



Informatique graphique avancée, animation et rendu

Compte-rendu TP1

Traitements géométriques de points 3D

Louis Jean
Master 2 IMAGINE
Université de Montpellier
N° étudiant : 21914083

13 septembre 2024

Table des matières

1	Introduction	2
2	Nuage de points et principe	2
3	Définition des poids	3
3.1	Formule générale des poids	3
3.2	Différents noyaux de poids	3
4	Méthode SPSS	4
4.1	Description de l'algorithme	4
4.2	Résultats	4
5	Méthode HPSS	5
5.1	Description de l'algorithme	5
5.2	Résultats	5
6	Ajout de bruit sur le nuage de points d'entrée	6
7	Conclusion	7

1 Introduction

L'objectif du TP est d'implémenter des méthodes de traitement géométrique de nuages de points en 3D, plus spécifiquement les algorithmes **SPSS** (Smoothed Point Set Surfaces) et **HPSS** (Hierarchical Point Set Surfaces).

Pour ce faire, nous avons utilisé une base de code fournie pour afficher des nuages de points, et nous avons fait évoluer ce code en ajoutant les algorithmes.

2 Nuage de points et principe

Un nuage de points est simplement une soupe de points distribuée dans l'espace, de manière ordonnée ou non. Ici, aucune notion de topologie, de triangle ou de polygone. Notre but est de projeter un ensemble de points générés aléatoirement sur un autre ensemble de points déjà existant.. Pour ceci, nous nous sommes servis durant tout le TP d'un nuage de points fourni dans la base de code, que voici.



Figure 1: dino.pn

3 Définition des poids

Les poids jouent un rôle essentiel dans la projection des points sur une surface à partir d'un nuage de points en 3D. Ils définissent l'influence de chaque point voisin sur un point donné lors des étapes de projection.

3.1 Formule générale des poids

Les poids sont définis par une fonction $\phi(x, p_i)$ qui décrit l'influence d'un point voisin p_i sur le point d'entrée x . Le poids w_i dépend de la distance $d = \|x - p_i\|$ entre le point d'entrée et son voisin. La taille des caractéristiques locales est définie par un paramètre r , qui peut être constant ou dépendre du point x .

La formule générale pour les poids est la suivante :

$$w_i = \phi\left(\frac{\|x - p_i\|}{r(x)}\right)$$

3.2 Différents noyaux de poids

Les différents noyaux de poids utilisés dans les méthodes **SPSS** et **HPSS** incluent les noyaux **Singulier**, **Gaussien**, et **Wendland**. Voici leurs définitions :

- **Noyau Singulier** :

$$\phi(x, p_i) = \left(\frac{r}{d}\right)^s$$

Ce noyau interpole les points d'entrée et leurs normales, avec s la rapidité avec laquelle l'influence décroît avec la distance.

- **Noyau Gaussien** :

$$\phi(x, p_i) = e^{-\frac{d^2}{r^2}}$$

Ce noyau diminue exponentiellement avec la distance, ce qui implique que l'influence des voisins les plus éloignés est de plus en plus réduite.

- **Noyau Wendland** :

$$\phi(x, p_i) = \left(1 - \frac{d}{r}\right)^4 \cdot \left(1 + 4\frac{d}{r}\right)$$

Le noyau Wendland devient nul lorsque $d \geq r$, ce qui veut dire qu'il offre un très bon contrôle de la zone définie par r .

Voici comment j'ai implémenté ces noyaux dans le code :

```
1 double SingularWeight(Vec3 const &inputPoint, Vec3 const &neighbor, double radius
  , double s = 2) {
2     double d = Vec3::euclideanDistance(inputPoint, neighbor);
3     return pow(radius / d, s);
4 }
5
6 double GaussianWeight(Vec3 const &inputPoint, Vec3 const &neighbor, double radius
  ) {
7     double d = Vec3::euclideanDistance(inputPoint, neighbor);
8     return exp(-pow(d,2) / pow(radius,2));
9 }
10
11 double WendlandWeight(Vec3 const &inputPoint, Vec3 const &neighbor, double radius
  ) {
12     double d = Vec3::euclideanDistance(inputPoint, neighbor);
13     return pow(1 - (d / radius),4) * (1 + 4 * (d / radius));
14 }
```

4 Méthode SPSS

4.1 Description de l'algorithme

L'algorithme SPSS est une méthode pour projeter des points aléatoires sur une surface définie par un nuage de points équipés de normales.

Les étapes principales de SPSS sont les suivantes :

1. **Calcul du centroïde** $c(x)$:

$$c(x) = \frac{\sum_{p_i \in \text{NN}(x)} w_i p_i}{\sum_{p_i \in \text{NN}(x)} w_i}$$

où p_i représente les points voisins et w_i sont les poids associés aux points voisins calculés via un des noyaux définis.

2. **Calcul de la normale de projection** $n(x)$:

$$n(x) = \frac{\sum_{p_i \in \text{NN}(x)} w_i n_i}{\sum_{p_i \in \text{NN}(x)} w_i}$$

où n_i est la normale associée à chaque point voisin p_i .

3. **Projection du point** : Le point d'entrée x est ensuite projeté sur le plan défini par $c(x)$ et $n(x)$, en répétant plusieurs fois l'itération pour améliorer la précision.

4.2 Résultats

En utilisant SPSS, on obtient une surface lisse qui approximativement passe par les points d'entrée. Les résultats dépendent de la taille du noyau, du type de noyau utilisé, du nombre d'itérations de l'algorithme, du nombre de voisins considérés et du nombre de points de sortie désiré. Ici, j'ai choisi d'utiliser 10 itérations et 200 voisins, pour 100 000 points en sortie. Pour chaque noyau, j'ai pris $r = 0.05$, car c'est avec ce paramètre que j'ai observé les plus grandes variations entre ces dits noyaux.

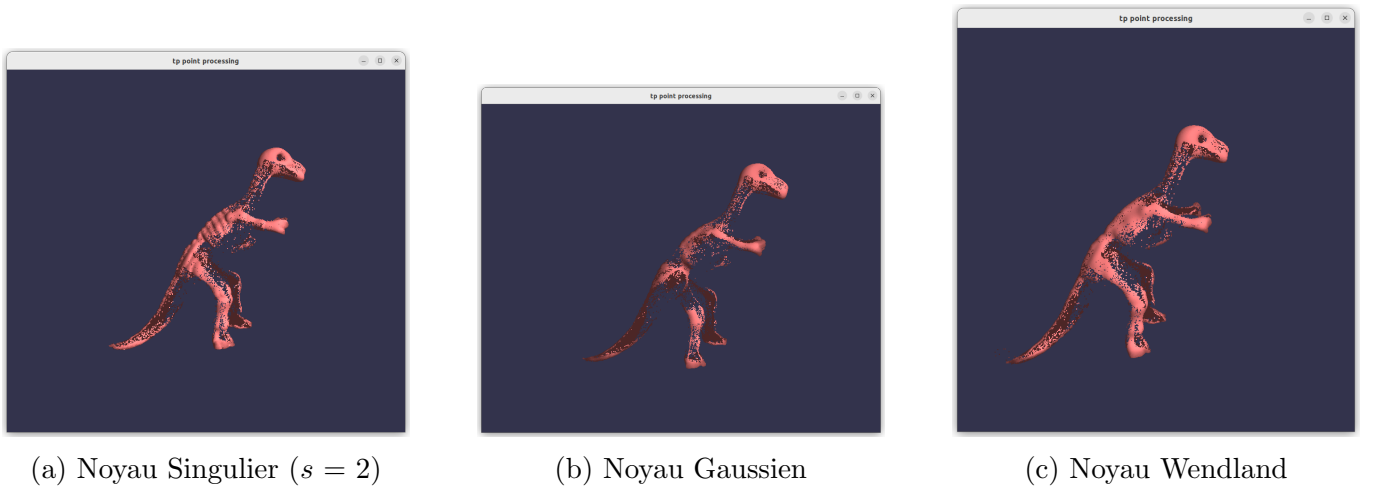


Figure 2: Résultats des différentes projections SPSS selon les types de noyaux

Le noyau Gaussien produit des surfaces plus précises mais moins lisses, tandis que le noyau Wendland a produit des surfaces plus douces. Enfin, le noyau Singulier lisse beaucoup les points de sorties, donnant lieu à des approximations visibles, mais ici il semble être le meilleur compromis.

5 Méthode HPSS

5.1 Description de l'algorithme

L'algorithme HPSS est similaire à SPSS, mais avec une étape supplémentaire pour prendre en compte la direction des normales des voisins lors de la projection. HPSS utilise une méthode plus avancée de projection qui projette d'abord chaque point p_i sur son plan tangent défini par sa normale n_i . Les étapes principales de HPSS sont les suivantes :

1. **Projection locale** : Chaque point voisin p_i est projeté sur son plan tangent :

$$\tilde{p}_i(x) = x - ((x - p_i)^T n_i) n_i$$

Cela permet de prendre en compte la géométrie locale plus précisément.

2. **Calcul du centroïde et de la normale** : Comme dans SPSS, on calcule ensuite le centroïde $c(x)$ et la normale $n(x)$ en utilisant les points projetés :

$$c(x) = \frac{\sum_{p_i \in \text{NN}(x)} w_i \tilde{p}_i(x)}{\sum_{p_i \in \text{NN}(x)} w_i}$$

$$n(x) = \frac{\sum_{p_i \in \text{NN}(x)} w_i n_i}{\sum_{p_i \in \text{NN}(x)} w_i}$$

3. **Projection du point** : Le point d'entrée x est ensuite projeté sur le plan défini par $c(x)$ et $n(x)$, en répétant plusieurs fois l'itération pour améliorer la précision.

5.2 Résultats

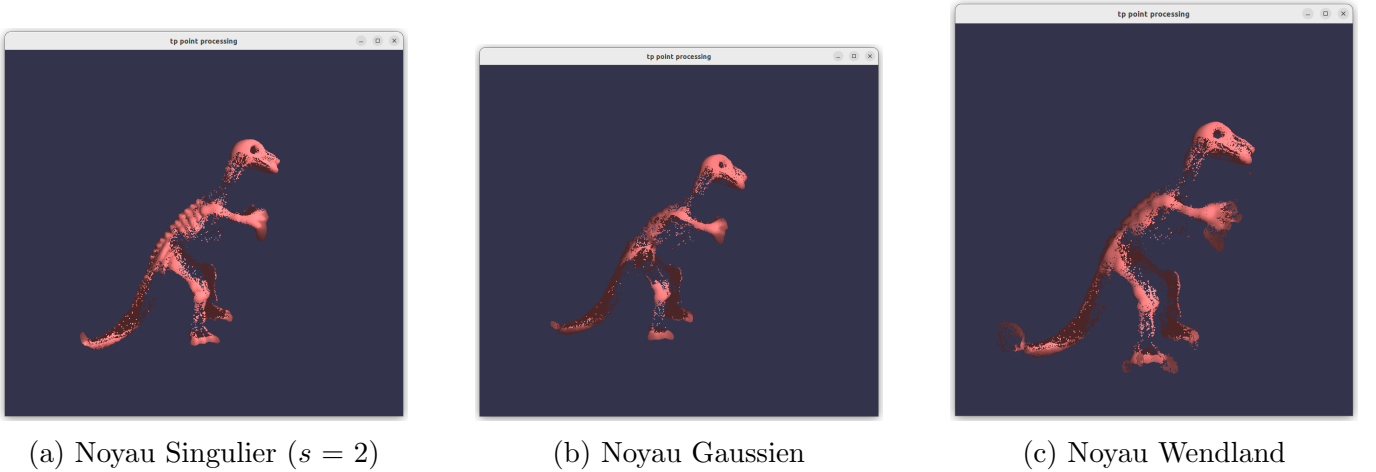


Figure 3: Résultats des différentes projections HPSS selon les types de noyaux

6 Ajout de bruit sur le nuage de points d'entrée

Pour cette dernière partie, il est proposé d'étudier l'impact du bruit appliqué aux points d'entrée sur les points de sortie.

Tout d'abord, pour générer du bruit, j'ai initialisé un générateur de nombres aléatoires compris dans $[-\alpha, \alpha]$ (avec α à définir dans le programme). J'ai ensuite perturbé la position de chaque point d'entrée en la déplaçant dans le sens de sa normale, avec une magnitude α .

```
1 void noiseAlongNormal(std::vector<Vec3> &points, std::vector<Vec3> &normals,
2 double range) {
3     std::mt19937 rng(time(NULL));
4     std::uniform_real_distribution<double> gen(-range, range);
5     for(int i = 0; i < points.size(); ++i) {
6         double randomNumber = gen(rng);
7         points[i] += randomNumber * normals[i];
8     }
9 }
```

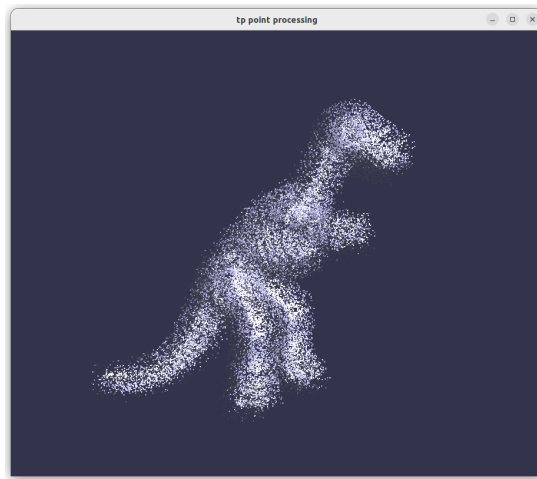
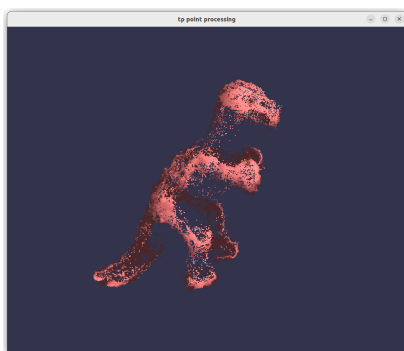
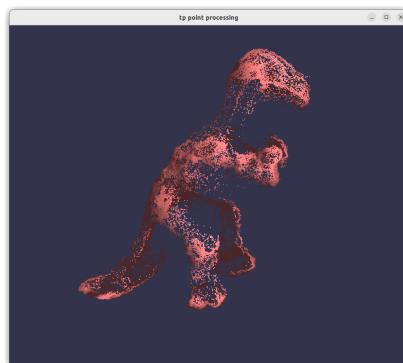


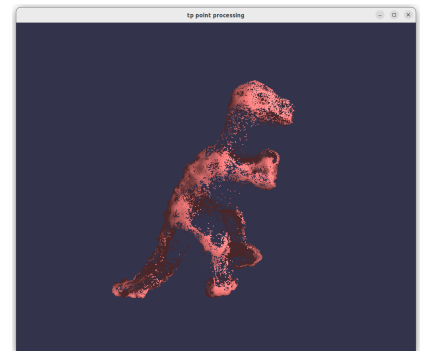
Figure 4: Nuage de points d'entrée observé avec $\alpha = 0.05$



(a) Noyau Singulier ($s = 2$)



(b) Noyau Gaussien



(c) Noyau Wendland

Figure 5: Résultats des différentes projections HPSS selon les types de noyaux sur des données bruitées avec $\alpha = 0.05$

On observe que le noyau Singulier est moins robuste au bruit, le noyau Gaussien lisse assez bien les perturbations liées au bruit, et le noyau Wendland semble être le meilleur choix lorsque du bruit est présent dans les données, respectant au mieux la géométrie d'origine.

7 Conclusion

Au cours de ce TP, nous avons implémenté et comparé les méthodes SPSS et HPSS pour le traitement de nuages de points 3D. En explorant différents noyaux de poids tels que le Singulier, le Gaussien et le Wendland, nous avons observé comment chacun influence la projection des points et la reconstruction de la surface.

Les expériences menées ont montré que :

- Le noyau Singulier tend à lisser fortement les points de sortie, ce qui peut conduire à une perte de détails géométriques.
- Le noyau Gaussien offre un bon compromis entre lissage et préservation des détails.
- Le noyau Wendland s'est révélé le plus efficace pour traiter des données bruitées, préservant la géométrie originale tout en lissant les anomalies.

La comparaison entre SPSS et HPSS a également mis en évidence que HPSS, en tenant compte des normales des points voisins lors de la projection, permet une reconstruction plus fidèle de la surface.

En conclusion, les méthodes SPSS et HPSS constituent des outils puissants pour le traitement et la reconstruction de surfaces à partir de nuages de points. Le choix du noyau de poids et la gestion des paramètres jouent un rôle crucial dans la qualité du résultat final.

Pour poursuivre ce travail, il serait intéressant d'implémenter APSS, qui se base sur des projections non pas sur des plans mais sur des sphères algébriques.