

Rapport de démarrage de projet NPM3D - Alexandre Lutt

1. Résumé de l'article

Ce projet de recherche consiste à étudier la reproductibilité l'article [2], ainsi que de proposer d'éventuelles pistes d'amélioration des performances. Cet article propose une méthode d'apprentissage profond (Deep Learning) pour déterminer une surface explicite à partir d'un nuage de points 3D. Jusque-là, la plupart des méthodes utilisaient l'algorithme "Marching Cubes" ([1]). Cet algorithme consiste à créer une représentation explicite (un maillage) d'une surface à partir d'une représentation implicite (typiquement, l'approximation d'une isosurface).

L'idée principale de [2] consiste à proposer une version de l'algorithme "Marching Cubes" qui soit à la fois généralisable à des topologies différentes, et différentiable, dans le but de proposer un réseau "end-to-end", capable d'apprendre des représentations des surfaces à partir d'une base de nuages de points. Ce réseau consiste en un premier bloc d'extraction de features pour chaque point, un auto-encodeur basé sur des convolutions, et enfin un bloc apprenant la représentation explicite de la surface à partir des déplacements des voxels et de l'occupation de la grille 3D.

2. Premiers tests d'implémentation

Il est intéressant de noter qu'à l'exception du bloc "auto-encodeur", la plupart des opérations introduites dans [2] ne sont pas immédiatement traduisibles en code PyTorch efficace. Par conséquent, un premier challenge a été d'apprendre à générer du code C++ et Cuda afin de développer une librairie capable de fournir les briques élémentaires nécessaires à la conception du réseau. Ayant accès au code de 2018, une partie importante de mon travail a été de refactoriser ce code, afin de profiter des nouvelles fonctionnalités de PyTorch (par exemple, la compilation de librairies C++/Cuda) pour développer un code plus clair, optimisé et moderne.

3. Suite envisagée

Une fois terminée la phase de développement du réseau, je compte l'entraîner sur un extrait de la base de données ShapeNet, à l'aide des GPUs disponibles sur Google Cloud Platform, afin de comparer mon implémentation aux résultats de [2]. Plusieurs paramètres utilisés étant disponibles dans [3], je compte vérifier la reproductibilité de la méthode originale. Enfin, je compte explorer différentes pistes d'amélioration, par exemple en proposant une régularisation, ou en modifiant l'architecture pour réduire le temps d'entraînement et améliorer les performances.

Références

- [1] Lorensen William E. and Cline Harvey E. Marching cubes : A high resolution 3d surface construction algorithm. In *SIGGRAPH*, 1987.
- [2] Liao Yiyi, Donné Simon, and Andreas Geiger. Deep marching cubes : Learning explicit surface representations. In *CVPR*, 2018.
- [3] Liao Yiyi, Donné Simon, and Andreas Geiger. Supplementary material for deep marching cubes : Learning explicit surface representations. In *CVPR*, 2018.