Crypto-Compression 3D

Compte Rendu 3

Potin Clément, Fournier Romain
Master 2 IMAGINE
Université de montpellier
2021

Problématique

L'objectif est de développer un algorithme qui, de manière conjointe, compresse avec ou sans pertes et chiffre un objet 3D. Les performances de cette méthode seront ensuite mesurées en termes de taux de compression et de qualité de reconstruction de l'objet 3D.

Décomposition du programme

Le logiciel, tel qu'on le voit aujourd'hui, sera décomposé en plusieurs étapes de compression et de chiffrement des données :

1. **Lecture du fichier clair (.obj)**
   Lire les données du fichier objet (.obj) et les stocker de façon à conserver l'intégrité du maillage (listes de sommets ("vertices"), normales à ces sommets, demi-segments ("half-edges") et faces) de façon à pouvoir travailler sur ces éléments
2. **Compression : quantification**
   Quantifier les données (réduire avec plus ou moins d'intensité la précision des positions des sommets (→ utile pour la prédiction à venir, et donc pour améliorer l'efficacité du codage de *Huffman*), utiliser une "carte des normales" pour remplacer chaque normale au sommet par la normale de la carte la plus proche (→ pour le codage de *Huffman* également), etc)
3. **Compression : *EdgeBreaker* + prédiction**
   Appliquer l'algorithme *Edgebreaker* sur ces éléments, pour grandement réduire la quantité d'information nécessaire pour décrire notre maillage, tout en conservant l'entièreté de ses données. On peut coupler à *EdgeBreaker* une méthode de prédiction pour ne plus stocker les coordonnées exactes des sommets, mais seulement un vecteur de correction par sommet, qui associé à la prédiction du point suivant, permettra de retrouver ses coordonnées exactes
4. **Chiffrement : transpositions**
   Appliquer un algorithme de transpositions avec clé privée (mélange des sommets) pour chiffrer notre maillage. Ces transpositions permettent de garder intactes les données, de façon à ce que l'objet soit toujours utilisable dans des logiciels, mais empêche d'obtenir le vrai maillage sans l'utilisation de la bonne clé de déchiffrement

5. **Écriture dans un fichier compressé (.bin ?)**
   Enregistrer les données de l'objet, compressées et chiffrées, dans un fichier limitant au maximum l'apport d'informations externes (idéalement, créer notre propre "writer/reader" et enregistrer nos informations dans un fichier binaire)
6. **Compression : codage entropique (Huffman)**
   Appliquer un niveau de compression supplémentaire à nos données en réalisant une compression via l'algorithme de *Huffman* sur notre fichier de sortie. Les étapes précédentes auront permis de préparer les données à un codage de *Huffman* le plus efficace possible

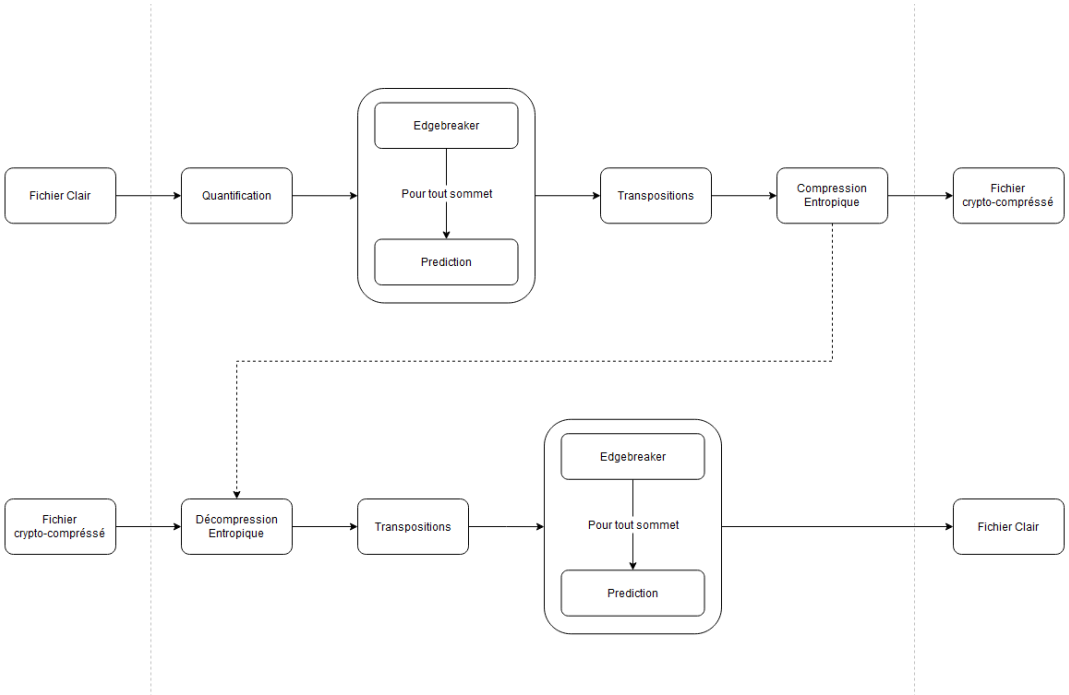

Schéma des principales étapes de logiciel

La décompression se fera dans le sens inverse (du fichier .bin au fichier .obj). Sans la bonne clé de déchiffrement, les données de l'objet recomposé seront similaires à celles qui auraient été obtenues avec la bonne clé de déchiffrement, mais les sommets seront dans le mauvais ordre. L'objet pourra donc être chargé dans n'importe quel logiciel, devrait conserver les mêmes attributs que l'objet d'origine tels que sa boîte englobante ("bounding box") ainsi que les mêmes sommets, mais ceux-ci resteront mélangés et l'objet ne sera donc pas utilisable tel quel.

Décomposition du projet

Liste des tâches

- T1 : Lire, récupérer les données, et écrire des fichiers .obj
- T2 : Quantifier les données obtenues après T1 (compression)
- T3.1 : Coder l'algorithme *EdgeBreaker* (compression/décompression)
- T3.2 : Implémenter la prédiction avec *EdgeBreaker* (compression/décompression)
- T4 : Créer un algorithme de transpositions (chiffrement/déchiffrement)
- T5 : Lire, récupérer les données, et écrire des fichiers compressés (.bin ?)
- T6 : Implémenter un algorithme de Huffman adapté aux fichiers créés/utilisés en T5 (compression/décompression)

Gantt

Tâches	S1 (25 Oct)	S2 (01 Nov)	S3 (08 Nov)	S4 (15 Nov)	S5 (22 Nov)	S6 (29 Nov)	S7 (06 Déc)	S8 (13 Déc)
T0								
T1								
T2								
T3.1								
T3.2								
T4								
T5								
T6								

Nom	Code couleur
Clément	
Romain	
Clément et Romain	

Tâches	Descriptions
T0	Étude de l'état de l'art, préparation du projet
T1	Lire, récupérer les données, et écrire des fichiers .obj
T2	Quantifier les données obtenues après T1 (compression)
T3.1	Coder l'algorithme EdgeBreaker (compression/décompression)
T3.2	Implémenter la prédiction avec EdgeBreaker (compression/décompression)
T4	Créer un algorithme de transpositions (chiffrement/déchiffrement)
T5	Lire, récupérer les données, et écrire des fichiers compressés (.bin)
T6	Implémenter un algorithme de Huffman adapté aux fichiers créés/utilisés en T5 (compression/décompression)

Décomposition prévisionnelle des tâches (14/11)

Méthodes d'évaluation

La semaine prochaine, en parallèle de notre travail sur l'implémentation des premières fonctionnalités de notre programme, nous avons dans l'idée de nous intéresser aux possibles méthodes existantes permettant d'évaluer le taux et la qualité de compression/reconstitution de notre objet, ainsi qu'aux méthodes évaluant la sécurité du chiffrement que nous appliquerons à nos données.

En ce qui concerne la qualité de la compression/reconstitution du modèle 3D, nous avons déjà certaines pistes, comme la Root-Mean-Square Deviation (RMSD), ou l'association de "Skeleton Trees" et de la distance de *Hausdorff* [14].

Lien du Git

Notre travail sera mis à jour au lien suivant :

https://github.com/Romimap/3D-CryptoCompression/

Références

Demos & Softwares

1. Vladimir Agafonkin, “Edgebreaker, the Heart of Google Draco” : https://observablehq.com/@mourner/edgebreaker-the-heart-of-google-draco

2. Google Draco: https://github.com/google/draco

Papers

3. Michael Deering, “Geometry Compression”, sun Microsystems, 1995 : http://web.cse.ohio-state.edu/~shen.94/Su01_888/deering.pdf

4. Chandrajit L Bajaj, Valerio Pascucci, Guozhong Zhuang, “Single Resolution Compression of Arbitrary Triangular Meshes with Properties”, University of Texas, 1997: https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.14.946&rep=rep1&type=pdf

5. Mike M. Chow, “Optimized Geometry Compression for Real-time Rendering”, Massachusetts institute of Technology, May 1997: https://www.semanticscholar.org/paper/Optimized-geometry-compression-for-real-time-Chow/39babe9519e6b58bae4350e4f57be86dc45ce6f9

6. Jarek Rossignac, “Edgebreaker: Connectivity compression for triangle meshes”, Georgia Institute of Technology, 1999 : https://www.cc.gatech.edu/~jarek/papers/EdgeBreaker.pdf

7. Daniel Cohen-Or, David Levin, Offir Remez, “Progressive Compression of Arbitrary Triangular Meshes”, Tel Aviv University, 1999: https://www.tau.ac.il/~levin/vis99-dco.pdf

8. Jarek Rossignac, Alla Safonova, Andrzej Szymczak, “3D Compression Made Simple: Edgebreaker on a Corner-Table”, Georgia Institute of Technology, 2001: https://www.researchgate.net/publication/3896746_3D_compression_made_simple_Edgebreaker_with_ZipandWrap_on_a_corner-table

9. Pierre Alliez, Mathieu Desbrun, “Progressive Compression for Lossless Transmission of Triangle Meshes”, University of Southern California, February 2002:
https://www.researchgate.net/publication/2534417_Progressive_Compression_for_Lossless_Transmission_of_Triangle_Meshes

10. Jarek Rossignac, “3D mesh compression”, College of Computing and GVU Center Georgia institute of Technology, January 2003:
https://www.researchgate.net/publication/27521282_3D_Mesh_Compression

11. Esam Elsheh, A. Ben Hamza, "Secret sharing approaches for 3D object encryption", Concordia Institute for Information Systems Engineering, Concordia University, Montréal, QC, Canada, 2011:
https://www.sciencedirect.com/science/article/abs/pii/S095741741100724X

12. In-Ho Lee and Myungjin Cho, "Optical Encryption and Information Authentication of 3D Objects Considering Wireless Channel Characteristics", Department of Electrical, Electronic, and Control Engineering, Hankyong National University, Ansung 456-749, Korea, October 2013:
https://www.osapublishing.org/DirectPDFAccess/766AED72-4C7E-47EE-BB28209C333886A8_276786/josk-17-6-494.pdf

13. Marc Éluard, Yves Maetz, and Gwenaël Doërr, "Geometry-preserving Encryption for 3D Meshes", Technicolor R&D France, November 2013:
https://www.researchgate.net/profile/Gwenael-Doerr/publication/273257218_Geometry-preserving_Encryption_for_3D_Meshes/links/54fc4b660cf2c3f52422a624/Geometry-preserving-Encryption-for-3D-Meshes.pdf

14. Xin Chen, Jingbin Hao, Hao Liu, Zhengtong Han and Shengping Ye, “Research on Similarity Measurements of 3D Models Based on Skeleton Trees”, School of Mechatronic Engineering, China University of Mining and Technology, Daxue Road 1, Xuzhou 221116, China, State Key Laboratory of Materials Forming and Mould Technology, Huazhong University of Science and Technology, Luoyu Road 1037, Wuhan 430074, China, April 2017:
https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&cad=rja&uact=8&ved=2ahUKEwi91fmtwZj0AhUI2BoKHYcKA1AQFnoECAMQAQ&url=http%3A%2F%2Fwww.mdpi.com%2F2073-431X%2F6%2F2%2F17%2Fpdf-vor&usg=AOvVaw0lfo-8VxxYFuVQTnyCt6JI

15. Ying Zhou, Lingling Wang, Lieyun Ding, Cheng Zhou, “A 3D model Compression Method for Large Scenes”, Huazhong Univ. of Science and Technology, 2018:
https://www.iaarc.org/publications/fulltext/ISARC2018-Paper207.pdf