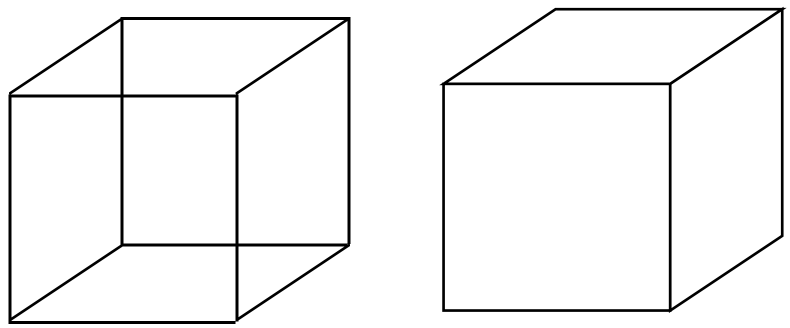
ELIMINATION DES LIGNES CACHÉES



Plusieurs algorithmes d'élimination de lignes cachées

- **dans l'espace "objet" (*object space* coordonnées du monde réel). Grande précision (celle de la machine): Roberts**
- **dans l'espace "image"(*"bit map"* coordonnées de l'écran) Précision faible:**
  1. **algorithme de l'horizon flottant**
  2. **Warnock**

Probème principal des algorithmes d'élimination des lignes (ou faces) cachées: le tri

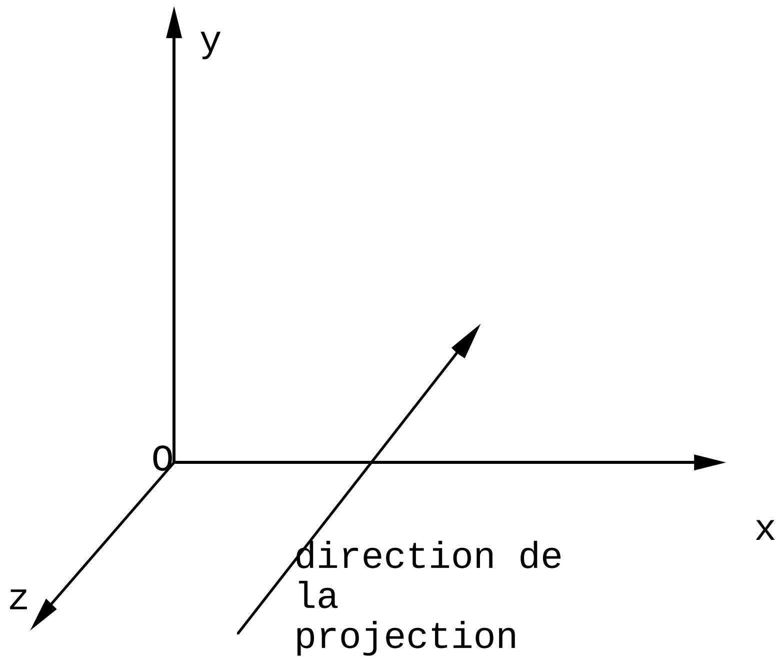
ALGORITHME DE ROBERTS

Hypothèses

- **les objets sont modélisés par des polyèdres *convexes*. On dispose d'une représentation qui permette de passer facilement des extrémités aux arêtes et aux faces.**
- **chaque face est définie par une équation du type:**

  $$ax + by + cz + d = 0$$

  où (a, b, c) est un vecteur normal orienté vers l'*intérieur* du polyèdre (déterminer le barycentre).



- **une projection orthogonale est fixée et son point de vision se situe à l'infini dans la direction des z positifs dans un repère "main droite"**

Problème

- **déterminer les segments visibles ou partiellement visibles à partir de l'infini vers les z positifs**

Méthode

- **pour chaque polyèdre, déterminer ses arêtes cachées par lui-même**

- **comparer chaque polyèdre à tous les autres pour déterminer les points d'intersection**

RAPPELS

1. **Equation d'un plan défini par un de ses points $\mathbf{P}_0 = (x_0, y_0, z_0)$ et un vecteur normal $\vec{u} = (a, b, c)$. Un point P$= (x, y, z)$ appartient au plan si et seulement si le produit scalaire $\mathbf{PP}_0.\vec{u}$ est nul:**

$$a(x - x_0) + b(y - y_0) + c(z - z_0) = 0$$

2. **Deux points $\mathbf{P}_0 = (x_0, y_0, z_0)$, $\mathbf{P}_1 = (x_1, y_1, z_1)$ sont de part et d'autre d'un plan d'équation**

$$f(x, y, z) = ax + by + cz + d = 0$$

si et seulement si

$$f(\mathbf{P}_0)f(\mathbf{P}_1) = f(x_0, y_0, z_0)f(x_1, y_1, z_1) < 0$$

3. **Deux vecteurs $\vec{u}_0 = (a_0, b_0, c_0), \vec{u}_1 = (a_1, b_1, c_1)$ font un angle aigu (inférieure à 90^0) si et seulement si leur produit scalaire est positif ou nul**

$$\vec{u}_0\vec{u}_1 = a_0a_1 + b_0b_1 + c_0c_1 \geq 0$$

UN POLYÈDRE SEUL

Première étape: pour chaque polyèdre, détermination les arêtes cachées par lui-même.

Equation d'une face

$$ax + by + cz + d = 0$$

On distingue

- **face *avant* (*front faces*): le point de vision se trouve du côté opposé à celui indiqué par la normale:** $c < 0$
- **face *arrière* (*back faces*): le point de vision se trouve du même côté que celui indiqué par la normale:** $c > 0$

Notion de visibilité d'une arête par rapport à une face F

- **un point A=** (x, y, z) **est *visible* par rapport à une face F s'il n'existe pas de point** (x, y, z') **de la face F tel que** $z' > z$
- **une arête est visible si tous ses points, sauf éventuellement une ou deux extrémités sont visibles**
- **une arête est *invisible* si tous ses points, sauf éventuellement une ou deux extrémités sont invisibles**
- **une arête est *partiellement visible* dans les autres cas**

Une arête est visible si au moins l'une des deux faces auxquelles elle appartient est avant. Elle est invisible dans le cas contraire.

POLYÈDRES ENTRE EUX

Deuxième étape: pour chaque polyèdre, comparer toutes ses arêtes restées visibles après la première étape, avec le volume des autres polyèdres et déterminer leur partie visible.

Tout se ramène à l'opération:

étant donnée une face avant d'équation $F(x, y, z) = 0$ et un segment AB déterminer si AB est

- **visible**
- **invisible**
- **partiellement visible**

Remarque: on suppose que la face n'est pas parallèle au plan y, z. Sinon AB est visible sauf s'il est dans le plan de la face (cas semblable à celui illustré dans le test 4, voir infra).

Méthode:

- **faire subir au segment AB quatre tests couvrant toutes les situations possibles**
- **arrêter la procédure dès que l'un des tests permet de conclure**
- **ordonner ces tests du moins coûteux vers le plus coûteux**

NOTATIONS

- **la face F (polygone convexe): $\mathbf{P}_1, \ldots, \mathbf{P}_n$ d'équation**

$$F(x, y, z) = 0$$

- **les extrémités de l'arête:**

$$\mathbf{A} = (x_A, y_B, z_A), \mathbf{B} = (x_B, y_A, z_B)$$

- **équation du plan parallèle à l'axe z et passant par le côté $\mathbf{P}_i\mathbf{P}_{i+1}$ du polygone, pour $i = 1, \ldots, n$:**

$$H_i(x, y, z) = 0$$

- **équation du plan parallèle à l'axe z passant par AB (identique à l'équation de la droite qui est la projection de AB sur x, y):**

$$Q(x, y, z) = x(y_B - y_A) - y(x_B - x_A) - x_A(y_B - y_A) + y_A(x_B - x_A) = 0$$

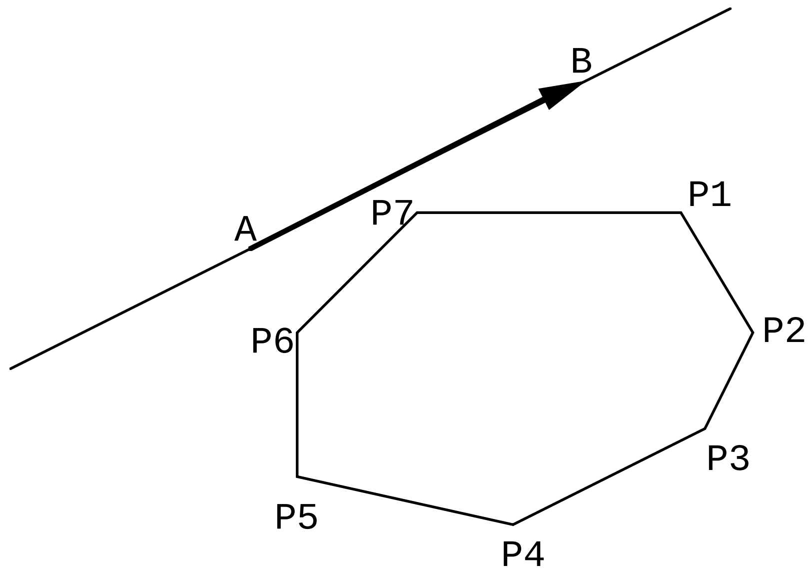
TEST 1

Les points A et B se trouvent devant le plan de la face:

$$F(\mathbf{A}) \leq 0 \textbf{ et } F(\mathbf{B}) \leq 0 \textbf{ ?}$$

Si oui, AB est *visible*

TEST 2



Le plan $Q(x, y, z) = 0$ **passant par l'arête AB ne coupe pas la face ?**

Si oui, AB est *visible*.

CALCUL DU TEST 2

- **le test équivaut à: tous les sommets P_i de la face sont strictement du même côté du plan à l'exception éventuellement de l'un d'entre eux qui est sur la plan.**
- **point de vue pratique: à tout sommet P_i de la face on associe l'entier**

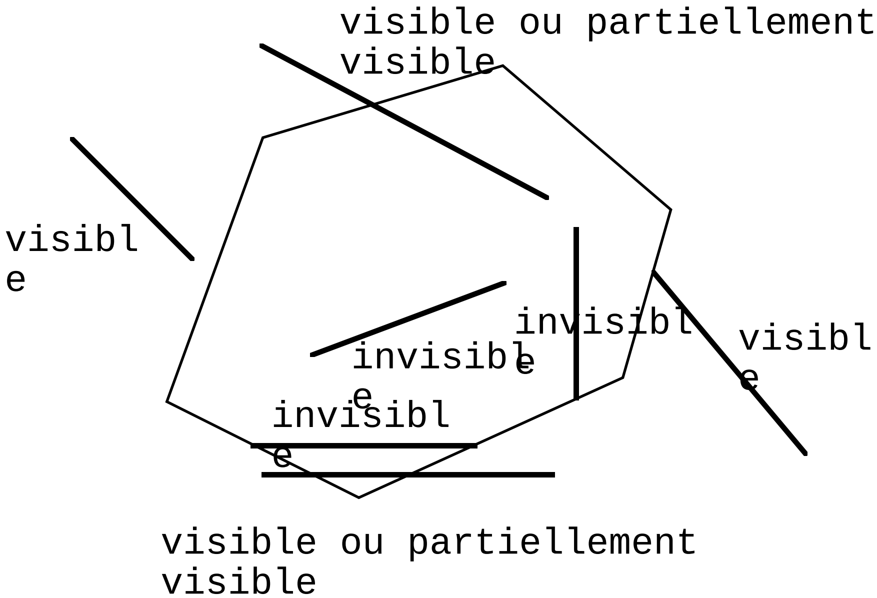
$$s(\mathbf{P}_i) = \begin{cases} 1 \ \textbf{si}\ Q(\mathbf{P}_i) > 0 \\ 0 \ \textbf{si}\ Q(\mathbf{P}_i) = 0 \\ -1 \ \textbf{si}\ Q(\mathbf{P}_i) < 0) \end{cases}$$

Alors AB est visible si

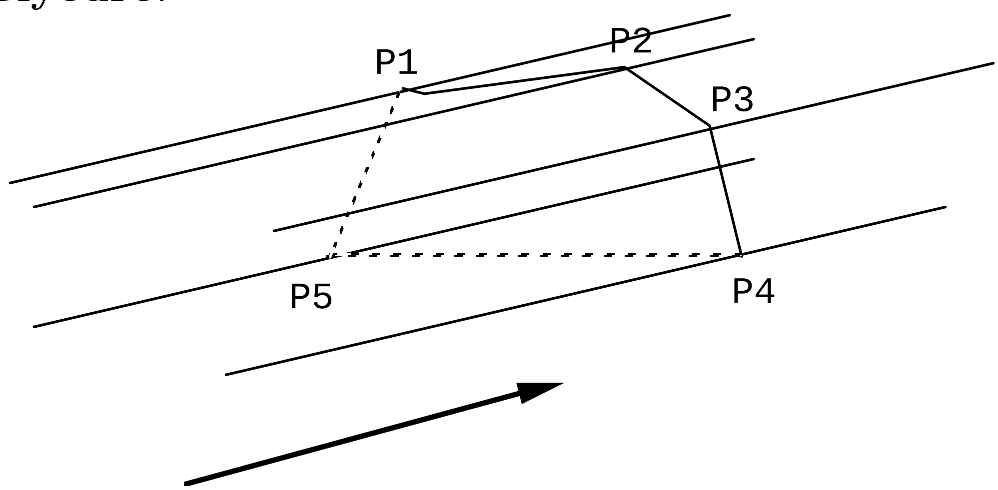
$$\sum_{1 \leq i \leq n} s(\mathbf{P}_i) = -n, -n+1, n-1 \ \textbf{ou}\ n$$

TEST 3

Cas restants à cette étape



Dans tous les cas le plan passant par AB intersecte le cylindre d'axe parallèle aux z, s'appuyant sur la face du polyèdre.



On donne

- $\mathbf{P}_i\mathbf{P}_{i+1}$ une arête
- $H_i(x, y, z) = 0$ l'équation du plan Π_i passant par $\mathbf{P}_i\mathbf{P}_{i+1}$ et parallèle à l'axe z
- $\mathbf{P}$ un sommet quelconque de la face, distinct de $\mathbf{P}_i$ et de $\mathbf{P}_{i+1}$
- A un point quelconque.

Le point A est

- ***extérieur*** au cylindre par rapport au plan Π_i si $H_i(\mathbf{A})H_i(\mathbf{P})$ 0 où
- ***intérieur*** au cylindre par rapport au plan Π_i si $H_i(\mathbf{A})H_i(\mathbf{P})$ 0
- ***sur*** le cylindre dans les autres cas

Est-il vrai que pour tout plan Π_j les deux points A et B sont intérieurs ou sur Π_j?

Si oui, le segment est invisible

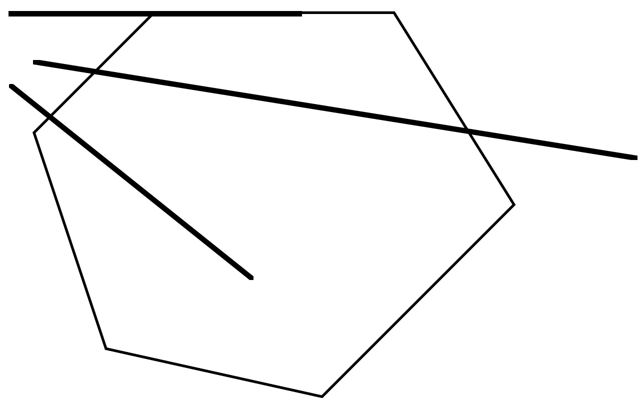
Existe-t'il un plan Π_j pour lequel l'un des points, disons A est extérieur et l'autre B est soit extérieur soit sur Π_j?

Si oui, le segment est visible

TEST 4

A cette étape, ou bien

- **ou bien il existe un** $i = 1, \dots, n$ **AB pour lequel le segment AB et** $\mathbf{P}_i\mathbf{P}_{i+1}$ **sont dans un même plan parallèle à l'axe des** z
- **ou bien le segment AB coupe le cylindre en 1 ou 2 points**



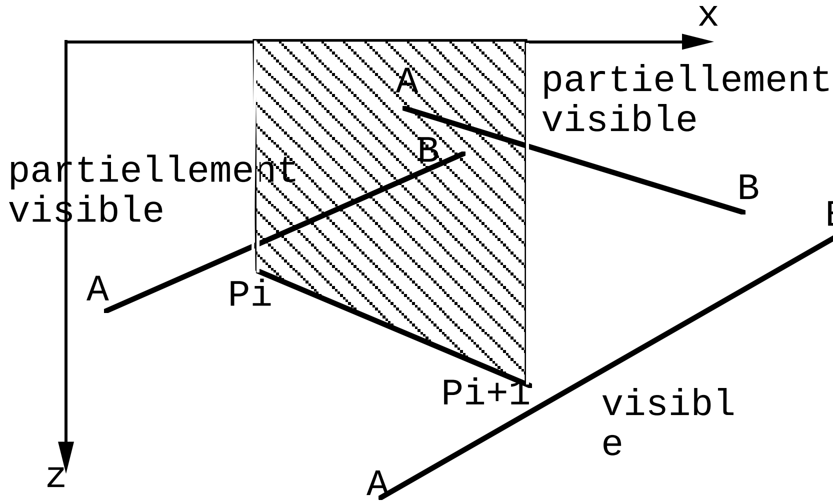
Equations paramétriques de la droite AB

$$\mathbf{R}(t) = \mathbf{A} + t\mathbf{AB}$$

Equations paramétriques de l'arête $\mathbf{P}_i\mathbf{P}_{i+1}$ pour $i = 1, \ldots, n$:

$$\mathbf{S}(s) = \mathbf{P}_i + s\mathbf{P}_i\mathbf{P}_{i+1}$$

Cas particulier: AB et $\mathbf{P}_i\mathbf{P}_{i+1}$ sont dans un même plan parallèle à l'axe des z:



Pour chaque $i = 1, \ldots, n$ faire

1. **Déterminer si le segment AB intersecte le plan $H_i(x, y, z) =$** 0

   - **valeur du paramètre t_i correspondant au point d'inters** **$\mathbf{R}(t_i) = \mathbf{R}_i$ de la droite support de AB avec le plan** $H_i(x, y, z) = 0$
   - **l'intersection est entre A et B si et seulement si** $0 \leq t_i \leq 1$

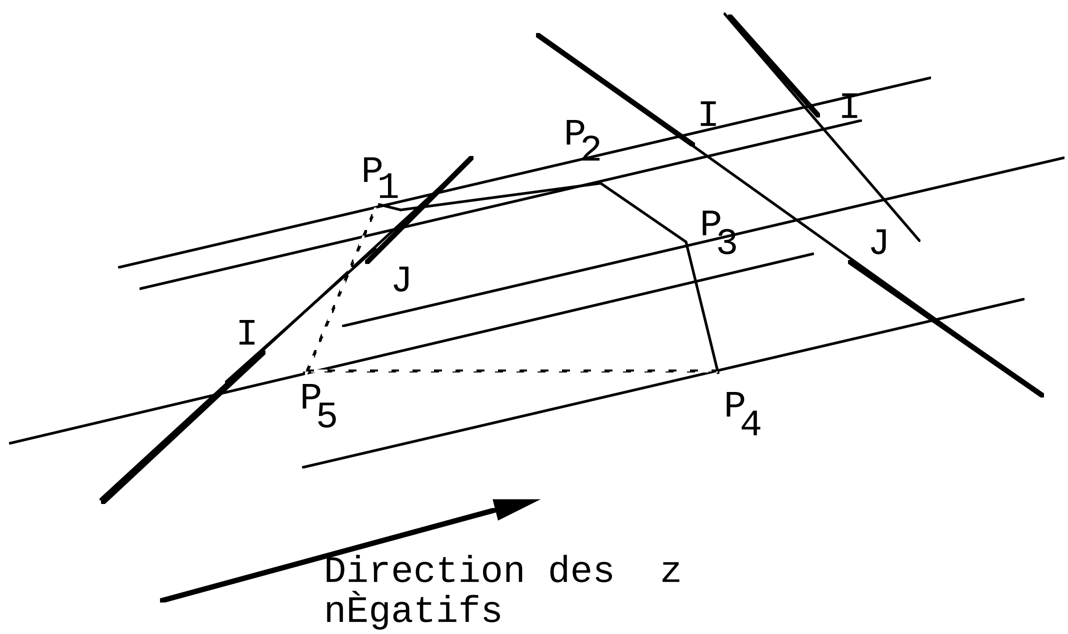
2. **Déterminer si l'arête $\mathbf{P}_i\mathbf{P}_{i+1}$ intersecte le plan passant par AB et parallèle à l'axe z**

   - **valeur du paramètre s_i correspondant au point d'inters** **$\mathbf{S}(t_i) = \mathbf{S}_i$ de l'arête $\mathbf{P}_i\mathbf{P}_{i+1}$ avec le plan passant par AB parallèle à l'axe z**
   - **l'intersection est entre $\mathbf{P}_i$ et $\mathbf{P}_{i+1}$ si et seulement si** $0 \leq s_i \leq 1$

3. **Remarque: les points $\mathbf{R}_i$ et $\mathbf{S}_i$ sont sur une même droite de direction z**

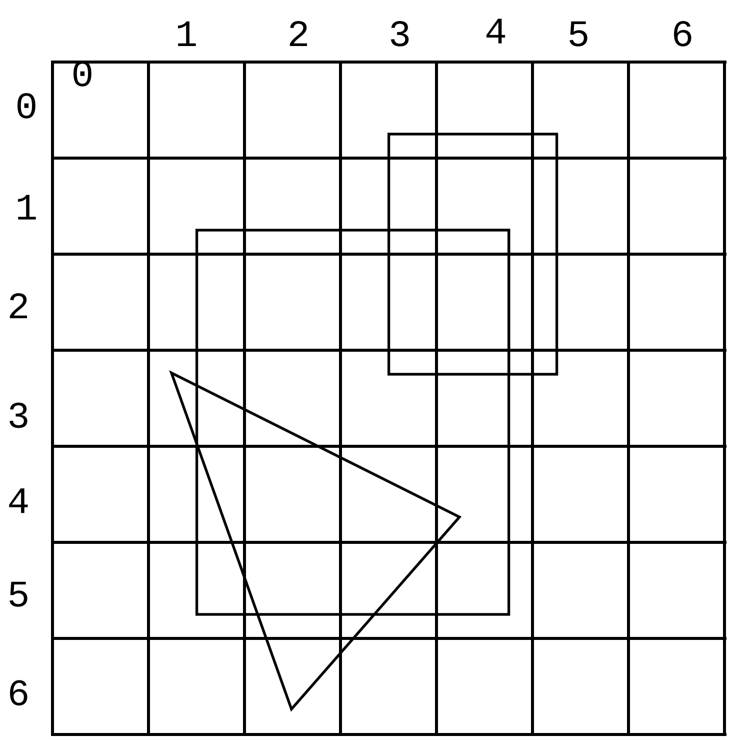
Trois cas

- **pour un certain** $i = 1, \ldots, n$ **on a** $0 \leq t_i \leq 1$ **et le point** $\mathbf{R}_i$ **est devant le plan de la face** $(F(\mathbf{R}_i) < 0)$**: le segment AB est lui-même devant la face, donc visible.**
- **sinon pour un unique** $1 \leq i \leq n$ **on a** $0 \leq t_i \leq 1$ **et le point** $\mathbf{R}_i$ **est derrière le plan de la face** $(F(\mathbf{R}_i) > 0)$**. Si A est invisible** $(H_i(\mathbf{A})H_i(\mathbf{P}) < 0$ **où** $\mathbf{P} \neq \mathbf{P}_i, \mathbf{P}_{i+1})$ **alors la partie visible correspond à** $t_i \leq t \leq 1$
- **sinon (pour 2 valeurs** $1 \leq i \neq j \leq n$ **on a** $0 \leq t_i < t_j \leq 1$**) et la partie visible est constituée des deux segments** $\mathbf{AR}_i$ **et** $\mathbf{R}_j\mathbf{B}$



ACCROÎTRE LA PERFORMANCE

- **But: éviter de comparer une arête à des faces qui de toute évidence ne la cachent pas**
- **Moyens: utiliser un quadrillage de la fenêtre de l'espace objet et réaliser l'algorithme des lignes cachées dans chaque pavé de ce quadrillage**



MISE EN ŒUVRE

Limiter la partie intéressant de la scène au rectangle

$$(xmin, ymin, xmax, ymax)$$

et considérer N^2 **pavés de largeur et de hauteur**

$$\Delta_x = \frac{xmax - xmin}{N}, \Delta_y = \frac{ymax - ymin}{N}$$

On applique l'algorithme des lignes cachées seulement à l'intérieur de chaque pavé qu'elle intersecte.

On utilise comme struture de données un tableau Objet$[0 \ldots N - 1, 0 \ldots N - 1]$ **dont l'entrée** (i, j) **consiste en**

- **l'ensemble des faces dont la projection sur la fenêtre intersecte le pavé de coordonnées** (i, j)
- **la face la plus proche qui recouvre le pavé**

Calcul effectif du tableau Objet

```
pour tout polygone faire début
   pour toute droite verticale d'abscisse i intersectant
                        le polygone faire début
      calculer les ordonnées j1 et j2 des deux points
                        d'intersection
      pour tout j1 ≤ j≤ j2 faire
         ranger le polygone dans Obj[i,j]
   fin
fin
```

En fait on détermine les deux points d'intersection en appliquant une procédure semblable à celle de Bresenham. Pour chaque polygone on fait

```
soit S_{1,g} le sommet d'abscisse minimale le plus bas
soit S_{1,d} le sommet d'abscisse maximale le plus bas
soit S_{2,g} le sommet d'abscisse minimale le plus haut
soit S_{2,d} le sommet d'abscisse maximale le plus bas

parcourir la "moitié" inférieure S_{1,g}S_{1,d} du polygone
      dans le sens inverse des aiguilles d'une montre et
      pour chaque [i,j], coordonnées du point courant faire
                      Bas[i]:=j

parcourir la "moitié" supérieure S_{2,g}S_{2,d} du polygone
      dans le sens des aiguilles d'une montre et
      pour chaque [i,j], coordonnées du point courant faire
                      Haut[i]:=j

pour tout Bas[i]≤ k ≤ Haut[i] faire
   ranger le polygone dans Objet[i,k]
```