

# Traitement de Nuage de Points

## TP3 - RANSAC

EPITA - Majeure IMAGE

Novembre/Décembre 2022

### Introduction

L'objectif de ce TP est d'implémenter l'algorithme RANSAC afin de détecter des primitives géométriques dans des nuage de points 3D.

- la documentation de la librairie C++ **Eigen** est accessible à [eigen.tuxfamily.org/dox](http://eigen.tuxfamily.org/dox)
- utiliser [MeshLab](#) ou [CloudCompare](#) pour visualiser un nuage de points 3D

### 1 Détection d'un plan unique

**Étape 1.** Implémenter un algorithme RANSAC qui détecte un seul plan unique dans un nuage de points 3D. Tester le programme sur le fichier `road_small.obj` puis sur le fichier `road_full.obj`.

Le programme peut soit colorier avec une couleur RGB chaque point du plan, soit diviser en deux les données sauvegardées dans deux fichiers `.obj`. De manière incrémentale, un algorithme naïf mais peu efficace peut être implémenté d'abord. Des optimisations ou des variantes de RANSAC peuvent être utilisées une fois un premier programme fonctionnel créé.

### 2 Détection de plusieurs plans

**Étape 2.** Adapter l'algorithme RANSAC pour détecter plusieurs plans. Tester le programme sur le fichier `church.obj`.

Dans un premier temps, le nombre de plans détectés peut être fixé a priori. Un critère d'arrêt peut être mis au point ensuite. Cette étape peut soit modifier le code de l'étape précédente, soit créer de nouvelles fonctions.

### Rendu

Il faut rendre les éléments suivants :

- le code source qui compile le ou les exécutables grâce à la commande `cmake ..` dans un dossier `build` (le dossier `data` n'étant pas nécessaire)
- un rapport de **2 pages maximum** (figures comprises s'il y en a) qui synthétise le ou les algorithmes développés, les variantes et optimisation choisies, les résultats obtenus, les différentes valeurs des éventuels paramètres, les limitations ou les problèmes non résolus