

Infographie - M1

Examen du 26/05/2010

2h - Notes de cours et TD autorisées.

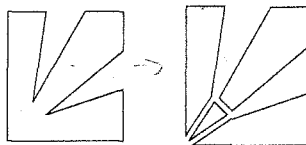
Les algorithmes demandés doivent être rédigés avec le même niveau de détail que ceux vus en cours : les opérations et constantes mathématiques de base (constante π , sin, cos, produit vectoriel $\wedge$, produit scalaire $\cdot$, produit croisé $\times$, etc) sont données, mais tous les autres calculs et traitements à effectuer (e.g. calculs d'intersections) doivent être décrits en détail, éventuellement de manière structurée (étapes numérotées, conditionnelles, itérations). Vous pouvez supposer que vous disposez de piles, listes et tableaux.

La correction tiendra compte de la clarté et de la précision de vos solutions. Inutile de descendre jusqu'au pseudo-code, mais vos algorithmes doivent être implémentables sans aucune réflexion supplémentaire (i.e. la simple description géométrique du traitement n'est pas suffisante).

Problème

Si vous ne parvenez pas à traiter une partie, vous pouvez la supposer résolue pour traiter les suivantes.

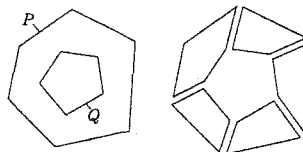
(I) Donnez un algorithme permettant de découper un polygone arbitraire du plan $P = (P_1, \dots, P_n)$ en polygones convexes. Votre algorithme devra renvoyer une liste de polygones convexes recouvrant exactement le polygone de départ.



La solution n'est pas unique. Le polygone initial étant précisément non nécessairement convexe, certaines idées immédiates peuvent ne pas fonctionner.

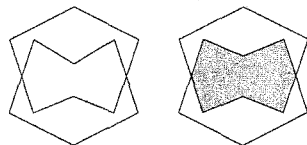
(II) Donnez un algorithme le plus simple possible (plus simple que l'algorithme du cours, gérant le cas d'un polygone non convexe) permettant de déterminer si un point M du plan est intérieur à un polygone convexe $P = (P_0, \dots, P_{n-1})$.

(III) Donnez un algorithme prenant en argument deux polygones du plan $P = (P_0, \dots, P_{n-1})$, $Q = (Q_0, \dots, Q_{m-1})$ tous les deux convexes et tels que Q soit intérieur à P , et renvoyant une liste de polygones convexes recouvrant exactement toute la surface de P extérieur à Q . Justifiez la correction de votre méthode.



(IV) Rappelons que l'algorithme de Weiler-Atherton permet de calculer le fenêtrage d'un polygone par un autre, dans le cas où une arête du premier intersecte au moins une arête du second. Notez que l'algorithme ne calcule rien d'autre

que les contours de la surface formée de tous les points du plan intérieurs aux deux polygones. Notez également que si les deux polygones sont convexes, cette surface recouvre un unique polygone convexe:



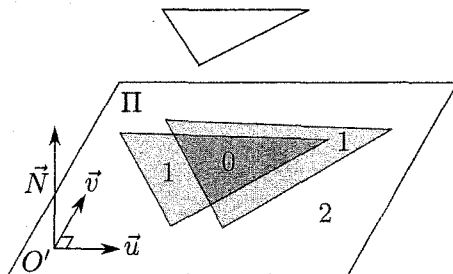
Montrer comment utiliser l'algorithme de Weiler-Atherton (et les précédents) pour calculer cette fois, étant donnés deux polygones convexes P, Q (éventuellement disjoints, ou encore, l'un des deux éventuellement inclus dans l'autre) une liste de polygones convexes recouvrant exactement la surface formée de tous les points du plan intérieurs à seulement l'un des deux polygones (mais pas intérieurs aux deux).

(V) À partir de Weiler-Atherton et de l'algorithme trouvé en (IV), montrer comment écrire un nouvel algorithme prenant en argument K polygones convexes du plan, et renvoyant K listes. Pour chaque $k \in [1, \dots, K]$, la liste numéro k devra contenir un ensemble de polygones convexes recouvrant exactement la surface formée de tous les points du plan intérieurs à exactement k polygones.

(VI) Cette dernière partie est une application de l'algorithme trouvé en (V). On suppose placés dans l'espace usuel:

- K sources lumineuses, chacune sans volume, chacune émettant de la lumière dans toutes les directions, représentées par les points $L_1, \dots, L_K$.
- une surface plane et sans bornes, représentée par un plan Π de normale $\vec{N}$ et contenant le point O' ; on suppose le plan Π muni d'un repère local orthonormé $(O', \vec{u}, \vec{v})$ (i.e. on suppose $\|\vec{u}\| = \|\vec{v}\| = 1$, avec $\vec{u}$ et $\vec{v}$ orthogonaux),
- un polygone convexe $(P_0, \dots, P_{n-1})$.

On pose la convention suivante : un point de Π est d'intensité lumineuse k s'il est éclairé par exactement k sources lumineuses. Dans la figure ci-dessous, deux sources lumineuses sont placées vers le haut, entre le plan et le polygone:



Donnez un algorithme permettant de calculer pour chaque $k \in [0, \dots, K]$ (notez le K exclus), une liste de polygones recouvrant exactement la portion de Π d'intensité lumineuse égale à k . Détaillez soigneusement vos calculs.