TME2. Enveloppe convexe.

BM Bui-Xuan

1 Enveloppe convexe

Problème de l'enveloppe convexe d'un ensemble de points : Un polygone est convexe si tout segment joignant deux points du polygone est entièrement contenu dans le polygone. L'enveloppe convexe d'un ensemble de points est le plus petit (polygone) convexe contenant tout point de l'ensemble. Le calcul de l'enveloppe convexe est un calcul fondamental en géométrie algorithmique.

Soit un ensemble $\mathcal{S}$ de points dans le plan.

Exercice 1 – Algorithme naïf

Deux points A et B de $\mathcal{S}$ est sur l'enveloppe convexe de $\mathcal{S}$ si tout autre point de $\mathcal{S}$ est strictement "du même côté" de la droite passant par A et B. Dans le cas où les points sont "du même côté", mais certains sont alignés avec A et B, les deux points A et B doivent être les deux points les plus éloignés parmis ces points.

1. Implanter l'algorithme naïf pour le calcul de l'enveloppe convexe en s'inpirant du produit vectoriel pour le test "du même côté".

   ***N.B.** : Le produit vectoriel de deux vecteurs colinéaire est nul.*

2. Analyser la complexité de cet algorithme.

1.1 Pré-calcul

Le principe du pré-calcul, ou *kernelization*, consiste à fournir une méthode pour réduire toute instance d'un problème algorithmique à une instance équivalente de taille plus petite. Pour le calcul de l'enveloppe convexe, il s'agit surtout des méthodes de filtrage : nous allons enlever les points "inutiles".

Exercice 2 – Tri par pixel

Soient A, B, C trois points de $\mathcal{S}$ ayant les mêmes abscisses.

1. Montrer qu'au plus deux de ces points appartiennent à l'enveloppe convexe de $\mathcal{S}$. En déduire une première méthode de filtrage pour le calcul de l'enveloppe convexe.
2. Appliquer le tri par pixel avant d'appeler l'algorithme naïf pour calculer l'enveloppe convexe de $\mathcal{S}$. Y a t il une différence concernant le temps de calcul ?

Exercice 3 – Filtrage Akl-Toussaint

Voir TME1.

1. Est il vraiment plus avantageux d'appliquer à la fois le tri par pixel et le filtrage Akl-Toussaint ? (Pourquoi ?)

1.2 Algorithmes

Exercice 4 – Algorithme Jarvis (*alias* algorithme du paquet cadeau)
Le principe est le suivant :

— Trouver un premier point P appartenant à l'enveloppe convexe : celui d'abscisse minimum fera l'affaire.

— Trouver un point Q tel que PQ forme un côté de l'enveloppe convexe : on peut utiliser le produit vectoriel.

— Parcourir $\mathcal{S}$ et trouver le point R tel que l'angle $(\overrightarrow{PQ}, \overrightarrow{QR})$ est minimum : on peut utiliser le produit scalaire.

— QR sera alors un côté de l'enveloppe convexe : on remplace P et Q par Q et R, respectivement.

— Répéter les deux étapes précédentes jusqu'à ce qu'on retombe sur le premier point P.

1. Implanter l'algorithme Jarvis et analyser sa complexité.

Exercice 5 – Algorithme Graham (modifié)
Le principe est le suivant :

— Trier les points selon le filtrage par pixel sur les abscisses : les points seront par exemple triés d'abord par abscisse croissant parmi ceux d'ordonnée minimum, puis par abscisse décroissant parmi ceux d'ordonnée maximum.

— Pour tout triplet de points successives qui "tourne à droite", enlever le point du milieu : on peut utiliser le produit vectoriel.

1. Implanter l'algorithme Graham et analyser sa complexité.

Exercice 6 – Algorithme QuickHull
Le principe est le suivant :

— Considérer les quatre points A, B, C, D définis dans le filtrage Akl-Toussaint (et appliquer ce filtrage).

— Pour chaque côté du quadrilatère, disons AB, trouver le point X le plus éloigné de AB "du côté extérieur" du quadrilatère : on peut utiliser le produit vectoriel.

— Enlever tout point appartenant au triangle ABX.

— Ajouter AX et BX aux côtés à traiter, remplacer le terme quadrilatère par pentagone et répéter.

1. Implanter l'algorithme QuickHull et analyser sa complexité.