Remplissage de polygones.....2
Première méthode.....2
Calcul du rectangle englobant 6
Appartenance d'un point à un polygone.....8
Remplissage de polygone ligne par ligne 10
Algorithme 16

Remplissage de polygones

Première méthode

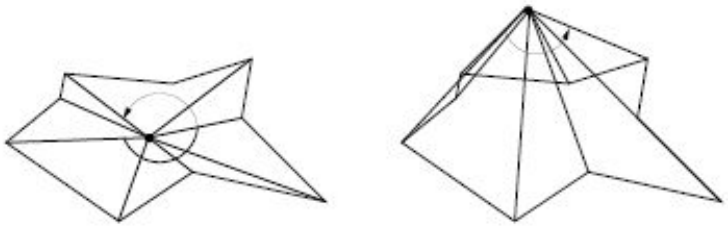
La plus simple méthode pour remplir un polygone consiste à examiner chaque pixel de l'écran (image ou trame) pour savoir s'il se trouve à l'intérieur de ce polygone.

Caractéristiques de la méthode

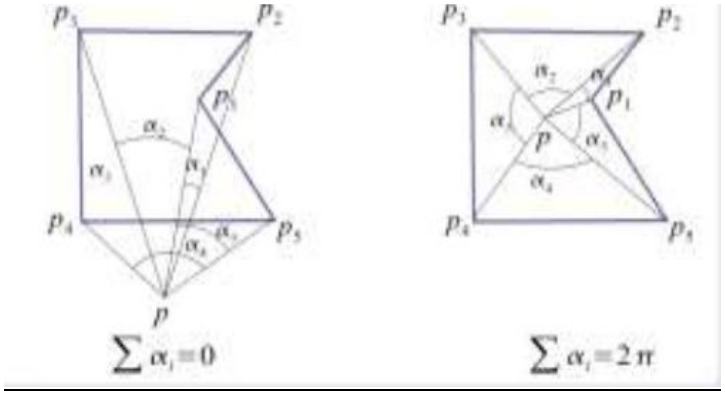
- simplicité
- conduit à un gaspillage

Rappel : appartenance d'un point à un polygone

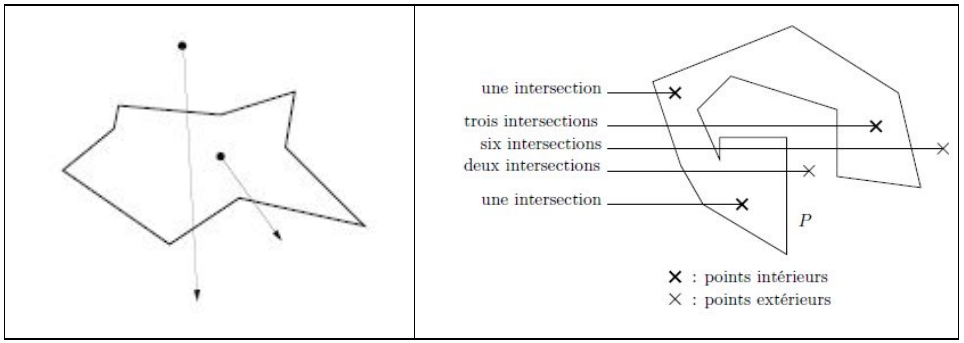
Première Méthode : basée sur l'angle sous lequel le point voit le polygone



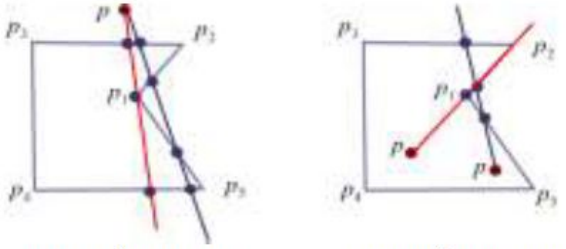
Points intérieurs : $\Sigma\ angles = 2\pi$; **points extérieurs** : $\Sigma\ angles = 0$.

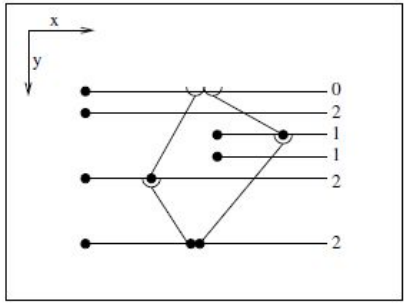


Deuxième méthode :



Points intérieurs : nombre d'intersections impair
Points extérieur : nombre d'intersections pair.





Les sommets du polygone constituent un cas particulier d'intersection.

En effet, suivant le cas l'intersection **est simple** : les deux cotés du polygone au sommet sont de part et d'autre de la demi-droite, ou **double** : les deux cotés du polygone sont du même coté de la demi-droite.

Amélioration

Réduire le volume englobant en déterminant un rectangle englobant.

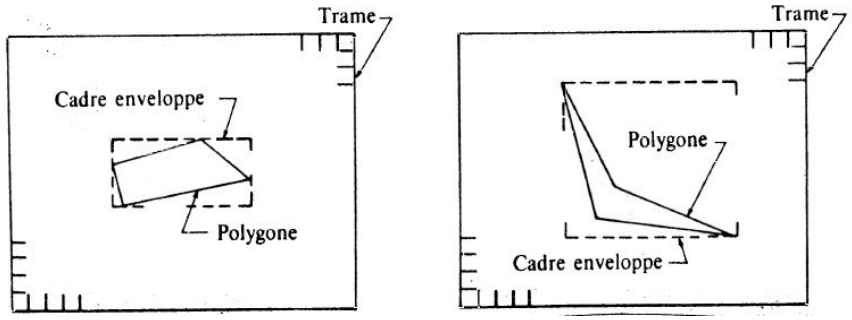
Le rectangle englobant est le plus petit rectangle qui contient le polygone.

Par suite, le test des points s'effectue uniquement par rapport aux points du rectangle englobant.

Remarque

La réduction du nombre de points extérieurs par rapport au nombre de points intérieurs dépend de la forme du polygone.

Exemples



Calcul du rectangle englobant

Pour déterminer le rectangle englobant d'un polygone en recherche les valeurs extrêmes des coordonnées des sommets.

RectangleEnglobant(Poly, RectEG)

Début

Xmin(RecEG) ← Max(Réel)

Ymin(RecEG) ← Max(Réel)

Xmax(RecEG) ← Min(Réel)

Ymax(RecEG) ← Min(Réel)

Pour i variant de 1 à nbsom(Poly) faire

Si X(Poly[i]) < Xmin(RecEG) alors

Xmin(RecEG) ← X(Poly[i])

Finsi

…

finpour

fin

Remplissage d'un polygone sans trous

Remplissage(RecEG, Poly, CR)

Début

```
Pour x variant Xmin(RecEG) à Xmax(RecEG) faire
    Pour y variant de Ymin(RecEG) à Ymax(RecEG) faire
        Si intérieur(x, y, Poly) alors
            AffichePixel(x, y, CR)
        Finsi
    Finpour
Finpour
```

Fin

Appartenance d'un point à un polygone

Méthode

Critère d'appartenance basé sur le calcul des angles

Pour chaque côté du polygone, l'angle sous lequel il est vu depuis le point à tester doit être déterminé avec son signe

Le calcul d'un angle d'un triangle à partir des coordonnées de ses 3 sommets (angle entre 2 vecteurs) peut se faire à l'aide du produit scalaire ou du produit vectoriel

Intérieur(x, y, Poly) → Booléen

Début

SomAngle ← 0

Pour i variant de 1à nbsom(Poly) faire

Ax ← X(Poly[i]) - x

Ay ← Y(Poly[i]) – y

Bx ← X(Poly[i+1]) - x

By ← Y(Poly[i+1]) – y

NormeA ← racineCarree(Ax*Ax + Ay*Ay)

NormeB ← racineCarree(Bx*Bx + By*By)

Si NormeA= 0 ou NormeB=0 alors

Renvoyer vrai

Sinon

ProduitScalaire_AB ← Ax*Bx + Ay*By

ProduitVectoriel_AB ← Ax*By – Ay*Bx

Sin ← ProduitVectoriel_AB /(NormeA*NormeB)

Cos ← ProduitScalaire_AB /(NormeA*NormeB)

SomAngle ← somAngle + arcTangente(cos/sin)

finsi

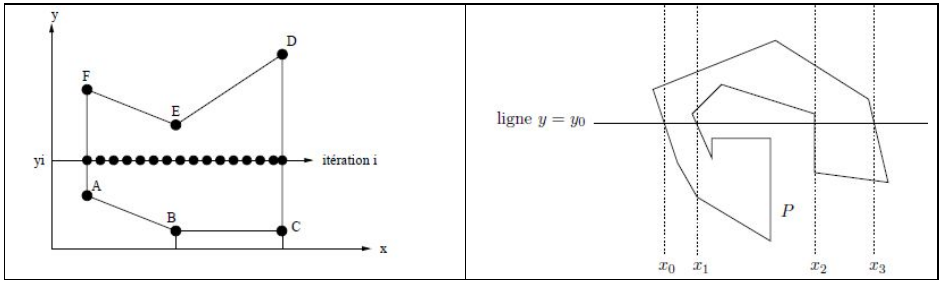
finpour

Renvoyer ← abs(SomAngle > Π)

Fin

Remplissage de polygone ligne par ligne

Principe



Ce type d'algorithme consiste à balayer les lignes horizontales appartenant au rectangle englobant le polygone à traiter et à afficher les pixels intérieurs au polygone sur chaque ligne :

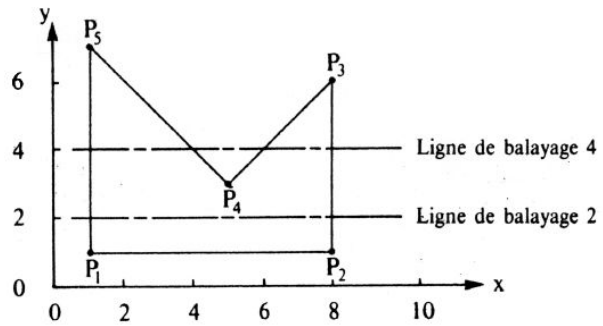
Balayage du polygone

Les caractéristiques des pixels d'une ligne de balayage donnée ne changent que là où côté du polygone coupe la ligne de balayage

Cette intersection divise la ligne de balayage en deux régions :

- région intérieur : pixels de la ligne de balayage qui sont à l'intérieur du polygone
- région extérieur : pixels de la ligne de balayage qui sont à l'extérieur du polygone

Exemple



La ligne de balayage 2 est divisée en 3 régions :

- ♦ x < 1 → Extérieur au polygone
- ♦ 1 <= x <= 8 → Intérieur au polygone
- ♦ x > 8 → Extérieur au polygone

La ligne de balayage 4 est divisée en 5 régions :

- ♦ x < 1 → Extérieur au polygone
- ♦ 1 <= x <= 4 → Intérieur au polygone
- ♦ 4 <= x <= 6 → Extérieur au polygone
- ♦ 6 <= x <= 8 → Intérieur au polygone
- ♦ x > 8 → Extérieur au polygone

➢ Les intersections de la ligne de balayage 4 ne sont pas nécessairement déterminées de la gauche vers la droite. En effet :

Si le polygone est décrit par la liste de ses sommets P1 à P5 Alors

- les intersections des côtés avec la ligne de balayage 4 seront déterminées dans l'ordre 8 6 4 1
- il faudra ensuite les trier dans l'ordre des x croissants (càd 1 4 6 8)
- traiter les intersections par paires :
  - ❖ les pixels 0 à 1 , 4 à 6, 8 à 10 sont mis à la couleur du fond
  - ❖ les pixels 1 à 4, 6 à 8 reçoivent la couleur de remplissage

Différentes étapes de l'algorithme de remplissage par ligne de balayage :

1) Déterminer pour chaque ligne de balayage, les points d'intersections avec le polygone
2) Les points obtenues doivent ensuite être triés d'abord selon Y (c à d par ligne de balayage), puis selon X
3) Les segments sont alors traités par une paire de points d'intersections consécutifs triés.

Remarques

- ♦ Les côtés horizontaux du polygone peuvent être ignorés, car ils sont parallèles aux lignes de balayage (ç à d ne peuvent pas couper une ligne de balayage)
- ♦ Un sommet appartenant à deux côtés suit un traitement spécial
- ♦ Afin d'éviter de devoir trier la liste des points lorsqu'elle est complète, il est préférable d'utiliser
  - ❖ un arbre binaire ordonné dans lequel chaque point sera inséré en respectant la relation d'ordre suivante :

    (x1, y1 < (x2, y2) si y1<y2 ou y1=y2 et x1<=x2
  - ❖ où une liste chaînée

Exemple

Remplissage par les lignes de balayage

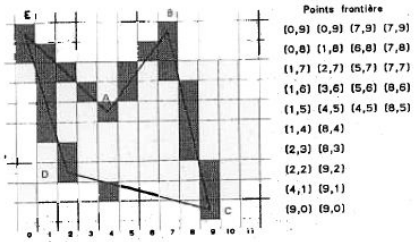


Fig.5.71 Points d'intersection

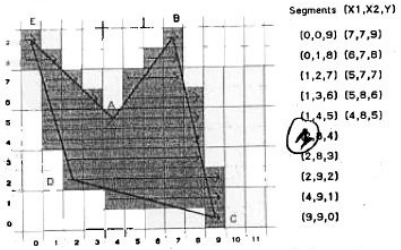


Fig.5.72 Points d'intersection triés et segments à remplir

Méthode de la liste d'activation des côtés

Le remplissage se fait par ligne de balayage, il est suffisant de ne stocker que les points relatifs (intersections) à cette ligne.

→ on décompose le polygone en tranches horizontales dans lesquelles seuls quelques côtés sont considérés.

Mise en œuvre

On utilise deux tables :

- LCA = Table des Côtés actifs

  = une liste chaînée pour gérer une suite de côtés actifs

  = cette liste contient les côtés ayant une intersection avec la ligne de balayage en cours de traitement.

  = cette liste est mise à jour à chaque changement de la ligne de balayage

  = Triée en permanence par ordre croissant des abscisses des points d'intersections.

- SI = Structure intermédiaire = contient les côtés du polygone triés suivant les y croissants

  = tableau de listes chaînées

  = contenant les informations nécessaires à la gestion de LCA

  = pour chaque ordonnée y, on va stocker l'ensemble des côtés c du polygone tels que y est l'ordonnée minimale Ymin des deux extrémités du côté c.

➢ En parcourant SI on pourra savoir quels côtés devront entrer dans LCA pour telle ou telle ordonnée.

- Pour gérer le retrait des côtés de LCA, on mémorise l'ordonnée maximale de chaque côté.
- On stocke donc pour chaque côté :
  - L'abscisse Xmin correspondant à Ymin
  - l'ordonnée ymax du côté
  - 1/m l'inverse du coefficient directeur (incrément en x pour un déplacement unitaire en y)

    yi+1 = yi + 1 → xi+1 = xi + 1/m

    En effet,

    yi = axi + 1

    yi+1 = axi+1 + 1

    = yi + 1

    → 1 = m(xi+1 – xi)

    → xi+1 = xi + 1/m

Les liste chaînées sont triées par ordre croissant de xmin, puis pour deux xmin identiques, par ordre croissant de la valeur 1/m.

Ce tri à pour but de faciliter le déplacement des côtés vers LCA car cet ordre de classement devra être respecté dans LCA.

Algorithme

Remplissage_ActivationCôtés(Poly, CR)

Début

```
// Création de la structure SI
```

SI ← Créat_SI(Poly)

```
//Initialisation  de la structure LCA à vide
```

LCA ← Init_LCA()

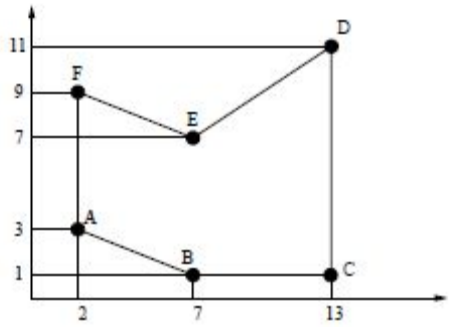
Pour chaque ligne de balayage y intersectant le polygone faire

- Gérer les entrés dans LCA à partir de SI (c à d mettre dans lCA les entrées de la table SI dont le Ymin = y)
- Gérer les sorties de LCA à partir de SI (c à d enlever de LCA les entrées de la table SI dont le Ymax=y
- Afficher tous les morceaux de la ligne de balayage décrits dans LCA (c à d remplir tous les pixels entre deux intersections qui se trouvent à l'intérieur du polygone en utilisant la règle de parité pour déterminer si un point est à l'intérieur d'une région : c'est à dire la parité est initialement paire, qu'à chaque intersection rencontré la parité est inversée et que les pixels sont tracés seulement quand la parité est impaire.

Finpour

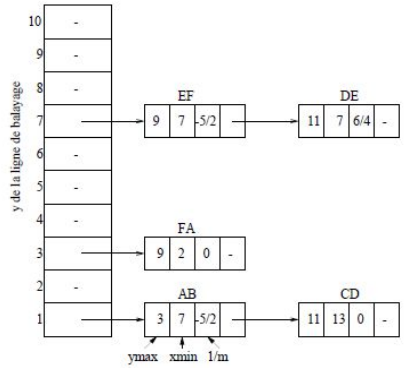
FIN

Exemple



Remarque : les côtés horizontaux du polygone peuvent être ignorés, car ils sont parallèles aux lignes de balayage.

SI=



Remplissage de la table LCA : voir le détail en CM