

TP 3 : Neighborhood descriptors

NPM3D - Février 2022

Mehdi Zemni & Chady Raach

mehdi.zemni@student-cs.fr chady.raach@student-cs.fr

A. Normals in CloudCompare

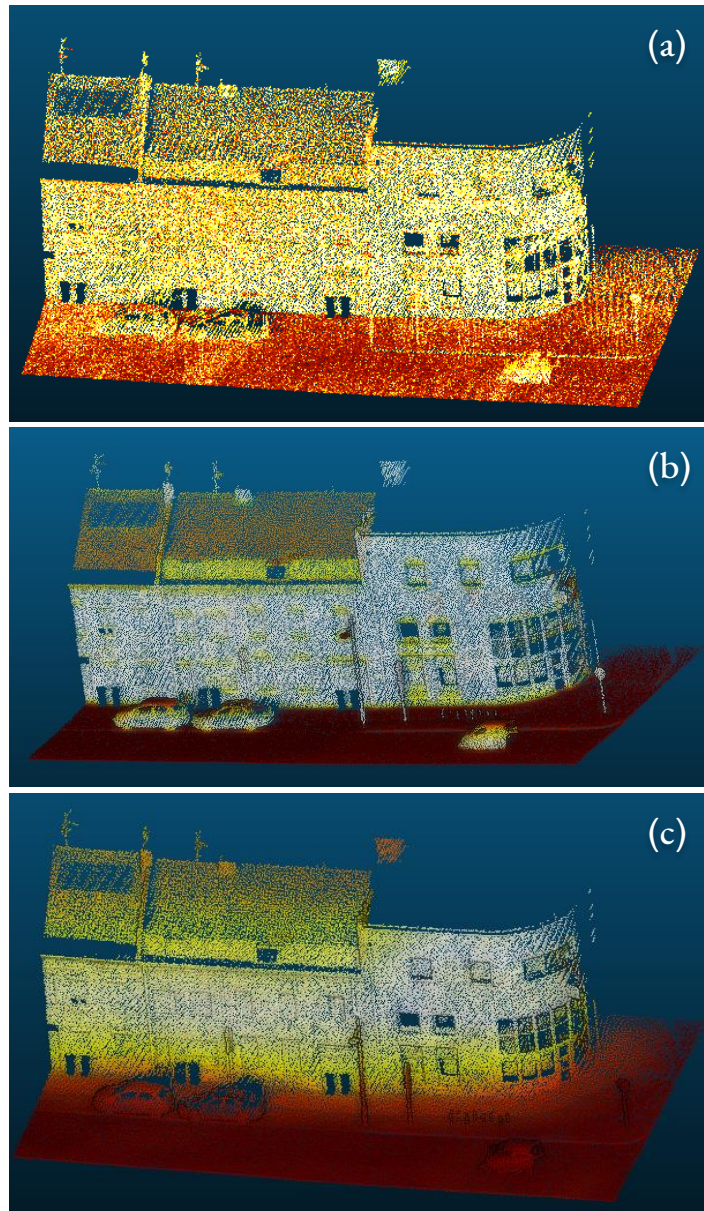


Figure 1 Normales obtenus avec CloudCompare avec un rayon de voisinage de (a) 5cm (b) 50cm et (c) 5m

- 1) Lorsque le rayon de voisinage est trop petit (5cm), nous obtenons des normales très bruitées car le nombre de points voisins au query point n'est pas suffisant pour avoir une estimation précise de la normale. Si le rayon est trop grand (5m), on utilise un voisinage grand pour estimer les normales et on observe un phénomène de lissage des normales. On voit sur la fig.1c que le toit et la façade du bâtiment ont presque les mêmes orientations des normales, pareil pour la rue et les voitures. En effet, les points appartenant à une voiture ont comme voisins des points appartenant à la voiture mais aussi à la rue. Ceux-ci sont prépondérants et impactent considérablement l'orientation estimée des normales sur la voiture.

En revanche sur la fig1.b ou on a utilisé un rayon de 50cm, on voit que les directions des normales sont mieux estimées même au niveau des bords et des angles car le rayon choisi est plus petit que l'ordre de grandeur des détails sur le nuage de point.

- 2) Une bonne façon de choisir le rayon serait de le prendre aussi petit que le plus gros détail que nous voulons analyser.

B. Local PCA and normal computation in Python

- 3) Le vecteur propre ayant la valeur propre la plus petite est le vecteur qui estime la normale à la surface puisque la valeur propre mesure la diffusion des points dans la direction du vecteur propre et cette diffusion doit être minimale dans la direction de la normale.
- 4) On utilise la fonction `compute_local_pca` qu'on a codé pour calculer le vecteur propre ayant la plus petite valeur propre pour chaque matrice de covariance de chaque point du nuage en utilisant 50 cm comme rayon pour le voisinage.

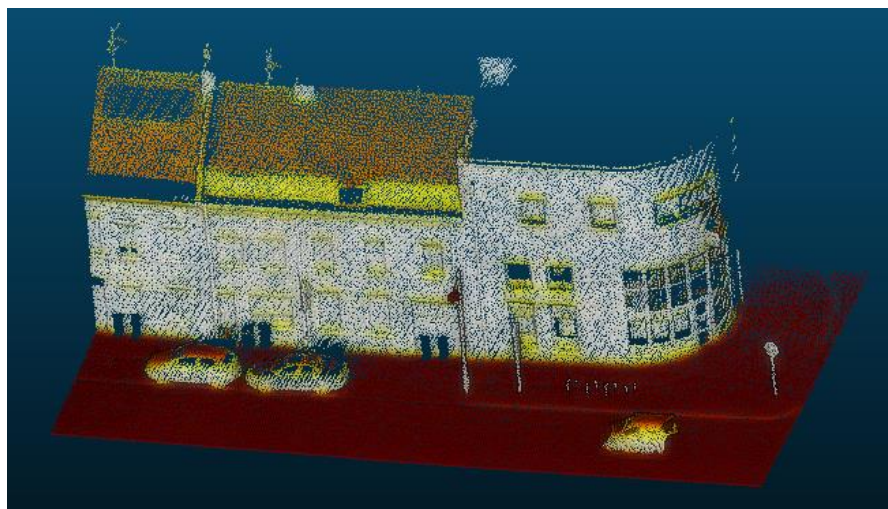


Figure 2 Normales obtenus avec 'Compute_local_PCA' et converties en champ scalaire "Dip" dans CloudCompare avec un rayon 50cm

5) On utilise cette fois un voisinage K-NN au lieu d'un voisinage avec rayon

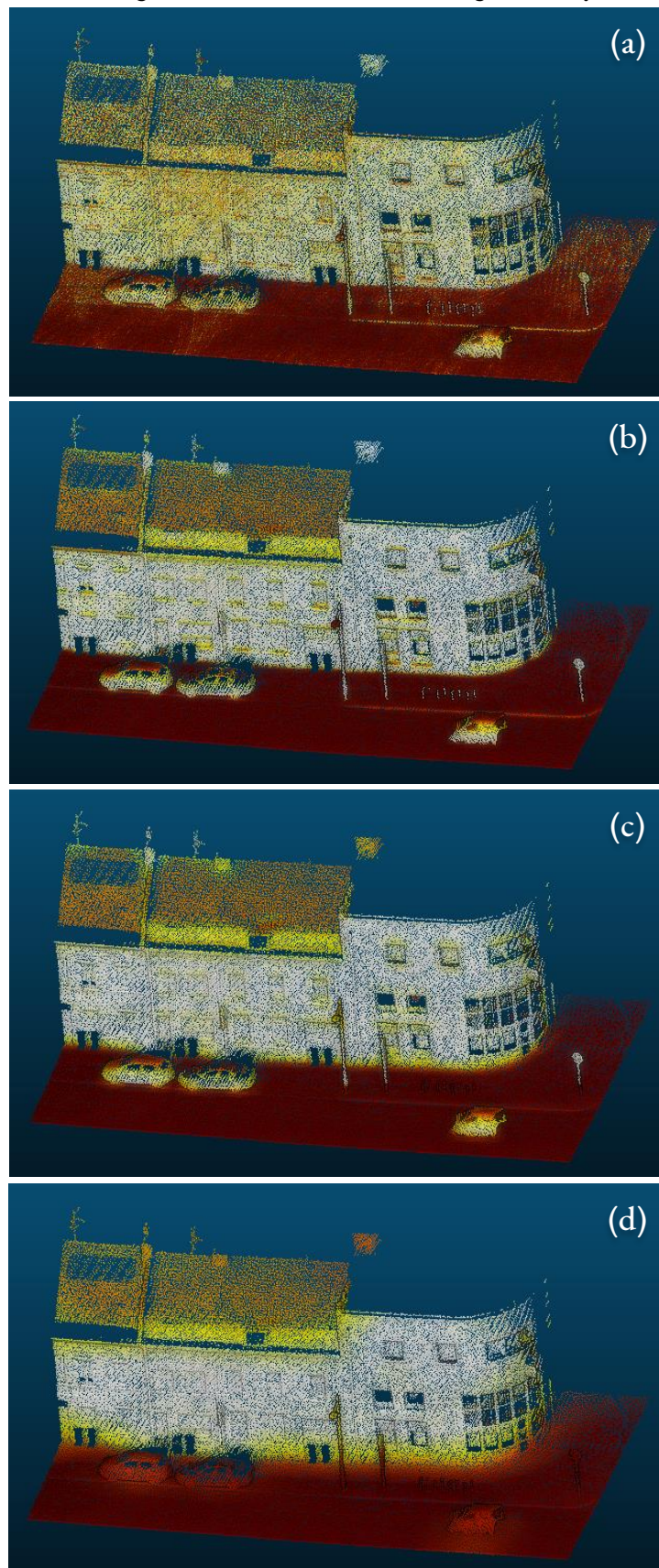


Figure 3 Normales obtenus avec 'Compute_local_PCA_knn' et converties en champ scalaire "Dip" dans CloudCompare avec (a) $k=5$, (b) $k=50$, (c) $k=500$ et (d) $k=5000$

Les mêmes observations de la première question s'appliquent lorsqu'on utilise un voisinage K-NN pour calculer les vecteurs propres de la matrice de covariance du voisinage à chaque point du nuage. Avec un nombre de voisins trop petit ($k=5$, fig.3 (a)), on observe un bruit dans l'estimation de la normale. Un nombre de voisins $k=50$ semble optimal pour ce nuage de point car on arrive à distinguer les bords nettement (bords des fenêtres, bords entre le trottoir et la façade du bâtiment). La différence par rapport au voisinage sphérique est que le voisinage K-NN n'est pas limité dans l'espace dans une boule de rayon fixe mais déterminé par le nombre de voisins.

C. Going further (BONUS)

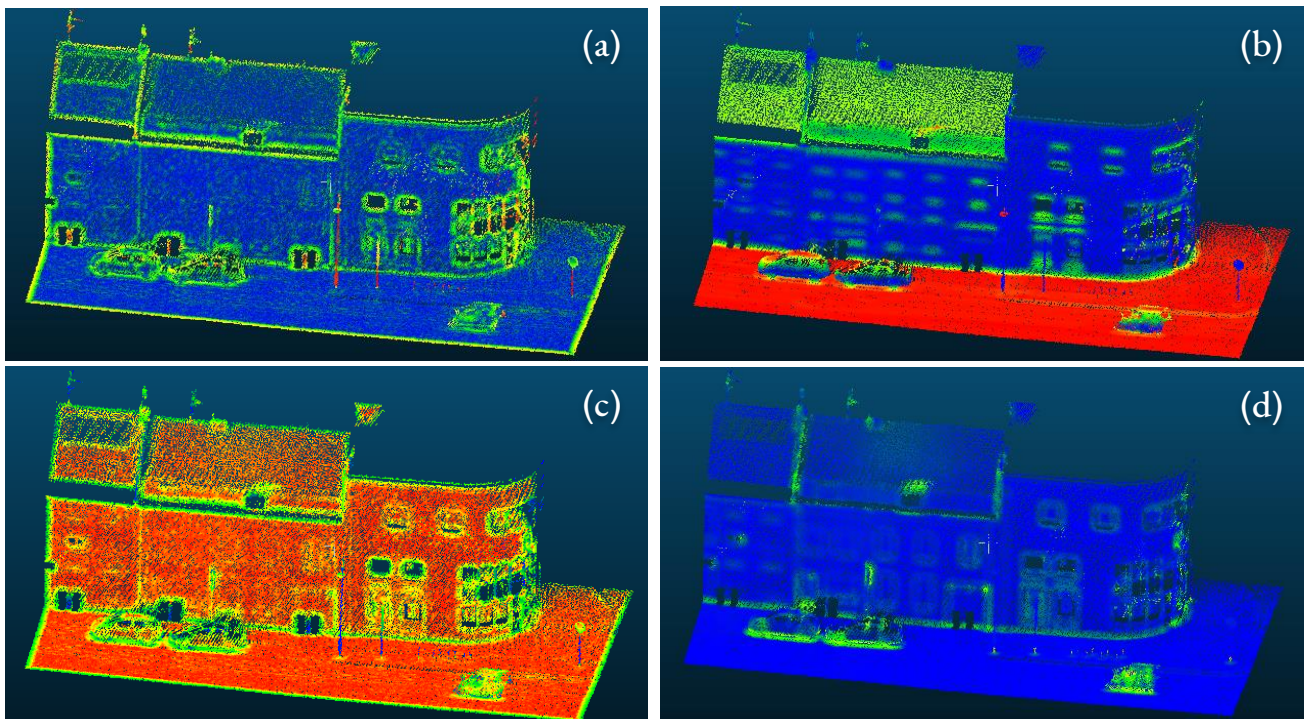


Figure 4 4 features as scalar fields (a) verticality, (b) linearity, (c) planarity, (d) sphericity

- La linéarité est égale à $1 - \lambda_2 / \lambda_1$, elle mesure la dispersion des points le long du deuxième axe principal par rapport au premier et donc si le coefficient de linéarité est grand alors la dispersion selon le premier axe est beaucoup plus importante et donc les points sont dispersés le long d'un axe (forme une ligne).
- La planéité mesure s'il y a des points le long de la normale (troisième PCA) comparé au deuxième axe.
- Si la sphéricité est élevée ≈ 1 alors $\lambda_1 \approx \lambda_3 \approx \lambda_2$, ce qui signifie que les points sont dispersés de manière isotrope (il n'y a pas de direction principale spécifique). Les points se trouvent donc distribués dans une sphère (La sphéricité est élevée dans les structures non planes comme les voitures).