

Infographie - M1

Examen du 26/05/2010

Indications de corrections

(I) On pose comme d'habitude $P_{n+k} = P_k$ pour tout $k > 0$. Le plus simple est encore de trianguler le polygone. On peut supposer que les sommets sont énumérés comme dans Weiler-Atherton (dans le sens direct). Si le polygone n'est pas réduit à un triangle, il suffit de trouver un $i \in [0, \dots, n[$ tel que les trois sommets P_i, P_{i+1}, P_{i+2} vérifient deux propriétés :

1. $P_i \vec{P}_{i+2}$ est intérieur au polygone - ce qui, avec la convention faite sur l'énumération des sommets, revient à $P_i \vec{P}_{i+1} \propto P_{i+1} \vec{P}_{i+2} > 0$,
2. $[P_i P_{i+2}]$ ne coupe aucune arête du polygone - cette seconde condition est nécessaire, et n'est pas impliquée par la première.

L'existence d'un tel i n'est pas trop dure à prouver (en admettant que tout polygone peut être triangulé, ce qui est intuitivement clair, tout polygone peut être construit par assemblage de triangles formant des polygones de plus en plus grands, et il suffit pour se convaincre de l'existence du trio de points cherché de considérer le dernier triangle ajouté à un tel assemblage). On découpe le polygone en deux parties suivant $[P_i P_{i+2}]$, et on triangule récursivement la partie ne contenant pas P_{i+1} .

A noter que dans certaines copies, le découpage se base bien sur l'une ou l'autre des conditions, mais pas sur les deux. D'autre part, certains algorithmes proposés ne se servent pas de la récurrence, ce qui rend le traitement assez laborieux.

(II) En posant $P_n = P_0, P_{n-1} = P_1$, il suffit par exemple de vérifier pour chaque $i \in [0, \dots, n[$ que $P_i \vec{P}_{i+1} \propto P_i \vec{P}_{i+2}$ et $P_i \vec{P}_{i+1} \propto P_i \vec{M}$ ont même signe.

(III) Il faut en principe gérer le cas singulier où Q n'est pas strictement intérieur à P : deux sommets de P, Q confondus, ou encore, un sommet de Q sur une arête de P , ou encore, un sommet de P sur une arête de Q . Dans ce cas, la surface de P extérieure à Q forme déjà un ensemble de polygones - il faut extraire d'une manière ou d'une autre ces polygones (ce qui n'est pas trop difficile) et les passer à (I).

Lorsque Q est strictement intérieur à P , le seul problème à résoudre est le fait que la surface de P extérieure à Q n'est pas un polygone. On peut par exemple choisir deux sommets distincts de P et relier chaque sommet au sommet de Q le plus proche (sachant que P et Q sont convexes, les deux segments construits ne peuvent intersecter une arête) : la surface de P extérieure à Q se retrouve alors découpée en deux polygones, que l'on passe à (I).

(IV) On peut lancer Weiler-Atherton, et examiner les intersections ajoutées aux deux polygones. Si le calcul d'intersections n'ajoute pas plus d'un sommet à chaque polygone, ou bien l'un des deux polygones est (faiblement) intérieur à l'autre et on peut utiliser (III) ou bien les deux polygones sont (faiblement) disjoints, et la surface cherchée est formée des deux polygones (le test (II) permet de trancher entre les trois situations possibles).

Sinon, la liste contient au moins deux intersections, et les polygones étendus permettent facilement de récupérer les polygones couvrant la surface cherchée - soit par calcul direct, soit en se contentant de modifier deux lignes de la boucle principale de Weiler-Atherton de manière à rester à l'extérieur du polygone considéré comme celui de fenêtrage et non à l'intérieur. Une fois de plus, on triangularise les polygones trouvés par (I).

(V) S'il n'y a que deux polygones, on utilise d'une part (IV) (pour la liste correspondant aux points intérieurs à exactement 1 polygone) d'autre part Weiler-Atherton (pour la surface intérieure aux deux). Dans chaque cas, (I) permet de sur-découper les polygones générés en polygones convexes. S'il y a $K > 2$ polygones, le traitement est un peu plus complexe. On isole un polygone (mettons, P_K), on traite récursivement les $K - 1$ polygones restants, et l'on obtient $K - 1$ listes $L_1, \dots, L_{K-1}$,

On part ensuite de $L_K = \emptyset$, $\mathcal{P} = \{P_K\}$, et on effectue le traitement suivant tant que $\mathcal{P}$ n'est pas vide (on admettra que le traitement termine, ce qui est à peu près intuitivement clair mais pas complètement immédiat) :

1. On retire un élément P de $\mathcal{P}$.
2. On invoque l'algorithme pour $K = 2$ successivement sur chacun des polygones des listes $L_1, \dots, L_k$, en s'arrêtant, s'il existe, au premier polygone Q tel que cet algorithme restreint renvoie un couple de listes (L'_1, L'_2) avec L'_1 non vide.

Si $Q \in L_k$, on retire Q de L_k , les éléments de L'_1 intérieurs à Q (utiliser (II)) sont ajoutés à L_k (ils sont couverts par Q , mais pas par P), les éléments de L'_1 intérieurs à P (utiliser encore (II)) sont ajoutés à $\mathcal{P}$ (ils sont couverts par P , mais pas par Q), les éléments de L'_2 sont ajoutés à L_{k+1} (ils sont couverts à la fois par P et Q , soit par un polygone de plus).

Si Q n'existe pas, on ajoute P à L_1 (P est disjoint de tout autre polygone présent dans $L_1, \dots, L_K$).

(VI) Il suffit de calculer l'ombre du polygone relativement à chaque source lumineuse (pour chaque k , pour chaque j , calcul du point d'impact du rayon d'origine L_k et de direction $L_j \vec{P}_j$ sur Π , c.f. TD 1), de transformer chaque ombre en un polygone du plan par passage aux coordonnées locales de Π (produits scalaires de $O'\vec{M}$ avec $\vec{u}, \vec{v}$), d'invoquer (V), puis de faire revenir aux coordonnées de l'espace les sommets des polygones obtenus ((x, y) donne $O' + x \times \vec{u} + y \times \vec{v}$).