

Reconstruction de surface à partir d'un nuage de points

Contenu

L'archive contient :

- un fichier *CMakeLists.txt* permettant d'ouvrir et de compiler le projet
- un dossier *header* contenant tous les fichiers .h
- un dossier *src* contenant tous les fichiers .c

Pour compiler le projet, ouvrir le *CMakeLists.txt* avec *QtCreator*.

Organisation du code

Le code est séparé en :

- *Hoppe* :
 - o classe permettant de dérouler l'algorithme
 - o contient toutes les fonctions correspondant aux différentes étapes
- *Plane*
 - o fonctions définissant les opérateurs sur les points et les normales (addition, soustraction, produit scalaire, produit dyadique, ...)
 - o fonction sur la matrice de covariance

Choix d'implémentations

Concernant la recherche du k-voisinage, nous avons choisi d'utiliser des kD-Trees afin de ne calculer qu'une seule fois les relations de voisinage et de pouvoir diminuer la complexité comparée à d'autres algorithmes de recherche de k-voisinage.

Utilisation des librairies

Nous avons choisi d'utiliser *PCL (Point Cloud Library)* pour :

- la structure *pcl::PointCloud*, qui permet de représenter un nuage de points et d'accéder à chacun des points grâce à un indice
- la structure *pcl::KdTreeFLANN*, qui contient le kD-Tree calculé à partir d'un nuage de points
- la structure *pcl::PointXYZ*, qui représente un point grâce à 3 attributs x, y et z (flottants)
- la fonction *nearestKSearch*, qui donne les k-voisins

Nous avons également utilisé *Eigen* (inclus dans *PCL*) pour :

- la structure *Eigen::Matrix3d* pour les matrices 3x3
- la structure *Eigen::Vector3f* pour les vecteurs de dimension 3
- la structure *Eigen::EigenSolver* permettant de récupérer les valeurs propres et les vecteurs propres d'une matrice

Algorithme

Même s'il produit des résultats approximant correctement une surface, cet algorithme peut produire des résultats incohérents dans plusieurs situations :

- si k est trop grand par rapport au nombre de points lors de la recherche du k -voisinage, l'approximation sera trop proche pour les différents points du nuage initial
- si la surface à reproduire contient des plis ou des trous, cela peut être mal interprété lors de la recherche de k -voisinage et des points pourraient être considéré comme appartenant au k -voisinage d'un point, faussant ainsi les résultats (disparition des plis ou des trous)

Par ailleurs, cet algorithme suppose d'avoir un nuage de points de densité uniforme en entrée, or cela est rarement le cas lors de la création des données.