

Chapitre 3 : Triangulation et applications

Arnault Ioualalen, Arnaud Mary, Hélène Amodéos, Benoit Lopez

1 Problème de la triangulation

But : Partitionner en triangles un polygone ou l'enveloppe convexe d'un ensemble de points.

Motivations :

- Imagerie 3D
- Décomposition d'un polygone : calcul d'aire, calcul de plus court chemin...
- Reconstruction 3D : construire un maillage réaliste à partir d'un ensemble de points.

Objectif : Faire de « belles » triangulations.

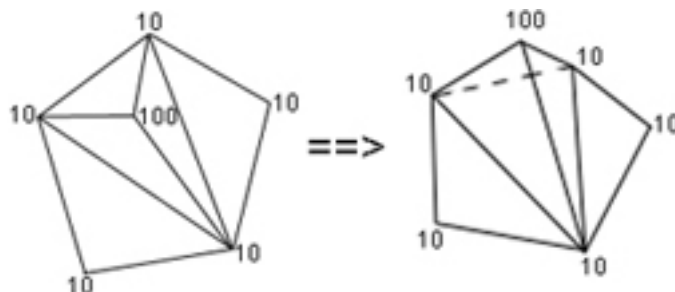


FIG. 1 – Exemple d'une mauvaise triangulation

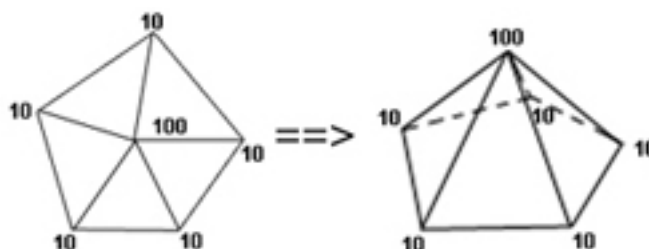


FIG. 2 – Exemple d'une bonne triangulation

Un problème connexe à la triangulation : Déterminer un régionnement du plan (diagramme de « Voronoï »).

Exemple 1.1 La triangulation permet notamment de :

- Trouver le site le plus proche d'un point du plan (bureau de poste).
- Trouver un chemin qui évite au mieux tous les sites (mines).

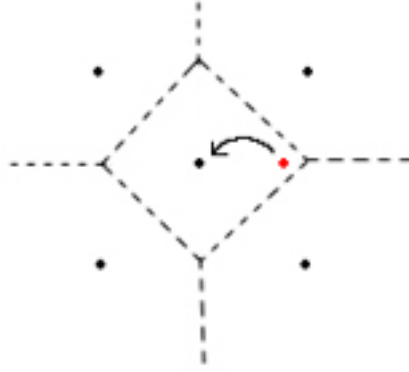


FIG. 3 – Exemple du bureau de poste

Définition 1.1 Soit P un ensemble de points. Une triangulation de P est un ensemble de triangles $\tau = (T_1, T_2, \dots, T_t)$ tels que :

1. les sommets des triangles sont des points de P ,
2. $\forall p \in P, \forall i \in [1..t] \Rightarrow p$ est un sommet de T_i ou $p \notin T_i$,
3. (T_1, T_2, \dots, T_t) est une partition de $EC(P)$.

Pour un polygone simple (i.e. sans trou) $Q = (p_0, p_1, \dots, p_r, p_0)$, une triangulation de Q est un ensemble de triangles $\tau = (T_1, T_2, \dots, T_t)$ tel que :

1. les sommets des triangles sont des points de P ,
2. $\forall p \in P, \forall i \in [1..t] \Rightarrow p$ est un sommet de T_i ou $p \notin T_i$,
3. (T_1, T_2, \dots, T_t) est une partition de l'intérieur du polygone.

Remarque : Seule la dernière condition change, les deux premières sont identiques.

Exemple 1.2

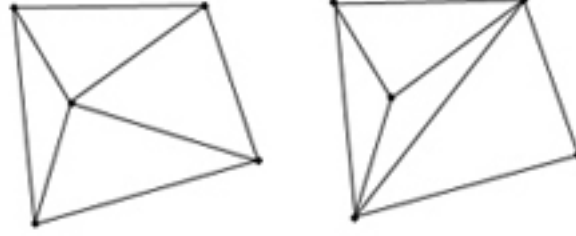


FIG. 4 – 2 triangulations possibles pour un même ensemble de points

Dans les 2 cas, il existe plusieurs triangulations possibles, mais on a toujours les invariants suivants (voir chapitre sur les graphes planaires) :

- Soit P un ensemble de n points, si on note :
 - n_e = le nombre de sommets de $EC(P)$.
 - m = le nombre de segments créés (à partir de $EC(P)$).
 - t = le nombre de triangles créés.
 alors on a les égalités suivantes :
 - $m = 3(n - 1) - n_e$.
 - $t = 2(n - 1) - n_e$.
- Soit Q un polygone à n sommets, si on note m_i le nombre de segments créés à l'intérieur de Q , on a les égalités suivantes :
 - $m_i = n - 3$.
 - $t = n - 2$.

Définition 1.2 Le dual d'une triangulation $\tau = (T_1, T_2, \dots, T_t)$ est le graphe dont les sommets sont $1, \dots, t$ et dont les arêtes sont les paires $\{i, j\}$ pour lesquelles T_i et T_j ont une frontière commune.

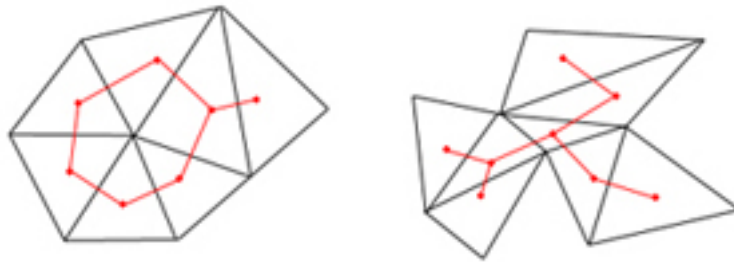


FIG. 5 – Duals de triangulations d'un ensemble de points et d'un polygone

2 Triangulation d'un ensemble de points : algorithme incrémental

On considère $P = (p_1, p_2, \dots, p_r)$ un ensemble de points du plan.

Principe de l'algorithme : On trie les points de P par ordre lexicographique croissant, i.e. $(x, y) <_{lex} (x', y') \Rightarrow (x < x') \text{ ou } (x = x' \text{ et } y < y')$. Puis on les insère dans la triangulation.

On note p_1, p_2, \dots, p_n les points triés par ordre croissant et on note EC_i l'enveloppe convexe $EC(p_1, p_2, \dots, p_i)$. On dit qu'un point p_j de EC_i est visible par p_{i+1} si le segment $[p_{i+1}p_j]$ ne coupe pas EC_i . Pour trianguler à l'étape $i + 1$ on cherche tous les points visibles par le point qu'on cherche à ajouter.

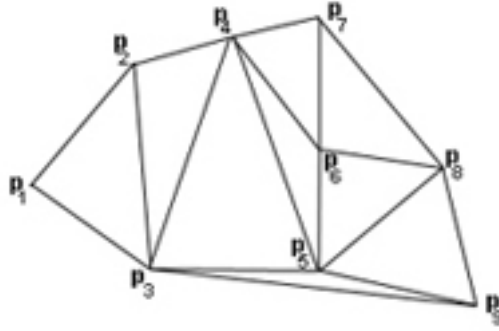


FIG. 6 – Illustration du principe de l'algorithme

Du fait que les points de P sont triés par ordre lexicographique on a le lemme suivant :

Lemme 2.1 $\forall i \in \{1, \dots, n\}$ on a :

- p_i est un point de EC_i .
- les points visibles de EC_i forment un intervalle.
- p_i est visible par p_{i+1} .

De ce fait lors du calcul de l'étape $i+1$ les points visibles de EC_i sont à rechercher autour de p_i

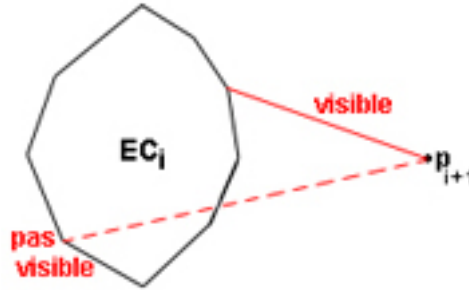


FIG. 7 – Points visibles par p_{i+1}

On a l'algorithme suivant :

Algorithme 1 TRIANGULATION INCREMENTALE

– **Données :** $P = (p_1, \dots, p_n)$ des points du plan
– **Sortie :** τ une triangulation de P

$\tau \leftarrow \{p_1, p_2, p_3\}$
 $EC \leftarrow (p_1, p_2, p_3)$ ou (p_1, p_3, p_2) (dans le sens direct)
pour $i=3$ à $n-1$ **faire**
 # ajout de p_{i+1}
 On note $EC = (q_1, \dots, q_l)$ avec $q_k = p_i$
 tant que $(\det(p_{i+1}q_k, p_{i+1}q_{k+1}) < 0)$ **faire**
 $\tau \leftarrow \tau \cup \{p_{i+1}q_kq_{k+1}\}$
 $k \leftarrow k + 1$
 fin tant que
 $k_{haut} \leftarrow k$
 Retrouver k tel que $q_k = p_i$
 tant que $(\det(p_{i+1}q_k, p_{i+1}q_{k-1}) > 0)$ **faire**
 $\tau \leftarrow \tau \cup \{p_{i+1}q_kq_{k-1}\}$
 $k \leftarrow k - 1$
 fin tant que
 $k_{bas} \leftarrow k$
 Mise à jour de EC : on remplace $(q_{k_{bas}+1}q_{k_{bas}+2} \dots q_{k_{haut}-1})$ par p_{i+1}
fin pour
Retourner : τ

Preuve : On admet la validité de l'algorithme.

Complexité :

- le tri des points peut se calculer en $O(n \log(n))$
- la mise à jour de l'enveloppe convexe peut s'implémenter en $O(1)$ avec une liste doublement chaînée.
- la boucle principale créera au plus $2n$ triangles.

Total : $O(n \log(n))$

3 Triangulation de Delaunay

Motivation : Améliorer l'algorithme précédent qui parfois rend des triangulations « moches »

Exemple 3.1 Problème de la cartographie 3D

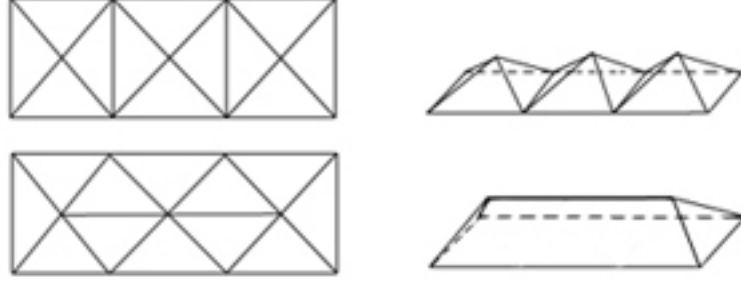


FIG. 8 – La première triangulation donne des pyramides alors que la réalité serait plutôt une chaîne de montagne (deuxième triangulation)

Remarque : Soit P un ensemble de n points du plan. Toute triangulation de P contient $t = 2(n - 1) - |EC(P)|$ triangles. On a donc toujours le même nombre d'angles : $3t$.

Pour une triangulation τ on note $A(\tau)$ la suite des angles de tous les triangles de τ , triée par ordre croissant. On dit que τ est meilleur que τ' si $A(\tau) >_{lex} A(\tau')$ (i.e. le premier terme qui diffère entre $A(\tau)$ et $A(\tau')$ est plus grand dans $A(\tau)$).

Exemple 3.2

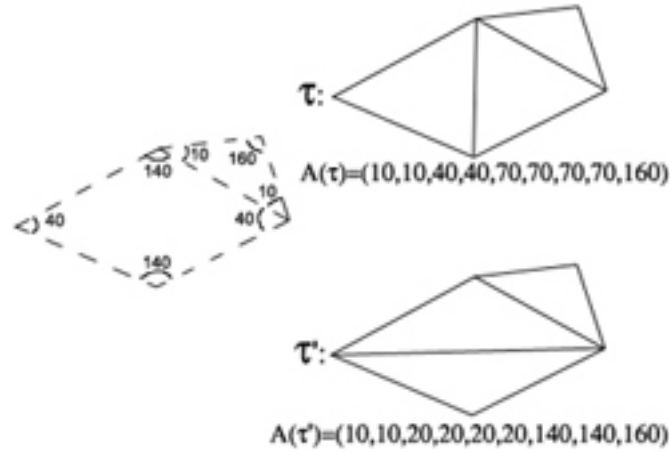


FIG. 9 – τ est meilleure que τ'

Définition 3.1 Une triangulation de Delaunay de P est une triangulation τ de P avec $A(\tau)$ maximum parmi toutes les triangulations.

Remarque : Si l'ensemble P ne contient aucun quadruplet de points cocycliques alors la triangulation de Delaunay est **unique**.

Problématique : Comment savoir si une nouvelle triangulation améliore le résultat ?

Lemme 3.1 Soit D un point du plan, si D est inclus strictement dans le cercle circonscrit du triangle ABC , alors la triangulation $\tau = \{ACD, ABD\}$ est meilleure que $\tau' = \{ACB, DCB\}$ (car $A(\tau) >_{lex} A(\tau')$)

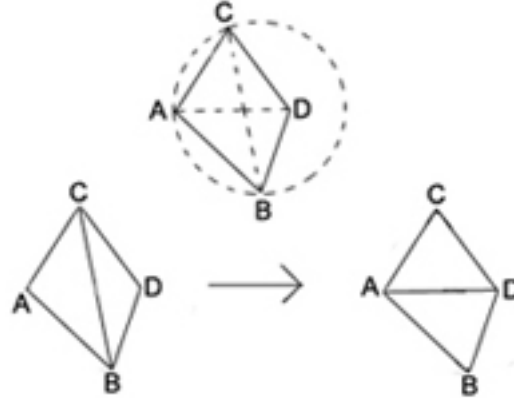


FIG. 10 – Exemple de flip

Caractérisation : Soit $\tau = (T_1, T_2, \dots, T_t)$ une triangulation d'un ensemble de points P , alors τ est une triangulation de Delaunay de $P \Leftrightarrow \forall i \in [1..t]$ le cercle circonscrit à T_i ne contient strictement aucun sommet de P . (i.e il n'y a plus aucun « flip » possible)

Remarques :

- l'implication \Rightarrow est immédiate : en effet si on a un flip qui est possible encore alors on peut faire croître strictement $A(\tau)$
- l'implication \Leftarrow est beaucoup moins évidente (notion de triangulation localement de Delaunay).

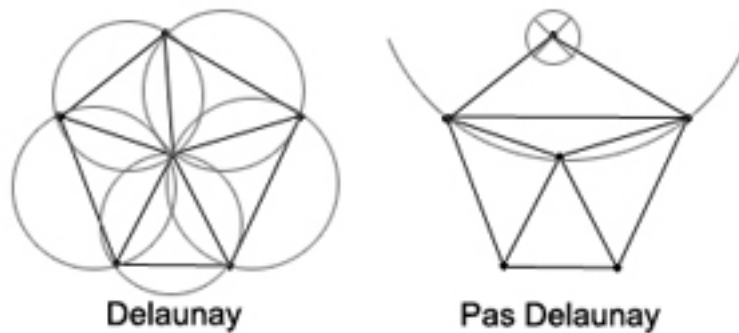


FIG. 11 – À gauche une triangulation de Delaunay, à droite une triangulation qui n'est pas de Delaunay

L'algorithme suivant permet d'obtenir une triangulation de Delaunay :

Algorithme 2 DELAUNAY-PAR-FLIP

- **Données** : $P = (p_1, \dots, p_n)$ des points du plan
 - **Sortie** : $\tau = (T_1, \dots, T_t)$ une triangulation de Delaunay de P
 - Construire une triangulation τ de P
 - tant que** un flip est possible dans τ **faire**
 - Faire le flip dans τ .
 - fin tant que**
 - Retourner** : τ
-

Remarques :

- L'algorithme se termine car la suite $A(\tau)$ croît strictement, mais quand ?
- On peut construire un algorithme polynomial : Bower-Watson.

Algorithme 3 BOWER-WATSON

- **Données** : $P = (p_1, \dots, p_n)$ des points du plan
 - **Sortie** : $\tau = (T_1, \dots, T_t)$ une triangulation de Delaunay de P
 - Trouver un triangle ABC contenant tous les sommets de P
 - $\tau = \{ABC\}$
 - Noter (p_1, \dots, p_n) les points de P à ajouter
 - pour** $i=1$ à n **faire**
 - Trouver tous les triangles de τ qui contiennent p_i dans leur cercle circonscrit
 - Noter q_1, \dots, q_l les sommets de ces triangles triés par ordre polaire croissant depuis p_i
 - Supprimer ces triangles de τ
 - $\tau \leftarrow \tau \cup \{q_1 q_2 p_i, q_2 q_3 p_i, \dots, q_{l-1} q_l p_i, q_l q_1 p_i\}$
 - fin pour**
 - Supprimer tous les triangles contenant A , B ou C
 - Retourner** : τ
-

Preuve : On admet la validité de l'algo.

Complexité :

- le tri de q_1, q_2, \dots, q_l peut s'implémenter en $O(n \log(n))$
- trouver les triangles contenant p_i se fait au pire en $O(n)$ car il y a un nombre linéaire de triangles en tout.
- La boucle principale s'exécute exactement n fois.

Total : l'algorithme de BOWER-WATSON s'exécute en $O(n^2 \log(n))$

Remarques :

- On peut améliorer la complexité car si on peut connaître les triangles voisins d'un triangle donné en $O(1)$, du coup on peut déterminer le tri de (q_1, q_2, \dots, q_l) en $O(n)$ (car les triangles à enlever forment un ensemble connexe étoilé)
- Si on améliore encore les structures de données utilisées on peut arriver à faire du $O(n \log(n))$ avec une même approche incrémentale.

4 Diagramme de Voronoï

Définition 4.1 On note :

$H_{ab} = \{p \in \mathbb{R}^2 : d(a,p) < d(a,b)\}$: le demi-plan défini par la médiatrice de $[ab]$ contenant a .

La cellule de Voronoï de a est : $V(a) = \bigcap_{b \in P, a \neq b} H_{ab}$.

Soit P un ensemble de n points du plan, on appelle diagramme de Voronoï l'ensemble des cellules de Voronoï $V(a)$ avec $a \in P$.

Exemple 4.1

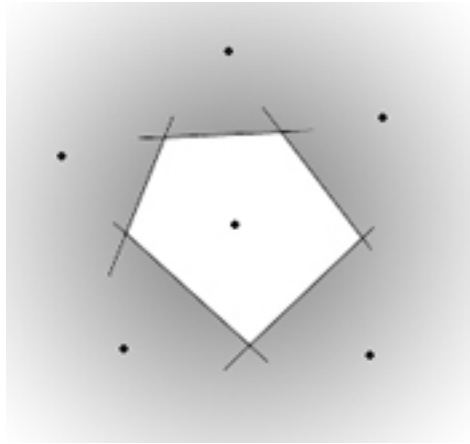


FIG. 12 – Exemple d'une cellule de Voronoï

Objectifs : Trouver une méthode pour calculer ces cellules, i.e trouver les médiatrices qui interviennent dans le diagramme.

Propriété 4.1 Soit P un ensemble de points du plan. Alors on a :

- Chaque cellule est convexe
- Les segments du diagramme de Voronoï sont portés par des médiatrices de paires de points de P .
- si P ne possède pas 4 points cocycliques alors les sommets du diagramme de Voronoï (au sens graphe) sont tous de degré 3.

Remarque : Un point v du diagramme de Voronoï est l'intersection des médiatrices des centres (2 à 2) des cellules contenant v . On a donc que v est le centre du cercle circonscrit au centre des cellules contenant v .

Théoreme 4.1 Si $v \in V(p_1) \cap V(p_2) \cap V(p_3)$ alors il n'existe pas de sommets de P strictement à l'intérieur du cercle circonscrit à p_1, p_2, p_3 (et de centre v).

Preuve : Si un tel sommet p existait alors on aurait $d(v,p) < d(v,p_i)$ pour $i=1,2,3$. Et donc on aurait $v \notin V(p_i)$ pour $i=1,2,3$.

Exemple 4.2

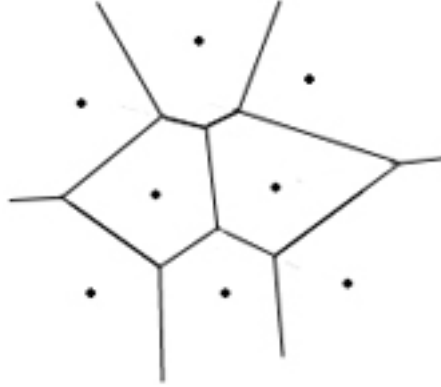


FIG. 13 – Exemple de diagramme de Voronoï

Corollaire : Si P ne contient pas 4 sommets cocycliques, la triangulation de Delaunay est le dual du diagramme de Voronoï.

Preuve : Comme il n'y a pas 4 sommets cocycliques les sommets du diagramme de Voronoï sont tous de degré 3 donc les faces (cf. chapitre sur les graphes planaires) du dual sont des triangles (sauf la face externe). Par le théorème précédent on a que pour chaque triangle, le cercle circonscrit ne contient pas de sommet de P ; on a donc la triangulation de Delaunay.

Remarque : Si il y a 4 points cocycliques dans P , le dual ne contient pas forcément que des triangles et n'est donc pas une triangulation.

Conclusion : On a donc un algorithme pour calculer le diagramme de Voronoï en 2 étapes :

- Calculer la triangulation de Delaunay \rightarrow en $O(n \log(n))$
- Calculer le dual en $O(n)$

Toutefois il existe des algorithmes permettant de calculer directement en $O(n \log(n))$ le diagramme de Voronoï (ex : algorithme de Fortune).

5 Triangulation de polygone

Problématique : Comment découper un polygone en triangles, est-ce toujours possible ? (oui).

Définition 5.1 Une diagonale d'un polygone $P = (P_1, P_2, \dots, P_n)$ est un segment reliant 2 points de P non consécutif et inclus dans l'intérieur de P .

Lemme 5.1 Tout polygone qui n'est pas un triangle admet une diagonale.

Preuve : On trouve A le sommet le plus à gauche de P et note B et C ses voisins. Si $[BC]$ est à l'intérieur de P , on a trouvé une diagonale, sinon on balaye ABC par un faisceau partant de A parallèle à (BC) . On note Z le premier sommet de P rencontré. A cet instant on note I et J les intersections du faisceau avec $[AB]$ et $[AC]$. Le triangle AIJ est inclus dans l'intérieur de P donc AZ est une diagonale de P .

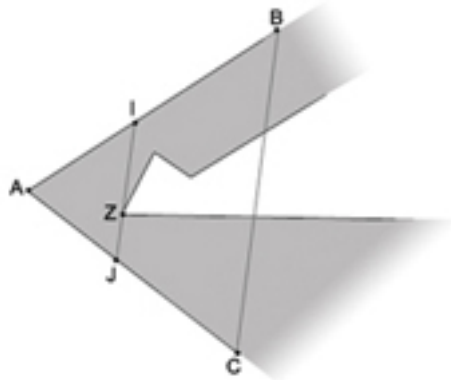


FIG. 14 – Illustration de la preuve

Corollaire : Tout polygone est triangulable et on a un algorithme en $O(n^2)$

Preuve : Une diagonale se trouve en $O(n)$, et elle sépare le polygone en 2 polygones à au plus $n-1$ sommets. On continue par récurrence...

Corollaire : Le dual d'une triangulation d'un polygone est un arbre.

Preuve : Une diagonale sépare P en P_1 et P_2 . Par récurrence les duals de P_1 et P_2 sont des arbres. On les relie par une seule arête (frontière formée par la diagonale), on obtient donc un arbre.

Exemple 5.1

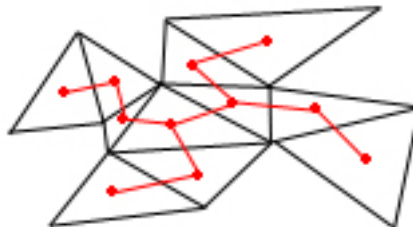


FIG. 15 – Construction du dual d'un polygone