Projet 2 : Triangulation de Delaunay

Antonin Garret, Nathan Thomasset

Novembre 2015

1 Introduction

La création d'une triangulation de Delaunay est un problème qui trouve des applications multiples, notamment dans les méthodes de modélisation et d'affichage. Ce projet étudie une méthode de construction d'une telle triangulation dans le plan, et sa généralisation dans des dimensions supérieures.

2 Triangulation de Delaunay dans le plan

2.1 Introduction

Une triangulation d'un nuage de points est un ensemble de triangles qui couvre l'enveloppe convexe du nuage et dont les sommets coïncident avec les points du nuage.

Une triangulation de Delaunay est une triangulation qui possède une propriété supplémentaire : Le cercle circonscrit de tout triangle ne contient aucun autre point de l'ensemble que les sommets du triangles

Dans cette première partie, l'objectif était de définir un algorithme permettant d'obtenir une triangulation de Delaunay à partir de n'importe quel ensemble de points fourni.

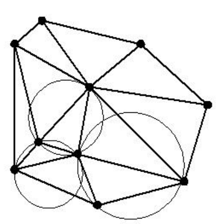

Figure 1 – Un exemple de triangulation de Delaunay

2.2 Description de l'algorithme

L'algorithme que nous avons utilisé pour créer la triangulation est incrémental. Il s'exécute comme suit :

- On crée une triangulation simple dont l'enveloppe convexe contient les points à trianguler
- On ajoute les points un à un en créant à chaque fois une nouvelle triangulation qui contient à la fois les points déjà triangulés et le nouveau point

Pour la première étape, nous avons utilisé les points extrémaux de la fenêtre d'affichage pour créer deux triangle qui forment une triangulation de Delaunay simple. Pour la deuxième étape, la question la plus importante est de savoir comment ajouter un point à la triangulation. Nous allons donc nous intéresser à la méthode utilisée.

2.3 Ajout d'un point à la triangulation

L'ajout d'un point s'effectue en trois étapes :

1. On détecte les triangles dont le cercle circonscrit contient le point à ajouter
2. On calcule le bord de cet ensemble
3. On supprime de la triangulation les triangles détectés et on ajoute les triangles formés, pour chaque arêtes du bord, des deux points de cette arête et du point à ajouter

FIGURE 2 – Étape 1 : Les triangles détectés sont en gras

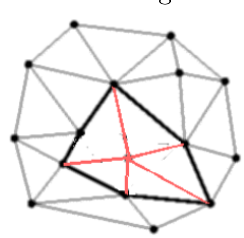

FIGURE 3 – Étape 3 : Les arêtes ajoutées sont en rouge

Pour l'étape 1, on utilise le fait que qu'un point D est dans le cercle circonscrit du triangle (A, B, C) si et seulement si :

$$\begin{vmatrix} A_x & A_y & A_x^2 + A_y^2 & 1 \\ B_x & B_y & B_x^2 + B_y^2 & 1 \\ C_x & C_y & C_x^2 + C_y^2 & 1 \\ D_x & D_y & D_x^2 + D_y^2 & 1 \end{vmatrix} \cdot \begin{vmatrix} B_x - A_x & C_x - A_x \\ B_y - A_y & C_y - A_y \end{vmatrix} > 0$$

On teste tous les triangles en utilisant cette formule, ceux pour lesquels elle est positives étant ceux dont le cercle circonscrit contient le point.

Pour l'étape 2, on remarque que les arêtes du bord sont celles qui appartiennent un unique triangle. Toutes les autres arrêtes appartiennent à exactement deux triangles. Il suffit donc de créer une liste contenant toutes les arêtes et de supprimer celles qui apparaissent deux fois.

En suivant cette méthode, on obtient bien une nouvelle triangulation contenant le point à ajouter

2.4 Exemples graphiques

Voici des images obtenues en appliquant l'algorithme à différents nuages de points, et en affichant la triangulation à l'aide du module graphics de OCaml :

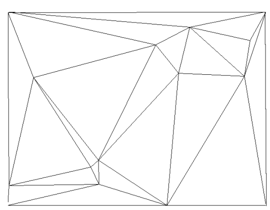

Figure 4 – Une triangulation à partir d'un nuage de 10 points

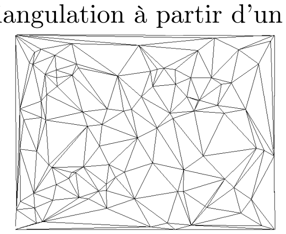

Figure 5 – Une triangulation à partir d'un nuage de 100 points

2.5 Le problème de l'exclusion des points extrémaux

Un inconvénient de l'algorithme utilisé est qu'il ajoute des point à la triangulation qui ne font pas partie de l'ensemble de départ.

Nous avons tenté de résoudre ce problème, mais malheureusement notre méthode n'a pas abouti. Celle-ci consistait à retirer les triangles contenant des points extrémaux et reposait sur la supposition erronée que l'enveloppe convexe du nuage de points faisait partie de la triangulation obtenue avec l'algorithme de base.

Comme ce n'est pas la cas, on obtient une triangulation à laquelle il manque l'enveloppe convexe du nuage, comme on peut le voir sur cette image :

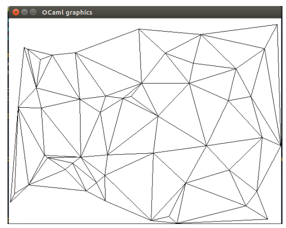

FIGURE 6 – Un exemple de triangulation incomplète

Une manière de corriger ce problème serait de calculer l'enveloppe convexe du nuage pour compléter notre triangulation, mais une fois implémenté le calcul de l'enveloppe convexe, une meilleur méthode serait alors de reprendre notre algorithme de base pour commencer avec une triangulation formée des triangles dont les sommets seraient les points des arrêtes de enveloppe auxquels on aurait ajouté le premier point du nuage.

3 Triangulation multi-dimensionnelle

3.1 Introduction

On cherche maintenant à généraliser cet algorithme à un nuage de points dans un espace de dimension quelconque d : il s'agit de toruver un ensemble de simplexes convrant l'enveloppe convex des points et dont les sommets coïncident avec les points du nuage, tels qu'il n'y ait aucun point du nuage dans l'hémisphère d'un simplex dont il n'est pas sommet.

3.2 Description de l'algorithme

L'algorithme utilisé est très similaire à celui déjà expliqué pour la triangulation en deux dimensions : on part d'une triangulation déjà déterminée et on ajoute un à un les points du nuage, en modifiant la triangulation en conséquence.

3.3 Triangulation de départ

Pour déterminer la triangulation de départ, on utilise un algorithme récursif. En dimension 2, il s'agit de la triangulation de départ déjà présentée. En dimension supérieure, on va trianguler un hypercube centré sur le point 0 contenant tout les points du nuage. Pour se faire, on commence par déterminer une triagulation de ses faces : cette triangulation nous permet d'obtenir une liste des hyperfaces des simplexes de départs. Il ne reste plus qu'à connecter ces hyperfaces au point 0, situé au centre de l'hypercube pour obtenir la triangulation souhaitée.

3.4 Ajout d'un point à la triangulation

Pour ajouter un point à une triangulation, on commence par rechercher les simplexes dont il est dans l'hémisphère. En deux dimensions, on regarder le signe de

$$\begin{vmatrix} A_x & A_y & A_x^2+A_y^2 & 1 \\ B_x & B_y & B_x^2+B_y^2 & 1 \\ C_x & C_y & C_x^2+C_y^2 & 1 \\ D_x & D_y & D_x^2+D_y^2 & 1 \end{vmatrix} \cdot \begin{vmatrix} B_x-A_x & C_x-A_x \\ B_y-A_y & C_y-A_y \end{vmatrix}$$

En effet, en retirant la dernière ligne à toutes les autres puis en effectuant un développement dernière colonne, on trouve :

$$\begin{vmatrix} A_x & A_y & A_x^2+A_y^2 & 1 \\ B_x & B_y & B_x^2+B_y^2 & 1 \\ C_x & C_y & C_x^2+C_y^2 & 1 \\ D_x & D_y & D_x^2+D_y^2 & 1 \end{vmatrix} = \begin{vmatrix} A_x-D_x & A_y-D_y & ||A||^2-||D||^2 \\ B_x-D_x & B_y-D_y & ||B||^2-||D||^2 \\ C_x-D_x & C_y-D_y & ||C||^2-||D||^2 \end{vmatrix}$$

En retirant la première ligne à toutes les autres, on obtient ensuite :

$$\begin{vmatrix} A_x & A_y & A_x^2+A_y^2 & 1 \\ B_x & B_y & B_x^2+B_y^2 & 1 \\ C_x & C_y & C_x^2+C_y^2 & 1 \\ D_x & D_y & D_x^2+D_y^2 & 1 \end{vmatrix} = \begin{vmatrix} A_x-D_x & A_y-D_y & ||A||^2-||D||^2 \\ B_x-A_x & B_y-A_y & ||B||^2-||A||^2 \\ C_x-A_x & C_y-A_y & ||C||^2-||A||^2 \end{vmatrix}$$

On remarque que ce déterminant est invariant par translation : si $A_x, B_x, C_x, D_x = A'_x+\alpha, B'_x+\alpha, C'_x+\alpha, D'_x+\alpha$ et $A_y, B_y, C_y, D_y = A'_y+\beta, B'_y+\beta, C'_y+\beta, D'_y+\beta$ on obtient :

$$\begin{vmatrix} A_x-D_x & A_y-D_y & ||A||^2-||D||^2 \\ B_x-A_x & B_y-A_y & ||B||^2-||A||^2 \\ C_x-A_x & C_y-A_y & ||C||^2-||A||^2 \end{vmatrix} = \begin{vmatrix} A'_x-D'_x & A'_y-D'_y & ||A'||^2-||D'||^2 \\ B'_x-A'_x & B'_y-A'_y & ||B'||^2-||A'||^2 \\ C'_x-A'_x & C'_y-A'_y & ||C'||^2-||A'||^2 \end{vmatrix}$$

En remplaçant les coordonnées des points par leurs coordonnées par rapport au point O, centre du cercle circonscrit au triangle, dans le déterminant précédent on obtient donc

$$\begin{vmatrix} A_x & A_y & A_x^2+A_y^2 & 1 \\ B_x & B_y & B_x^2+B_y^2 & 1 \\ C_x & C_y & C_x^2+C_y^2 & 1 \\ D_x & D_y & D_x^2+D_y^2 & 1 \end{vmatrix} = \begin{vmatrix} A_x-D_x & A_y-D_y & OA^2-OD^2 \\ B_x-A_x & B_y-A_y & OB^2-OA^2=0 \\ C_x-A_x & C_y-A_y & OC^2-OA^2=0 \end{vmatrix}$$

D'où

$$\begin{vmatrix} A_x & A_y & A_x^2+A_y^2 & 1 \\ B_x & B_y & B_x^2+B_y^2 & 1 \\ C_x & C_y & C_x^2+C_y^2 & 1 \\ D_x & D_y & D_x^2+D_y^2 & 1 \end{vmatrix} \cdot \begin{vmatrix} B_x-A_x & C_x-A_x \\ B_y-A_y & C_y-A_y \end{vmatrix} = (OA^2-OD^2) \begin{vmatrix} B_x-A_x & C_x-A_x \\ B_y-A_y & C_y-A_y \end{vmatrix}^2$$

par développement dernière colonne. Dans le cas général, il suffit de tester le signe du déterminant de la même matrice à laquelle on rajoute les lignes et colonnes contenant les coordonnées et points du simplexe supplémentaire, et vérifier qu'il est du même signe que $(-1)^d$, un facteur $(-1)^{d+2}$ apparaissant dans le développement dernière colonne mentionné. S'ils sont du même signe, alors le point à ajouter est dans le simplexe.
Une fois déterminés les simplexes à retirer, on détermine les hyperfaces formant la frontière, que l'on connecte au point à ajouter pour former les nouveaux simplexes.

4 Conclusion

On dispose donc d'un algorithme permettant de déterminer la triangulation d'un nuage de points en dimension 2 tout d'abord, puis en dimension quelconque, toutefois quelques points noirs sont à signaler :

— Le calcul du déterminant n'est pas du tout optimisé : on applique un algorithme récursif naïf de développement première colonne de complexité $\mathcal{O}(d!)$; l'implémentation d'un pivot de Gauss serait préférable.

— La triangulation de départ ajoute de nombreux points, deux solutions sont envisageables : en dimension 2, partir comme proposé précédemment de l'enveloppe convexe des points ; en dimension supérieure on pourrait au lieu de notre algorithme récursif partir d'un simplexe contenant tous les points du nuage (au lieu de l'hypercube et de tous les points que l'on ajoute par application de notre algorithme récursif)