

Une méthode de radiosité de faible complexité

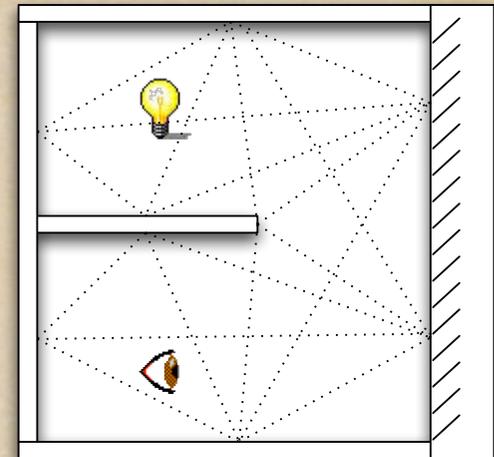
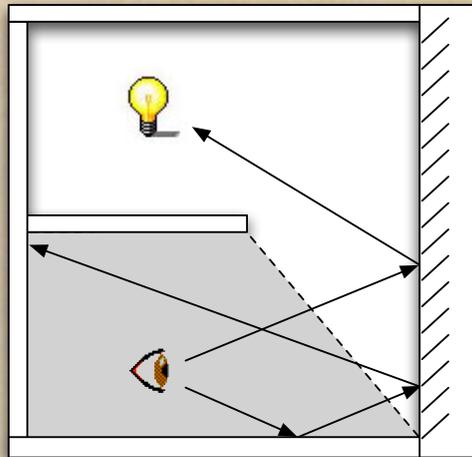
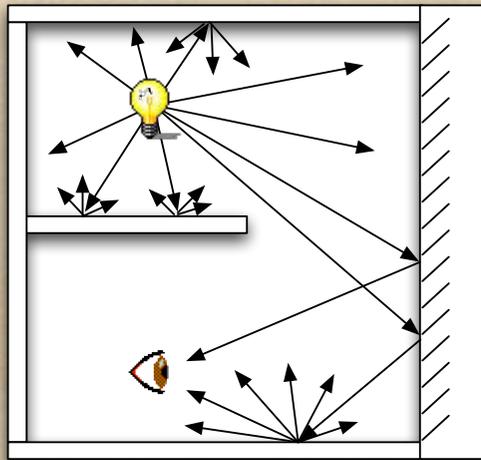
Pierre Chatelier : chatelier@llic3.u-clermont1.fr

Rémy Malgouyres : remy.malgouyres@llic3.u-clermont1.fr

Plan

- ◆ Radiosité basée sur des voxels
- ◆ Utilisation de la géométrie discrète
- ◆ Optimisation de l'algorithme

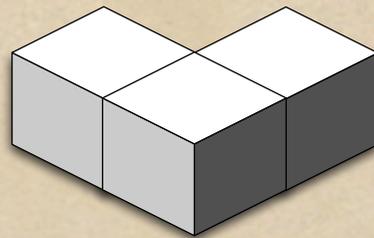
Ray-casting, ray-tracing, radiosit 



- ◆ Ray casting : calcul prohibitifs
- ◆ Ray tracing : rayons de lumi re renvers s, mauvaise gestion du diffus
- ◆ Radiosit  : approche globale contr l e

Radiosité basée sur les voxels

- ◆ Radiosité : une équation caractérisant les transferts d'énergie
- ◆ Discrétisation de l'équation continue sur des volumes élémentaires (voxels)



- ◆ Résolution itérative

Équation de radiosité

$$B(x) = E(x) + \rho_d(x) \int_{y \in \text{scene}} B(y) \frac{\cos \theta_x \cos \theta_y}{\pi r^2} V(x, y) dy$$

x : point

$B(x)$: radiosité

$E(x)$: émittance

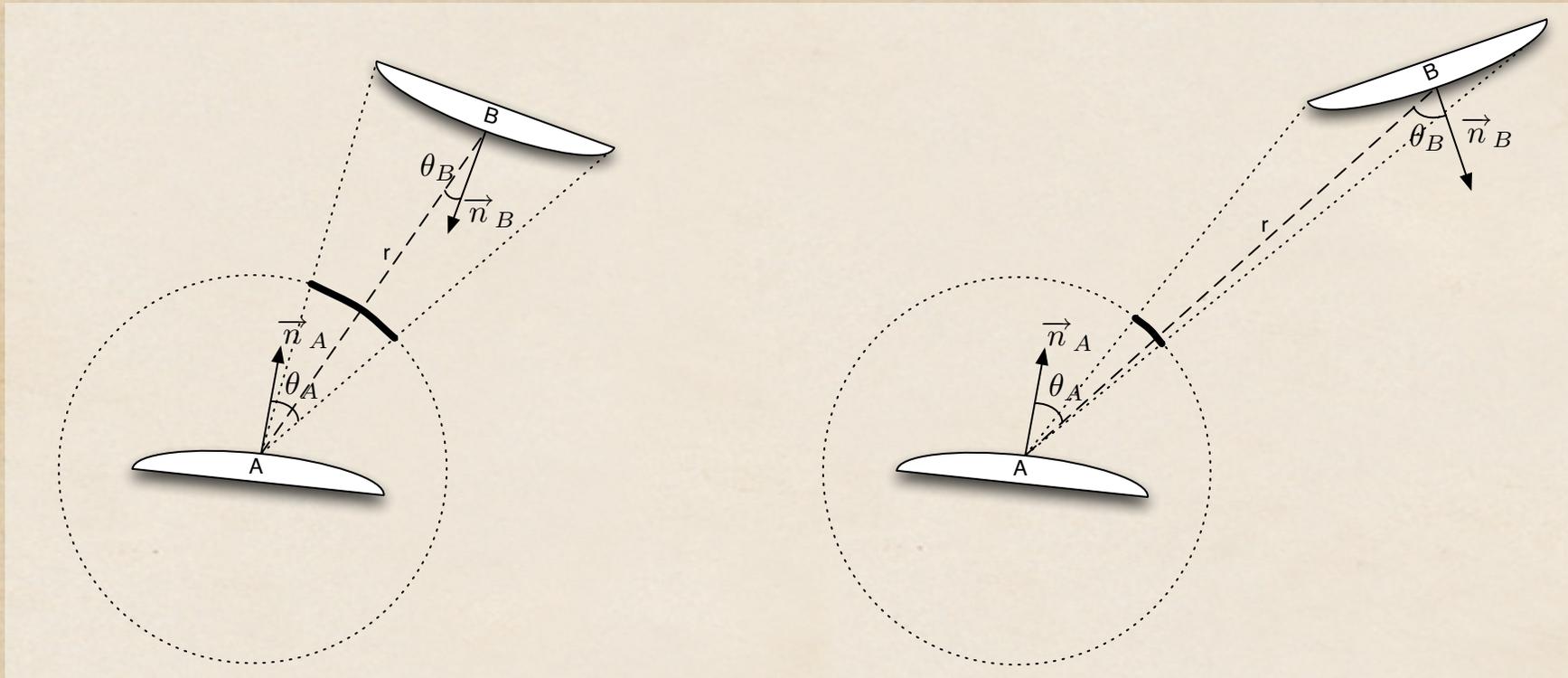
$\rho_d(x)$: coefficient de réflexion
diffuse

θ_x : angle avec la normale

r : $\|x-y\|^2$

$V(x, y)$: visibilité (0 ou 1)

Angles solides



- ◆ Reflète bien distance et orientation

Équation de radiosité

$$B(x) = E(x) + \rho_d(x) \int_{y \in \text{scene}} B(y) \frac{\cos \theta_x \cos \theta_y}{\pi r^2} V(x, y) dy$$

x : point

$B(x)$: radiosité

$E(x)$: émittance

$\rho_d(x)$: coefficient de réflexion
diffuse

θ_x : angle avec la normale

r : $\|x-y\|^2$

$V(x, y)$: visibilité (0 ou 1)

Équation de radiosité

$$B(x) = E(x) + \rho_d(x) \int_{y \in \text{scene}} B(y) \frac{\cos \theta_x \cos \theta_y}{\pi r^2} V(x, y) dy$$

x : point

θ_x : angle avec la normale

$B(x)$: radiosité

r : $\|x-y\|^2$

$E(x)$: émittance

$V(x, y)$: visibilité (0 ou 1)

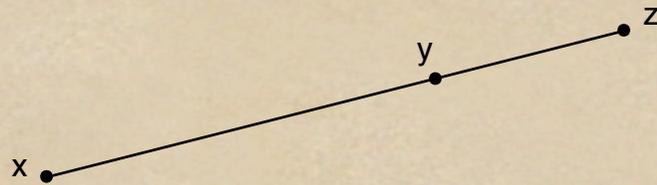
$\rho_d(x)$: coefficient de réflexion
diffuse

- ◆ Importance du cas diffus idéal, d'objets opaques
- ◆ Problème de la visibilité $V(x, y)$

Discretisation (1)

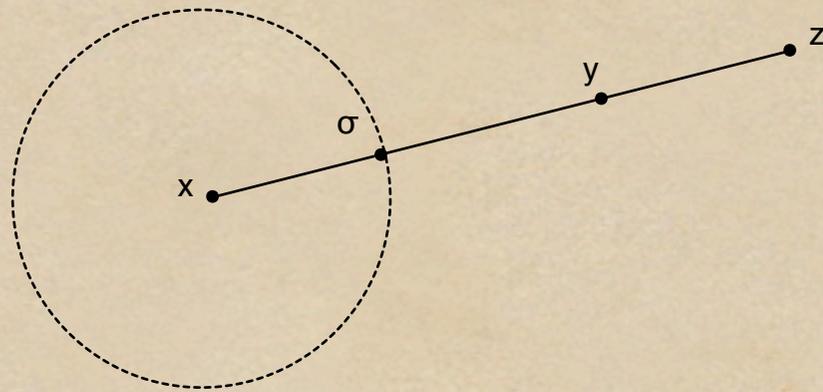
$$B(x) = E(x) + \rho_d(x) \int_{y \in scene} B(y) \frac{\cos \theta_x \cos \theta_y}{\pi r^2} V(x, y) dy$$

Visibilité



- ◆ Un seul point est vu dans une direction donnée
- ◆ On déduit la visibilité de la direction plutôt que du point “cible”

Visibilité



- ◆ Un seul point est vu dans une direction donnée
- ◆ On déduit la visibilité de la direction plutôt que du point "cible"

Discrétisation (1)

$$B(x) = E(x) + \rho_d(x) \int_{y \in scene} B(y) \frac{\cos \theta_x \cos \theta_y}{\pi r^2} V(x, y) dy$$

Discretisation (1)

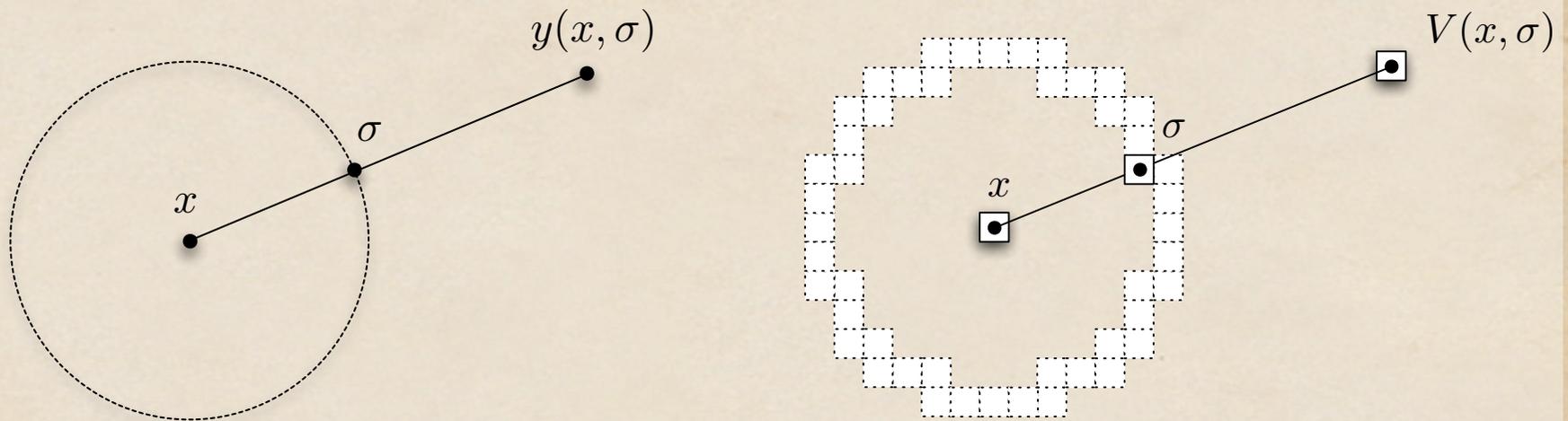
$$B(x) = E(x) + \rho_d(x) \int_{y \in \text{scene}} B(y) \frac{\cos \theta_x \cos \theta_y}{\pi r^2} V(x, y) dy$$

$S_R(x)$: Sphère de rayon R centrée sur x

$y(x, \sigma)$: premier point rencontré depuis x dans la direction de σ

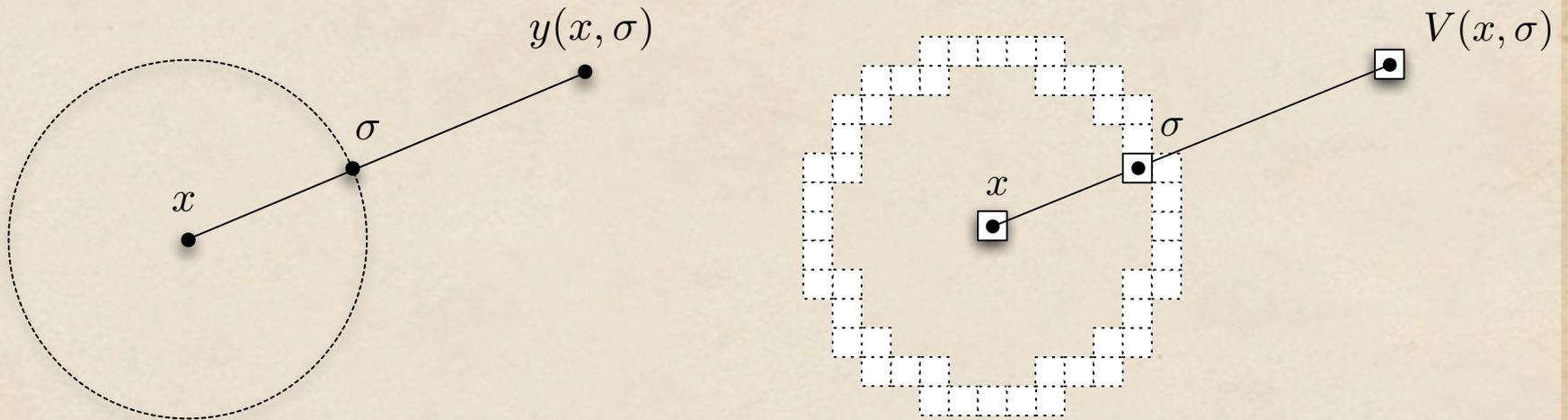
$$\int_{\sigma \in S_R(x)} B(y(x, \sigma)) \frac{\cos \theta_x}{\pi} \underbrace{\left(\cos \theta_y \frac{d(y(x, \sigma))}{\|x - y(x, \sigma)\|^2} \right)}_{\text{angle solide de } y \text{ vu depuis } x}$$
$$\int_{\sigma \in S_R(x)} B(y(x, \sigma)) \frac{\cos \theta_x}{\pi} d\sigma$$

Discrétisation (2)



- ◆ Utilisation d'une sphère discrète
- ◆ Un angle solide est alors attaché à un voxel de la sphère
- ◆ Notion de "facteur de direction"

Discrétisation (3)



$$\int_{\sigma \in S_R(x)} B(y(x, \sigma)) \frac{\cos \theta(x, y(x, \sigma))}{\pi} d\sigma$$

$$\approx \sum_{\sigma \in \Sigma_R(x)} B(V(x, \sigma)) \frac{\cos \theta(x, V(x, \sigma))}{\pi} A(x, V(x, \sigma))$$

Résolution itérative (1)

- ◆ Résoudre

$$B(x) = E(x) + \rho_d(x) \sum_{\sigma \in \Sigma_R(x)} B(V(x, \sigma)) \frac{\cos \theta(x, V(x, \sigma))}{\pi} A(x, V(x, \sigma))$$

- ◆ Utiliser une suite convergente

$$B_{n+1}(x) = E(x) + \rho_d(x) \sum_{\sigma \in \Sigma_R(x)} B_n(V(x, \sigma)) \frac{\cos \theta(x, V(x, \sigma))}{\pi} A(x, V(x, \sigma))$$

- ◆ C'est une succession de propagations;
chaque itération est un rebond

Résolution itérative (2)

$$B_{n+1}(x) = E(x) + \rho_d(x) \sum_{\sigma \in \Sigma_R(x)} B_n(V(x, \sigma)) \frac{\cos \theta(x, V(x, \sigma))}{\pi} A(x, V(x, \sigma))$$

*[calculer les visibilités respectives $V(x, \sigma)$
(informations coûteuses en temps et en espace)]*
...

[Résoudre l'équation]

Pour chaque itération **faire**

Pour chaque voxel **faire**

Pour chaque direction **faire**

 Mettre à jour B_{n+1}

Fin Pour

Fin Pour

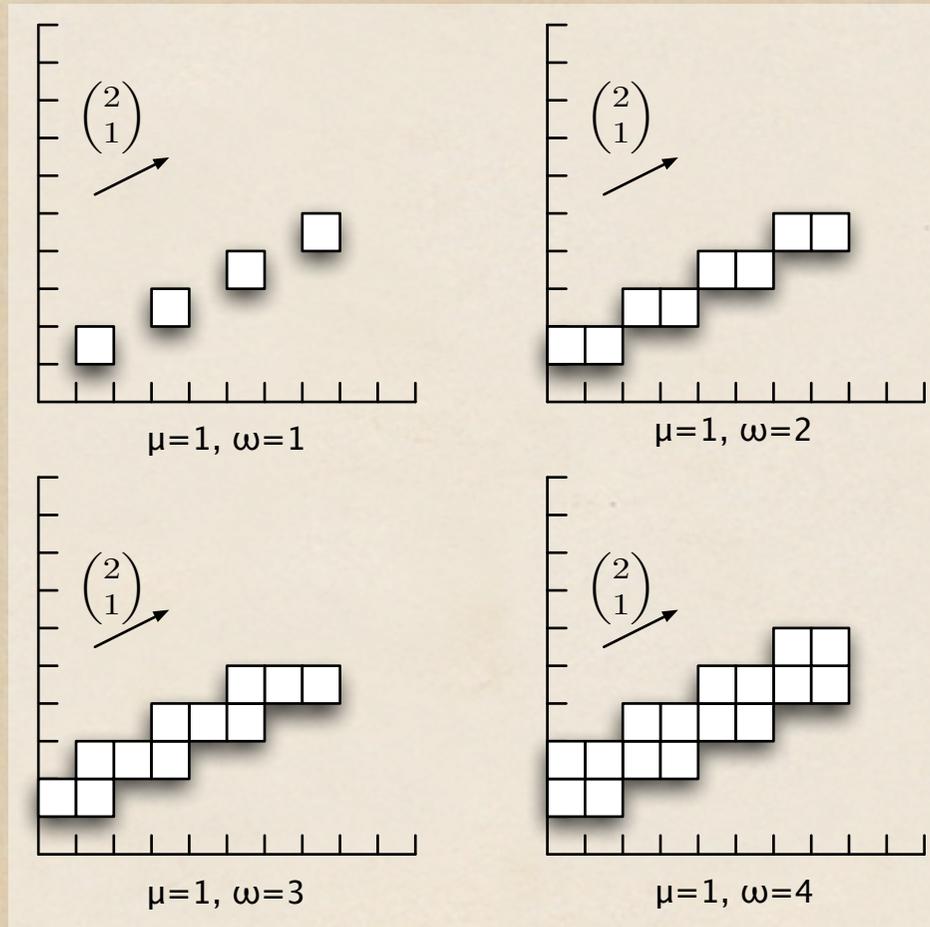
Fin Pour

Utilisation de la géométrie discrète

- ◆ *Système de coordonnées entières*
- ◆ *Utilisation de droites discrètes*
- ◆ *Discretisation implicite de l'espace en plus des surfaces*

Droites discrètes 2D

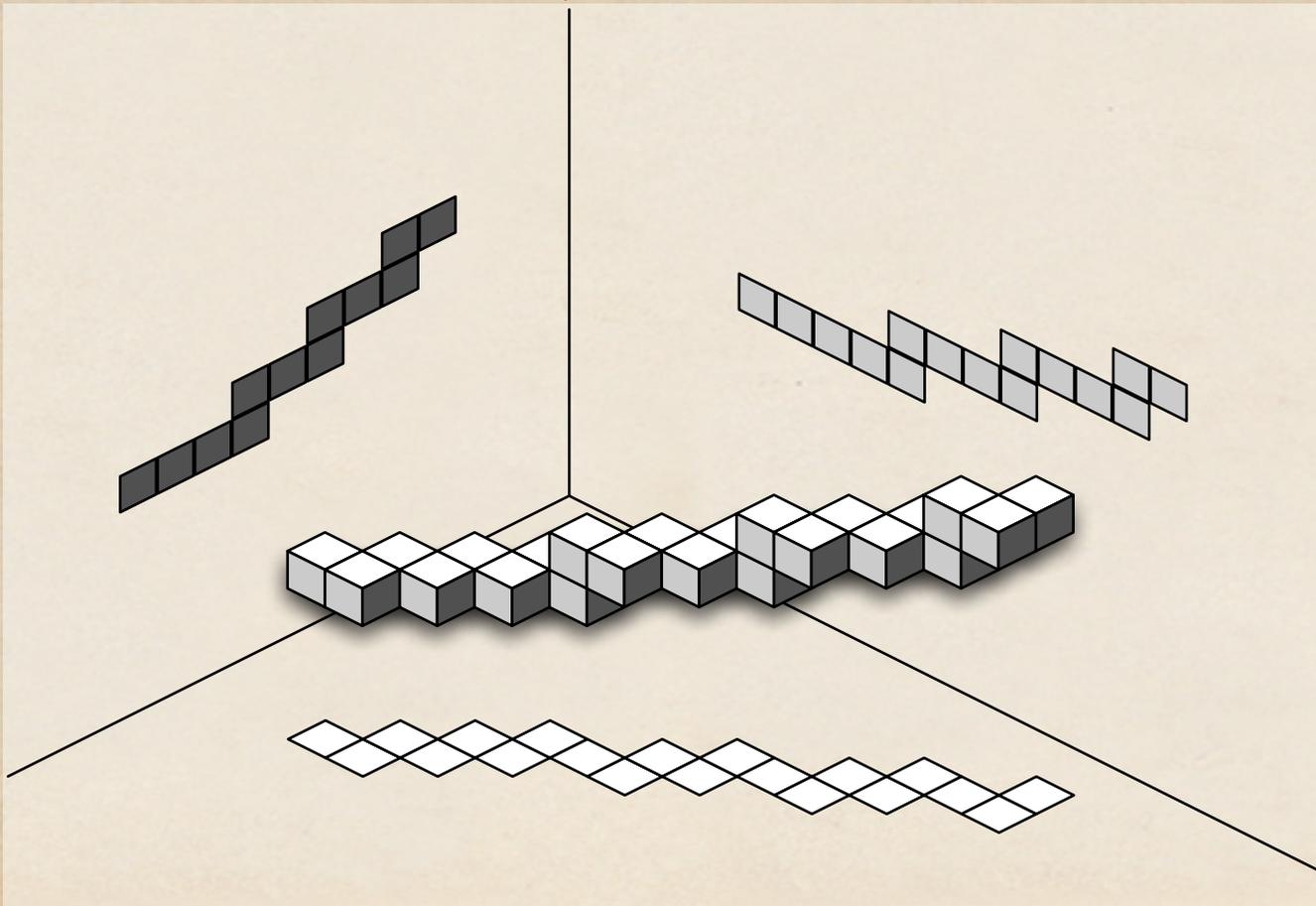
$$(x, y) \in \mathbb{Z}^2 / \mu \leq ax - by < \mu + \omega$$



Droites discrètes 3D (1)

Direction (a, b, c) $a \geq b \geq c$

$$(x, y, z) \in \mathbb{Z}^3 / \begin{cases} \mu \leq cx - az < \mu + \omega \\ \mu' \leq bx - ay < \mu' + \omega' \end{cases}$$



Droites discrètes 3D (2)

Direction (a, b, c) $a \geq b \geq c$

$$(x, y, z) \in \mathbb{Z}^3 / \begin{cases} \mu \leq cx - az < \mu + \omega \\ \mu' \leq bx - ay < \mu' + \omega' \end{cases}$$

- ◆ Se projette en droites discrètes 2D
- ◆ Deux droites discrètes 2D permettent de reconstruire la droite discrète 3D



Partitionnement de l'espace

- ◆ On partitionne l'espace en droites discrètes 3D

Direction (a, b, c) $a \geq b \geq c$

$$L(\mu, \mu') : (x, y, z) \in \mathbb{Z}^3 / \begin{cases} \mu \leq cx - az < \mu + \omega \\ \mu' \leq bx - ay < \mu' + \omega' \end{cases}$$

$$L(i * \omega, j * \omega') : (x, y, z) \in \mathbb{Z}^3 / \begin{cases} i * \omega \leq cx - az < (i + 1) * \omega \\ j * \omega' \leq bx - ay < (j + 1) * \omega' \end{cases}$$

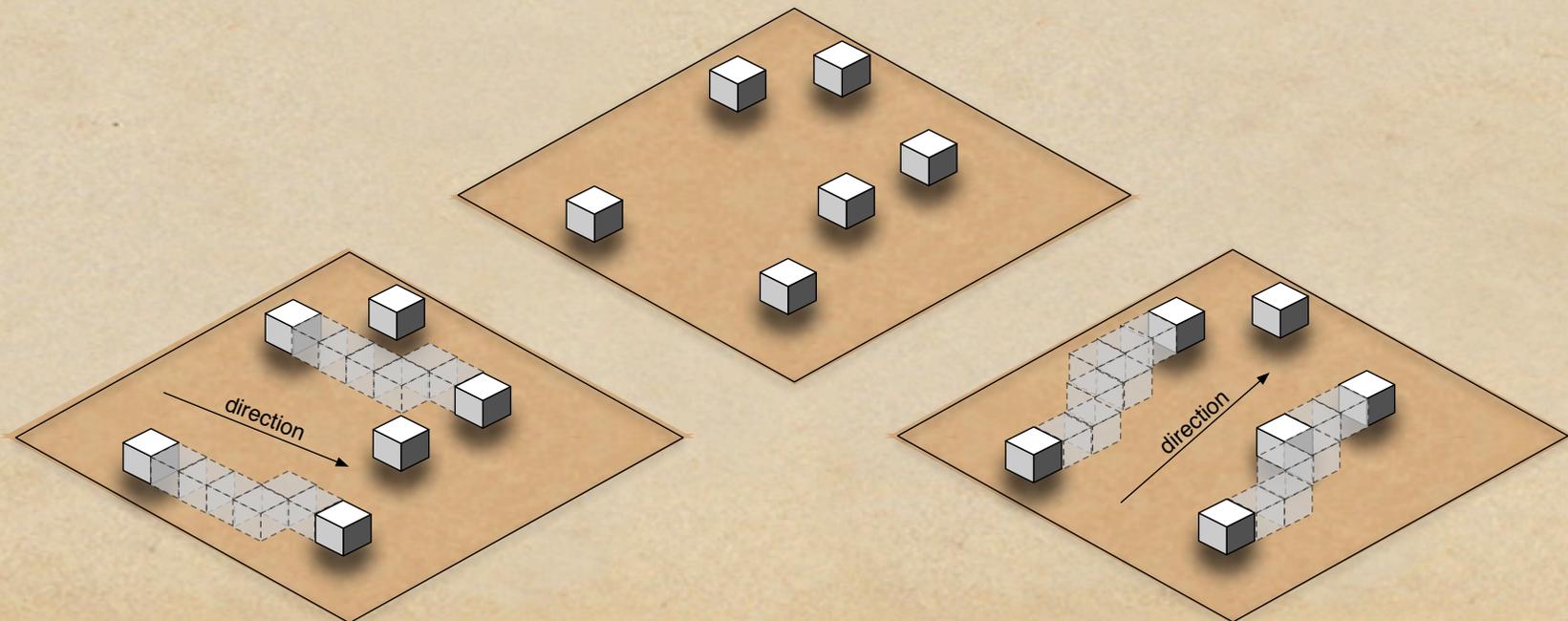
Pour un voxel (x, y, z) donné : $i = \left\lfloor \frac{cx - az}{\omega} \right\rfloor$ $j = \left\lfloor \frac{bx - ay}{\omega'} \right\rfloor$

Optimisation de l'algorithme

- ◆ Privilégier les directions
- ◆ Utiliser un ordre de parcours intelligent
- ◆ Conséquences : aucun pré-calcul coûteux et une complexité quasi optimale

Privilégier les directions (1)

- ◆ Traiter la visibilité direction par direction
- ◆ Pour chaque direction, partitionner en listes de voxels de même faisceau



Privilégier les directions (2)

Pour chaque itération **faire**

Pour chaque direction **faire**

[Répartition ($O(N)$)]

Pour chaque voxel (dans l'ordre choisi) **faire**

 | L'insérer dans la liste codant le faisceau

Fin Pour

[Tri ($O(N \log N)$)]

 Trier chaque liste

[Propagation ($O(N)$)]

Pour chaque liste (triée!) **faire**

 | Deux voxels successifs sont mutuellement visibles

 | Exécuter les propagations entre ces paires de voxels

Fin Pour

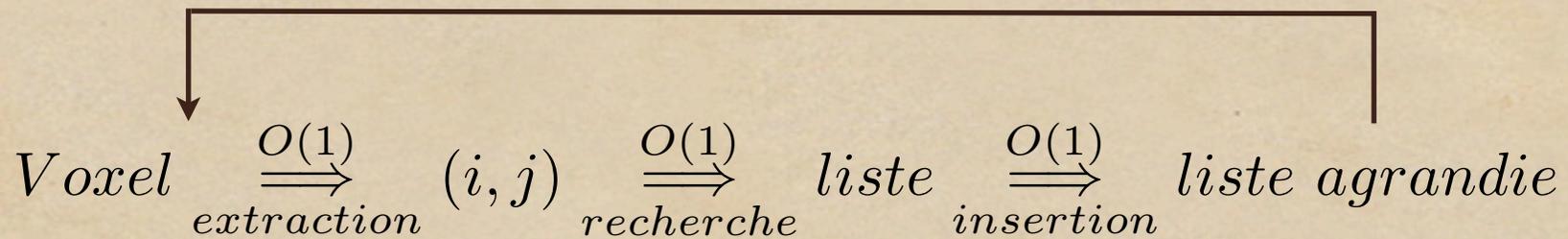
Fin Pour

Fin Pour

- ◆ Disparition des pré-calculs de visibilité et du stockage associé

Privilégier les directions (3)

◆ Répartition : N fois : $O(N)$



◆ Tri : m listes de tailles variables $O(N \ln N)$

◆ Propagation : $O(N)$

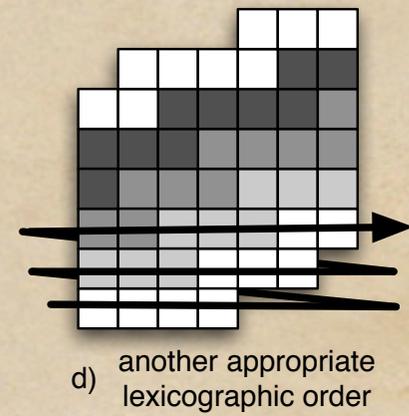
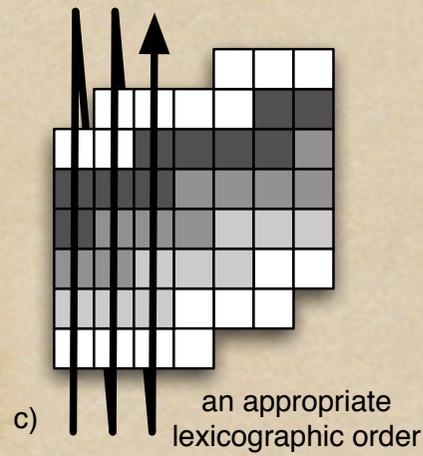
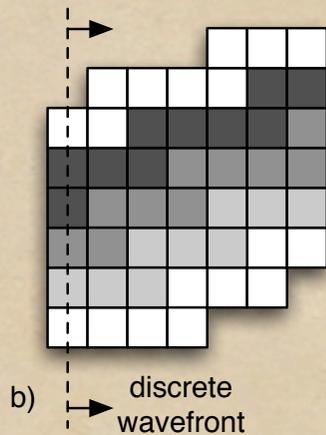
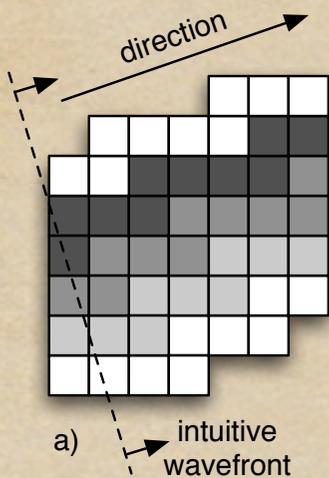
◆ Nombre d'itérations I et de directions D constants; bilan : $O(I \times D \times N \ln N)$

Privilégier les directions (4)

- ◆ Une étape de tri des listes a été nécessaire
- ◆ On s'en passe en traitant les voxels dans un ordre adapté
- ◆ Les listes sont alors remplies dans le bon ordre, l'étape de tri devient superflue

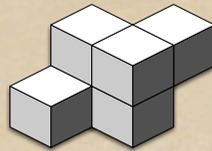
Parcours intelligent (1)

- ◆ Un “front d’onde” lexicographique convient



Parcours intelligent (2)

- ◆ L'épaisseur restreinte empêche les bulles



- ◆ L'ordre respectif de x , y , z importe peu : réduit à 8 ordres lexicographiques
- ◆ symétrie : 4 ordres lexicographiques suffisent
- ◆ La préparation de ces informations coûte $O(N \log N)$ en temps et $O(N)$ en espace.

Algorithme final

[Préparation ($O(N \log N)$)]

Calculer les 4 ordres lexicographiques

[Résolution]

Pour chaque itération **faire**

Pour chaque direction **faire**

 Choisir l'ordre lexicographique adapté

[Répartition ($O(N)$)]

Pour chaque voxel (dans l'ordre choisi) **faire**

 | L'insérer en fin de la liste codant le faisceau

Fin Pour

[Propagation ($O(N)$)]

Pour chaque liste (automatiquement triée!) **faire**

 | Deux voxels successifs sont mutuellement visibles

 | Exécuter les propagations entre ces paires de voxels

Fin Pour

Fin Pour

Fin Pour

Complexité (1)

Complexité spatiale

- ◆ L'ensemble des listes contient une et une seule référence vers chacun des N voxels
- ◆ Complexité spatiale $O(N)$

Complexité (2)

Complexité temporelle

- ◆ Complexité en $O(N \ln N) + O(I \times D \times N)$
- ◆ $O(N \log N)$ négligeable vu les constantes
- ◆ Complexité quasi linéaire

Conclusion

- ◆ Disparition des précalculs et complexité spatiale optimale ($O(N)$)
- ◆ Complexité temporelle quasi linéaire
- ◆ Redondance des calculs à chaque itération, mais peu d'itérations, possibilité de sauvegarde
- ◆ Algorithme bon candidat à la parallélisation

Illustration



$2 \cdot 10^6$ voxels, $15 \cdot 10^3$ directions, 6 itérations : 27h

Perspectives

- ◆ Expérimenter la parallélisation
- ◆ Ne plus se limiter au cas diffus idéal