

RAPPORT DE PROJET 2014/2015

Triangulation de Delaunay

Aurélien Argoud - Joël Randrianandrasana

Master 2 Informatique - Spécialité Image

Table des matières

1	Introduction	2
2	Les sommets	2
2.1	Description de la structure vertex	2
2.2	Génération aléatoire de sommets	3
3	Les simplexes	3
3.1	Description de la structure triangle	3
3.2	Indexation des sommets et des voisins d'un triangle	4
3.3	Candidats : Affectation à leurs triangles hôtes	4
3.4	Candidats : Distance verticale d'un sommet à son triangle hôte	6
4	Algorithme de Delaunay	6
4.1	Affinage du maillage en plusieurs itérations	6
4.2	Triangulation initiale	7
4.3	Traitement du triangle de priorité maximum : Division d'un triangle en trois triangles	8
4.4	Maintien de la propriété de Delaunay : Correction du maillage	9
5	Résultats et performances	11
6	Conclusion	12

1 Introduction

La triangulation de Delaunay est un algorithme de maillage de forme géométrique visant à générer un maillage homogène à partir d'un ensemble de sommets en entrée. Les maillages résultant de cet algorithme sont appréciables visuellement et permettent de limiter les erreurs de calcul liées à l'interpolation de valeurs dans une triangulation. De plus, l'algorithme permet de générer des maillages adaptés à des niveaux de détails, qualité appréciable en terme d'occupation mémoire ou encore en terme de performances graphiques.

Ce rapport présente les structures de données utilisées dans le cadre de l'implémentation de ce projet, les opérations liées à celles-ci ainsi qu'une description de l'algorithme central.

2 Les sommets

2.1 Description de la structure vertex

Les sommets utilisés pour la génération du maillage du terrain sont représentés grâce à la structure vertex. Cette structure encapsule les informations suivantes :

- Les 3 composantes définissant les coordonnées cartésiennes du sommet.
- Les différentes relations de navigation du sommet vers d'autres sommets, navigables dans les deux sens (avant et arrière) :
 - Relation de navigation arbitraire.
 - Relation par ordre polaire.
 - Relation lexicographique.
 - Relation de navigation par ordre de « candidats » d'un même triangle hôte.

2.2 Génération aléatoire de sommets

L'application prend en charge la génération aléatoire de n sommets dont 4 sommets correspondent aux sommets du carré unité. Les abscisses et ordonnées des $n-4$ sommets supplémentaires appartiennent à l'intervalle $]0,1[$. Les altitudes de tous les sommets sont générées entre 0 et une altitude maximum paramétrable par l'utilisateur via l'option `-a` de la ligne de commande.

Les coordonnées des sommets sont générées aléatoirement avec une double précision selon une loi de probabilité uniforme sans vérification de l'existence d'un sommet possédant déjà les mêmes abscisses et ordonnées (contrainte de 2,5 D). La double précision nous permet d'éviter ce test de complexité minimale $O(n \log(n))$ et ainsi d'obtenir une génération de sommets en $O(n)$.

3 Les simplexes

Les simplexes définissant les faces générées du terrain sont décrites sous la forme de triangles.

3.1 Description de la structure triangle

Chaque triangle est défini comme suit :

- Les trois sommets du triangle listés dans l'ordre trigonométrique.
- Ses trois triangles adjacents, indexés de manière opposée aux sommets du triangle. Ceux-ci sont dénommés « voisins » du triangle et peuvent être nuls.
- La liste des sommets n'appartenant à aucune face courante du maillage et se projetant verticalement sur ce triangle, avec en tête de liste, le sommet le plus distant verticalement du triangle. Ces sommets sont dénommés « sommets candidats du triangle » ou encore « candidats » dans le présent document.

- La distance maximum séparant le triangle de ses sommets candidats.

3.2 Indexation des sommets et des voisins d'un triangle

Les sommets et triangles adjacents sont indexés de la manière suivante, une absence de triangle adjacent étant indiquée par une valeur nulle à l'indice correspondant :

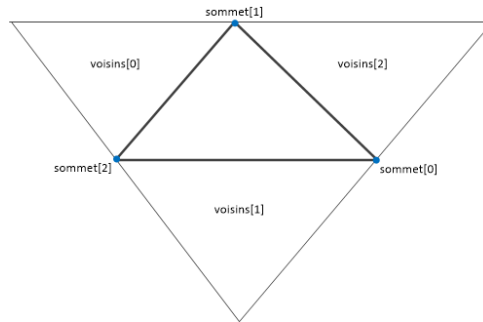


FIGURE 1 – Triangles - Indexation

3.3 Candidats : Affectation à leurs triangles hôtes

Les sommets n'appartenant pas encore aux faces du maillage sont affectés à leurs triangles hôtes respectifs par projection verticale :

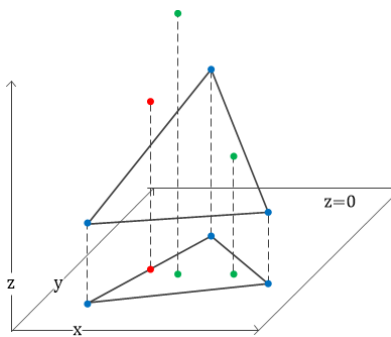


FIGURE 2 – Candidats - Affectation aux triangles hôtes par projection verticale

Comme illustré sur la figure précédente, ce test d'appartenance s'effectue via la projection sur le plan $z=0$. Tout sommet se projetant sur une arête du triangle est ignoré lors des traitements suivants. Dans le plan, un point est considéré à l'intérieur d'un triangle si celui-ci se trouve du même côté de chaque arête :

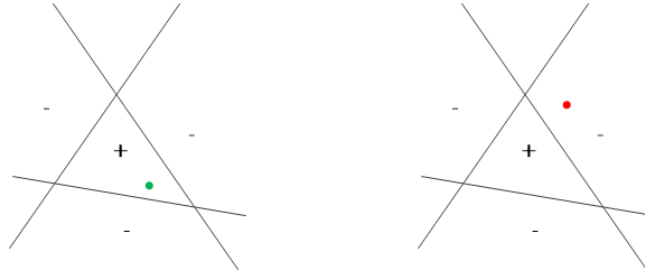


FIGURE 3 – Candidat - Test d'inclusion - Partie 01

Ainsi sur la partie gauche de l'illustration précédente, le point testé appartenant à deux côtés différents de deux arêtes du triangle, celui-ci est considéré à l'extérieur du triangle. La localisation d'un point par rapport à un segment (demi-plan positif, négatif ou encore aligné) s'effectue simplement via un produit vectoriel dans \mathbb{R}^3 (donc un calcul de déterminant).

Dans l'exemple ci-dessous, la composante z positive du produit vectoriel $\vec{AB} \times \vec{AP}$ indique un point P situé dans le demi-plan positif du segment $[AB]$ (inversement pour le point Q) :

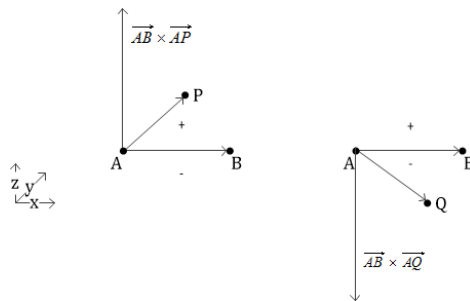


FIGURE 4 – Candidat - Test d'inclusion - Partie 02

3.4 Candidats : Distance verticale d'un sommet à son triangle hôte

Le calcul de la distance verticale d d'un candidat P à son triangle hôte s'effectue en trois étapes :

- Calcul de l'équation du plan Π auquel appartient le triangle
- Calcul du projeté vertical P' du point P sur le plan Π (calcul de P'_z avec $P'_x = P_x$ et $P'_y = P_y$)
- Calcul de la valeur $|P_z - P'_z|$

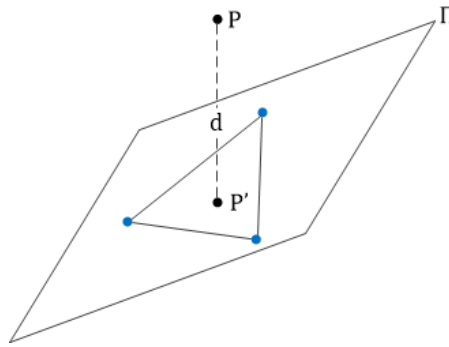


FIGURE 5 – Candidat - Distance verticale au triangle hôte

4 Algorithme de Delaunay

4.1 Affinage du maillage en plusieurs itérations

La triangulation de Delaunay se fait en plusieurs étapes où, à chacune d'elle, le nombre de triangle augmente jusqu'à satisfaction d'un critère d'arrêt. Ci-dessous un maillage après plusieurs itérations successives de Delaunay avec 50 vertex au départ :

A chaque itération, la triangulation comptabilise deux triangles de plus. L'algorithme continue d'itérer tant qu'il reste des sommets à insérer au maillage ou tant qu'un critère d'arrêt spécifié par l'utilisateur n'est pas satisfait.

Les deux triangulations obtenues ci-dessus sont maximales : tous les sommets ont été insérés aux maillages.

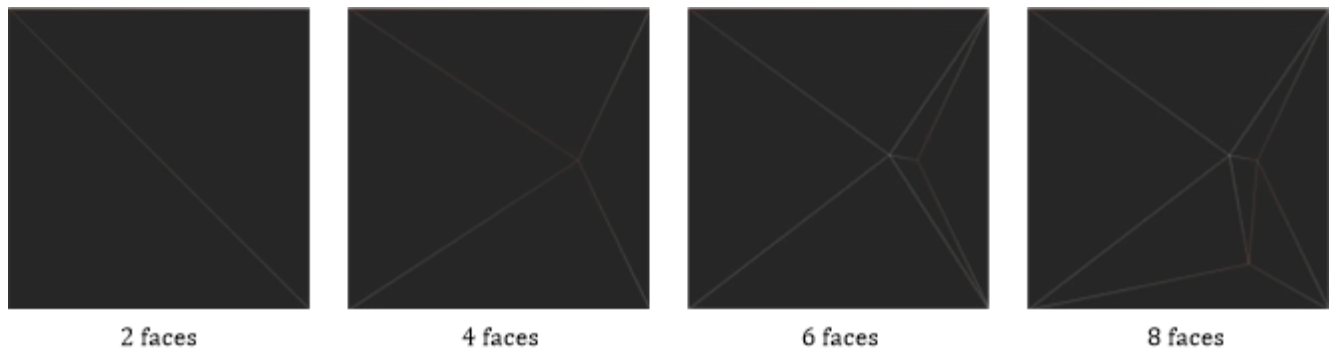


FIGURE 6 – Delaunay - Itérations successives

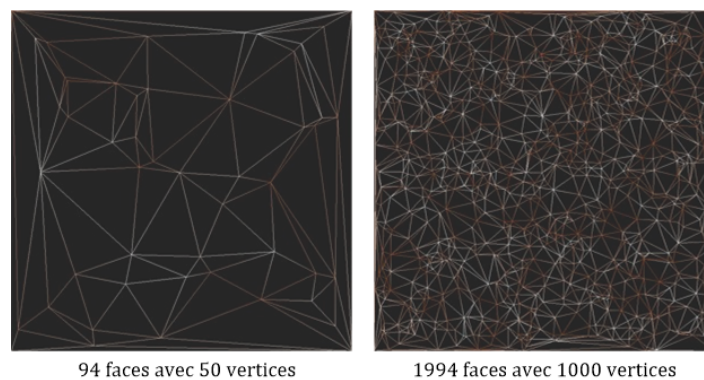


FIGURE 7 – Delaunay - Itérations finales

4.2 Triangulation initiale

Au commencement de notre triangulation, nous avons nos quatre sommets formant un carré. En divisant celui-ci en deux via un segment entre deux sommets opposés, nous obtenons nos deux triangles initiaux :

Une fois cette triangulation initiale obtenue, l'ensemble des sommets restants sont affectés en tant que candidats à l'un de ces deux triangles selon la projection décrite dans la section précédente.

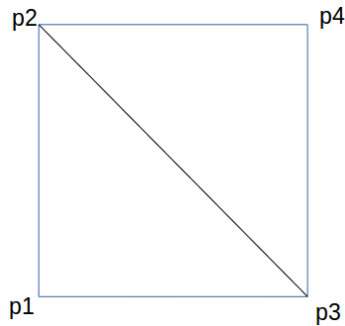


FIGURE 8 – Delaunay - Triangulation initiale

4.3 Traitement du triangle de priorité maximum : Division d'un triangle en trois triangles

La première étape d'une nouvelle itération aboutissant à une nouvelle triangulation, est de subdiviser le triangle de priorité maximum, donc possédant le candidat le plus éloigné de son triangle, en trois nouveaux triangles. Pour commencer, on supprime le triangle que l'on va modifier de la file de priorité. Puis, on sort les candidats de ce triangle pour pouvoir les partager sur trois nouveaux triangles. Le candidat le plus éloigné est mis à part pour former ces trois triangles. Pour les créer, chaque arête du triangle initial est relié à notre candidat mis à part. Cela donne donc nos trois triangles. Afin d'optimiser le traitement, seuls deux nouveaux triangles sont créés (modification du triangle existant pour le troisième nouveau triangle). Les candidats de l'ancien triangle sont ensuite dispatchés par projections verticales sur les trois nouveaux triangles obtenus.

Les triangles adjacents au triangle initial doivent également être maintenus de manière à tenir compte de la nouvelle triangulation (mise à jour de leurs relations de voisinage). De plus, la propriété de listing trigonométrique des sommets des nouveaux triangles doit également être maintenue.

Ces trois triangles sont ensuite insérés dans la file de priorité. Ceux-ci remettre en cause la propriété de Delaunay, ces trois nouveaux triangles sont également insérés dans la pile dédiée à la correction du maillage décrite dans la section suivante.

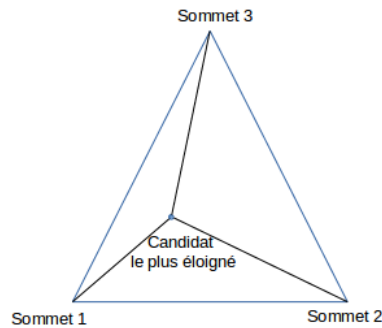


FIGURE 9 – Delaunay - Subdivision du triangle de priorité maximum

4.4 Maintien de la propriété de Delaunay : Correction du maillage

À chaque itération, les trois nouveaux triangles créés sont insérés dans la pile de Delaunay. Cependant, lors d'une itération, de nouveaux triangles peuvent être empilés à chaque correction (de nouveaux triangles susceptibles de remettre en cause la propriété de Delaunay du maillage). La pile est donc initialisée selon une taille fixe de manière à pouvoir accueillir, dans le pire des cas, l'ensemble des triangles courants de la triangulation soit d'une capacité de $2n - 6$ éléments ou d'une capacité égale au nombre de faces maximum indiqué par l'utilisateur en ligne de commande.

Chaque triangle de la pile est dépilé et traité tant que celle-ci n'est pas vide. Pour corriger un triangle, on regarde d'abord le point qui était précédemment un candidat. Grâce à cela, on peut savoir le triangle voisin à l'opposé de ce sommet, triangle avec lequel on peut corriger la triangulation.

Deux cas de figure :

- Dans le premier cas extrême, Le triangle est sur le bord de la triangulation et n'a pas de voisin de ce côté donc on passe au prochain triangle dans la pile.
- Dans les autres cas, on recherche le sommet opposé à notre triangle sur le triangle voisin.
 - Ce sommet opposé est en dehors du cercle circonscrit du premier triangle : aucune correction n'est nécessaire : on passe au prochain triangle dans la pile.
 - Enfin, si le sommet opposé est dans le cercle circonscrit du premier triangle,

on corrige la triangulation de manière à éviter ce cas de figure (sommet dans le cercle circonscrit d'un triangle).

Pour cela, on enlève l'arête qui sépare les deux triangles, puis on rajoute une arête dans l'autre sens, c'est de notre point de départ à notre côté opposé sur le voisin. Cependant, d'un point de vue algorithmique, cela peut demander un grand nombre de vérification de condition pour associer les bons sommets dans le bon triangle et bien déterminer quel voisin va avec quel sommet opposé.

Pour simplifier ça et supprimer les conditions inutiles, on fait tourner les indices des sommets et voisins des deux triangles pour avoir les deux cotés opposés à l'indice 1. Ainsi, avec très peu de code et aucune vérification de condition, on peut facilement déterminer quel triangle voisin va avec quel sommet.

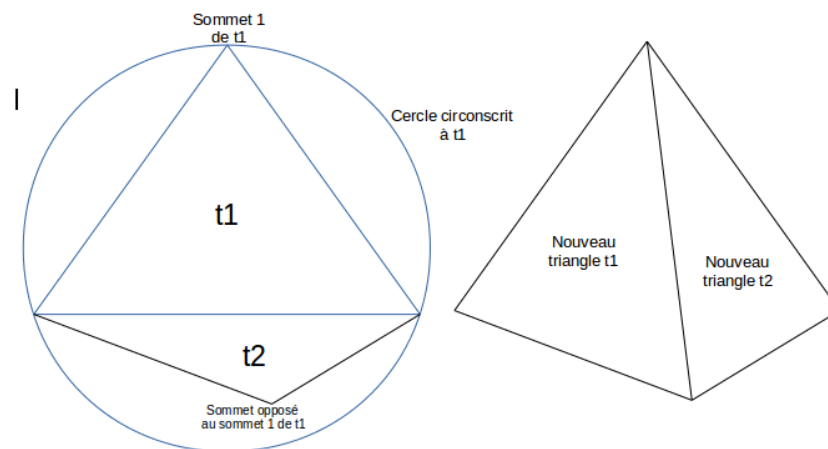


FIGURE 10 – Delaunay - Correction

Les deux triangles ont été modifié. Leurs candidats respectifs sont donc re-dispatchés de manière adéquate sur les deux nouveaux triangles obtenus. Une fois cette opération effectuée, la priorité des deux triangles dans la file est mise à jour (élévation ou enfoncement) en faisant attention aux cas de conflits pouvant survenir lors de cette mise à jour. Pour contourner ce problème, on met à jour une seconde fois la position du premier triangle qui a été mise à jour. De plus, les deux triangles voisins ayant vu leurs voisins passer de t1 à t2 et inversement, sont également mis à jour. Enfin, on ajoute ces deux triangles à la pile.

On effectue cette opération jusqu'à ce que la pile soit vide et donc qu'il n'y ait plus de triangle à corriger.

5 Résultats et performances

Ci-dessous, des exemples de triangulations obtenues (intensité lumineuse du terrain linéairement proportionnelle à l'altitude des sommets) :

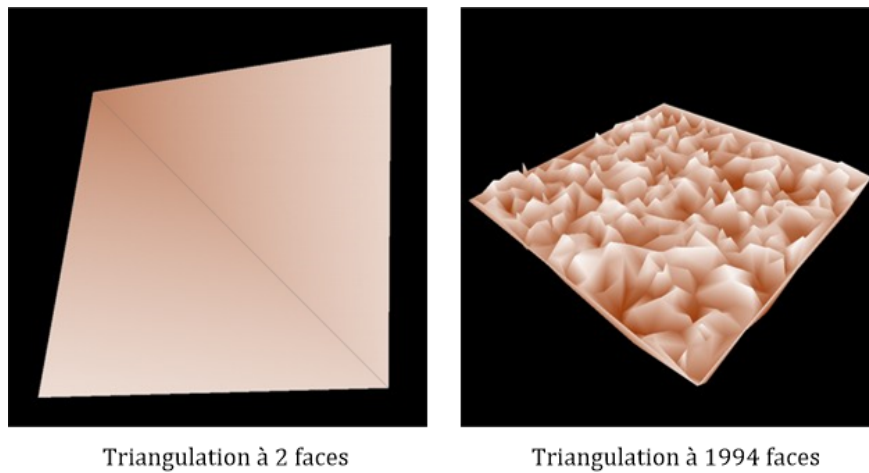


FIGURE 11 – Delaunay - Exemples de triangulations obtenues

De plus, les conditions d'arrêt suivantes sont disponibles :

- La génération d'un nombre maximum de faces paramétrable par l'utilisateur
- L'arrêt en fonction d'un goodness-of-fit paramétrable par l'utilisateur
- Un affichage en 2D ou en 3D avec gestion de la caméra
- La mise en évidence des triangles dans l'affichage 3D
- Le masquage des faces sur la bordure de la triangulation

L'algorithme génère une triangulation de Delaunay en un temps de complexité $O(n \log(n))$:

Les parties de l'algorithme occupant le plus de ressources résident dans :

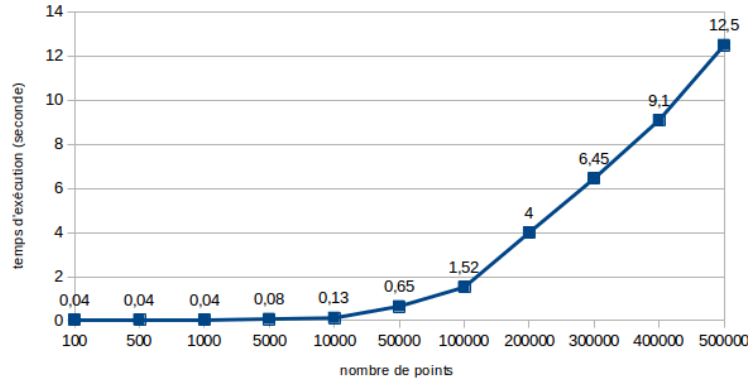


FIGURE 12 – Résultats - Complexité

- L'affectation des candidats à leurs triangles hôtes : Ce traitement en $O(n)$, avec n le nombre de candidats à dispatcher, décroît au fil des itérations car le nombre de candidats par triangle diminue. Cependant ce traitement étant appelé de manière massive dans l'algorithme impacte significativement les performances générales de l'algorithme.
- Les opérations sur la file de priorité en $O(\log(n))$, chaque itération de Delaunay pouvant engendrer plusieurs opérations celle-ci.

Optimisation possible

Le calcul et le stockage de la normal des triangles permettrait d'optimiser les performances de l'application. Les normales permettent de déterminer les équations des plans des triangles, pour calculer la distance verticale entre un triangle et ses candidats respectifs. La connaissance de cette équation est obligatoire pour ce calcul et n'aurait pu être déterminée qu'une seule fois (à la construction ou la modification du triangle).

6 Conclusion

L'algorithme de génération de maillage via Delaunay peut concurrencer les algorithmes standards, en assurant des performances correctes ainsi que l'obtention d'un maillage final aux propriétés appréciables :

- Maillage d'allures visuelles correctes et homogènes.
- Minimisant les problèmes liés aux erreurs d'interpolation, les faces obtenues limitant les difformités de triangulation.