

# Projet – Transmission Progressive de Modèles 3D

## Compression et Décodage Progressif de Modèles 3D

Septembre 2023

**But du projet** Le but de ce projet est de transmettre progressivement un modèle 3D représenté par un fichier OBJ. Nous nous intéresserons seulement à des données géométriques (maillages). Une représentation adaptée sera développée en 'inversant' un algorithme de compression de l'état de l'art.

**Déroulement du projet** Le projet sera réalisé par groupes de 4 étudiants. Le projet sera présenté lors du premier cours de Compression, Streaming, Interactions Video (semaine du 25 Sept.). Chaque groupe d'étudiant devra alors prendre connaissance des articles mis à leur disposition sur Moodle, et en choisir un qui sera le papier qu'ils devront implémenter durant ce projet. Une séance de TP-Projet aura lieu le **23 octobre**. Cette séance sera une évaluation d'étape durant laquelle, chaque groupe devra faire une présentation du papier choisi, ainsi qu'une présentation de leurs avancées en terme d'implémentation de l'algorithme. Il est attendu que le contenu du papier soit maîtrisé et l'implémentation et la répartition du travail déjà définie. L'évaluation du projet finale aura lieu le **24 novembre**. Elle prendra la forme d'une présentation devant la classe entière, durant laquelle les étudiants devront à la fois présenter le papier choisi, leur implémentation, ainsi que les résultats obtenus.

**Travail attendu** Vous devez développer un algorithme permettant de transformer un fichier OBJ en un fichier OBJA (OBJ Augmenté) qui permettra la transmission, et la visualisation progressive de l'objet 3D par un client. L'objectif est d'obtenir le meilleur ratio débit/distortion. Le code sera soumis en ligne sous Moodle avant la veille de la soutenance, et des présentations de chaque projet seront faites le 19 novembre. De plus, une évaluation en ligne de vos performances pourra être utilisée tout au long de votre développement pour tester l'efficacité de votre codage (<https://csi.alcouffe.eu/>).

**Code fourni** Vous trouverez le code fourni sous le dépôt suivant <https://gitea.tforgione.fr/tforgione/obja>.

- `decimate.py` : un script d'exemple utilisé pour décimer un modèle 3D de manière naïve;
- `/example` : des exemples de fichiers obj et obja, générés à partir du script `decimate.py`;
- `server.py` : le fichier contenant le client de visualisation;
- `obja.py` : parser python pour les modèles OBJ et la création de modèles OBJA.

## 1 Transmission d'un maillage 3D

Une première solution qui transmet un fichier OBJA naïf (mono-résolution) vous est fournie (`decimate.py`). L'implémentation ainsi que la documentation sont disponibles ici <https://gitea.tforgione.fr/tforgione/obja>. Vous trouverez dans le code fourni, une extension du format OBJ en OBJA (pour "augmenté") qui permet de représenter et transmettre progressivement un maillage triangulaire.

L'objectif est donc de développer une représentation progressive et compacte. Ainsi, le fichier sera transmis progressivement, par paquets, pour que l'objet soit visualisé au fur et à mesure de la réception des données. Vous pourrez vérifier et visualiser le bon fonctionnement de vos algorithmes en utilisant le client mis à disposition sur le dépôt git (`server.py`).

Vous pourrez évaluer la qualité de votre modèle simplifié, donc la capacité de compression, et sa progressivité en le faisant passer par le site proposé.

## 2 Décompression et modèle progressif

Afin de développer un modèle 3D progressif, nous vous proposons de vous baser sur un algorithme de compression de maillage 3D, qui permet de représenter un objet 3D à plusieurs niveaux de détail. Plusieurs articles sont proposés, et chacun des groupes choisira un article ou un groupe d'article différent. Voici les idées proposées :

- *Progressive meshes*, avec des critères de décimation à choisir (4 articles),
- *Squeeze & CPM*, papiers de Pajarola et Rossignac (2 articles),
- *Arbitrary triangular meshes*, papier de Cohen-Or et Levin (1 article),
- *Lossless transmission*, papier de Alliez et Desbrun (1 article),
- *MAPS, remeshing*, papier de Lee and co. (1 article),
- *Compression basée remeshing*, papier de Payan et co. (1 article).

Merci de trouver le détail des références bibliographiques, de faire des groupes de 4 et de vous inscrire sur le document [Inscription](#) et récupérer les papiers (qui sont aussi disponibles en ligne depuis l'n7) dans le dossier [ici](#).

En inversant le procédé de compression, vous définirez des opérations de raffinement du modèle, envoyées progressivement au serveur comme une suite d'instruction `obja`. Une bibliothèque python `obja.py` vous est fournie pour vous aider dans l'inversion du schéma de compression. Elle comprend la version python du parser OBJ, et un outil pour la gestion des indices (des sommets et des faces), utile lors de la suppression d'éléments.