Reconstruction 3D
Modélisations de surfaces
Triangulation de Delaunay

TIPE 2010 – Adrien EUDES

Problématique du projet

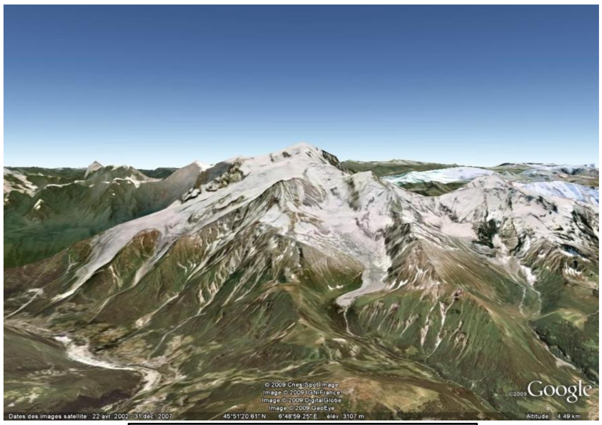
- Dans de nombreux domaines techniques, la numérisation de surfaces est devenue obligatoire. Prenons les exemples des jeux vidéos, de l'imagerie médicale, ou de la reconstruction de paysages
- Par quels procédés ces « images » sont-elles créées par l'ordinateur? Quelles sont les critères pour que le résultat soit le plus précis possible?

Les jeux vidéos sont de plus en plus réalistes

L'imagerie médicale offre la possibilité d'examiner l'intérieur d'un patient sans l'opérer

Modélisation d'un paysage: mode d'emploi

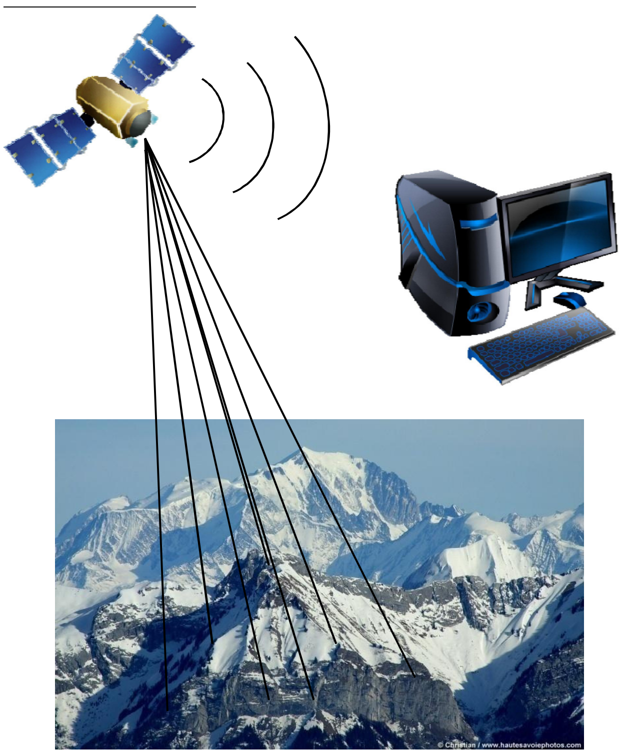
Photo du Mt-Blanc (Chamonix)



Le Mt-Blanc par Google Earth ©

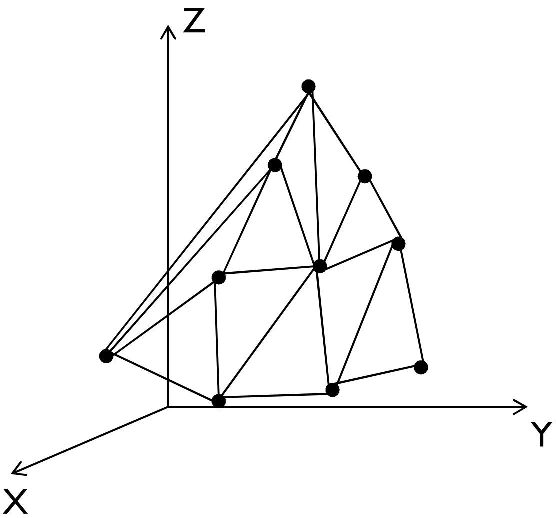
Acquisition d'un nuage de point

- Grâce aux satellites et aux photos aériennes, il est aujourd'hui possible
  - D'avoir des clichés de toutes la surface du globe
  - De connaitre la position exacte d'un point considéré (grâce à la superposition de plusieurs prises)
- On peut créer un nuage de points du paysage à modéliser



Exploitation du nuage de points: construction d'un maillage

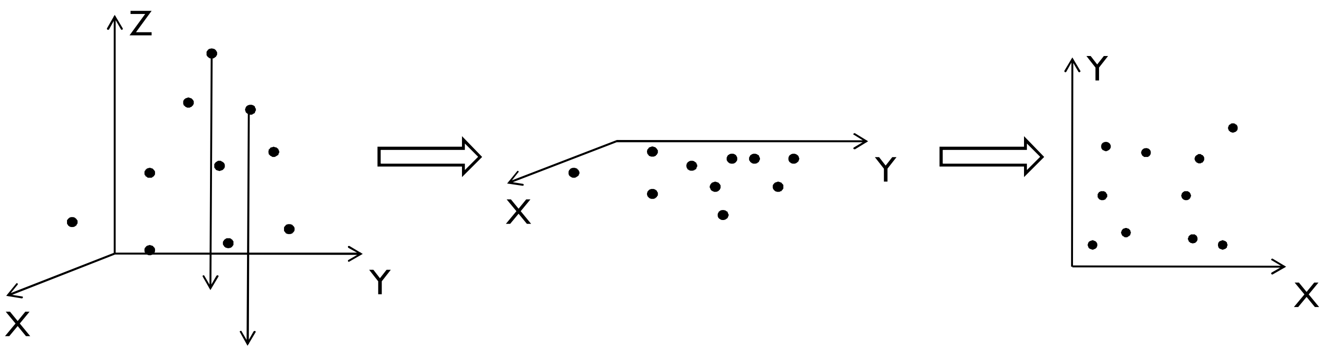
- Le maillage permet de créer une surface du nuage de points
- Le maillage ne doit pas être aléatoire
- On doit trouver un moyen de créer un maillage *unique* et *total*
- Le maillage doit rendre compte de la surface que l'on modélise



Maillage d'un nuage de point de manière aléatoire

Passage par la 2D pour mailler le nuage

- Afin de simplifier le maillage, comme la surface considérée (celle d'un paysage) représente une nappe surfacique, on effectue ce dernier en dimension 2.



Du diagramme de Voronoï à la triangulation de Delaunay

- Pour effectuer le maillage, il a été décidé dans ce projet de se servir de la *triangulation de Delaunay*.
- La triangulation de Delaunay est directement relié au diagramme de Voronoï, elle est sont dual.

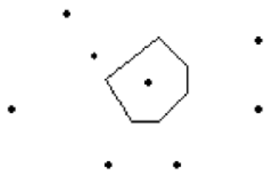
Boris Nikolaïevitch Delaunay, Mathématicien russe du XXè siècle

Georgi Fedoseevich Voronoï, Mathématicien russe du XIXè siècle

Diagramme de Voronoï

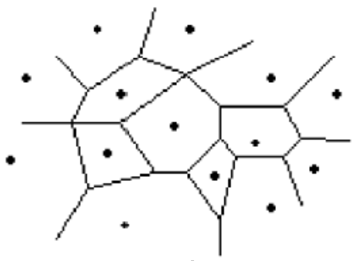
- Le diagramme de Voronoï permet de diviser une surface en polynômes convexes.
- On appelle polygone de Voronoï associé au site Pi de la région Vor(Pi) tel que chaque point P à pour plus proche site Pi. (*d* représente la distance euclidienne)

$$Vor_P(P_i) = \left\{x \in R^2, d(x, P_i) \leq d\left(x, P_j\right), \forall j \in P - P_i\right\}$$

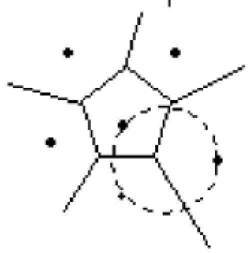


- On décrit le diagramme de Voronoï comme l'union des régions de Voronoï de tous les points.

$$Vor_N(P) = \bigcup_{p \in P} Vor_p(P)$$

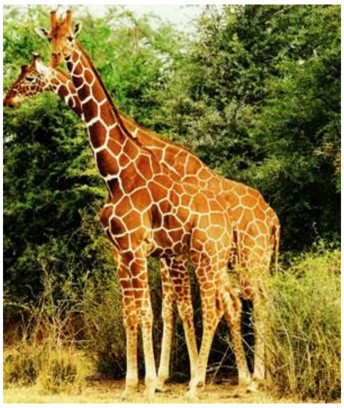


- Propriétés:
  - Pour chaque sommet S du diagramme de Voronoï, le cercle passant par les trois points voisins à ce sommet, ne contient aucun autre point de P. (voir les preuves)
  - Une arrête de Voronoï sépare tout point de son plus proche voisin.



Suppléments au diagramme de Voronoï

- On retrouve des diagrammes de Voronoï dans la nature. Notamment sur le cou des girafes réticulées et sur les tortues.
- Les diagrammes de Voronoï expliqués en dimension 2 peuvent se généraliser en dimension n.



Dual du diagramme de Voronoï

- Théorème:

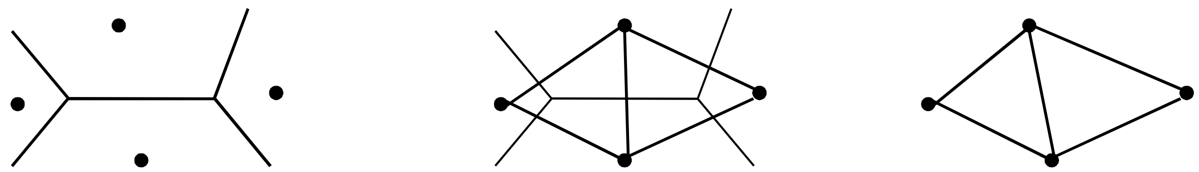
  Le dual du diagramme de Voronoï est une triangulation sur l'ensemble des points.

  - Ce théorème est démontré en vérifiant que ce dual définit une partition du domaine intérieur à l'enveloppe convexe de l'ensemble des points. En ayant remarqué de manière préliminaire qu'à chaque sommet de Voronoï correspondait un triangle du dual, on vérifie pour cela que
    - Un triangle intersectant l'intérieur de l'enveloppe convexe n'est pas plat.
    - Pour une paire de sommet de Voronoï donnée, si les cercles associés à ces deux sommets ont une intersection sachant qu'ils ne peuvent être l'un à l'intérieur de l'autre, alors les deux points d'intersections qu'ils définissent forment un segment séparant les deux triangles correspondants.

Dual du diagramme de Voronoï (suite)

- Chaque point intérieur à l'enveloppe convexe est aussi contenu dans au moins un triangle.
- On définit la *triangulation de Delaunay* d'un ensemble de points du plan comme étant la dual du diagramme de Voronoï correspondant.

- On peut, par dualité et non colinéarité de tous les points et non cocyclicité de quatre points, déduire des résultats portant sur le diagramme de Voronoï les propriétés suivantes:
  - La triangulation de Delaunay est <u>unique</u>.
  - Elle est <u>complète</u>.
  - **Les cercles passant par les trois sommets de chaque triangle ne contiennent aucun autre site en leur intérieur.**

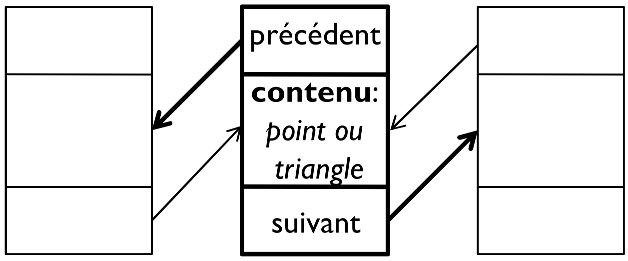


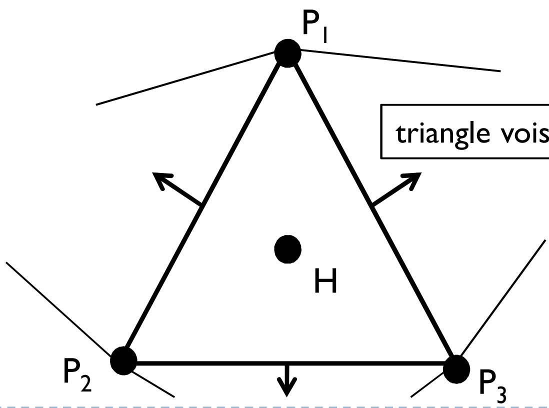
Programmation de la triangulation de Delaunay

- L'objectif du programme écrit dans le cadre du projet et de prendre en entrée une liste de points et de rendre en sortie une liste de triangles. Ces triangles représentent la surface représentée par les points.
- La méthode employée est itérative, elle place les points d'une liste les un après les autres et s'assure qu'après chaque insertion la triangulation est toujours de Delaunay.
- Le programme est écrit en langage *Pascal* et compilé avec Gnome Pascal (GPC).

Structures des données utilisées

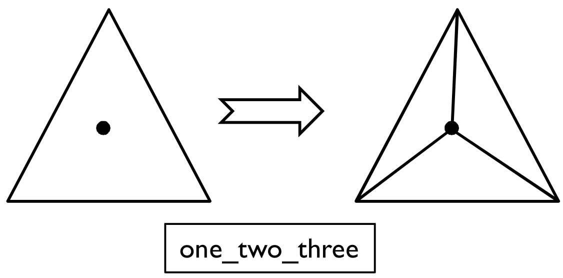
- Listes doublement chaînées: la cellule contient un élément (point ou triangle) et pointe sur les éléments qui la précède et la succède dans la liste.
- Triangles: un triangle est composé de trois sommets (liste de points) de trois voisins (liste de triangles) et d'un point représentant le centre de son cercle circonscrit.

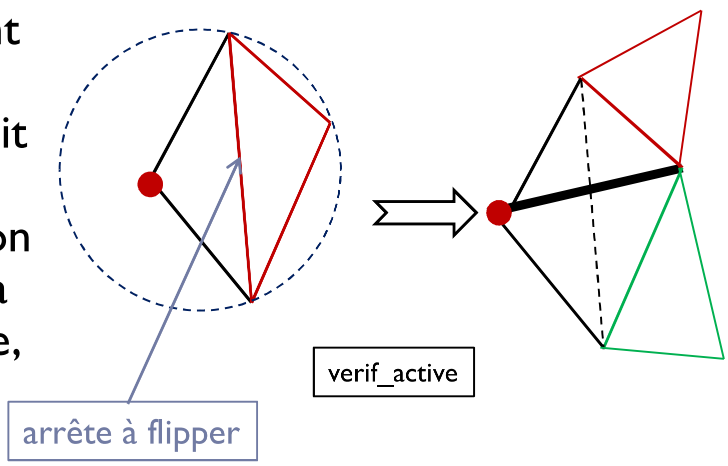




Deux procédures importantes

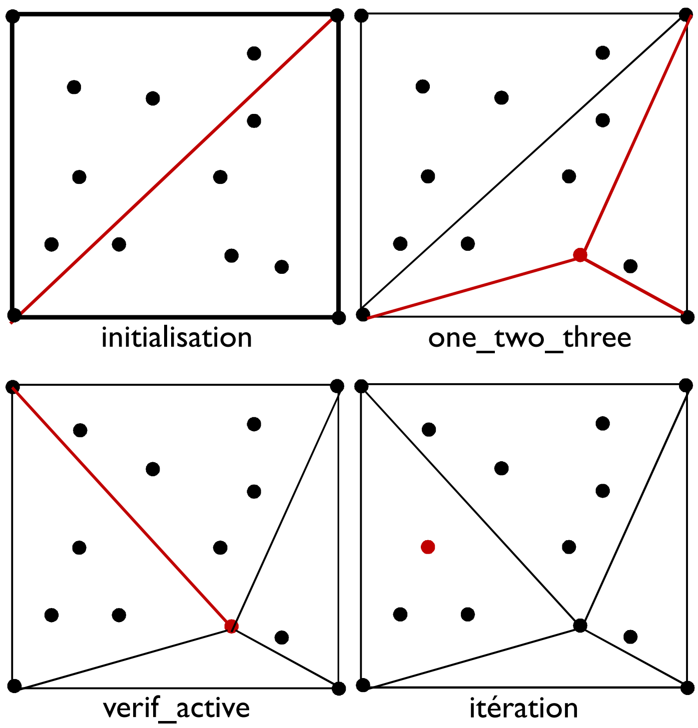
- *one_two_three*: à partir d'un triangle et d'un point à l'intérieur de ce triangle, elle crée trois plus petits triangles.
- *verif_active:* à partir d'un triangle et d'un point sommet d'un triangle voisin, si le point est à l'intérieur du cercle circonscrit au triangle, elle fait « flipper» une arête. Elle applique ensuite la vérification aux voisins plus éloignés. A la fin du passage de verif_active, la triangulation est dite de Delaunay.





Algorithme

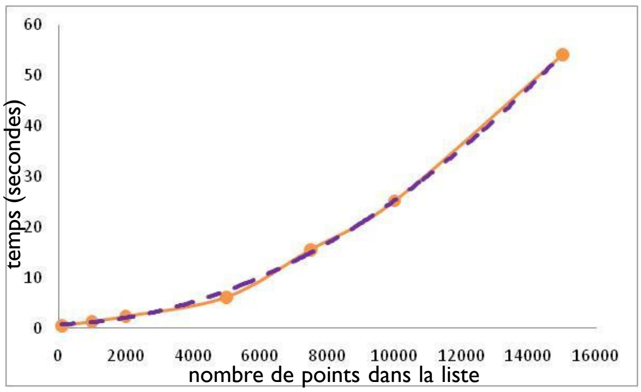
- On initialise le processus en créant un rectangle englobant tous les points et qui définit les deux premiers triangles.
- On place ensuite les points les uns après les autres:
  - on trouve dans quel triangle est contenu le nouveau point
  - on effectue one_two_three.
  - on effectue verif_active.
- On supprime enfin la boite englobante et les triangles du contour pour obtenir un polygone convexe.



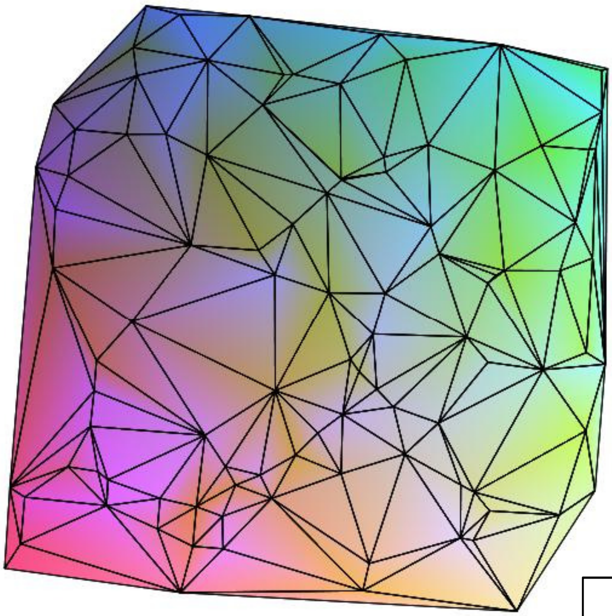
Test du programme

- Les premiers tests ont été effectués sur des listes de points à peu d'éléments, un cas relativement simple afin de corriger les bugs du programme et de vérifier que la triangulation était bien celle de Delaunay.
- Des tests ont ensuite pu être réalisés sur des listes de points rentrés aléatoirement.
- *Evaluation des performances*:
  - On s'attendait à trouver un temps en $O(n^{3/2})$ grâce au guidage lors de la recherche du triangle contenant le point inséré. Les résultats présentent un temps en $O(n^2)$

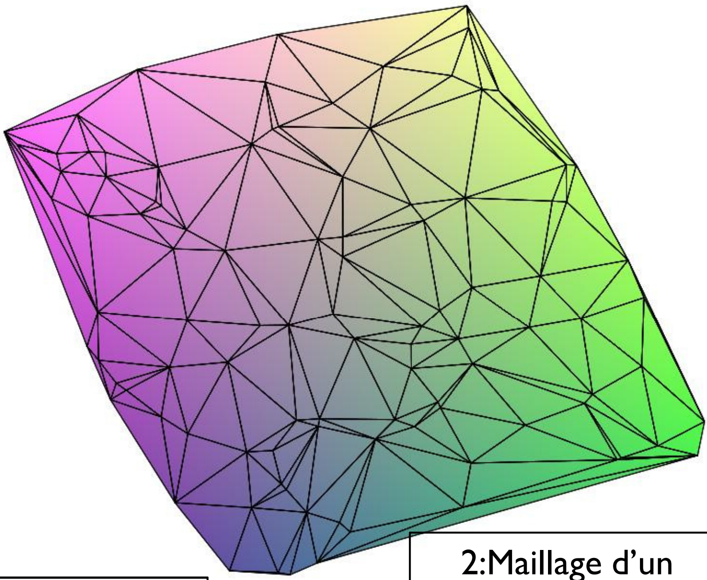
- 100 points→0,50s
- 1000 points→1,38s
- 2000 points→2,38s
- 5000 points→6,19s
- 7500 points→15,56s
- 10000 points→25,18s
- 15000 points→54,06s



Exemples



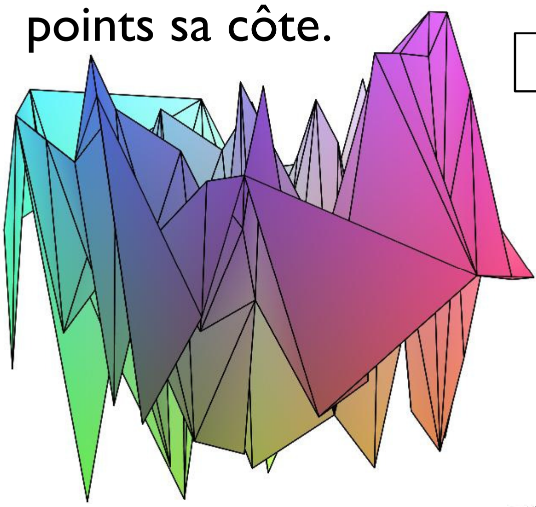
1:Maillage d'un nuage de 100 points aléatoires



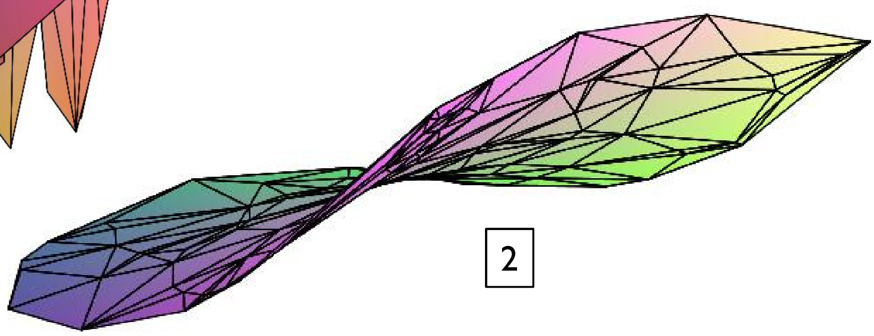
2:Maillage d'un nuage de 100 points d'un paraboloïde hyperbolique

Retour à la 3D

- Une fois le maillage effectué, on peut rendre à chacun des points sa côte.



1



2

Plus ou moins de points

- Un nouveau test a été effectué pour tester le nombre de points qui sont nécessaires pour obtenir un résultat satisfaisant.

Conclusion

- Ce projet permet de transformer un nuage de points en une surface.
- Les points sont relevés à partir de satellites ou de photos aériennes. Ils sont ensuite traiter par l'algorithme de Delaunay.
- On remarque que le rendu est assez bon pour le paraboloïde hyperbolique.
- Bien sûr, plus le nombre de points relevé est élevé, plus le rendu est précis. Toutefois la triangulation choisie permet d'obtenir un résultat tout à fait satisfaisant à partir de peu de points, et cela dans un temps tout à fait raisonnable.