

HAI809I Codage et compression multimédia

TP 5 - Compression 3D

Andrew Mansour

1 Initialisation

Les métriques RMSE et Hausdorff (codées dans les fichiers metrics.cpp et son header metrics.h), sont calculés et implémentés utilisant les formules données en cours.

Le RMSE est une métrique utile pour trouver la différence globale entre deux maillages, elle est calculée avec cette formule.

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n \|y_i - \bar{y}_i\|^2}$$

La distance de Hausdorff, quand à elle, prend en compte les plus grandes erreurs/différences entre nos deux maillages, sa formule est :

$$d_H(A, B) = \max \left(\sup_{a \in A} \inf_{b \in B} d(a - b), \sup_{b \in B} \inf_{a \in A} d(b - a) \right)$$

L'implémentation de cette formule a été moins directe que le RMSE, car il faut trouver la plus grande valeur de a et b donnant une distance minimale, ce qui fait que le calcul de cette métrique est plus long à exécuter.



Figure 1: Images des deux maillages identiques, on obtient $\text{RMSE} = \text{Hausdorff} = 0$, qui est le résultat attendu vu qu'il n'y a aucune différence entre les deux maillages.

2 Quantification Draco

La quantification (quantification.cpp et quantification.h) se fait en utilisant les valeurs des sommets du maillage, d'un facteur de quantification données en entrée ainsi que les coordonnées minimales de la boîte englobante et le plus grand côté de la boîte englobante.

Pour générer une boîte englobante, (Code dans la classe BoundingBox.h) on va créer une classe qui contient deux attributs, un vecteur du coin du haut, et un vecteur pour le coin du bas. Pour "englober" notre maillage, on va simplement agrandir ces bords en agrandissant la boîte au plus petit possible pour contenir chaque sommet de notre maillage. (La classe a été faite pour le projet RayTracer le semestre dernier, je l'ai réutilisé pour éviter d'en recréer une)

La valeur $bbmin$ de la formule correspond ainsi à l'attribut min de notre classe. La valeur $range$ correspond au plus grand côté, c'est à dire la plus grande distance en une coordonnée entre nos vecteurs min et max .

Après avoir calculé ses valeurs, on applique simplement la formule, et on arrondi ces valeurs en entiers pour obtenir un maillage voxélisé.

La déquantification (quantification.cpp et quantification.h) revient simplement à appliquer la formule sur notre maillage quantifié, en prenant les mêmes valeurs qp , $bbmin$ et $range$, ce qui va nous donner un maillage que l'on peut voir.

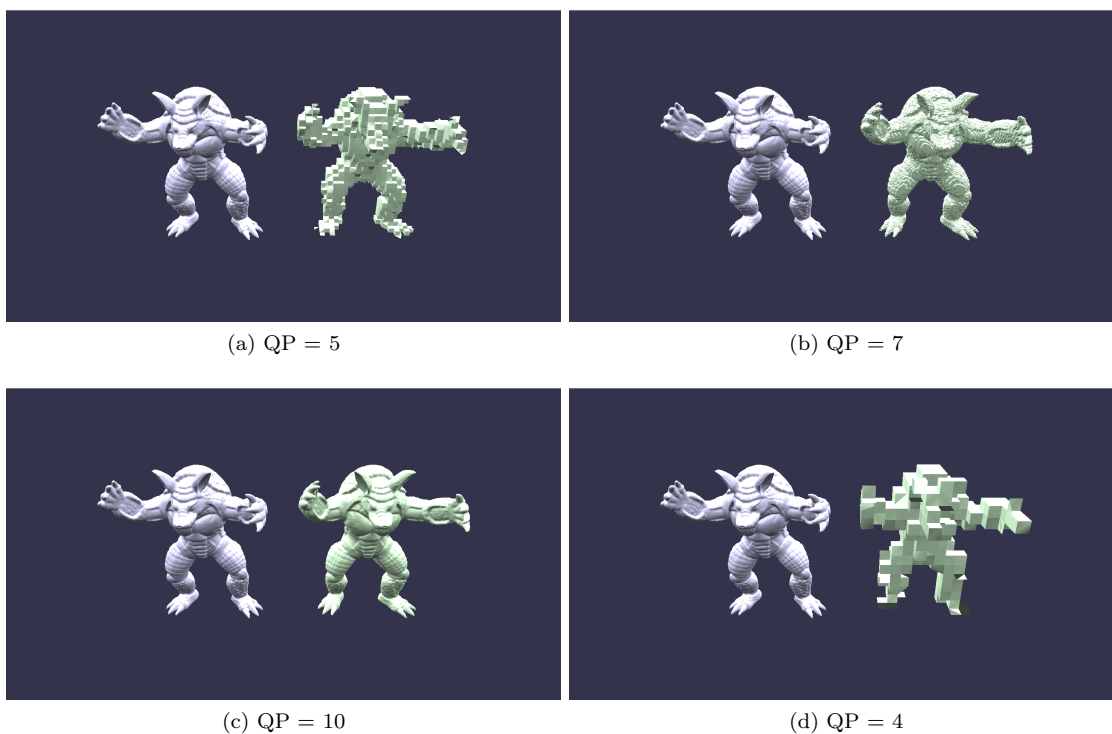


Figure 2: Comparaison des maillages avec de différentes valeurs du facteur de quantification. On observe que plus le facteur est bas, plus on voit une forte différence avec le maillage, avec des valeurs trop basses, on ne reconnaît presque plus notre maillage original, et avec des valeurs très hautes, on obtient deux maillages très similaires.

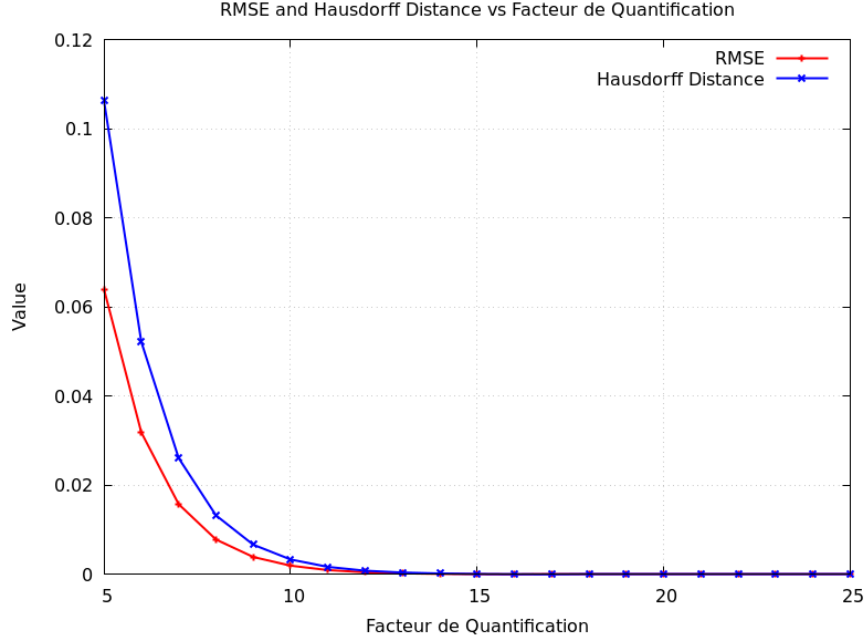


Figure 3: Courbe montrant les métriques RMSE et Hausdorff en fonction d'un facteur de quantification compris entre 5 et 25, on remarque que plus on augmente ce facteur, plus notre maillage va ressembler à notre maillage d'origine, résultant en des valeurs qui se rapprochent de zéro pour les deux méthodes de mesure.

La courbe représentant le taux de compression n'a pas été faite car on ne supprime jamais de sommets dans ce TP, on obtiendra donc toujours un taux de compression égale à 1. Une solution possible aurait été de supprimer tout les sommets qui se trouvent à la même position après avoir arrondi leurs valeurs en int.

3 Codage rANS

L'objectif du codage rANS (implémentation dans rans.h et rans.cpp) est de stocker toutes les valeurs possibles des coordonnées de nos sommets en une seule valeur. Le décodage permet d'obtenir de nouveau nos anciennes valeurs à partir de la valeur encodée.

L'étape préalable est de calculer les différentes listes nécessaires pour ce calcul. D'abord, il faut créer une liste d'entiers contenant toutes les coordonnées de nos sommets appelée la séquence, ici, on a une liste de la forme :

$$S = x_1, y_1, z_1, x_2, y_2, z_2, \dots, x_n, y_n, z_n$$

Notre alphabet sera tous les entiers compris entre 0 et le facteur de quantification au carré, pour avoir toutes les valeurs possibles pour nos coordonnées. Et les fréquences sont calculées en vérifiant pour chaque valeur dans l'alphabet le nombre d'occurrences dans notre séquence. La fréquence pondérée est simplement la liste de la somme des fréquences de chaque index inférieur. Après avoir calculé toutes ces valeurs nécessaires, on applique la formule :

$$x_i = \left\lfloor \frac{x_{i-1}}{F_i} \right\rfloor \cdot M + C_i + \text{mod}(x_{i-1}, F_i)$$

Le défaut de ce codage est qu'il nécessite des valeurs bien plus grandes que ce qui peut être stocké dans un int. Il faut donc utiliser des maillages plus petits (comme l'armadillo dans la base de modèles 3D) et des unsigned long long int, pour pouvoir les stocker. Pour l'armadillo, on obtient une valeur

$$x = 12401974005966427793.$$

Le décodage se fait en appliquant les opérations inverses du codage, on utilise les mêmes listes d'alphabet et de fréquences et on applique la formule :

$$x_{i-1} = \left\lfloor \frac{x_i}{M} \right\rfloor \cdot F_{st} + slot - C_{st}$$

$$st = C^{-1}(slot)$$

$$slot = \text{mod}(x_i, M)$$

L'implémentation du décodage n'a pas été réussi, chaque tentative de réglage d'un problème menait à un autre, actuellement, il y a un problème d'erreur de segmentation, mais due au manque de temps, ceci n'a pas été réglé avant l'heure du dépôt.