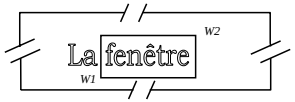
5. Le fenêtrage

5.1 Introduction

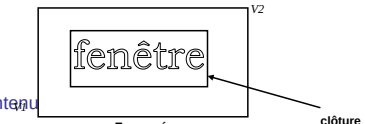
L'affichage d 'un modèle implique la mise en correspondance des coordonnées des points et des lignes du modèle avec les coordonnées appropriées du dispositif où l 'image doit être visualisée.

- Il faut définir
  - 2 espaces de coordonnées (utilisateur et écran)
  - la fenêtre
  - la clôture



Espace utilisateur

- Fenêtre : Domaine à visualiser (utilisateur)
- Clôture : zone où sera projeté le contenu de la fenêtre (écran)

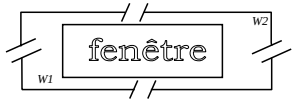


Espace écran

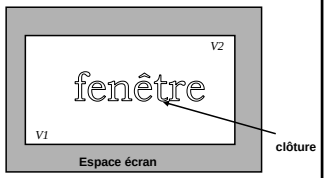
1

- Remarques :
  - la fenêtre et la clôture sont homothétiques
  - le découpage des parties n'appartenant pas à la fenêtre s'appelle le fenêtrage (clipping)
- Transformation des coord. utilisateur en coord. écran

  soit W une fenêtre définie par les 2 points $W1$ et $W2$ et V une clôture définie par $V1$ et $V2$



Espace utilisateur



Espace écran

$$W_1=\begin{pmatrix}x_{w_1}\\ y_{w_1}\end{pmatrix}\quad W_2=\begin{pmatrix}x_{w_2}\\ y_{w_2}\end{pmatrix}\quad V_1=\begin{pmatrix}x_{v_1}\\ y_{v_1}\end{pmatrix}\quad V_2=\begin{pmatrix}x_{v_2}\\ y_{v_2}\end{pmatrix}$$

- Le passage fenêtre-clôture transforme le point R de W en un point E de V

$$x_e=\frac{x_r-x_{w_1}}{x_{w_2}-x_{w_1}}(x_{v_2}-x_{v_1})+x_{v_1}$$

$$y_e=\frac{y_r-y_{w_1}}{y_{w_2}-y_{w_1}}(y_{v_2}-y_{v_1})+y_{v_1}$$

3

OpenGL

- **Transformation de cadrage**

  glViewport();

  Cette fonction permet de déterminer la zone où sera affichée l'image.

- **Coordonnées normalisées**

  OpenGL calcule de façon transparente les coordonnées normalisées (division par la 4ieme composante des coordonnées homogènes)

4

5.2 Le fenêtrage 2D

- fenêtrage rectangulaire : les cotés de la fenêtre sont parallèles aux axes des coordonnées

- fenêtrage d 'un point

  le point $P(x,y)$ est *visible* si $x \in [x_{w1}, x_{w2}]$ et $y \in [y_{w1}, y_{w2}]$

- fenêtrage d 'un segment de droite :

  un segment (P_1, P_2) est invisible si

$$x_1, x_2 \geq x_{w2} \quad \text{ou} \quad x_1, x_2 \leq x_{w1}$$
$$\text{ou} \quad y_1, y_2 \geq y_{w2} \quad \text{ou} \quad y_1, y_2 \leq y_{w1}$$

visible si

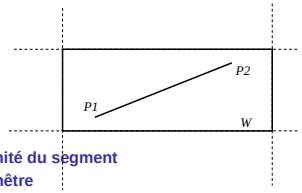
$$P_1 \text{ et } P_2 \in W$$

5

Fenêtrage de segments

Algorithme de Cohen-Sutherland : fenêtrage rectangulaire de segment

Principe On subdivise le segment en plusieurs petits segments ne pouvant appartenir qu'aux catégories *visible* ou *invisible*.



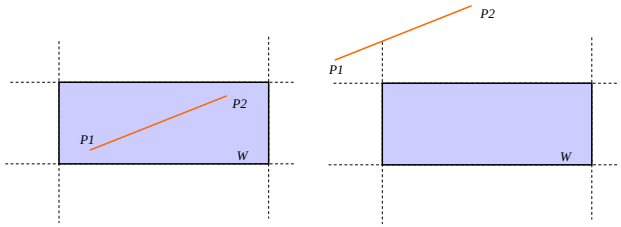
procède en 3 étapes:

- **1. Affecter un code à 4 bits à chaque extrémité du segment**
  bit 1 = 1 : l'extrémité est au dessus de la fenêtre
  bit 2 = 1 : l'extrémité est en dessous de la fenêtre
  bit 3 = 1 : l'extrémité est à droite de la fenêtre
  bit 4 = 1 : l'extrémité est à gauche de la fenêtre

6

Algorithme de Cohen-Sutherland

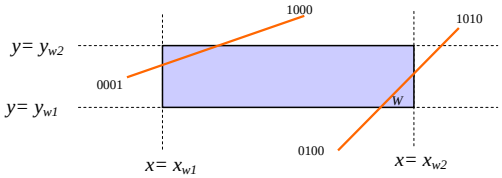
- **2. On classe le segment**
  si $code1 = 0000$ ***et*** $code2 = 0000$ **alors le segment est visible totalement**
  sinon si $(code1$ ***et*** $code2) \neq 0000$ **alors il est invisible**
  sinon le segment est partiellement visible ⟹ fenêtré (étape 3)



7

Algorithme de Cohen-Sutherland

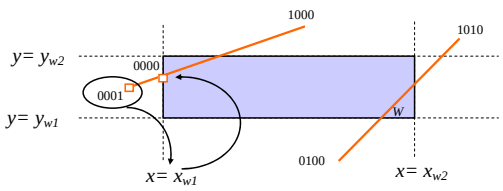
- **3. fenêtrage**
  Si $bit1 = 1$, **intersection possible avec** $y = y_{w2}$
  Si $bit2 = 1$, **intersection possible avec** $y = y_{w1}$
  Si $bit3 = 1$, **intersection possible avec** $x = x_{w2}$
  Si $bit4 = 1$, **intersection possible avec** $x = x_{w1}$



8

Algorithme de Cohen-Sutherland

- **4. On remplace l'extrémité par le point d'intersection et on recommence avec le nouveau segment**



Fenêtrage généralisé de segments en 2D par des contours convexes

- Définition
  On dit qu'un polynôme est convexe si 2 points quelconques situés à l'intérieur du polynôme déterminent un segment lui même toujours situé tout entier dans le polynôme.

- Algorithme de Cyrus-Beck

  Modélisation paramétrique d 'un segment de droite :

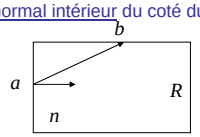
$$x(t) = x_1 + (x_2 - x_1)t$$
$$y(t) = y_1 + (y_2 - y_1)t$$

  Dans un système de coordonnées cartésiennes en 2D

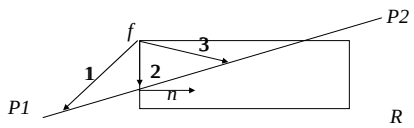
$$\forall b \in R, \quad n \cdot (b - a) \geq 0$$

  On utilise n le vecteur normal intérieur du coté du contour convexe R

$$\forall b \in R, \quad n \cdot (b - a) \geq 0$$



Algorithme de Cyrus-Beck



cas 1 : $n \cdot [P(t) - f] < 0$ le point est à l'extérieur de la fenêtre
cas 2 : $n \cdot [P(t) - f] = 0$ le point est sur la fenêtre
cas 3 : $n \cdot [P(t) - f] > 0$ le point est à l'intérieur de la fenêtre

- **Calcul de l 'intersection entre le bord et le segment.**

$$n \cdot [P(t) - f] = 0 \quad \Leftrightarrow \quad t = -\frac{w \cdot n}{D \cdot n} \qquad D \neq 0$$

avec $w = P_1 - f$
et $D = P_2 - P_1$ (direction du segment)

si $D \cdot n > 0 \Rightarrow t$ est une borne inférieure
si $D \cdot n < 0 \Rightarrow t$ est une borne supérieure

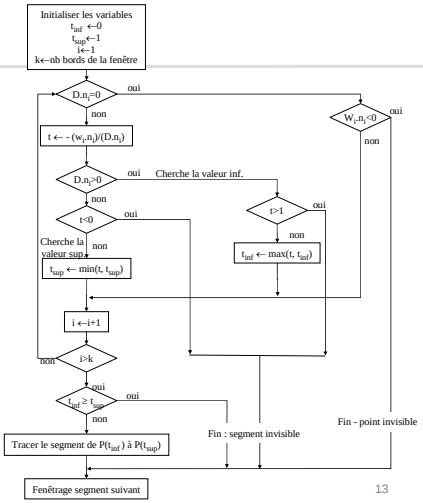
Algorithme de Cyrus-Beck

Pour un même segment, il faut appliquer la méthode pour tous les bords de la fenêtre.

Vérifier que les intersections entre les valeurs possibles de t ne sont pas nulles

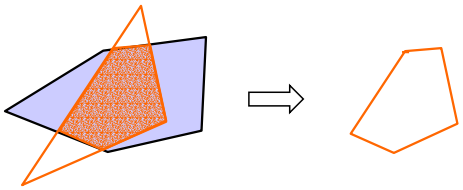
si borne inférieure > borne supérieure
segment invisible
sinon
segment visible pour t ∈ [borne inf, borne sup]

Algorithme de Cyrus-Beck



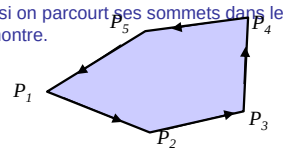
13

Fenêtrage de polygones par des polygones convexes



Définitions

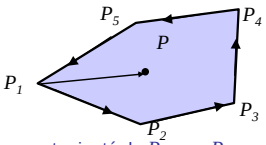
- On dit qu'un polygone de sommets P_1, P_2,... P_n (et de cotés $P_{i-1}P_i$ et P_nP_1) est orienté positivement (+) si on parcourt ses sommets dans le sens inverse des aiguilles d 'une montre.



14

Fenêtrage de polygones par des polygones convexes

Définitions



- Soit P_1 et P_2 les extrémités d'un segment orienté de P_1 vers P_2, un point P sera à gauche du segment si

$$\overrightarrow{P_1P_2} \wedge \overrightarrow{P_1P} > 0$$

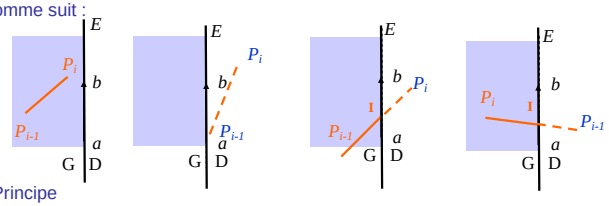
$$(x_2-x_1)(y-y_1) - (y_2-y_1)(x-x_1) > 0$$

- Si P est à gauche de chacun des cotés d'un polygone orienté positivement, il est à l'intérieur du polygone.

15

Algorithme de Sutherland-Hodgman

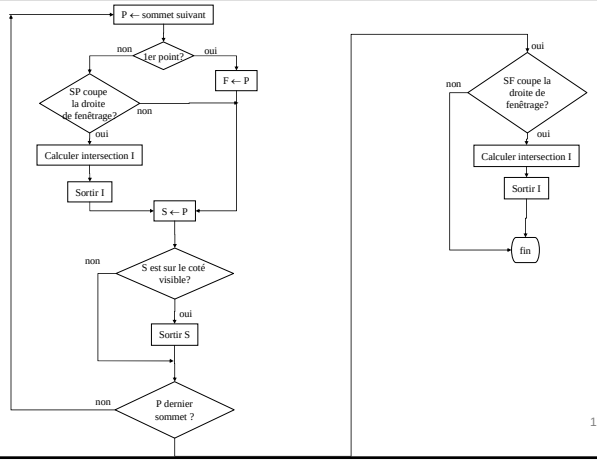
- Soient P_1, P_2,... P_n la liste des sommets de polygone à fenêtrer et $E(a,b)$ un coté du polygone (+) de fenêtrage. Nous fenêtrons chaque côté du polygone par E pour former un nouveau polygone dont les sommets sont déterminés comme suit :



- Principe

1) si P_{i-1} et P_i sont à gauche de E, on place le sommet P_i dans la table de sortie
2) si P_{i-1} et P_i sont à droite de E, la table de sortie ne change pas
3) si seul P_{i-1} est à gauche de E, on place le point I dans la table de sortie
4) si seul P_i est à gauche de E, on place les points I et P_i dans la table de sortie

Algorithme de Sutherland-Hodgman



Algorithme de Sutherland-Hodgman

- Diagrammes donnés pour un bord (+) de la fenêtre
- A répéter pour tous les bords de la fenêtre.
- Chaque table de sortie devient la table d'entrée du bord suivant

5.3 Le fenêtrage 3D

En 2D : fenêtrage (le plus souvent rectangulaire)
En 3D : fenêtrage par un volume de vision

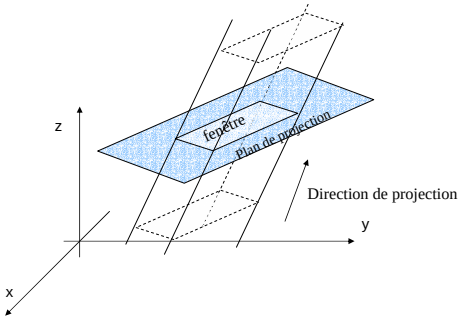
- Intérêt du découpage dans l'espace : on élimine les parties de la scène qui ne seront pas visualisées dans l'image avant de faire tous les calculs
- Exemples :
  - suivant la position de l 'observateur, on ne peut pas visualiser toute la scène car le champ de vision est limité.
  - les objets qui sont très éloignés seront très petits, voire invisibles (perspective)

Le fenêtrage 3D

- Définition du volume de visualisation
  - fenêtre rectangulaire appartenant au plan de projection
  - projection parallèle : parallélépipède dont les arêtes sont parallèles à la direction de projection
  - projection perspective : pyramide infinie dont le sommet est le centre de projection

Formes du volume de fenêtrage

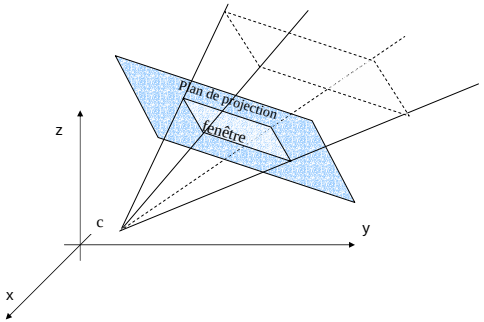
- parallélépipédique



21

Formes du volume de fenêtrage

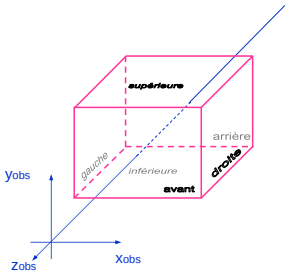
- pyramidal

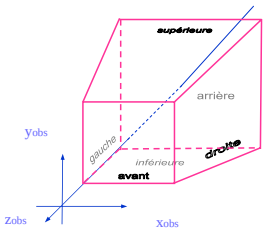


22

Volumes de fenêtrage

- Généralement:
  - Volumes de fenêtrage finis
  - Donnés dans le repère observateur





23

Algorithme de Cohen-Sutherland

- **1. Affecter un code à 6 bits à chaque extrémité du segment**
  bit 1 = 1 : l'extrémité est à gauche du volume
  bit 2 = 1 : l'extrémité est à droite du volume
  bit 3 = 1 : l'extrémité est en dessous du volume
  bit 4 = 1 : l'extrémité est au dessus du volume
  bit 5 = 1 : l'extrémité est devant le volume
  bit 6 = 1 : l'extrémité est derrière le volume

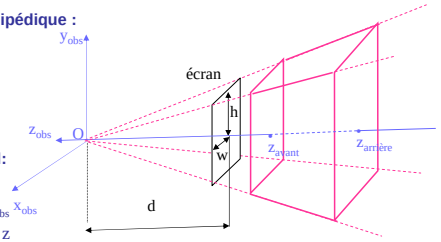
- **2. Classer le segment**
  si $code1 = 000000$ ***et*** $code2 = 000000$ **le segment est visible totalement**
  sinon si $(code1$ ***et*** $code2) \neq 000000$ **il est invisible**
  sinon le segment est (potentiellement) partiellement visible

- **3. Examiner les segments restants :**
  - **calcul de l'intersection du segment avec le plan du volume concerné**
  - **calculer son code d'extrémité**
  - **aller en 2 avec le nouveau segment.**

24

Calcul des intersections : équations des plans

- **Pour un volume parallélépipédique : équations des plans triviales**



- **Pour un volume pyramidal:**

projections sur le plan $Ox_{obs}z_{obs}$

plan de gauche : $x = -w/d\ z$

plan de droite : $x = w/d\ z$

projections sur le plan $Oz_{obs}y_{obs}$

plan supérieur: $y = h/d\ z$

plan inférieur : $y = -h/d\ z$

valeurs de Z

plan avant: $z = z_{avant}$

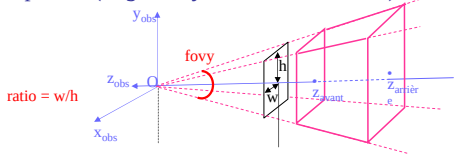
plan arrière : $z = z_{arrière}$

25

Les volumes de fenêtrage avec OpenGL

- **Projection perspective**

glFrustum(left, right,bottom,top,zNear,zFar);
gluPerspective(angleFovy, ratio, zNear, zFar);



- **Projection orthogonale**

glOrtho (left, right, bottom,top,zNear,zFar)

26