NPM3D - TP 3: Neighborhood descriptors

Adrien Golebiewski

IASD 2022-23

1 Question 1

The normal estimation is approximating the surface with a neighborhood chosen inside a sphere of a certain radius. If the radius is too small, only noise will appear ; but if the radius is too big, the approximation is too big and we loose information on the surface of the objects.

We see on fig.1 that the roof and the facade of the building have almost the same orientation of the normals, same for the street and the cars. Indeed, the points belonging to a car have as neighbors points belonging to the car but also to the street. These are preponderant and impact considerably the estimated orientation of the normals on the car.

On the other hand on the other figures where we used a radius of 50cm, we see that the directions of the normals are better estimated even at the level of the edges and the corners because the chosen radius is smaller than the order of magnitude of the details on the point cloud.

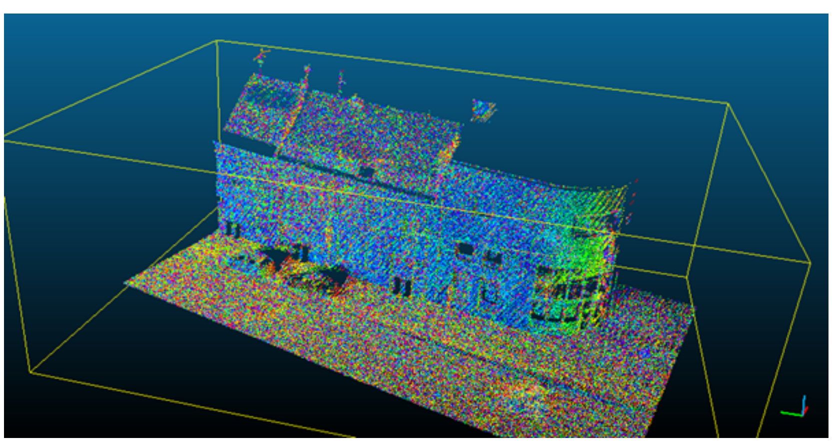

Figure 1: Dip scalar field with a radius of 2 cm

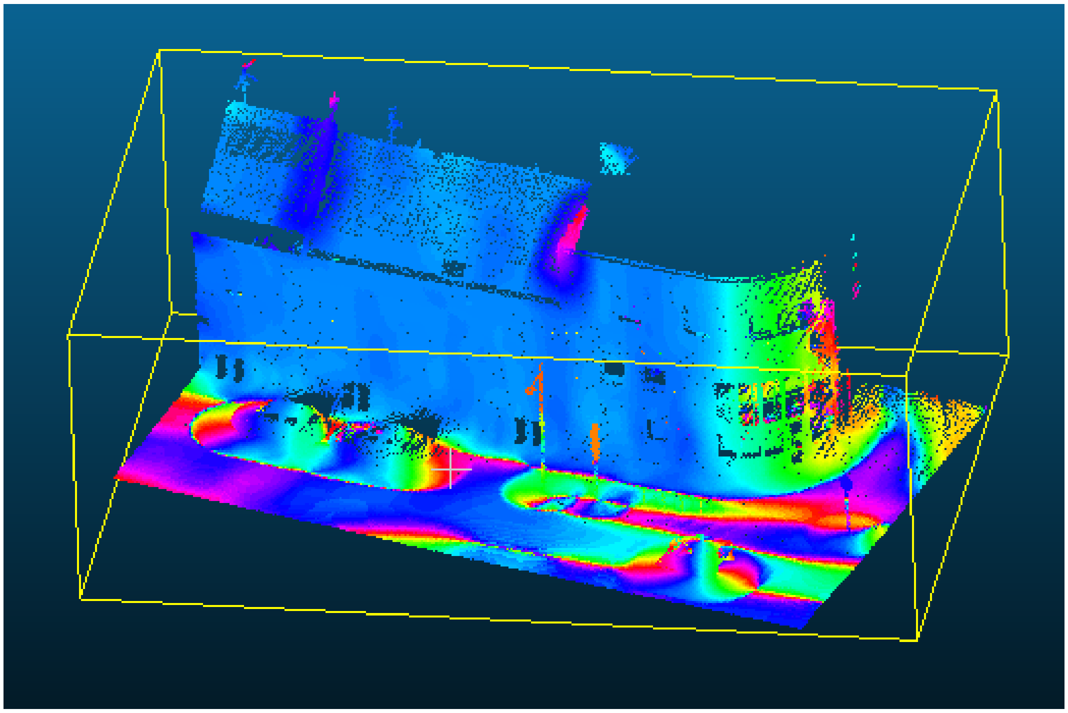

Figure 2: Dip scalar field with a radius of 2 m

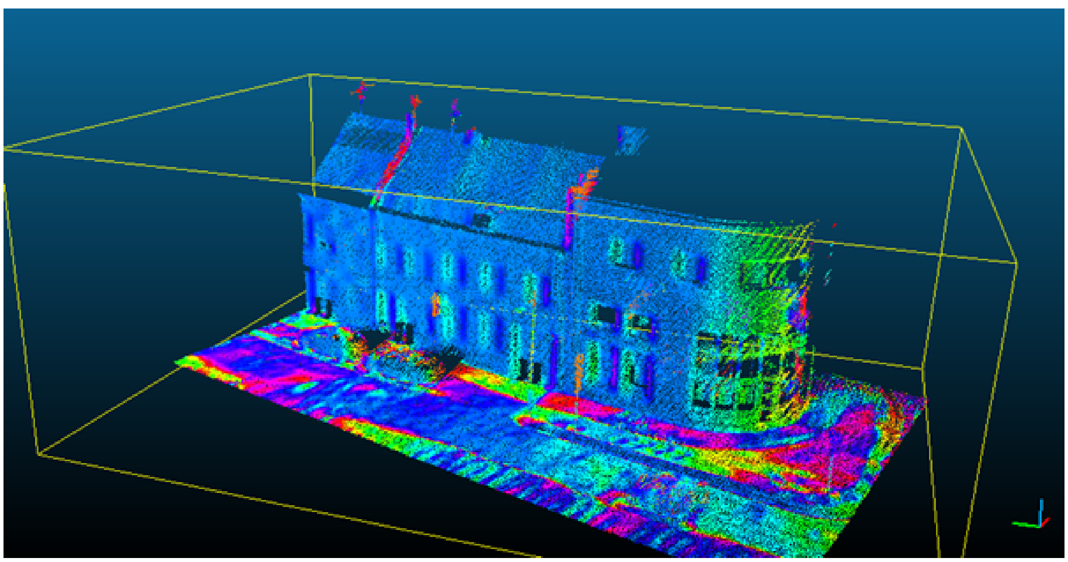

Figure 3: Dip scalar field with a radius of 50cm

2 Question 2

To choose the neighborhood scale for a good normal estimation, we should decide what is the order of magnitude of the smallest object we want to approximate the surface in the scene. 50cm is a reasonable order of magnitude to be able to see the windows, the cars, details on the roof.

3 Local PCA and normal computation in Python

After apply our PCA to Lille street small.ply cloud, we obtain the same results as expected :

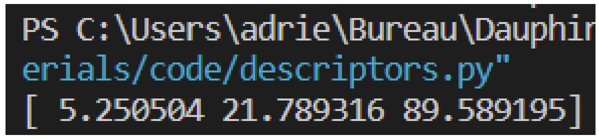

Figure 4: PCA verification results

4 Which one of the three eigenvectors is expected to be the normal of the surface?

The eigenvector with the **smallest eigenvalue** is the vector that estimates the normal to the surface since the eigenvalue measures the scattering of the points in the direction of the eigenvector. This scattering must be minimal in the direction of the normal.

5 Question 3

We use the compute local pca function that we have coded to compute the eigenvector with the smallest eigenvalue for each covariance matrix of each point of the cloud using using 50 cm as the radius for the neighborhood.

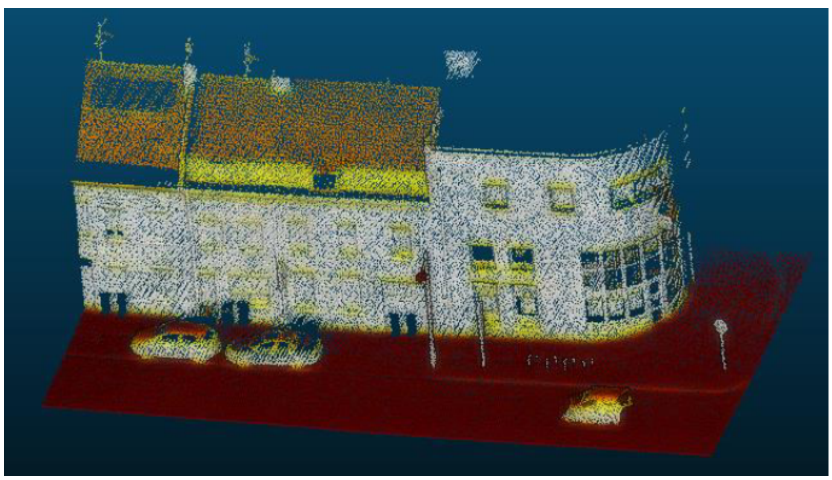

Figure 5: Dip scalar field of the normals computed with our local PCA algorithm

6 Question 4

We change the neighborhood definition of our normal by using k nearest neighbors instead of a radius.

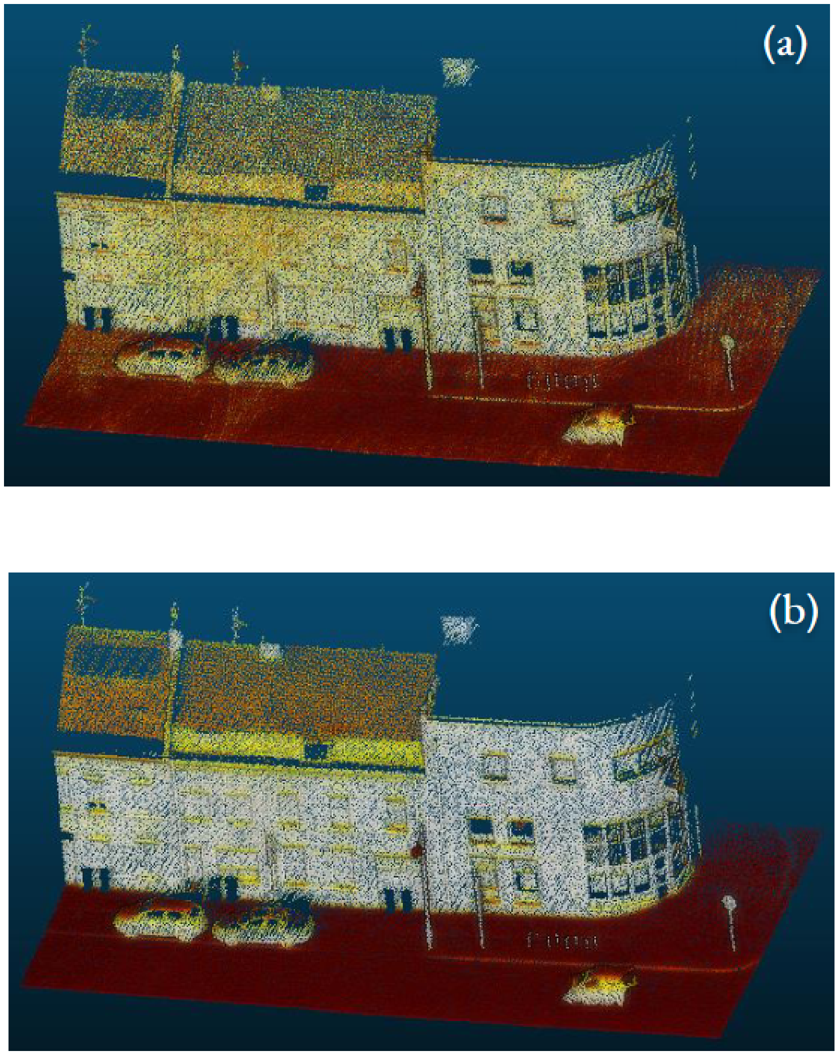

Figure 6: Normals obtained with 'Compute local PCA knn' and converted to scalar field "Dip" in CloudCompare with (a) k=5, (b) k=30

The same observations of the first question apply when using a KNN neighborhood to compute the eigenvectors of the neighborhood covariance matrix at each point of the cloud. With a too small number of neighbors (k=5, fig.3 (a)), we observe noise in the estimation of the normal. A number of neighbors k= 50 seems optimal for this point cloud because we can distinguish the edges clearly (window edges, edges between the sidewalk and the building facade. The difference with the spherical neighborhood is that the K-NN neighborhood is not spatially limited in a ball of fixed radius but determined by the number of neighbors.

7 Bonus

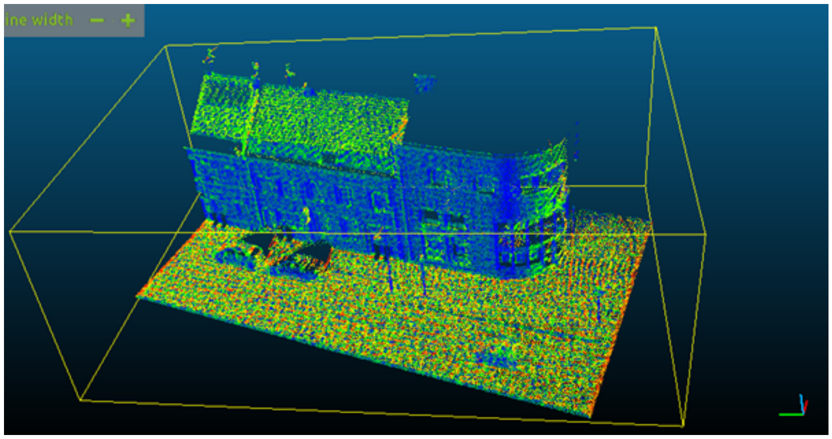

Figure 7: Verticality scalar field

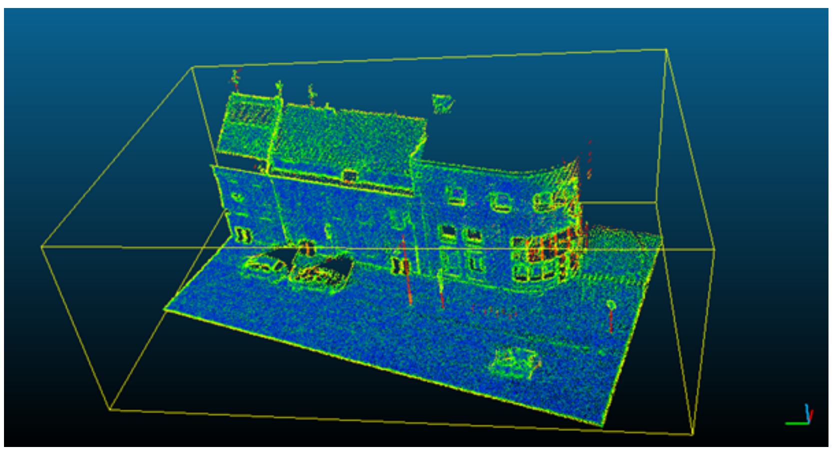

Figure 8: Linearity scalar field

Linearity is a scalar that takes values between 0 and 1. When linearity goes to zero, we can verify that the two biggest eigenvalues are of the same order of magnitude, so the points are equally distributed in the two directions. When the linearity is close to 1, lambda2 is small compared to lambda1. So the point clouds has more variability along one direction, which is typical of a line, like the streetlamps' rods.

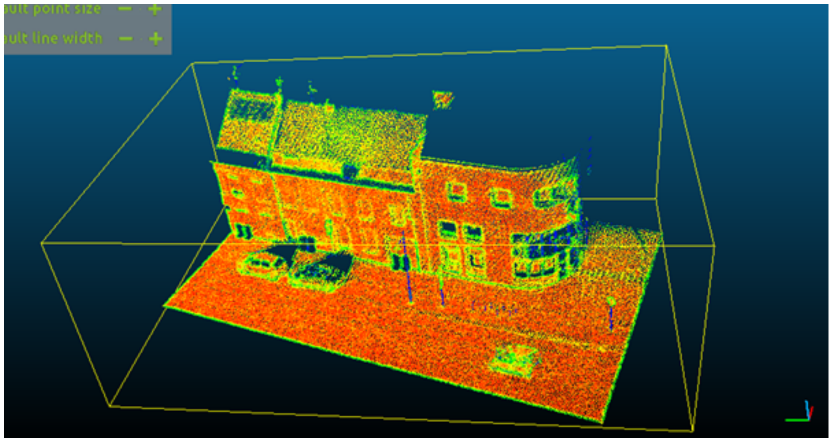

Figure 9: Planarity scalar field

Planarity is a scalar between 0 and 1. It is close to 0 when lambda2 is close to lambda3. In this case, the local variability of the point cloud is mainly along one direction (like for a thin line or a stick). It is close to 1 if lambda2 is close to lambda1, and lambda3 is small compared to them. In this case, the point cloud are distributed uniformly along the two main directions, given a planar shape. We can see with the streetlamps' rods that the planarity is close to 0 for example, whereas it is close to 1 on the walls on the on ground (planar shape).

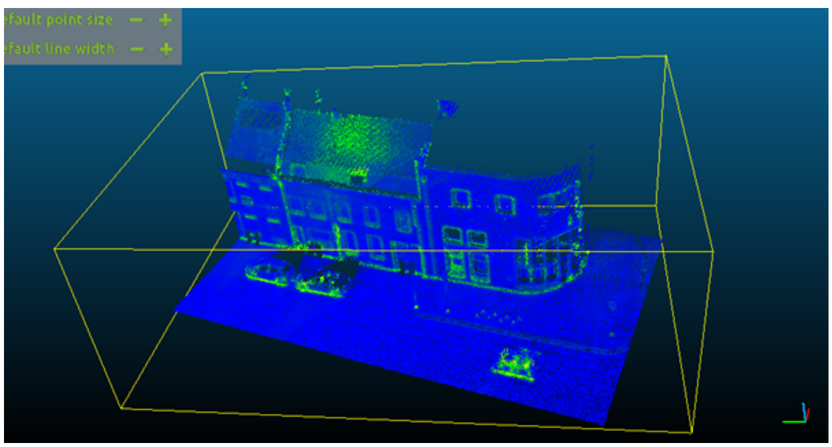

Figure 10: sphericity scalar field

The sphericity is a scalar between 0 and 1. When it is close to 0, the normal direction's eigenvalue is small compared to the biggest eigenvalue, which corresponds more to a planar shape. When it is close to 1, all the eigenvalues will have around the same value (lambda2 is between lambda3 and lambda1). So we have points clouds distributed in the same way in all directions, which correspond to the shape of a sphere or a ball. We can see that on the wheels of the cars for instance, slightly on the gutters or on the top of the roof.