

README for Server part

Partie serveur géré par NODE JS.

C'est la partie où le nuage de point est envoyé de l'IPAD au serveur pour faire les calculs (retrieval par similarity search et SVM).

L'utilisateur upload le point cloud (par HTTP POST) et choisit ensuite deux options :

- Retrieval par similarity search
- Retrieval par SVM

Ensuite l'utilisateur récupère les résultats (par HTTP GET) selon l'option choisie.

1. Descriptions des dossiers

Module : contient le code qui fait appel aux binaires C++ pour la recherche par similarity search ou SVM

node_modules : les modules propres à NodeJS

public : les pages HTML NodeJS

results : un fichier json qui correspond au résultat similarity search et un fichier correspondant au résultat du SVM c'est à dire un label

routes : les différentes routes NodeJS. Une route upload.js qui s'occupe de la partie upload, une route download.js qui s'occupe de la partie download et une route index.js qui référence ces deux routes et s'occupe d'être l'interface.

uploads: le nuage de point qui a été uploads

view : différentes view au format jade pour NodeJS

2. Explication rapide des différents fichiers importants

retrieval.js

Contient les appels aux binaires C++ pour réaliser les calculs

trainDB → Run train_similaritySearch (render the database and build the tree)

similaritySearchPrediction → Run retrieval_similaritySearch (from a query point cloud, find the closest point cloud)

svmPrediction → From a query point cloud, try to retrieve the category using a trained SVM

obj2pcl → Convert the obj point cloud from the structure sensor to a pcd file. Run obj2pcl

transform_point_cloud → run transform_point_cloud_main. Permet de faire une transformation d'un nuage de point (rotation et/ou translation)

upload.js

router.post('/', function (req, res)

C'est la partie qui gère l'upload du point cloud de l'IPAD à l'ordinateur.

On récupère les informations du formulaire HTTP c'est à dire le descripteur que l'utilisateur a choisi mais également la méthode (svm ou similarity search) et le dataset (structure sensor ou rgbd)

Ensuite on convertit le nuage de point qui est au format obj au format pcd car c'est ce dernier qu'on va utiliser

Enfin, selon la méthode choisie on appelle le programme similaritySearch ou SVM

Paramètre à mettre :

Il faut préciser les chemins de la base de données des descripteurs pour chacun des bases de données (structure sensor et RGBD)

download.js

```
router.get('/', function(req, res)
```

C'est la partie qui gère le download des résultats du serveur à l'ipad pour la partie similaritySearch.

On récupère le fichier Json et on l'envoie à nouveau sur l'ipad pour le parsing