HAI918I - Image, Sécurité et Deep Learning
TP6

Ingo Diab

2023

Table des matières

1 Lien Github

Lien du TP : `https://github.com/IngoDiab/HAI918I_Image_securite_Deep_Learning/tree/main/TP6`

2 Méthode de Cho

Dans ce tp, nous devions utiliser la méthode de Cho afin de cacher un message dans un modèle 3D. Cette méthode consiste à :

- Convertir chaque vertex du modèle en coordonnées sphériques.
- Créer un histogramme que l'on divise en X bins de tailles égales (avec X = le nombre de bits du message à cacher).
- Trier chaque vertex selon sa norme et le ranger dans le bin correspondant.
- Normaliser chaque bin pour avoir des normes entre 0 et 1.
- Pour chaque bin nous calculons la moyenne. Si le bit à insérer est 1 et tant que la moyenne est inférieur à 0.5 + un certain alpha alors nous élevons les normes du bin à la puissance k (fixé à 1 et qui est décrémenté de delta_k à chaque itération). Si le bit à insérer est 0 et tant que la moyenne est supérieur à 0.5 - un certain alpha alors nous élevons les normes du bin à la puissance k (fixé à 1 et qui est incrémenté de delta_k à chaque itération).
- Après avoir fait cette opération sur tous les bits, dénormaliser les bins afin de retrouver les normes d'avant normalisation.
- Mettre les vertex des bins dans l'ordre pour avoir le mesh de base.
- Convertir les coordonnées sphérique en scalaire.

Pour extraire un message, nous devons :

- Convertir chaque vertex du modèle en coordonnées sphériques.
- Créer un histogramme que l'on divise en X bins de tailles égales (avec X = le nombre de bits du message à cacher).
- Trier chaque vertex selon sa norme et le ranger dans le bin correspondant.
- Normaliser chaque bin pour avoir des normes entre 0 et 1.
- Pour chaque bin, si la moyenne est supérieur à 0.5 alors le bit est 1 sinon le bit est 0

3 Expérimentations

Dans cette partie nous allons cacher différents messages dans le modèle 3D (le modèle de base est le lapin). Nous utiliserons la RMSE comme métrique de distorsion afin de voir la ressemblance entre notre modèle de base et le modèle avec le message caché. Nous allons faire plusieurs expérimentations en fixant 2 paramètres et en en faisant varier 1 (parmi la taille du message, la force d'insertion alpha et delta_k).

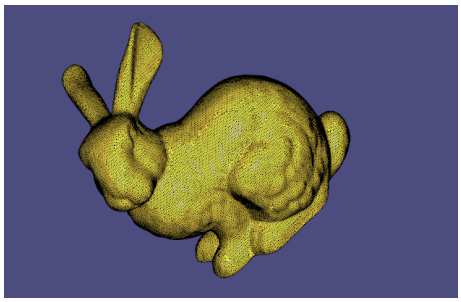

FIGURE 1 – Modèle de base

3.1 Variation de la capacité

Nous fixons alpha et delta_k et faisons varier le nombre de bit du message.

Nb bits	Alpha	Delta_k	RMSE
8	0.3	0.01	0.029
40	0.3	0.01	0.006
88	0.3	0.01	0.003
152	0.3	0.01	0.002
216	0.3	0.01	0.001

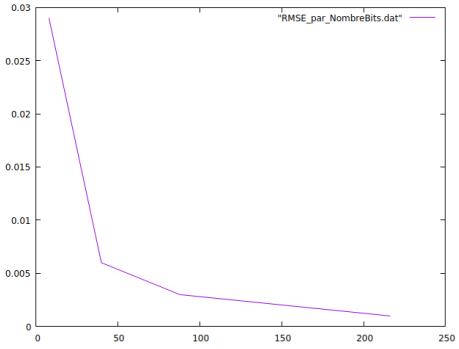

FIGURE 2 – RMSE en fonction du nombre de bits du message

Ici nous pouvons voir que plus le message est long, plus le modèle tatoué ressemble à l'originale. Cela s'explique par le fait que plus le message est long, plus les bins sont nombreuses et petites. Ce type de bin permet d'avoir un effet moins "bloc" et conserve donc plus les détails du mesh d'origine et nous avons donc un RMSE plus bas.

FIGURE 3 – Modèle avec 8 bits tatoués

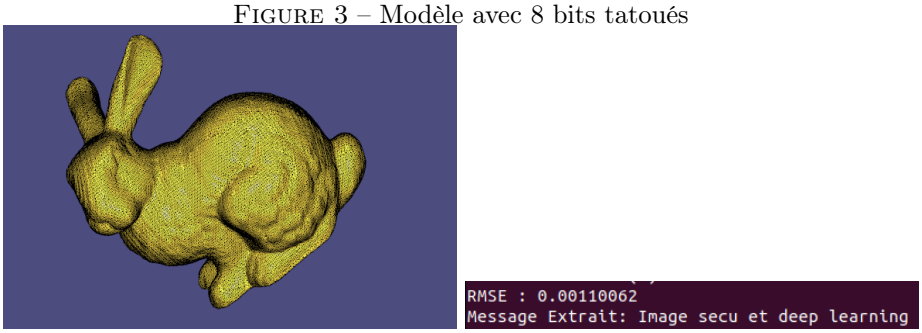

FIGURE 4 – Modèle avec 216 bits tatoués

3.2 Variation de la force alpha

Nous fixons le nombre de bits du message et delta_k et faisons varier la force d'insertion alpha.

Nb bits	Alpha	Delta_k	RMSE
40	0.01	0.01	0.0005
40	0.05	0.01	0.001
40	0.1	0.01	0.002
40	0.2	0.01	0.004
40	0.4	0.01	0.008

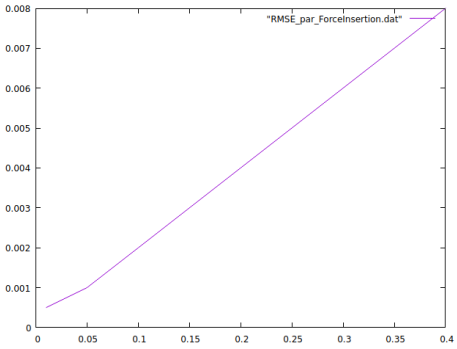

FIGURE 5 – RMSE en fonction de la force d'insertion

Ici nous pouvons voir que plus la force d'insertion est forte plus le modèle est distordu. Cela s'explique par le fait que lorsque la force alpha est grande alors pour insérer un 1, le bin devra avoir une moyenne plus grande ($> 0.5f +$ alpha) et pour insérer un 0, le bin devra avoir une moyenne plus petite ($< 0.5f$ - alpha). Ce qui fait que les vertex ont des normes plus différentes lorsque alpha est grand, ce qui créé une effet de relief visuellement.

FIGURE 6 – Modèle tatoué avec une force alpha de 0.01

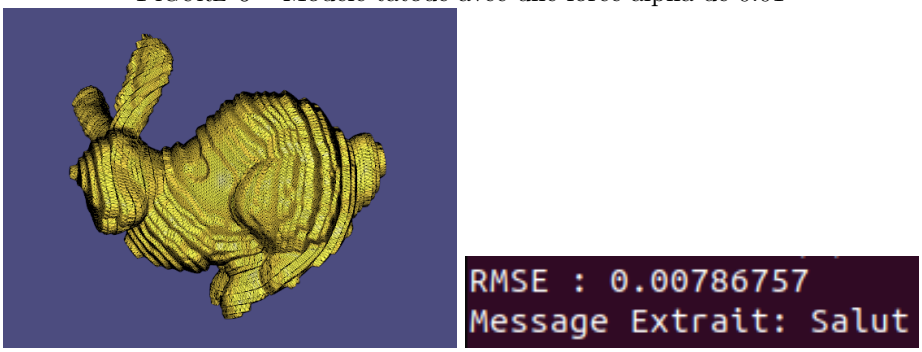

FIGURE 7 – Modèle tatoué avec une force alpha de 0.4

3.3 Variation de delta_k

Nous fixons le nombre de bits du message et la force d'insertion alpha et faisons varier delta_k.

Nb bits	Alpha	Delta_k	RMSE
40	0.1	0.01	0.002
40	0.1	0.02	0.002
40	0.1	0.05	0.002
40	0.1	0.2	0.003
40	0.1	0.5	0.003

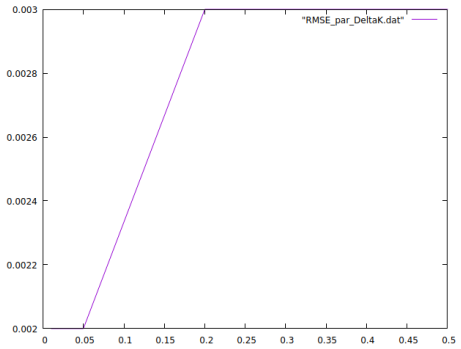

FIGURE 8 – RMSE en fonction du delta_k

Ici nous pouvons voir que plus le delta_k est fort plus le modèle est distordu mais la différence ne se fait quasiment pas ressentir. Cela s'explique par le fait que lorsque nous insérons le watermark et que nous élevons les normes des vertex à la puissance k, la puissance k ayant été trop décrémenté/incrémenté d'un coup, nous n'avons pas la valeur optimale de la puissance k pour satisfaire notre objectif (comme lorsque le learning rate est trop grand lors de l'utilisation de CNNs).

FIGURE 9 – Modèle tatoué avec une force alpha de 0.01

FIGURE 10 – Modèle tatoué avec une force alpha de 0.4

4 Conclusion

Dans ce tp, nous avons vu la méthode de Cho afin d'insérer un message dans un modèle 3D. Au cours de nos expérimentations, nous avons pu voir que 3 paramètres (surtout le nombre de bits du message et la force d'insertion) jouent sur la qualité de notre modèle final.