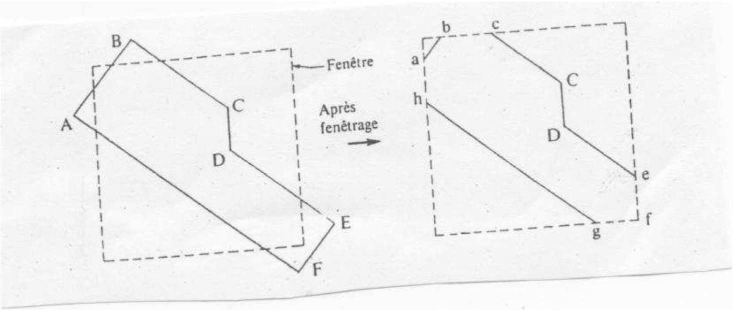
FENÊTRAGE DE POLYGONES

Exemples

Lorsque on fenêtre un polygone en le considérant comme un ensemble de segments, il se trouve transformé en une ou plusieurs lignes brisées.

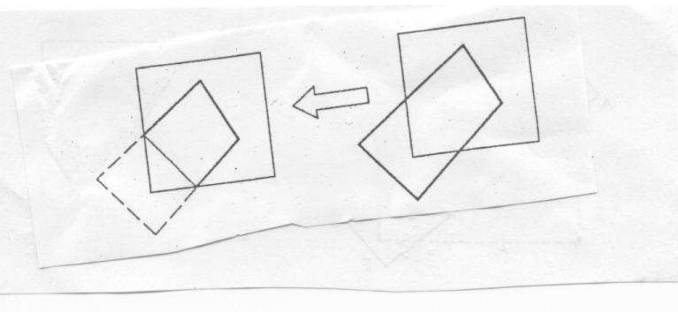
Il faut tracé les segments bc, ef, fg et ha

Par contre, le tracé des segements ef et fg est difficile.



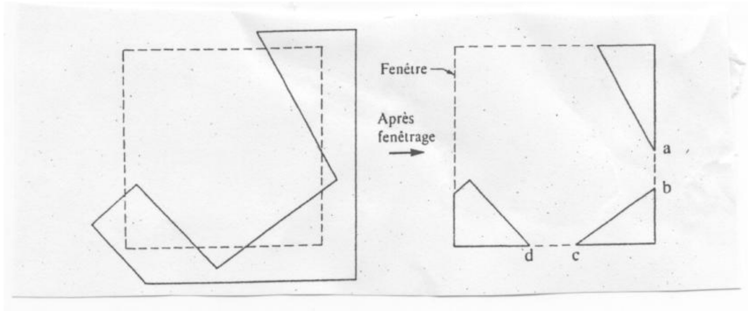
Lorsque ces polygones sont considérés comme des régions pleines, il devient nécessaire de leur conserver des contours fermés.

Si on utilise un algorithme de remplissage du polygone après découpage il nous faut joindre les intersections ? afin d'obtenir un polygone découpé (fermé).

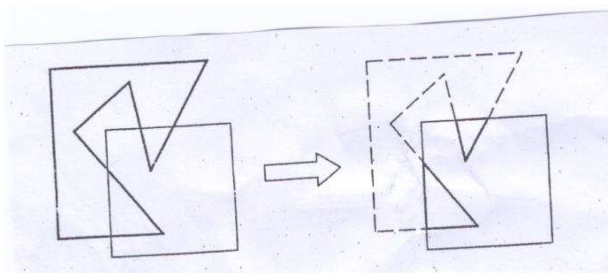


Fenêtrage d'un polygone génère plusieurs petits polygones disjoints.

Problème : les segments AB et CD apparaissent dans la description du polygone fenêtré.



Fenêtrage d'un polygone concave peut produire plusieurs polygones disjoints.



Algorithme de Sutherland et Hodgman

Principe

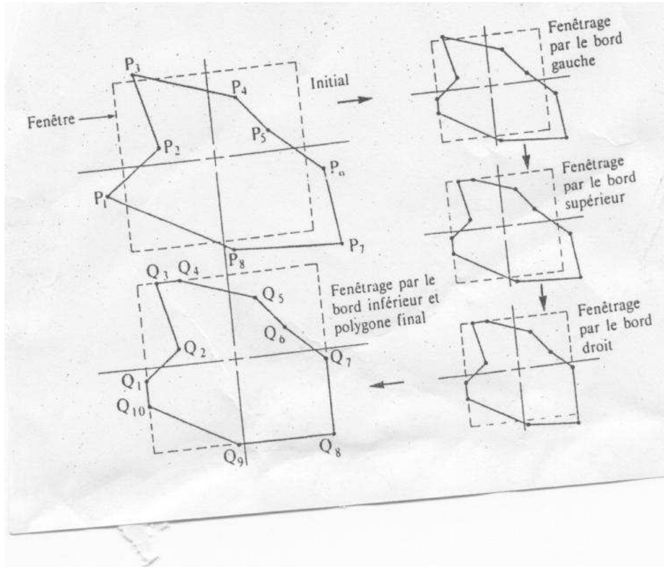
Sutherland et Hodgman coupent le polygone successivement par chacun des côtés de la fenêtre en réapliquant récursivement le procédé pour chacun des sous polygones obtenus :

Les sommets à l'intérieur de la fenêtre restent non modifiés

Les sommets à l'extérieur sont remplacés par les points d'intersection avec la fenêtre.

Travail côté par côté de la fenêtre : traiter tout le polygone à découper.

Exemple



Polygone Initial :

P1, …, P8 : Sommets du polygone

P1P2, , P8P1 : Liste des côtés du polygone

Fenêtrage du polygone par le bord gauche de la fenêtre

→ Polygone Intermédiaire = PI

Fenêtrage du PI par le bord supérieur de la fenêtre

→ Polygone Intermédiaire = PI

On répète le processus jusqu'à ce que le polygone ait été fenêtré par tous les bords de la fenêtre.

Remarques

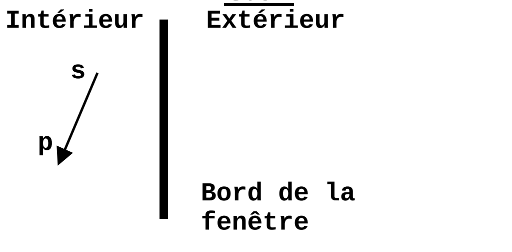
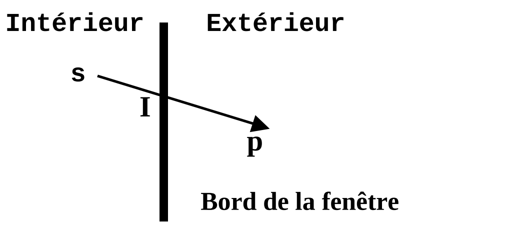
L’algorithme peut fenêtré n’importe quel polygone convexe ou concave, par n’importe quelle fenêtre polygonale convexe .

l’ordre suivant lequel le polygone est fenêtré par les différents côtés de la fenêtre est sans importance.

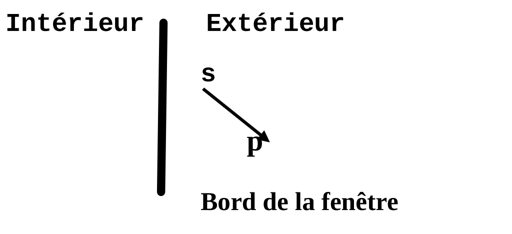
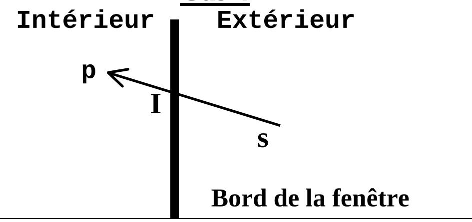
L’algorithme retourne une liste de sommets du polygone qui sont tous visible sur le côté d’un plan de fenêtrage.

A chaque étape : On examine la relation entre les sommets successifs du polygone et un côté de la fenêtre

4 cas possibles sont à analyser :

Cas 1  	Si l’arête [s, p] du polygone se situe entièrement à l’intérieur de la fenêtre, alors le sommet p est ajouté à la liste de sommets.
Cas 2  	L’arête [s, p] coupe la fenêtre → le point d’intersection I est considéré comme un nouveau sommet.

Remarque : Le sommet s ait été traité dans l’itération précédente.

Cas 3  	Les deux sommets sont à l’extérieur de la fenêtre : → Aucun sommet n’est donc ajouté.
Cas 4  	L’arête [s, p] coupe la fenêtre → le point d’intersection I et p sont ajoutés à la liste de sommets.

→ On rajoute : zéro, un ou deux sommets au polygone fenêtré.

Remarques

- En ce qui concerne, le premier point S du polygone, il suffit de vérifier la visibilité du point :

S'il est visible, le point S est ajouté dans la liste du polygone intérmédiaire

S'il n'est pas, on la stocke comme point de départ.

- Le côté final PnP1 doit subir un traitement spécial consistant à stocker le premier point en tant que F. le côté final devient alors PnF et peut être traité comme n'importe quel autre côté .

Actions de l'algorithme

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

Détermination de la visibilité d'un point

Détermination de l'intersection d'un côté du polygone avec le plan de fenêtrage.

Visibilité d'un point

Cela revient à déterminer sur quelle face du plan de fenêtrage il se trouve.

1^er^ méthode

Test du signe du produit scalaire du vecteur normal par le vecteur mené d'un point du segment ou du plan au point dont il s'agit.

2^ème^ méthode

Substitution des coordonnées du point dans l'équation du segment ou du plan.

3^ème^ méthode

utilisation du produit vectoriel.

Rappel : Produit vectoriel

Soit P1(x1, y1, z1) et P2(x2, y2, z2) deux poins de l'epace.

Le produit vectoriel entre P1 et P2 est : $P1 \otimes P2 = \begin{pmatrix} y1z2 - z1y2 \\ z1x2 - x1z2 \\ x1y2 - y1x2 \end{pmatrix}$

Simplification

Si P1 et P2 appartienne au plan (O, X, Y) alors z1=z2=0.

Dans ce cas $P1 \otimes P2 = \begin{pmatrix} 0 \\ 0 \\ x1y2 - y1x2 \end{pmatrix}$

Le signe de x1y2-y1x2 nous dit :

P2 est gauche

Où P2 est à droite

3ème Méthode (suite)

Une autre technique consiste à examiner le signe de la composante z du produit vectoriel du deux vecteurs du plan.

Si P1 et P2 sont deux points du plan de fenêtrage, et P3 le point en cours de traitement

Alors P1, P2 et P3 définissent un plan

Si l'on considère ce plan comme le plan (OXY)

Alors le produit vectoriel P1P2⊗P1P3 n'a qu'une composante non nulle z donnée par :

$$P1P2 \otimes P1P3 = \begin{pmatrix} x2 - x1 \\ y2 - y1 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} x3 - x1 \\ y3 - y1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ (x2 - x1)(y3 - y1) - (y2 - y1)(x3 - x1) \end{pmatrix}$$

si z > 0 → P3 est à droite du segment P1P2

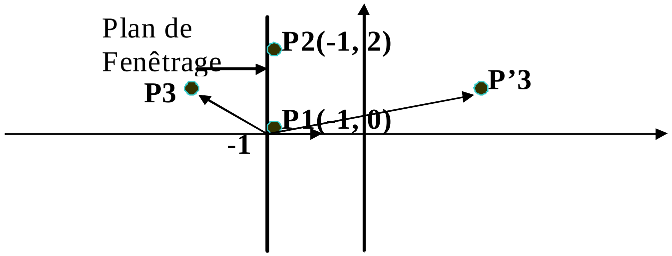
si z=0 → P3 est sur le segment P1P2

si z<0 → P3 est à gauche du segment P1P2

Exemple

Plan de fenêtrage en X=W=-1 perpendiculaire à l'axe des X.

Déterminer la position des points P3(-2,1) et P'3(2,1) par rapport à ce plan.



<u>Calcul du produit vectoriel</u>

Point P3 P1P2⊗P1P3=(0, 0, -2) Z=-2 < 0 → P3 est à gauche

Point P'3 P1P2⊗P1P'3=(0, 0, 6) Z=6 > 0 → P'3 est à droite

<u>Technique de substitution</u>

Point P3 x3-W = -2 – (-1) = -1 < 0 → P3 est à gauche

Point P'3 x3-W = 2 – (-1) = 3 > 0 → P'3 est à droite

<u>Technique du normale</u>

Normale intérieur en Ni=[1 0] et le point de fenêtrage en f(-1, 0)

Le produit scalaire des vecteurs donne :

Ni . [P3 – f] = -1 < 0 → P3 est à gauche

Ni . [P'3 – f] = 3 > 0 → P'3 est à droite

Rappel

- Un côté est visible si ses extrémités sont toutes deux visibles
- un côté est invisible si ses extrémités sont toutes deux invisibles
- si l'une des extrémités est visible et l'autre invisible, alors le côté coupe le plan de fenêtrage, on calcul donc le point d'intersection.

L'intersection de deux droites paramétriques dans le plan en 2 dimension

Equation paramétrique du segment [P1,P2] est :

P(s)=P1+(P2-P1)*s avec 0<=s<=1

Equation paramétrique du segment [P3,P4] est :

P(t)=P3+(P4-P3)*t avec 0<=t<=1

au point d'intersection on a P(s)=P(t)

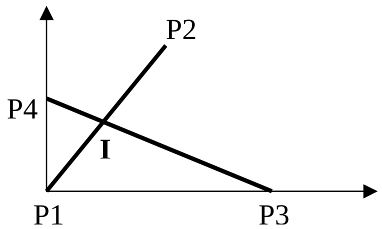
s'il n'y a pas de solution, c'est que les segments sont parallèles.

si s ou t $\notin$ [0, 1], c'est que les segments ne sont pas sécants.

Exemple

P1=[0 0]   P2=[3 2]   P3=[3 0]   P4=[0 2]

$$P1P2 = \begin{bmatrix} 3-0 \\ 2-0 \end{bmatrix} = \begin{bmatrix} 3 \\ 2 \end{bmatrix} = P2 \qquad P3P4 = \begin{bmatrix} 0-3 \\ 2-0 \end{bmatrix} = \begin{bmatrix} -3 \\ 2 \end{bmatrix}$$



P(s)=[0 0] + [3 2]*s → $\begin{cases} x(s) = 3s \\ y(s) = 2s \end{cases}$

P(t)=[3 0]+[-3 2]*t → $\begin{cases} x(s) = 3-3t \\ y(s) = 2t \end{cases}$

Si intersection on a $\begin{cases} 3s = 3-3t \\ 2s = 2t \end{cases} \Rightarrow s = t = 1/2$

Le point d'intersection est Pi(s)=[3/2 1]

Algorithme de Sutherland et Ogdman

Découpage d'un polygone PL par une fenêtre polygonale PW convexe

PL : Ligne polygonale de N1 sommets (Entré)

PW : Ligne polygonale fenêtre de N3 Sommets (le dernier sommet de PW est identique au premier) (Entré)

PS : Ligne polygonale de sortie de N2 sommets (Sortie).

Début

// Boucles sur les côtes de la fenêtre : Pour chaque côté de la fenêtre PW

Pour i variant de 1 à (N3-1) faire

    N2 ← 0

    PS ← vide

    // Pour chaque côté de la ligne polygonale PL

    Pour j variant de 1 à N1 faire

        Si j=1 alors F ← Pj   // Sauver le 1er=Dernier sommet

        Sinon

            Si coupe(S, Pj, Fi, Fi+1) alors

                I ← Intersection(S, Pj, Fi, Fi+1)

                Charger(I, N2, PS)

            Finsi

```
        Finsi
        S ← Pj
        Si visible(S, Fi, Fi+1) >= 0 alors
            Charger( S, N2, PS)
        Finsi
    Finpour
    Si (N2 > 0) alors
//Traitement du dernier côté de la ligne polygonale
        si coupe(S, F, Fi, Fi+1) > 0 alors
            I ← Intersection(S, F, Fi, Fi+1)
            Charger(I, N2, PS)
        Finsi
        // Découpage pour chacun des sous polygones obtenus
        LP1 ← LP2
        N1 ← N2
    Finsi
Finpour
```

FIN


```
Coupe(S, P, F1, F2) → Entier
Début
    V1 ← visible(S, F1, F2)
    V2 ← visible(P, F1, F2)
    Renvoyer (v1 <0 et v2 >0) ou (v1>0 et v2<0)
Fin
```

// Utilisation du produit vectoriel pour déterminer la visibilité d'un point P par rapport à un plan de fenêtrage défini par les points F1 et F2

```
Visible(P, F1, F2) → Entier
Début
    Renvoyer signe (x2-x1)(y-y1) – (x-x1)(y2-y1)
Fin
```

//Intersection de deux segments

Intersection(P1, P2, P3, P4) → Retourne le point d'intersection des segements P1P2 et P3P4

Début

P(s) ← P1+(P2-P1)*s = [x1 y1] + [x2-x1 y2-y1]*s

P(t) ← P3+(P4-P3)*t = [x3 y3] + [x4-x3 y4-y3]*t

// P(s) = P(t) →

$$\begin{bmatrix} x2-x1 & -(x4-x3) \\ y2-y1 & -(y4-y3) \end{bmatrix} \begin{bmatrix} s \\ t \end{bmatrix} = \begin{bmatrix} x3-x1 \\ y3-y1 \end{bmatrix}$$

//A*X=b → X = A^{-1}*b

$$\begin{bmatrix} s \\ t \end{bmatrix} = \begin{bmatrix} x2-x1 & -(x4-x3) \\ y2-y1 & -(y4-y3) \end{bmatrix}^{-1} * \begin{bmatrix} x3-x1 \\ y3-y1 \end{bmatrix}$$

I ← P1+(P2-P1)*s // ou I ← P3+(P4-P3)*t

Renvoyer I

Fin

Bibliographie

Livre : Algo pour l'infographie [David ROGERS]

Livre : Images de Synthèse M. Lucas