

Élimination des parties cachées

Élimination des parties cachées

- Il est important d'identifier les parties de la scène qui sont visibles à partir d'un point de vue donné.
- Il existe plusieurs méthodes à cet effet.
- Le choix de la méthode dépend de la nature de la scène et des objets

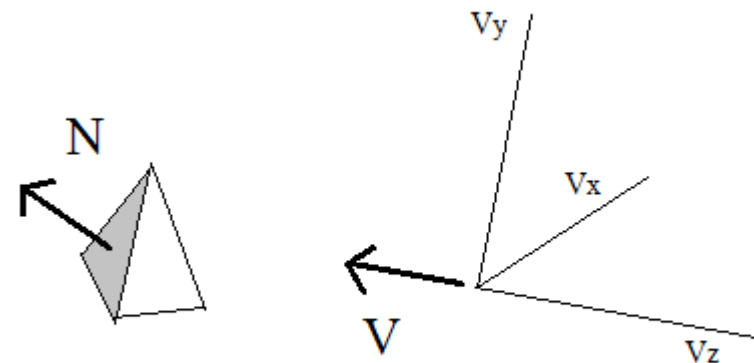
Élimination des parties cachées

Les algorithmes de détection des surfaces visibles peuvent s'appliquer :

- Sur les objets directement, c'est les méthodes espace-objet.
- Ou bien sur leur projection, c'est les méthodes opérant sur l'espace-image
- Des méthodes peuvent opérer de manière combinée sur l'espace image et l'espace objet

Détection des faces arrières

- Ou *back-face detection*, c'est une méthode simple et rapide qui opère sur l'espace objet
- Les faces arrières d'un polyèdre (des polygones) ne seront pas affichées
- En considérant le vecteur normal $N=(A, B, C)$ à la surface d'un polygone et V le vecteur de direction de l'observation.
- Si $V \cdot N > 0$ alors ce polygone est une face arrière.



Détection des faces arrières

Lorsque les coordonnées sont transformées en coordonnées de projection, la direction du regard est parallèle à l'axe Z, alors

$$\begin{aligned} V &= (0, 0, V_z) \\ V \cdot N &= V_z \cdot C \end{aligned}$$

Donc, on se base sur C pour définir si le polygone est une face cachée ou pas.

Un polygone est une face arrière si $C \leq 0$

Détection des faces arrières

Cette méthode s'applique sur les objets convexe où les facette sont complètement visibles ou pas

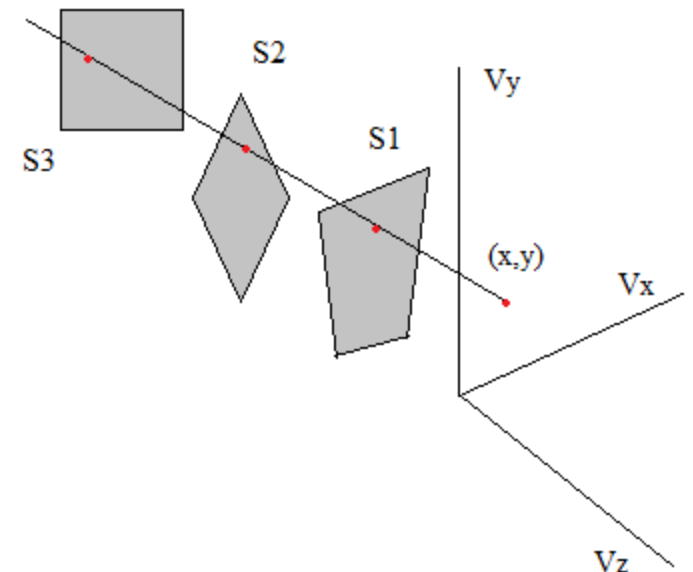
Le tampon de profondeur

Ou *depth-buffer*, c'est une méthode qui opère dans l'espace-image, appelée aussi *z-buffer*.

Chaque surface est traitée point par point.

Pour chaque point sa profondeur est stockée (z-buffer) en plus de sa couleur (frame-buffer).

Ces valeurs sont celle de la surface la plus proche du plan de visualisation



Le tampon de profondeur

Pour chaque point (x,y) de l'image

$z\text{-buffer}(x,y) = \text{infini}$

$\text{frame-buffer}(x,y) = I_{\text{background}}$

fin pour

Pour chaque surface S

 pour chaque point (x,y)

 calculer la profondeur z

 si $(z < z\text{-buffer}(x,y))$ alors

$z\text{-buffer}(x,y) = z$, $\text{frame-buffer}(x,y) = I_S(x,y)$

 fin pour

fin pour

Le tampon de profondeur

Calcul de la profondeur :

La profondeur d'un point projeté en (x,y) est calculée à partir de l'équation de chaque surface :

$$z = \frac{-Ax - By - D}{C}$$

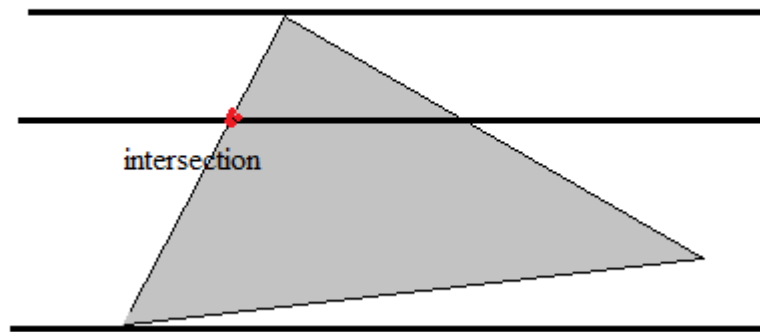
La profondeur d'un autre point de la surface peut être calculée par :

$$\Delta z = -\frac{A}{C}\Delta x - \frac{B}{C}\Delta y$$

Le tampon de profondeur

On parcourt la surface ligne par ligne, c'est le balayage de lignes.

- Déterminer l'intersection des coupes avec les bords de la surface
- Calculer la profondeur de chaque pixel à partir du pixel adjacent



Le tampon de profondeur

- Méthode simple pour la modélisation en surfaces polygonales
- Les points non affichés peuvent être éliminés
- Nécessite de la mémoire

Trie de la profondeur

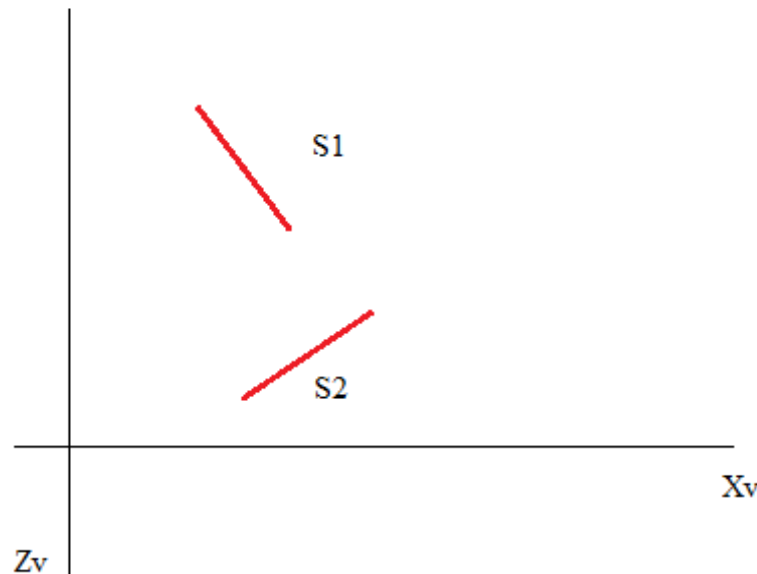
Méthode qui agit dans l'espace objet ainsi que dans l'espace image. Appelée aussi, *technique du peintre*.

Les surfaces sont triées selon l'ordre décroissant de leur profondeur

Elles sont affichées ensuite dans l'ordre en commençant par la plus éloignée

Trie de la profondeur

S'il n'y a pas de chevauchement entre une surface et celles qui suivent, alors la 1^{ère} (avec la profondeur la plus grande) est affichée.

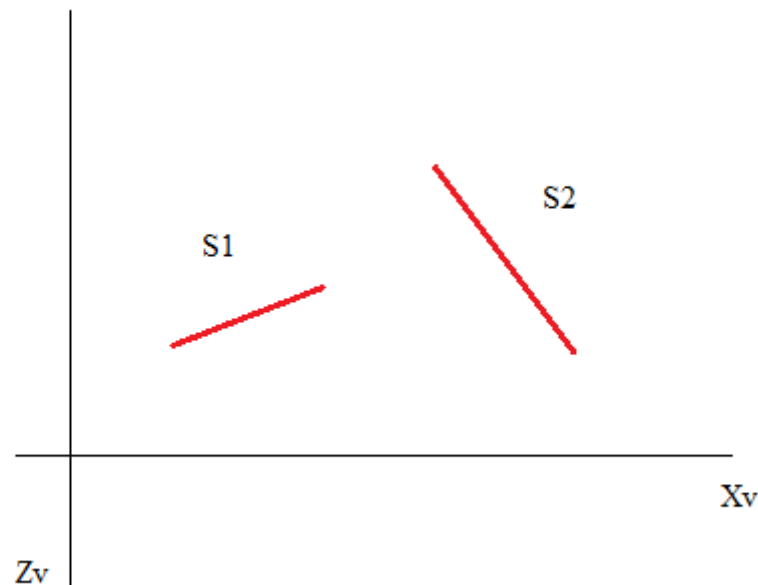


S'il y a des chevauchements dans la profondeur, entre une surface S et les surfaces qui la suivent, alors des testes sont effectués. Si au moins un teste est réussi, on ne change pas l'ordre.

Trie de la profondeur

Si :

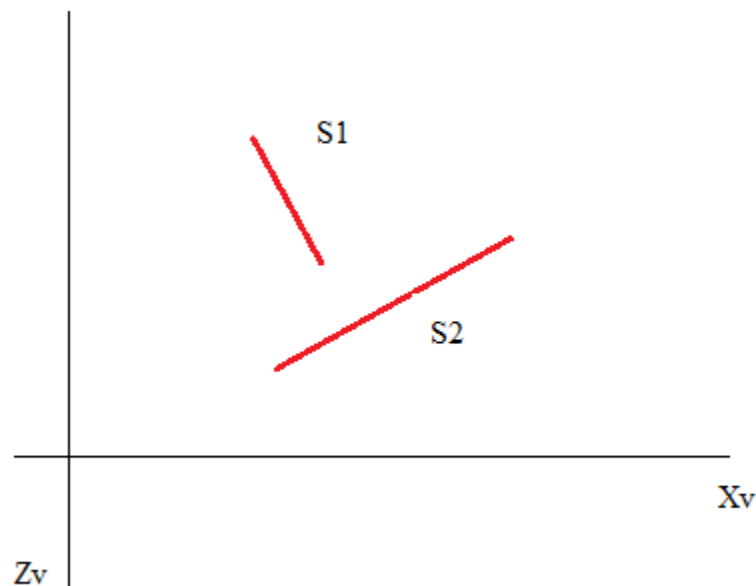
1- les rectangles englobant des surfaces dans le plan XY, ne se chevauchent pas



Trie de la profondeur

Si :

2- La surface S1 est complètement derrière les surfaces avec lesquelles elle chevauchent



Trie de la profondeur

Si :

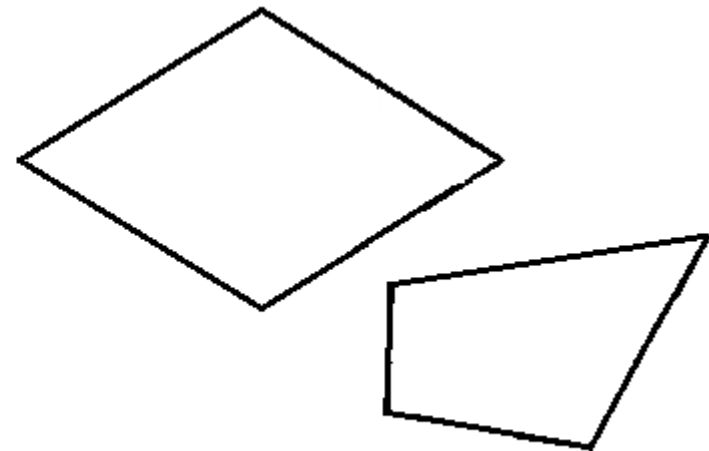
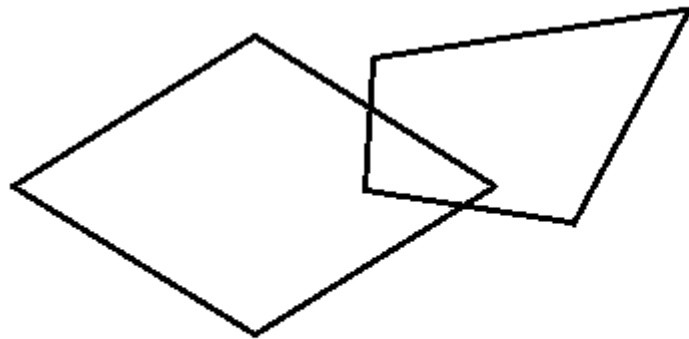
3- Les surfaces qui chevauchent sont complètement devant S1



Trie de la profondeur

Si :

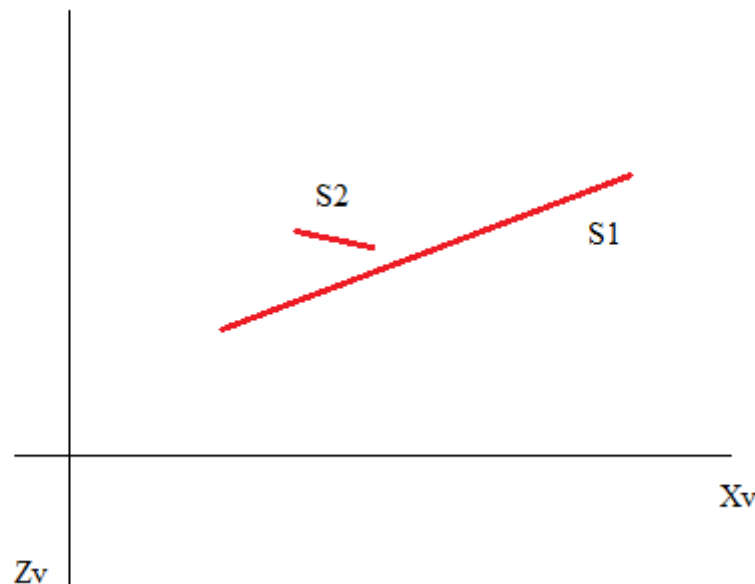
4- Les projections des surfaces sur le plan XY ne se chevauchent pas



Trie de la profondeur

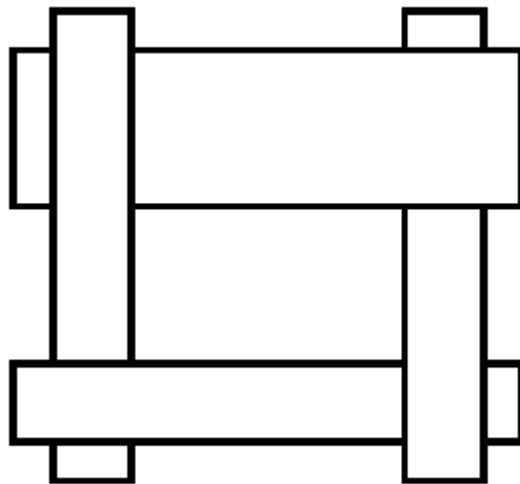
Si au moins un teste est réussi, on ne change pas l'ordre

Si tous les testes échouent, on inter-change les deux surfaces dans la liste de trie et on répète les testes pour les surfaces réordonnées



Trie de la profondeur

L'algorithme risque de boucler à l'infinie



Solution : subdiviser les facettes, utiliser les BSP pour trier

L'arbre BSP (Binary Spatial Partitionning)

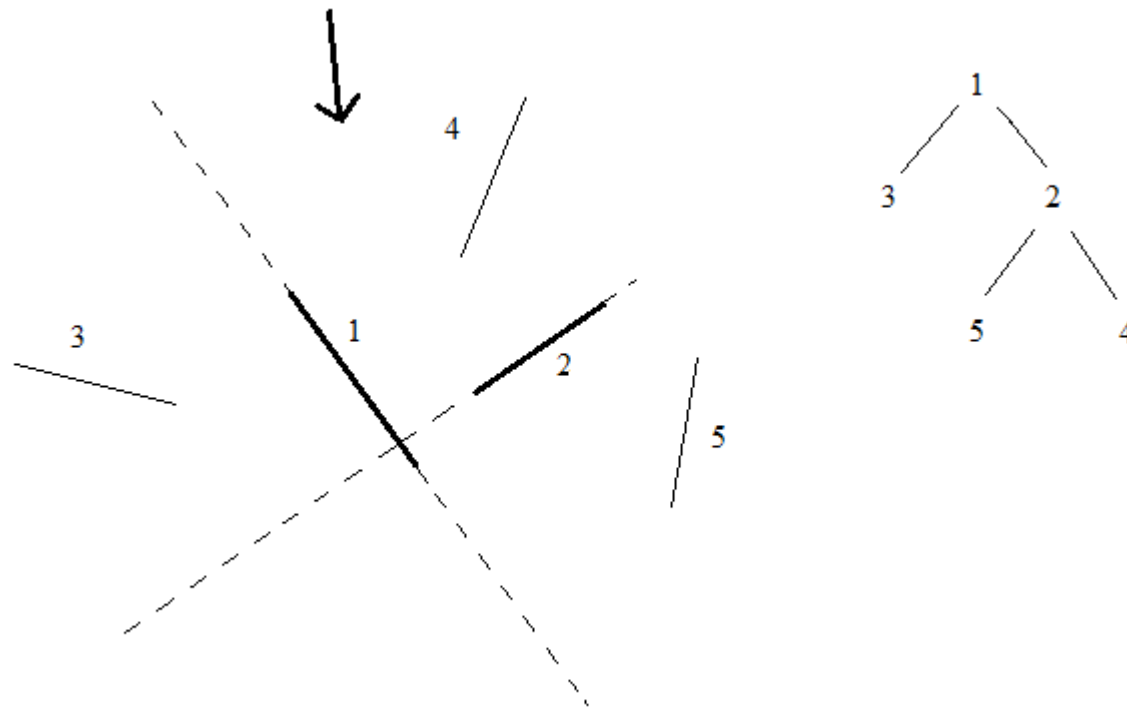
Les surfaces sont affichées à partir de l'arrière vers l'avant (technique du peintre)

Lorsque l'espace est divisé en deux parties, on définit celle qui est à l'intérieur et celle à l'extérieur relativement au plan de partitionnement

On fait un parcours infixe de l'arbre pour afficher

L'arbre BSP (Binary Spatial Partitionning)

Exemple

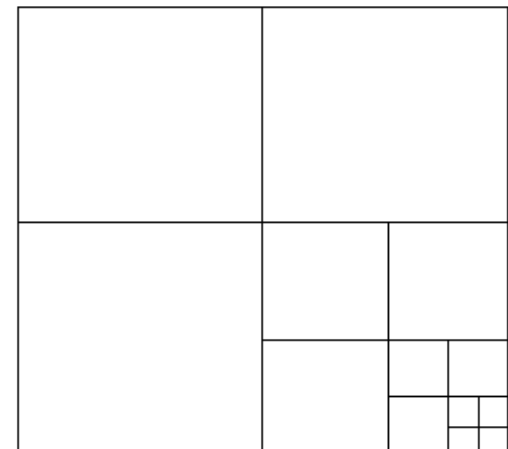


La subdivision de l'image

C'est une méthode qui opère essentiellement sur l'espace image mais utilise des opérations de l'espace objet

Chaque subdivision du plan image est examinée, si la visibilité ne pose pas de problèmes spécifiques alors l'affichage est réalisé dans cette région de l'image.

Sinon la région est divisée de nouveau en sous-régions et on réitère le processus de manière récursive.

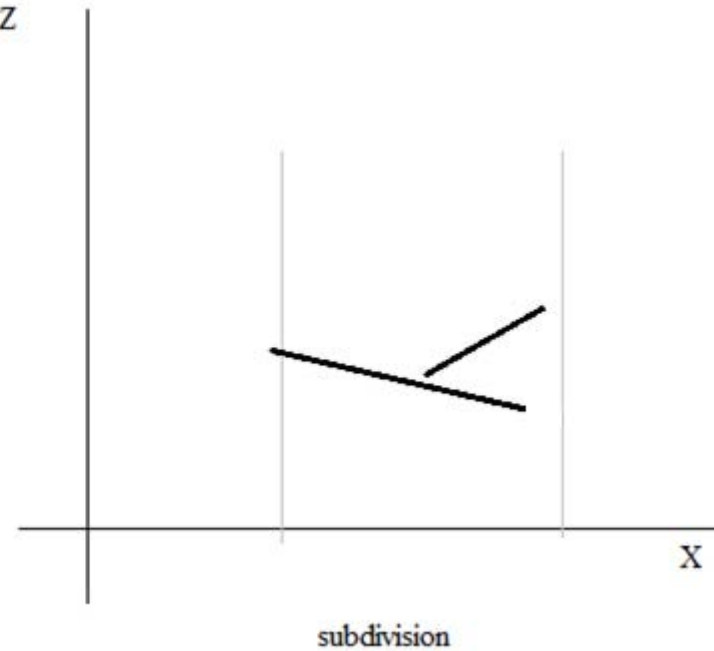
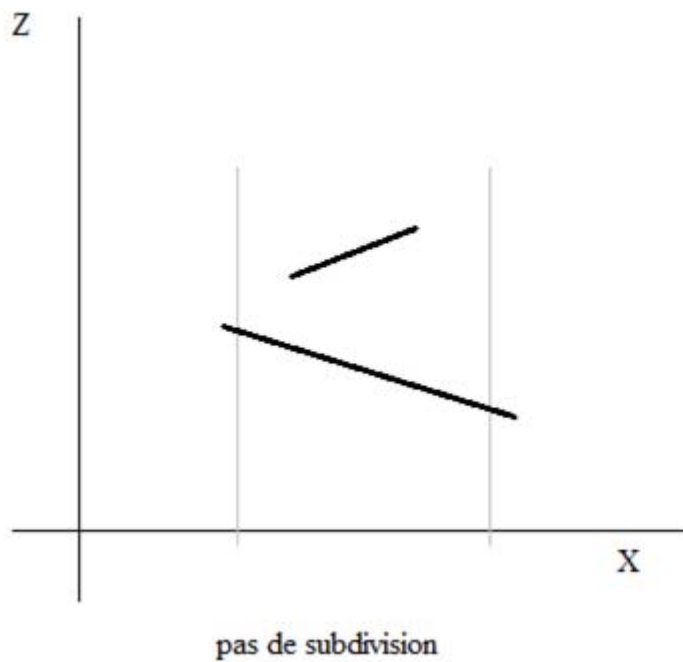


La subdivision de l'image

A chaque itération et pour chaque région, on effectue les testes suivants :

- Si tous les polygones sont à l'extérieur de la région alors pas d'affichage.
- Si un et un seul polygone intersecte la région, ou est contenu dans la région, ou contient la région, alors la partie du polygone correspondante est affiché.
- Si plus d'un polygone intersecte, ou est contenu, ou contient la région, alors si un de ces polygones est devant tous les autres, ce dernier est affiché, autrement, une subdivision est effectuée.

La subdivision de l'image



Le lancé de rayon

En considérant la ligne de vue qui va du plan de visualisation vers la scène, les objets qui sont en intersection avec cette ligne peuvent être définis.

Après le calcul de toutes les intersections entre les rayons et les surfaces, les surfaces visibles sont celles dont l'intersection est la plus proche du pixel.

