C++ / Rendu de Modèle Numérique de Terrain

Robotique/UE 3.1 - Projet - Décembre 2020

Supports de cours disponibles sur
`www.simon-rohou.fr/cours/c++`

L'objectif de ce projet est de gagner en autonomie et de proposer une solution pour la génération d'une image 2D représentant, par niveaux de couleurs, les différentes altitudes d'un terrain 2.5D (voir par exemple la Figure 1). Un Modèle Numérique de Terrain (MNT) du lac de Guerlédan vous est fourni pour ce projet.

Modèle numérique de terrain

Le modèle numérique de terrain est dit en « 2.5 dimensions » puisqu'il associe une seule altitude à une position horizontale donnée. Il ne s'agit donc pas de véritable 3D.

Le fichier qui vous est fourni rassemble les données en ASCII (format lisible). Par exemple :

```
48.29762887 -004.41737340 14.334
48.29762971 -004.41735997 14.379
48.29763698 -004.41738809 14.452
...
```

Les deux premiers champs correspondent aux coordonnées géographiques (respectivement la latitude et la longitude du point d'élévation). Le troisième champ est une altitude en mètres par rapport à une référence donnée (par exemple, le *zéro hydrographique*).

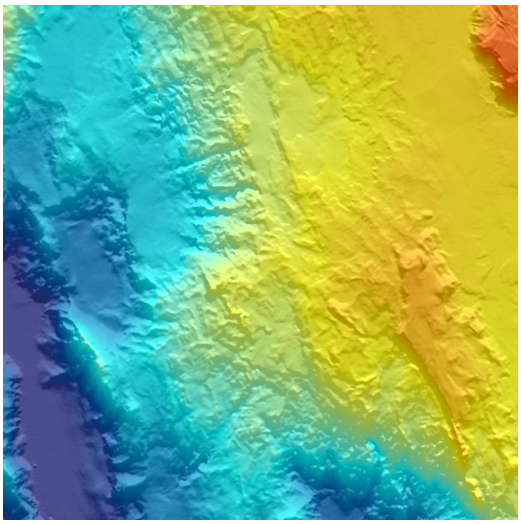

FIGURE 1 – Rendu d'un terrain sous-marin.

Les coordonnées géographiques (WGS84) sont faites pour positionner un point sur Terre ; elles ne correspondent donc pas à un référentiel cartésien comme attendu dans notre image. Il est préférable d'utiliser une projection adaptée à la situation. Nous pourrons par exemple travailler en Lambert93. Il existe des algorithmes pour convertir une position géographique $(long, lat)$ en coordonnées Lambert93 (x, y). Il est conseillé d'utiliser une bibliothèque externe réalisant cette projection pour vous, comme par exemple Proj.

— construisez votre projet avec CMake pour faciliter l'intégration de bibliothèques extérieures ;
— utilisez pour cela la commande CMake `target_link_libraries()` ;
— vous êtes libres de choisir une projection qui vous semble adaptée (UTM, Lambert93, ...).

Maillage

Un rendu de MNT est une image pour laquelle chaque pixel correspond à une zone carrée dans la projection choisie. La génération de cette image consiste à calculer, pour chaque pixel, une couleur synthétisant les élévations de la zone carrée correspondantes du MNT. Par exemple, les pixels clairs représentent une haute altitude et les pixels foncés le fond d'une vallée.

Les points d'un MNT peuvent former une grille régulière. Dans ce cas, l'association entre les pixels de l'image et les points du fichier MNT peut être simple. Lorsque la grille n'est pas régulière, il est préférable de réaliser un maillage à l'aide, par exemple, d'un algorithme de triangulation de Delaunay.

— notez que le passage d'une projection à l'autre déforme potentiellement la régularité d'une grille.

— La Figure 2 donne une indication de la procédure à suivre pour déterminer chaque pixel de l'image, avec ou sans maillage. La méthode avec maillage (suite à une triangulation de Delaunay, par exemple) offrira de meilleurs résultats.

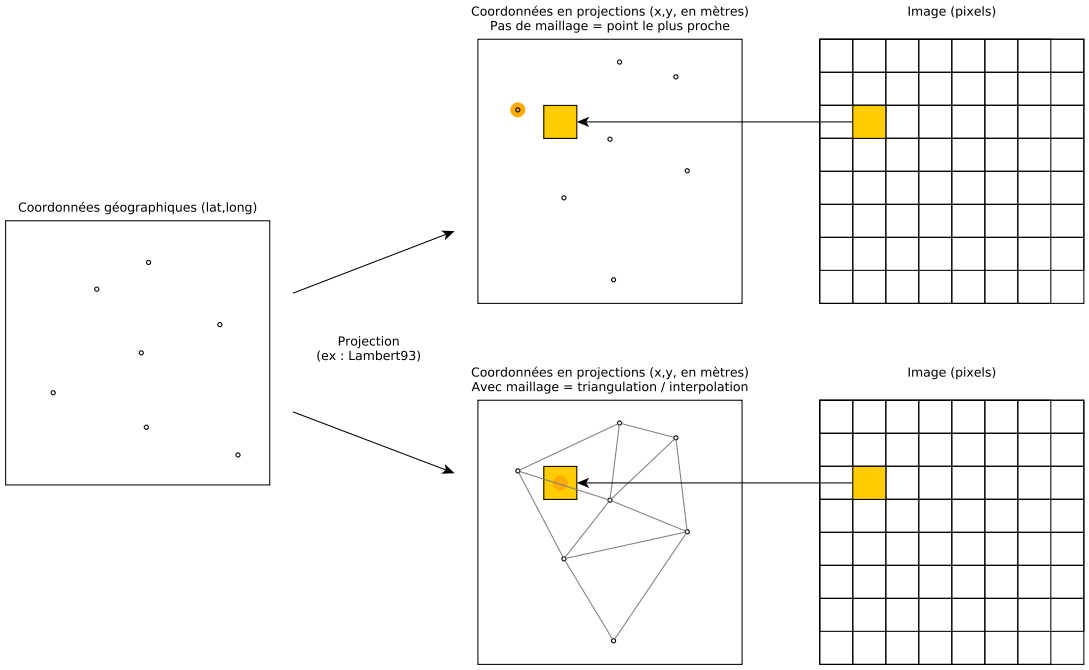

FIGURE 2 – Passage de coordonnées géographiques aux pixels d'une image. La fonction recherchant un point projeté à partir des coordonnées d'un pixel de l'image peut être longue à l'exécution. On pourrait réfléchir à une optimisation permettant d'éviter de parcourir tous les points projetés pour trouver le point correspondant à notre pixel (sans optimisation : complexité en $\mathcal{O}(n)$).

Format d'image

Le rendu sera généré par votre exécutable avec l'écriture d'un fichier de données. On retient les formats simples PGM (Portable GrayMap) ou PPM (Portable PixMap) qui sont deux types de fichiers graphiques faciles à écrire.

— dans un premier temps, il est préférable de ne s'intéresser qu'au format PGM, qui représente des images en nuances de gris sur un seul canal de couleur.

Ces formats d'images peuvent être générés en ASCII (texte lisible *via* un éditeur de texte) ou en binaire.

— pour faciliter le développement, commencez par le format ASCII et par la génération d'une image simple 3×3 contenant quelques niveaux de gris arbitraires.

Couleurs

Une fonction linéaire permet d'associer une nuance de gris à une altitude donnée, connaissant les altitudes minimale et maximale du terrain. Pour obtenir une image colorée, il faut utiliser une *color map* avec une transition progressive d'une couleur à l'autre. La fonction associée n'est plus un simple calcul linéaire.

— proposez une implémentation vous permettant de générer votre carte en couleur ;

— la *color map* « Haxby » fournit généralement un bon rendu pour les MNT :

Travail attendu à la fin du projet

Sont attendues : les sources d'un exécutable générant une image en couleurs de la projection d'un MNT. L'image devra être générée en binaire (PPM P6), un format plus compressé que la solution ASCII.

Les points suivants sont conseillés :

— développez avec le logiciel de gestion de versions Git ;

— testez votre code à chaque étape à l'aide de tests unitaires ;

— documentez votre implémentation.

Entrées / sorties du programme (à respecter strictement)

Votre programme se nommera `create_raster` et prendra en paramètres :

– le chemin/nom du fichier de données ;

– la largeur de l'image générée, en pixels. [1]

Exemple d'exécution : `./create_raster Guerledan_Feb19_50cm_wgs84.txt 800`

Fichiers complémentaires

En plus de vos sources, vous joindrez un script `build.sh` permettant la compilation de votre projet. Les fichiers temporaires ou compilés ne doivent pas être rendus.

Tout projet qui ne compile pas ne sera pas évalué.

Si votre projet nécessite l'installation d'une bibliothèque extérieure, veillez à le renseigner dans une documentation avec les directives d'installation.

1. la résolution et la hauteur de l'image seront déduites de ce paramètre de largeur en fonction des dimensions du MNT.

Grille d'évaluation du projet

../2	Organisation propre des sources en classes, structures, fichiers, répertoires
../3	Génération de l'image en binaire
../3	Génération de l'image en couleurs
../2	Compilation avec CMake
../2	Utilisation d'une projection officielle (Lambert93, Mercator, *etc.*)
../2	Documentation du code (sources commentées et structure documentée)
../2	Lisibilité du code
../2	Respect des entrées/sorties du programme
../2	Contours du MNT proprement rendus (si enveloppe non convexe)
../20	**Total**

../0.5	*Bonus :* génération d'un fichier de documentation (en HTML ou PDF)
../1	*Bonus :* implémentation de tests unitaires
../2	*Bonus :* génération avec ombrages [2]
../-1	*Malus :* rendu de votre exécutable ou de fichiers temporaires ou compilés (`.o`, *etc.*)

Votre projet doit compiler dans tous les cas.

2. Aussi appelé *hill shade* : assombrissement des pixels en fonction de l'orientation d'une surface par rapport à un soleil fictif. Cette fonctionnalité prend deux paramètres représentant la position relative du soleil : son altitude et son azimuth. La Figure 1 a été générée avec un ombrage, ce qui fait ressortir la rugosité du terrain, non visible avec une simple palette de couleurs.