Chapitre 2 : fenêtrage, algorithme de Sutherland et Hodgman

1 Présentation et analyse

1.1 Exemples

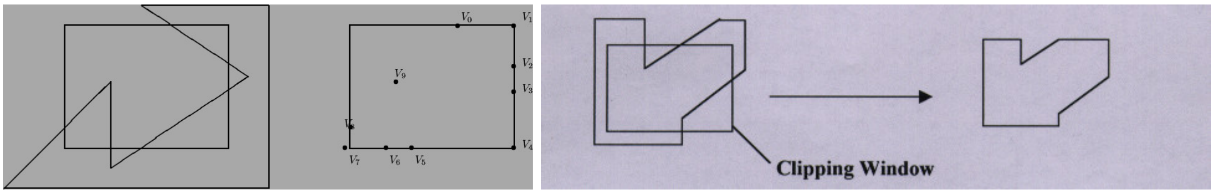

FIGURE 1 – Sommets ordonnés du Clipping et clipping window

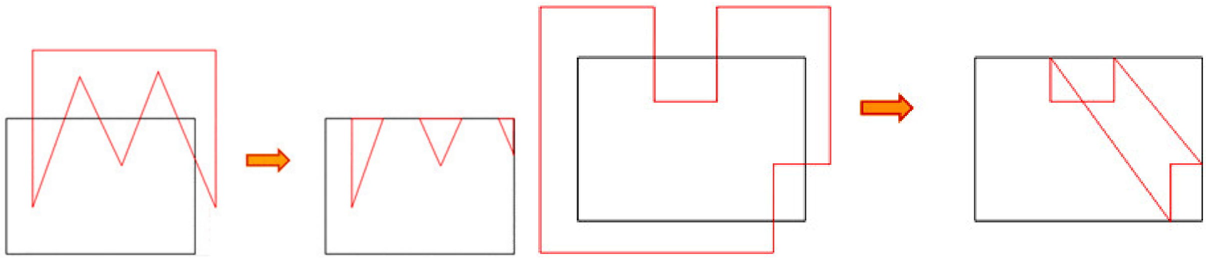

FIGURE 2 – Clipping correct et incorrect

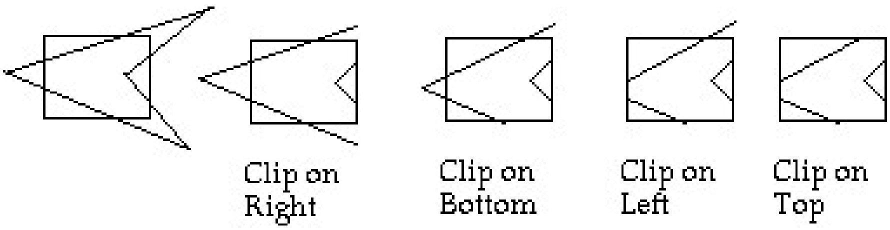

FIGURE 3 – Clipping bords fenêtre

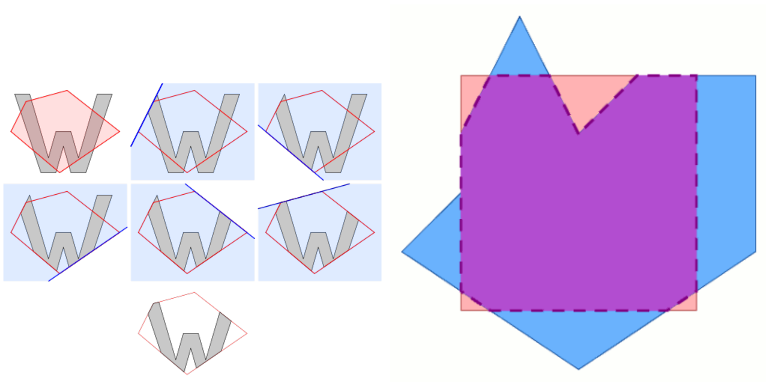

FIGURE 4 – Clipping et remplissage

On pose :

- Ligne polygonale d'entrée : PL à $N1$ sommets
  $P_1, ..., P_{N1}$: sommets de PL
  $[P_1 P_2], ..., [P_{N1} P_1]$: segments de PL
- Ligne polygonale de sortie (découpé) : PS, initialement vide, de $N2$ sommets
- Polygone fenêtre (window) convexe : PW à $N3$ sommets
- Fenêtrage de PL avec les bords ordonnés de la fenêtre :
  bord 1 :
    Construction de PS : enregistrement des intersections et des points visibles
    Découpage de PL, $PL \longleftarrow PS$
  ...
  bord $N3$
    Construction d'un nouveau PS : enregistrement des intersections et des points visibles
    Découpage de PL, $PL \longleftarrow PS$

1.2 Positionnement segment-fenêtre

On suppose que la position du point S a été traitée dans l'itération précédente, recherchons la position de P_j.

Quatre cas possibles sont à analyser :

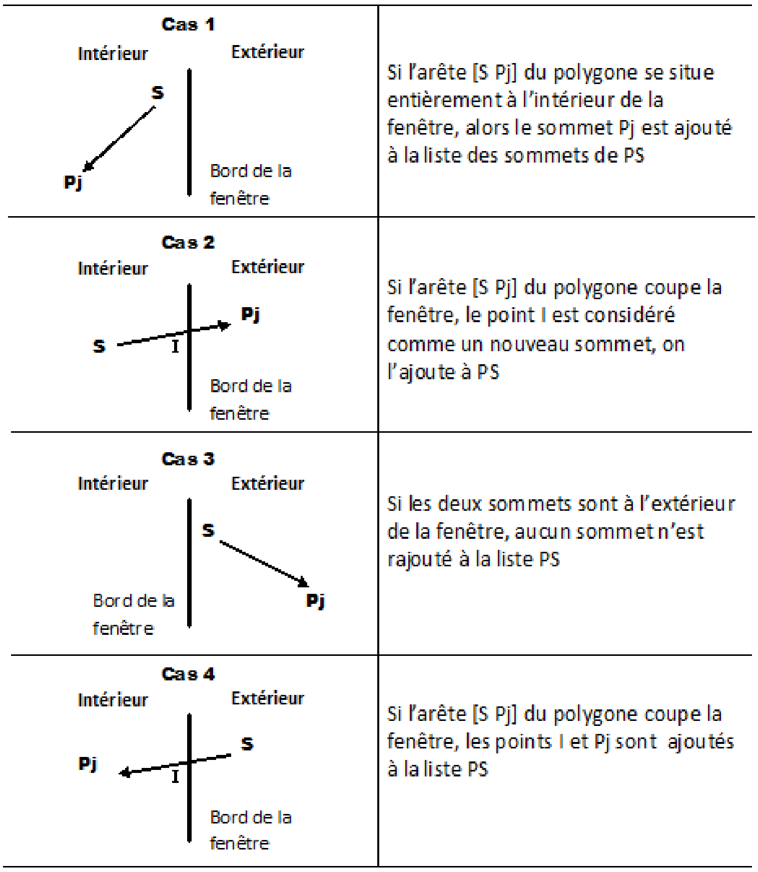

FIGURE 5 – Positionnement segment-fenêtre

Remarques :

- En ce qui concerne P_1, il suffit de vérifier sa visibilité par rapport à la fenêtre :
  1) S'il est visible, il est ajouté dans PS
  2) Sinon, on le stocke juste comme point départ
- Le côté final $[P_{N1}P_1]$ doit subir un traitement spécial consistant à stocker P_1 dans F avant le test, puis à considérer comme segment final $P_{N1}F$ pour effectuer le dernier test.

2 Actions de l'algorithme

Il y a deux actions à étudier avant l'écriture de l'algorithme complet :

1) Détermination de la visibilité d'un point
2) Détermination de l'intersection d'un segment du polygone avec un bord de la fenêtre

2.1 Visibilité d'un point

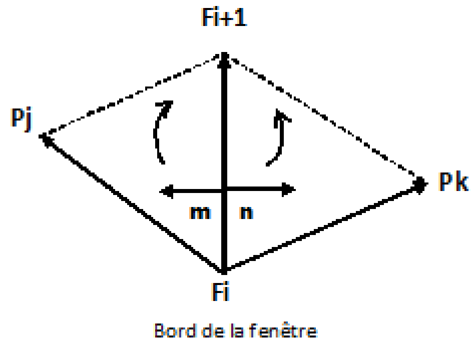

FIGURE 6 – Positionnement d'un point par rapport à un segment orienté

Deux méthodes sont possibles :

- On pose $\overrightarrow{n}$ normale intérieure et $\overrightarrow{m}$ normale extérieure de $[F_iF_{i+1}]$
Le produit scalaire des vecteurs donne :
  Si $\overrightarrow{n} \cdot \overrightarrow{F_iP_k} > 0$, P_k à droite
  Si $\overrightarrow{n} \cdot \overrightarrow{F_iP_j} < 0$, P_j à gauche
  Si $\overrightarrow{n} \cdot \overrightarrow{F_iP_l} = 0$, P_l sur le segment

- On détermine l'orientation des triangles :
  1) $|\overrightarrow{F_iP_k}\ \overrightarrow{F_iF_{i+1}}| > 0$, avec $\overrightarrow{F_iP_k}(x, y)$ et $\overrightarrow{F_iF_{i+1}}(x', y')$
  $\iff xy' - yx' > 0$
  $\iff F_iP_kF_{i+1}$ orienté positivement
  $\iff P_k$ à droite
  2) $|\overrightarrow{F_iP_j}\ \overrightarrow{F_iF_{i+1}}| < 0$, avec $\overrightarrow{F_iP_j}(x'', y'')$ et $\overrightarrow{F_iF_{i+1}}(x', y')$
  $\iff x''y' - y''x' < 0$
  $\iff F_iP_jF_{i+1}$ orienté négativement
  $\iff P_j$ à gauche

Un segment est :
- visible si ses deux extrémités sont visibles
- invisible si ses deux extrémités sont invisibles
- partiellement visible si l'une des extrémités est visible et l'autre invisible. On calcule alors le point d'intersection.

2.2 Intersection de deux droites paramétriques dans le plan

On s'intéresse à l'intersection de deux droites paramétriques du plan (P_1P_2) et (P_3P_4).
On pose $P_1(x_1,y_1)$, $P_2(x_2,y_2)$, $P_3(x_3,y_3)$ et $P_4(x_4,y_4)$. On a :
$(P_1P_2): \ P(t) = P_1 + (P_2 - P_1)t,\ t \in \mathbb{R}$ et $(P_3P_4): \ Q(s) = P_3 + (P_4 - P_3)s,\ s \in \mathbb{R}$.
Au point d'intersection, on a $P(t) = Q(s)$.

Soient $t, s \in [0,1]$,
$$\begin{aligned} & P(t) = Q(s) \\ \iff & P_1 + (P_2 - P_1)t = P_3 + (P_4 - P_3)s \\ \iff & (P_2 - P_1)t + (P_3 - P_4)s = P_3 - P_1 \\ \iff & \underbrace{\begin{pmatrix} x_2 - x_1 & x_3 - x_4 \\ y_2 - y_1 & y_3 - y_4 \end{pmatrix}}_{\Lambda} \underbrace{\begin{pmatrix} t \\ s \end{pmatrix}}_{X} = \underbrace{\begin{pmatrix} x_3 - x_1 \\ y_3 - y_1 \end{pmatrix}}_{b} \\ \iff & X = \Lambda^{-1}b, \text{ si } \Lambda \text{ est inversible, c'est à dire : } |\Lambda| \neq 0. \end{aligned}$$

Remarques :

. $|\Lambda| = 0$ signifie que les deux droites sont parallèles ou confondues
. Si $t, s \in [0,1]$, les segments $[P_1P_2]$ et $[P_3P_4]$ sont sécants
. Si $t \notin [0,1]$ ou $s \notin [0,1]$, l'intersection est sur le prolongement d'au moins un des segments

3 Algorithme de Sutherland-Hodgman

Notation : S point courant du polygone, P_j les autres points du polygone, F_i les points de la fenêtre.

On se munit de trois fonctions :

1. **coupe**(S, P_j, F_i, F_{i+1}) : retournant un booléen suivant l'intersection possible entre le côté $[SP_j]$ du polygone et le bord prolongé (une droite) (F_iF_{i+1}) de la fenêtre
2. **intersection**(S, P_j, F_i, F_{i+1}) : retournant le point d'intersection $[SP_j] \cap (F_iF_{i+1})$
3. **visible**(S, F_i, F_{i+1}) : retournant un booléen si S est visible par rapport à (F_iF_{i+1})

Algorithme de Sutherland-Hodgman

/* Découpage d'un polygone d'entrée $PL = \{P_1, ..., P_{N1}\}$ par une fenêtre polygonale convexe $PW = \{F_1, ..., F_{N3}\}$ */

PF :
  PL : liste de N1 sommets (Entrée)
  PW : liste de N3 sommets, avec $F_1 = F_{N3}$ (Entrée)
VI :
  i, j : entiers
  N2 : entier /* nombre de points de la liste PS */
  S, F, I : points
  PS : liste de points du polygone de sortie (PL découpé) */

```
Début
    /* Pour chaque point de la window PW */
    Pour i variant de 1 à (N3 − 1) Faire
        N2⟵0
        PS ⟵ vide
        /* Pour chaque point du polygone PL */
        Pour j variant de 1 à N1 Faire
            Si (j=1) Alors
                F ⟵ P_j /* Sauver le 1er = dernier sommet */
            Sinon
                Si coupe(S, P_j, F_i, F_{i+1}) Alors
                    I ⟵ intersection(S, P_j, F_i, F_{i+1})
                    Charger(I, PS)
                    N2++
                FinSi
            FinSi
            S ⟵ P_j
            Si visible(S, F_i, F_{i+1}) Alors
                Charger(S, PS)
                N2++
            FinSi
        FinPour
        Si (N2 > 0) Alors
            /* Traitement du dernier côté de PL */
            Si coupe(S, F, F_i, F_{i+1}) Alors
                I ⟵ intersection(S, F, F_i, F_{i+1})
                Charger(I, PS)
                N2++
            FinSi
            /* Découpage pour chacun des sous polygones */
            PL ⟵ PS
            N1 ⟵N2
        FinSi
    FinPour
Fin
```