

# M2-Images

J.C. Iehl, B Péroche

November 16, 2007

# Structures accélératrices et lancer de rayons

## pourquoi ?

- ▶ calculer toutes les intersections est trop long,
- ▶ trouver l'intersection la plus proche revient à chercher le minimum d'un ensemble d'intersections.

## idées

- ▶ ne calculer que ce qui est nécessaire,
- ▶ trouver efficacement le minimum des résultats.

# Structures accélératrices et lancer de rayons

## organiser

- ▶ mêmes idées que pour la recherche dichotomique,
- ▶ trier les données pour faire une recherche efficace.

## exploiter l'organisation

- ▶ pour ne calculer que ce qui est nécessaire,
- ▶ pour traiter les résultats du min au max (et donc trouver facilement le min)

# Intuition

## dichotomie en 1D

- ▶ rechercher une valeur dans un intervalle  $[x_{min} \ x_{max}]$ ,
- ▶ on connaît : l'intervalle de l'ensemble (l'englobant),
- ▶ l'ordre des intervalles dans l'ensemble (du min au max).

## extension à la 3D

- ▶ rechercher un point dans un cube aligné sur les axes  $[x_{min} \ x_{max}] \times [y_{min} \ y_{max}] \times [z_{min} \ z_{max}]$ ,
- ▶ on connaît : le cube de l'ensemble (l'englobant),
- ▶ quelle notion d'ordre en 3D ?

# Intuition

## ordre en 3D

- ▶ ?
- ▶ utiliser la direction du rayon pour ordonner les cubes,
- ▶ parcourir les cubes en s'éloignant de l'origine du rayon,
- ▶ cf. *trouver le min efficacement*.

# Structures Classiques

## des arbres

- ▶ octree : récursion sur le volume englobant de la scène, les 3 axes simultanément,
- ▶ kd-tree : récursion sur le volume englobant de la scène, un axe à la fois,
- ▶ BVH : récursion sur les objets, construit a posteriori les volumes englobants, les 3 axes simultanément,
- ▶ BIH : récursion sur les objets, construit a posteriori les intervalles englobants, un axe à la fois

des grilles, des grilles à plusieurs niveaux, des tables de hachage, etc.

# Parcours

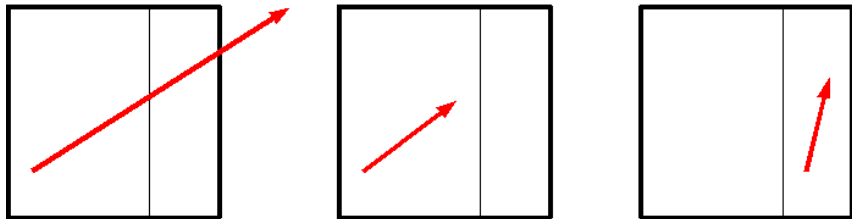
rayon : origine + direction

- ▶ déterminer si un fils peut intersecter le rayon,
- ▶ respecter l'ordre : s'éloigner de l'origine.

exploiter les propriétés de la structure

- ▶ structures récursives sur le volume :  
les fils ne s'intersectent pas,
- ▶ structures récursives sur les objets :  
les fils peuvent s'intersecter.

## Parcours (volume)

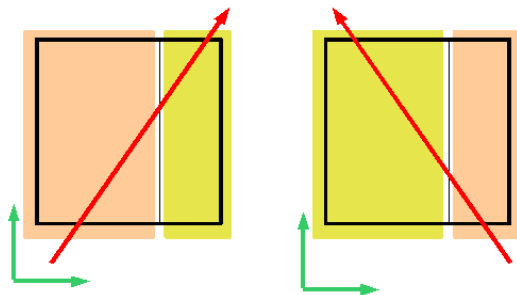


déterminer quels fils parcourir

- ▶ un seul,
- ▶ les deux ...
- ▶ ou aucuns.



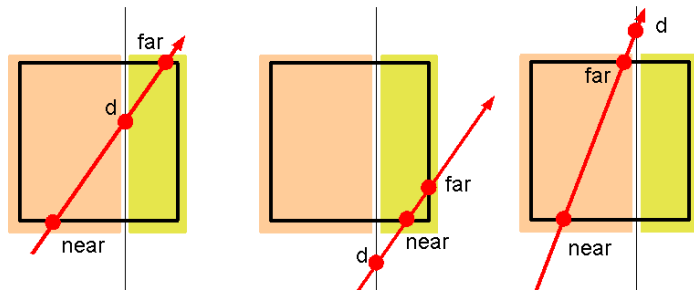
# Parcours (volume)



dans quel ordre

le signe de la direction du rayon suffit ...

# Parcours (volume)

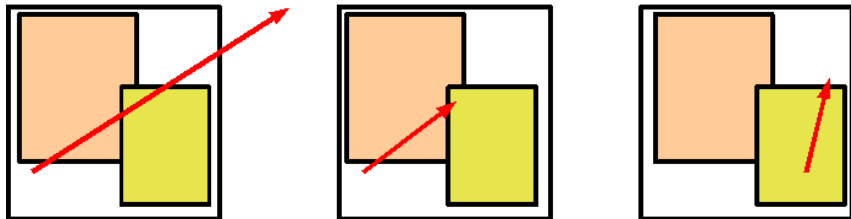


quels fils ?

dépend de la position de  $d$  par rapport à  $far$  et  $near$

- ▶  $d > far$
- ▶  $d < near$
- ▶  $near \leq d \leq far$

## Parcours (objets)



déterminer quels fils parcourir

calculer l'intersection du cube et du rayon pour les deux fils

$[near\ far]_{gauche}$ ,  $[near\ far]_{droit}$

# Parcours (objets)

dans quel ordre

ordre sur les intervalles  $[near\ far]_{gauche}$ ,  $[near\ far]_{droit}$

précalculer

- ▶ les fils ne changent pas de place pendant le calcul d'une image,
- ▶ inutile de calculer plusieurs fois leur disposition, déterminer une relation avec la direction du rayon.

# Construction

découper l'espace en cubes disjoints,

ou, grouper les objets et déterminer l'espace occupé,

trop de possibilités, comment choisir a priori la meilleure hiérarchie ?

# BVH : les détails

cas simple : 2 fils, volumes englobants : cubes alignés sur les axes.

## Construction

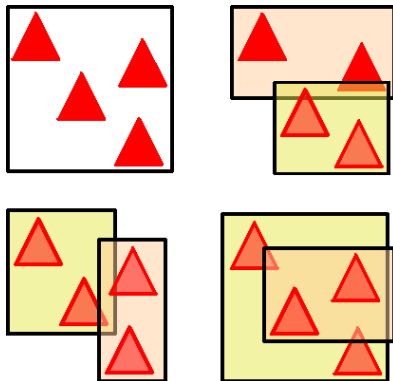
en fonction du volume occupé par les fils, et du temps de calcul de l'intersection du rayon avec les objets associés aux fils.

## Parcours ordonné

pas obligatoire, mais beaucoup plus efficace.

# Construction

trouver la meilleure répartition des objets pour chaque noeud  
choisir un plan candidat et répartir les objets.



# Construction

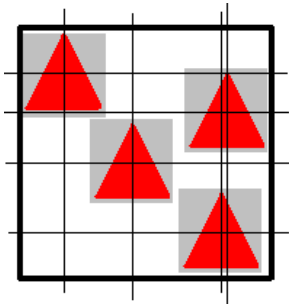
minimiser  $T$  : trouver le meilleur candidat

$$T = T_{gauche} + T_{droit}$$
$$T_{fils} = T_{cube} + \frac{A(fils)}{A(noeud)} N(fils) \times T_{objet}$$



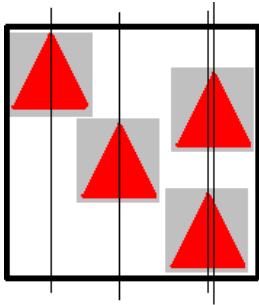
## Construction : algorithme

- ▶ travaille sur les cubes englobants de chaque objet,
- ▶ les candidats sont les 3 plans passants par le centre des cubes englobants.



## Construction : algorithme

- ▶ travaille sur un axe à la fois,
- ▶ teste tous les centres,
- ▶ évalue  $T$  à chaque fois et garde le meilleur (sur les 3 axes).



# Construction : algorithme

- ▶ construction récursive :
- ▶ critère d'arrêt ? lorsqu'il n'est plus intéressant de continuer.

# Evaluer T : algorithme

$$T_{\text{fils}} = T_{\text{cube}} + \frac{A(\text{fils})}{A(\text{noeud})} N(\text{fils}) \times T_{\text{objet}}$$

déterminer les 2 sous ensembles d'objets + cubes englobants

- ▶ naif : re-trier à chaque fois, cf. sujet TP,
- ▶ incrémental : trier une seule fois par axe, puis exploiter l'ordre pour contruire les cubes englobants,
- ▶ il est facile de calculer le min et le max d'un ensemble lorsqu'on insère un élément,
- ▶ mais pas le contraire (lorsqu'un supprime un élément) ?

## Evaluer T : algorithme

- ▶ parcourir de min vers max et construire la partie gauche et son cube englobant :
- ▶  $N(\text{fils}_{\text{gauche}}), A(\text{fils}_{\text{gauche}})$
- ▶ parcourir de max vers min pour la partie droite :
- ▶  $N(\text{fils}_{\text{droit}}), A(\text{fils}_{\text{droit}})$
- ▶ tous les termes de T sont connus pour tous les candidats,
- ▶ finir l'évaluation de T pour chaque candidat,
- ▶ garder le meilleur, répartir les objets en 2 sous-ensembles,
- ▶ recommencer