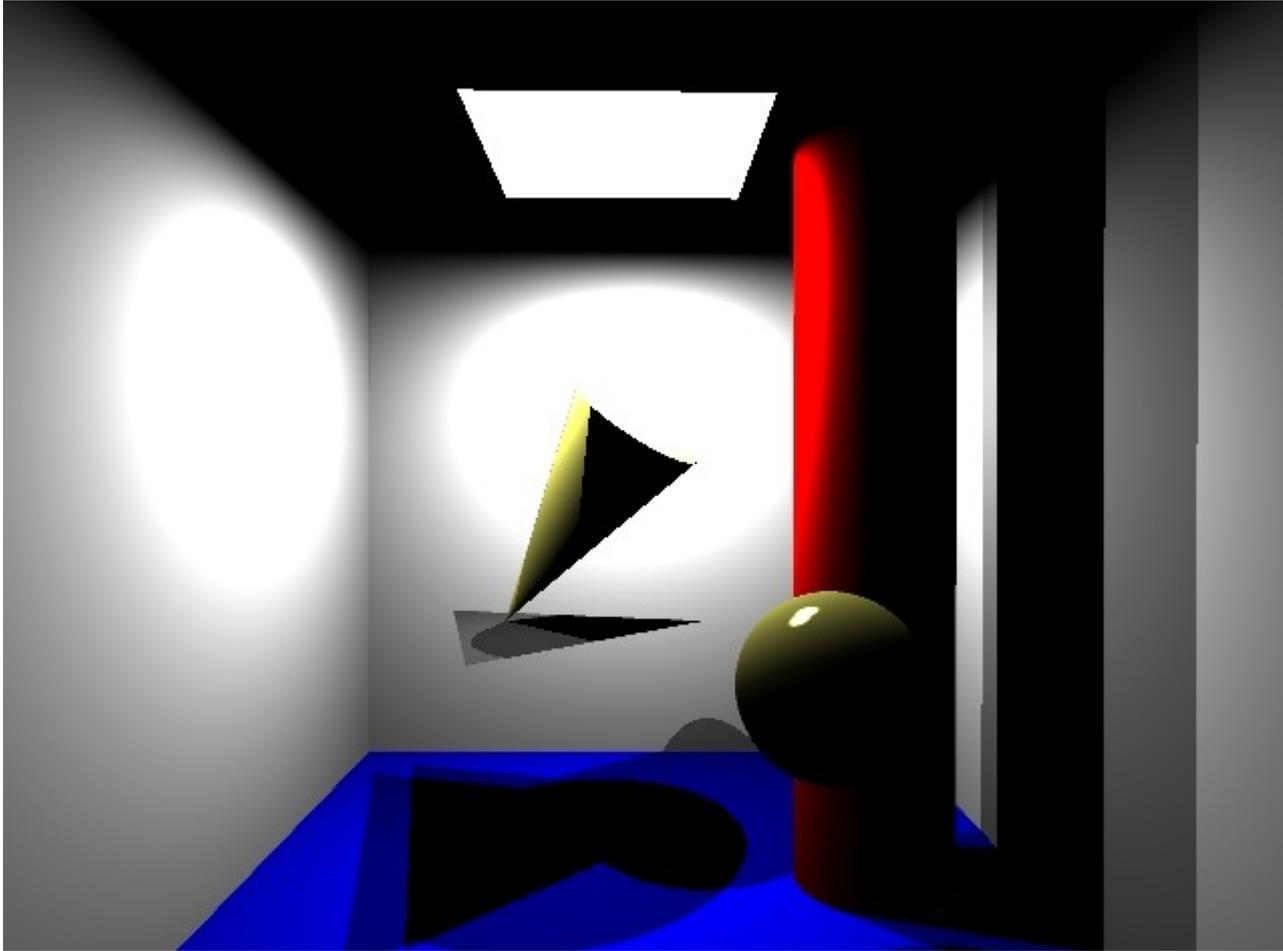
- Plusieurs rayons d'ombres



Soft Shadows

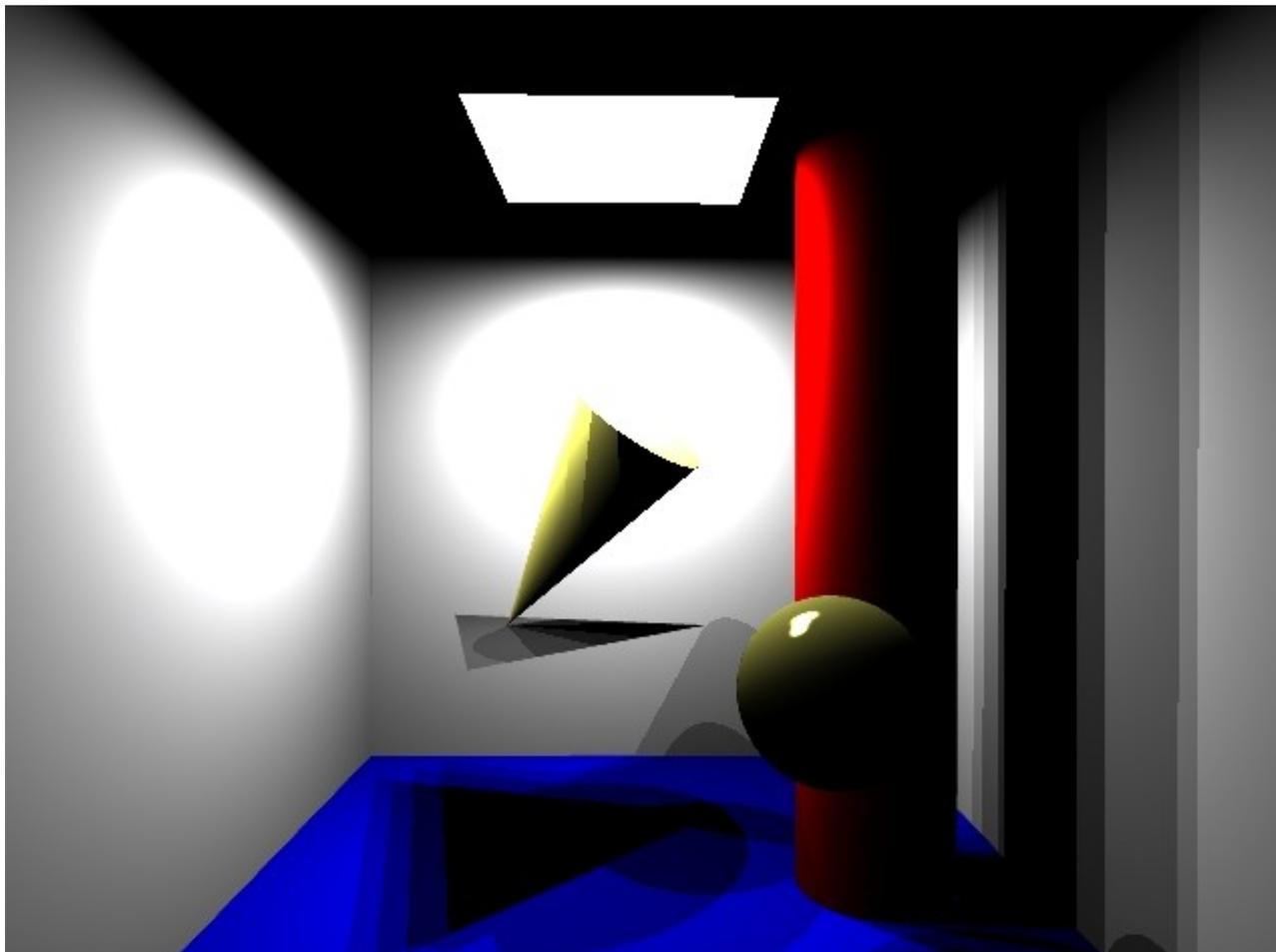


$L_p + 2$ sources ponctuelles



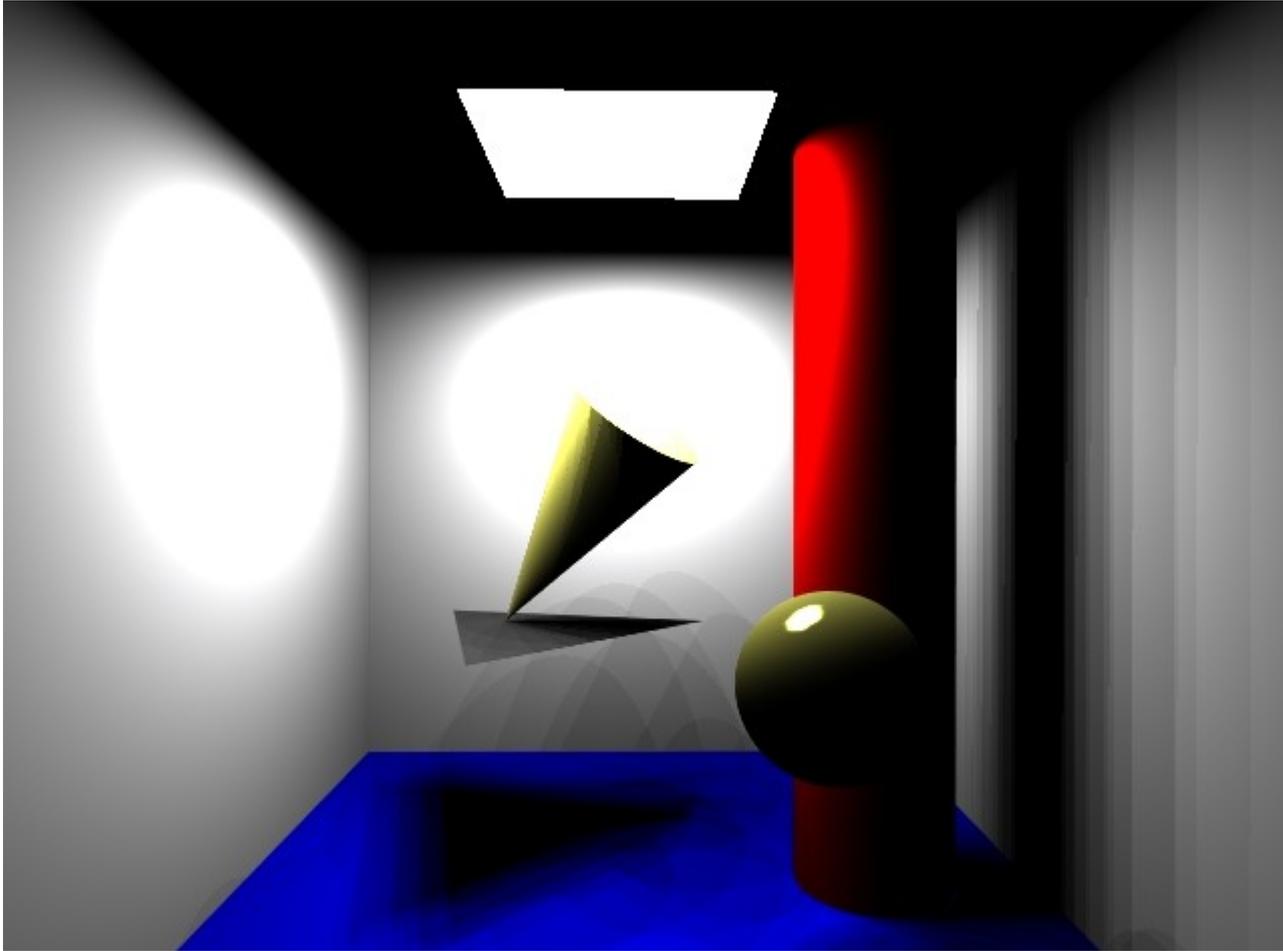
2 sources

Lp + 4 sources ponctuelles



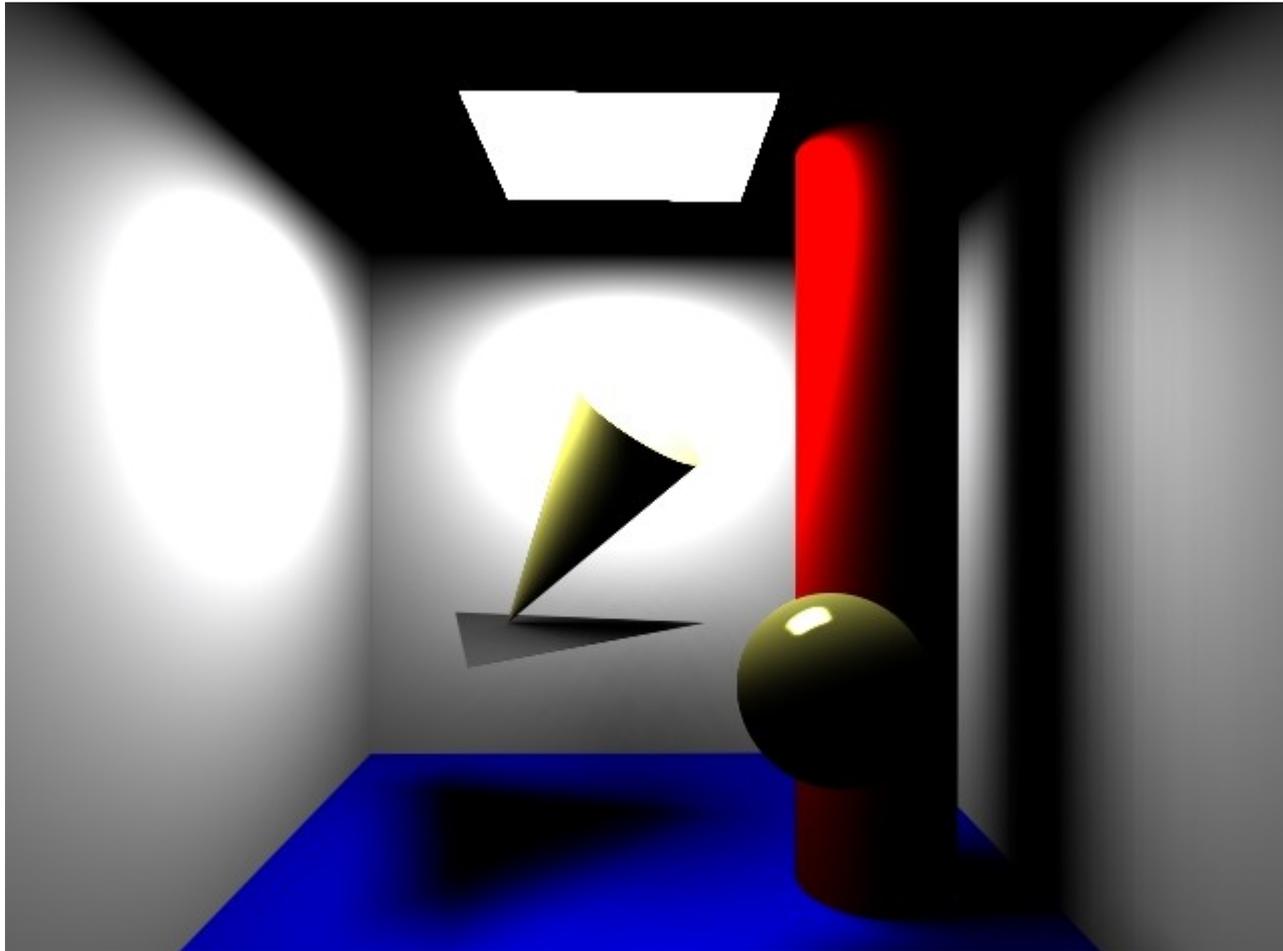
4 sources

Lp + 16 sources ponctuelles



16 sources

Lp + 256 sources ponctuelles

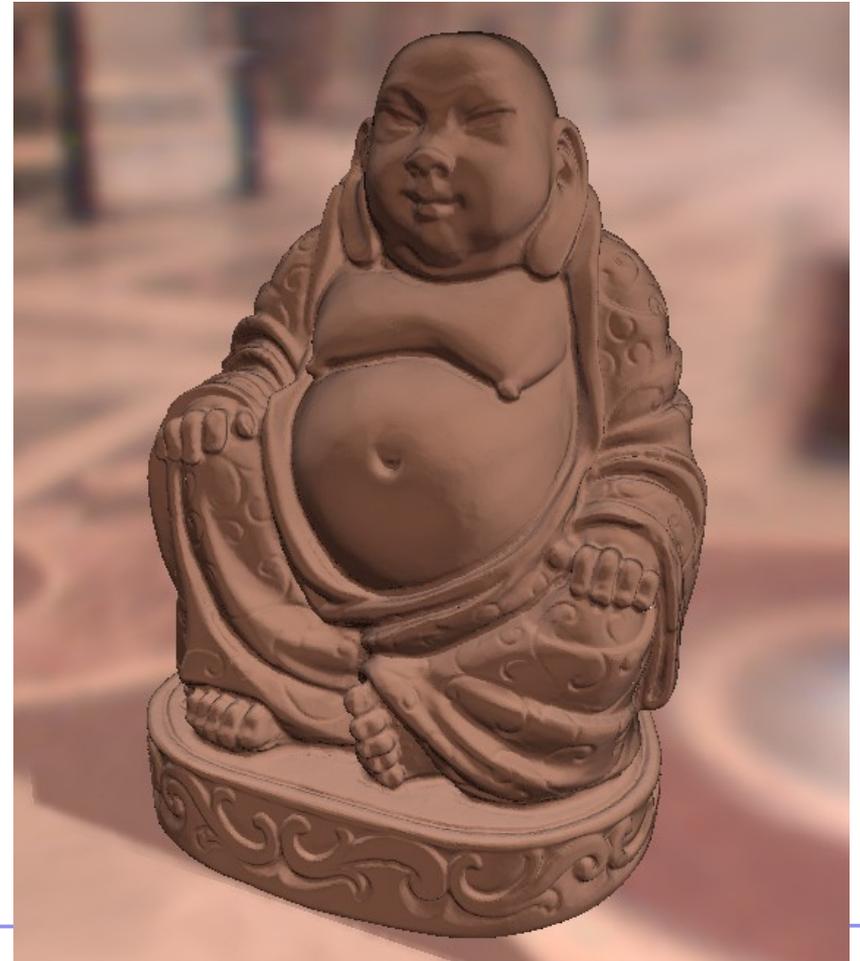


256 sources

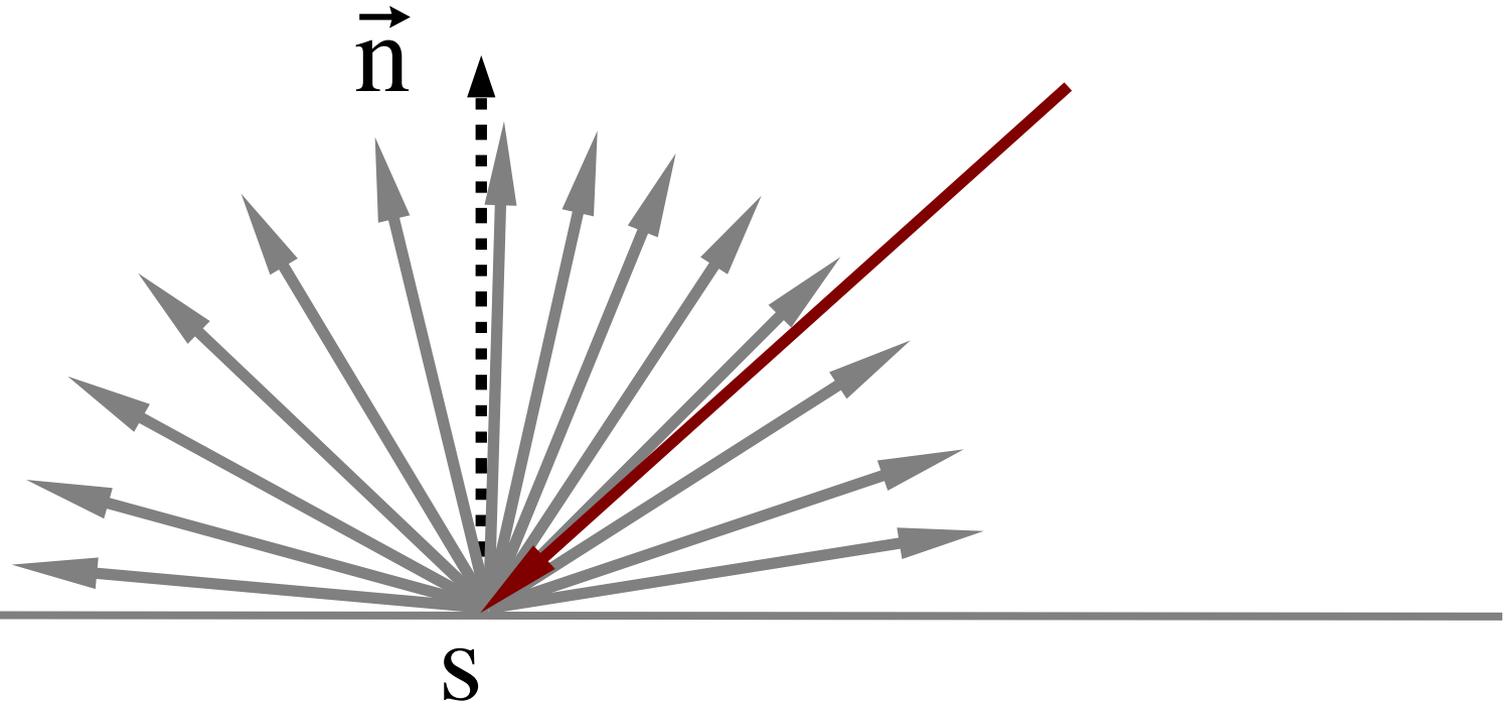
=> simple mais pas utilisable pour les applications temps réels !

Propriétés de Réflexion

- Diffus - Lambertien
 - Indépendant du point de vue



Diffus/Lambertien



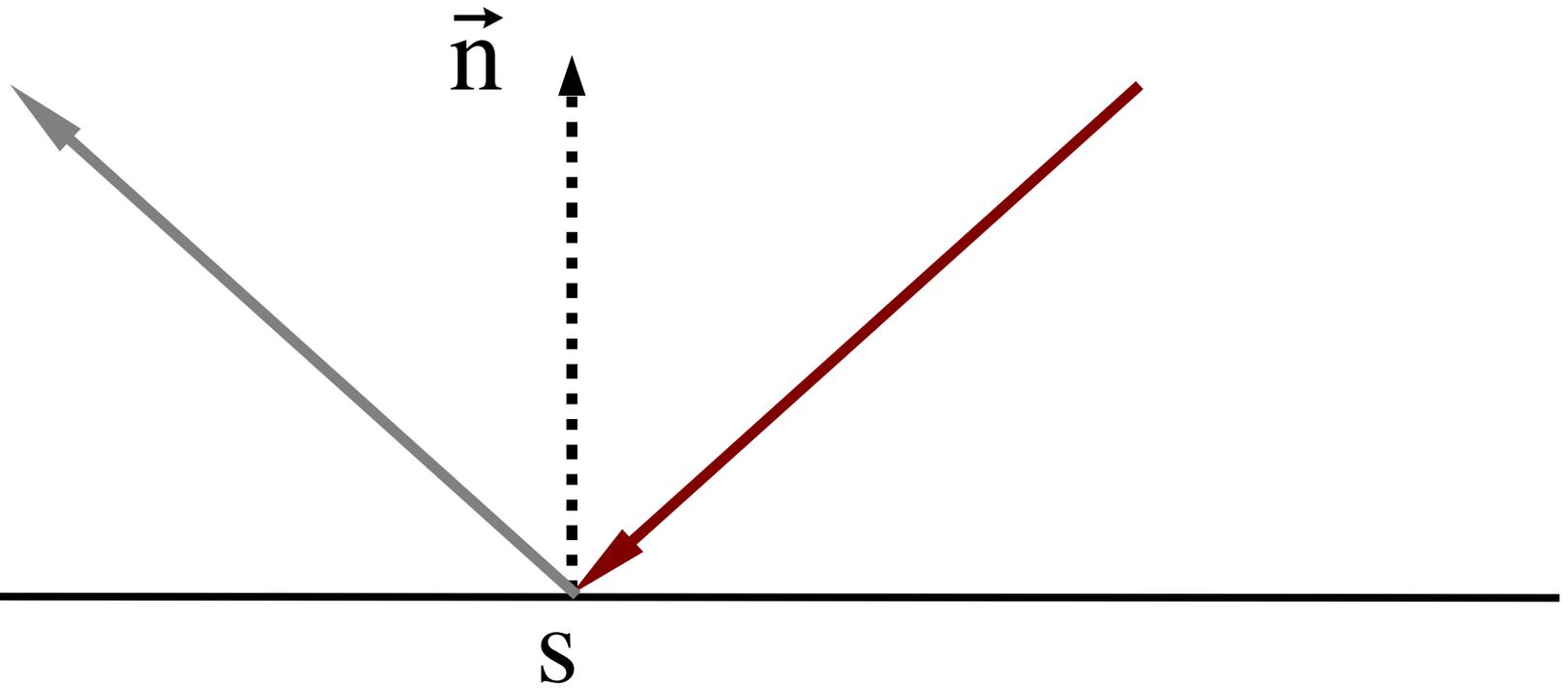
$$\rho \left(\vec{i}_l \rightarrow s \rightarrow \vec{o} \right) = \rho_d$$

Propriétés de Réflexion

- Diffus - Lambertien
 - Indépendant du point de vue
- Miroir – Spéculaire
 - Dépendant du point de vue

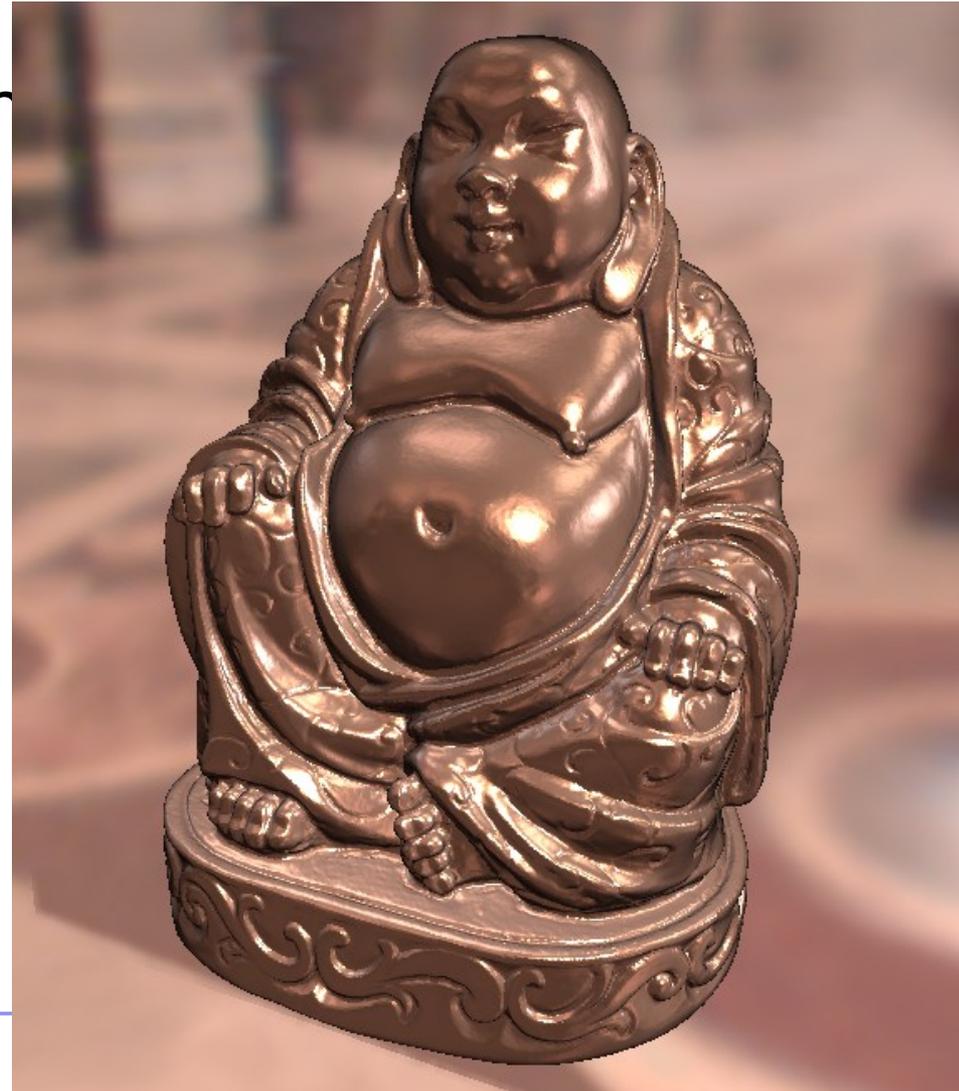


Spéculaire

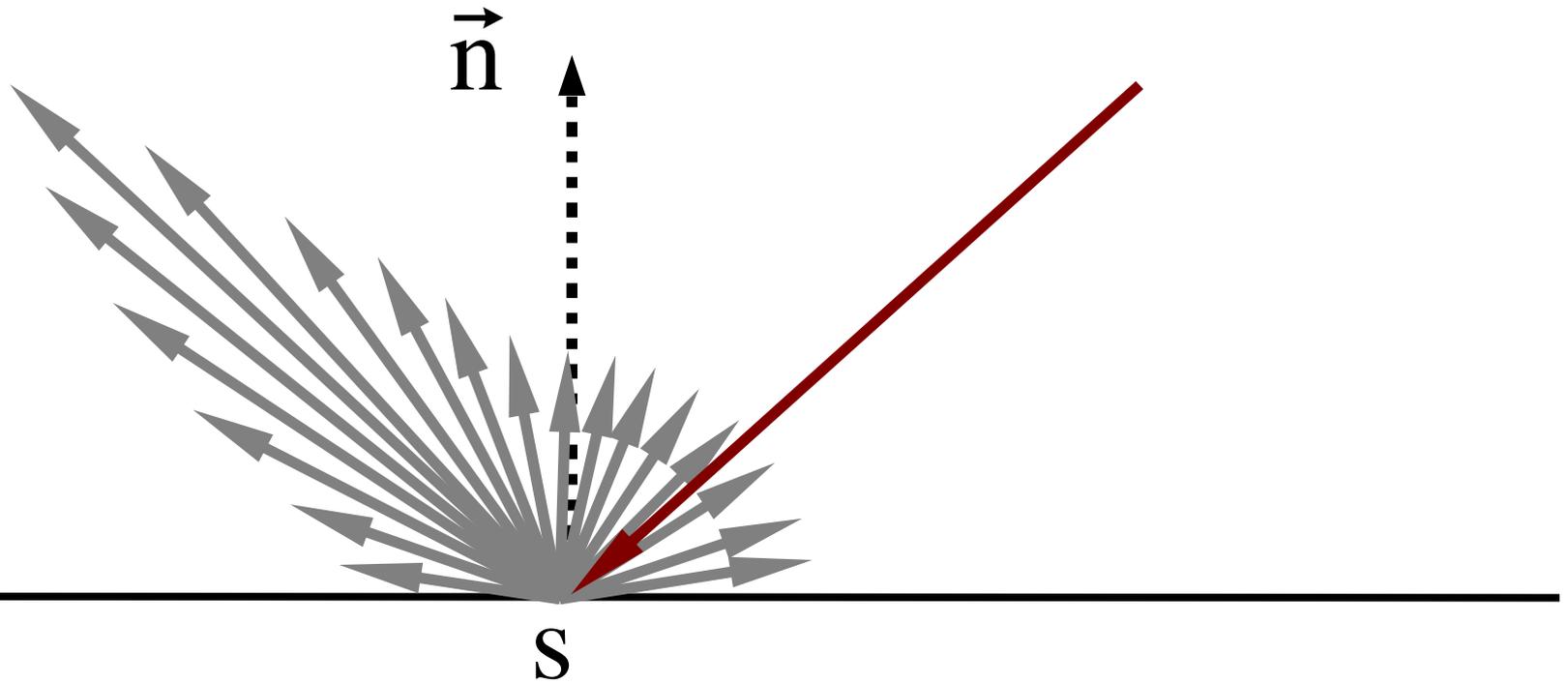


Propriétés de Réflexion

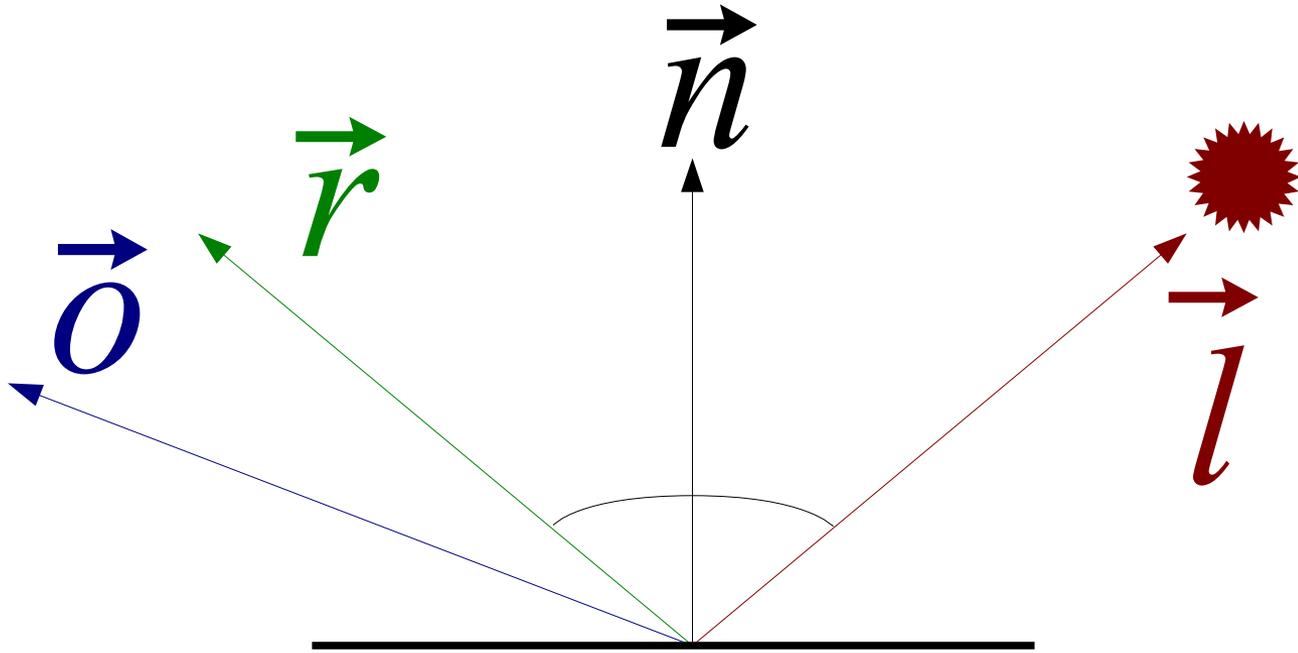
- Diffus - Lambertien
 - Indépendant du point
- Miroir – Spéculaire
 - Dépendant du point
- Glossy/Brillant
 - Dépendant du point
 - Lobe
 - Phong



Glossy/Brillant

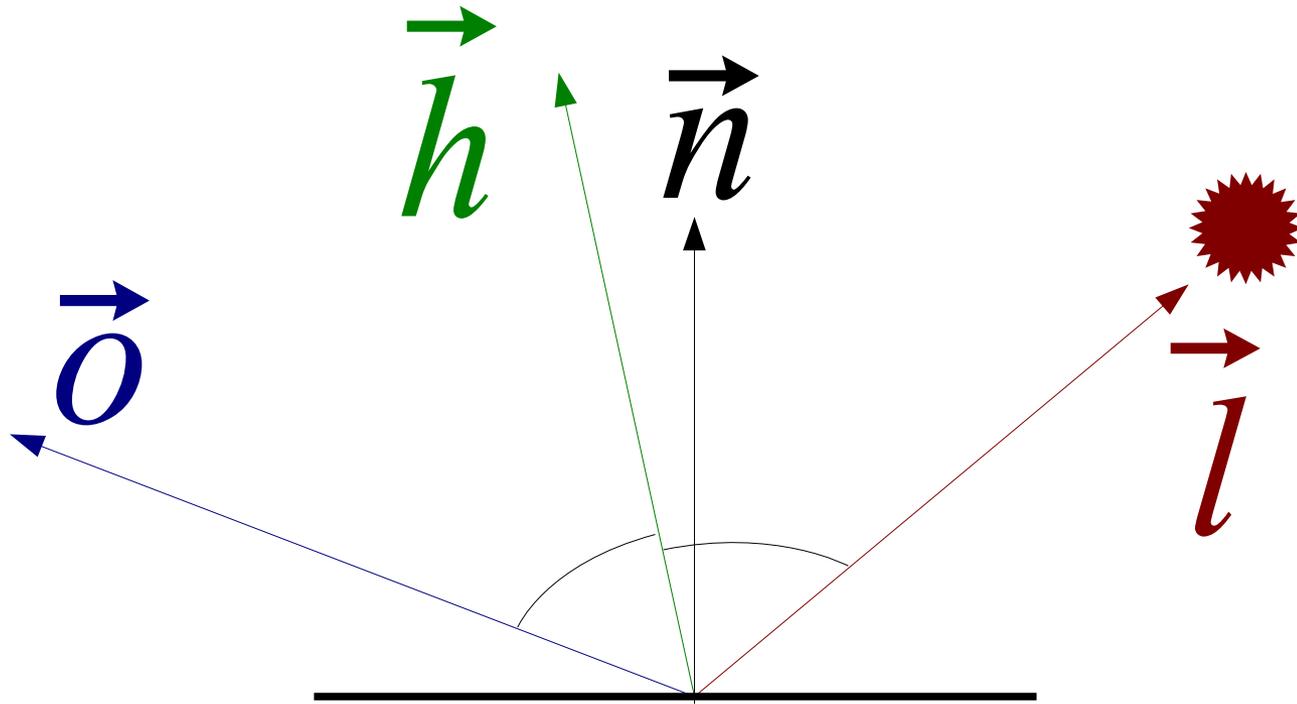


Modèle de Phong



$$\rho \left(\vec{i}_l \rightarrow s \rightarrow \vec{o} \right) = \rho_s \left\langle \vec{o} \cdot \vec{r} \right\rangle^n$$

Modèle de Blinn-Phong



$$\rho \left(\vec{i}_l \rightarrow s \rightarrow \vec{o} \right) = \left\langle \vec{n} \cdot \vec{h} \right\rangle^n$$