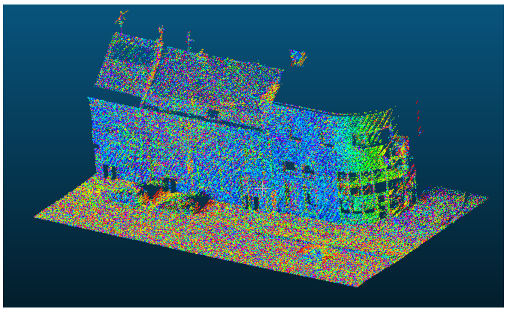
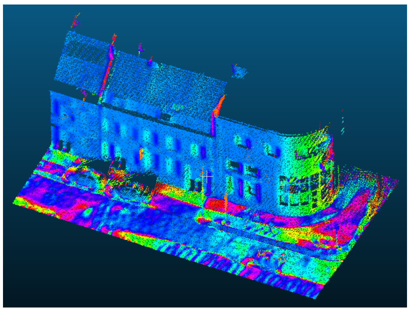
Mines Paris — PSL
Intelligence Artificielle, Systèmes et Données (IASD)
Nuages de Points et Modélisation 3D
Students: Alexandros Kouvatseas & Luka Lafaye de Micheaux

Question 1: If you use a too small / too big radius, what is the effect on the normal estimation? Use screenshots to support your claims.

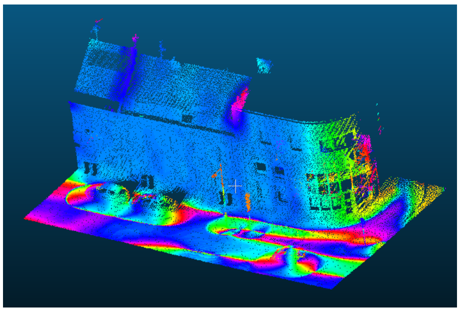
By observing *figure 1*, for each radius used, we can come to the conclusion that if it's too small, the normals tend to be very sensitive to noise and small variations in the point cloud, leading to a very rough and noisy appearance, as seen in the first screenshot (a). On the other hand, if the radius is too large, the estimated normals become overly smooth, and finer details are lost, as seen in the last image (c). A moderate radius (such as in the second image (b), radius = 0.5) balances detail preservation and smoothness, providing a more accurate normal estimation.



(a)



(b)



(c)

Figure 1: (a): Normals of the Lille_street_small.ply file, with radius = 0.05m of neighborhoods, (b): Radious = 0.5m, (c): Radious = 2m

Question 2: How would you choose the neighborhood scale for a good normal estimation?

To choose a good neighborhood scale for normal estimation, we need to balance between capturing enough points for a stable normal and not making it too large that details get lost. If the radius is too small, the normals will be very noisy and sensitive to tiny variations. If it's too big, everything gets smoothed out, and sharp edges disappear. A good approach is to test different values and look at how smooth or noisy the normals appear, but in most general cases, radious = 0.5 could be the most balanced solution between the two.

Question 3: Show a screenshot of your normals converted as "Dip" scalar field in CloudCompare with radius = 0.50m.

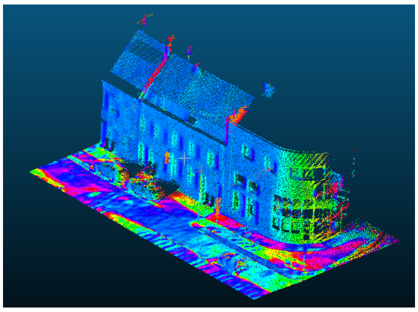


Figure 2: Normals converted as "Dip" scalar field obtained by the results of calculating the eigenvalues and eigenvectors of each query point neighborhood in the cloud.

Question 4: Change the neighborhood definition of your normal by using k nearest neighbors instead of a radius. Show a screenshot of your normals converted as "Dip" scalar field in CloudCompare with k = 30. What are the differences with the radius method and why?

By looking at *figure 2* and *figure 3* we observe that there are some key differences in smoothness and consistency. The radius method produces smoother normals, particularly on large flat surfaces like the ground and the building facades, while the kNN method introduces local variations, especially in areas with non uniform point densities. This is because the radius method adapts to the structure of the point cloud while kNN always selects a fixed number of neighbors, leading to more fluctuations in regions where the point density varies. In conclusion, the radius based normals are more stable, but kNN preserves some finer details that may lead to more noise.

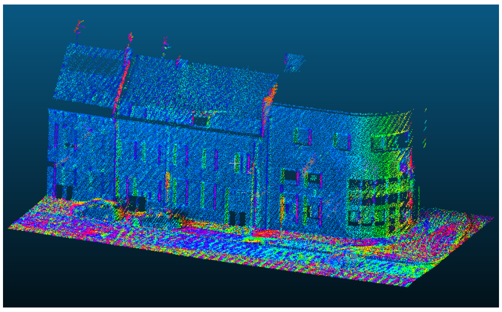
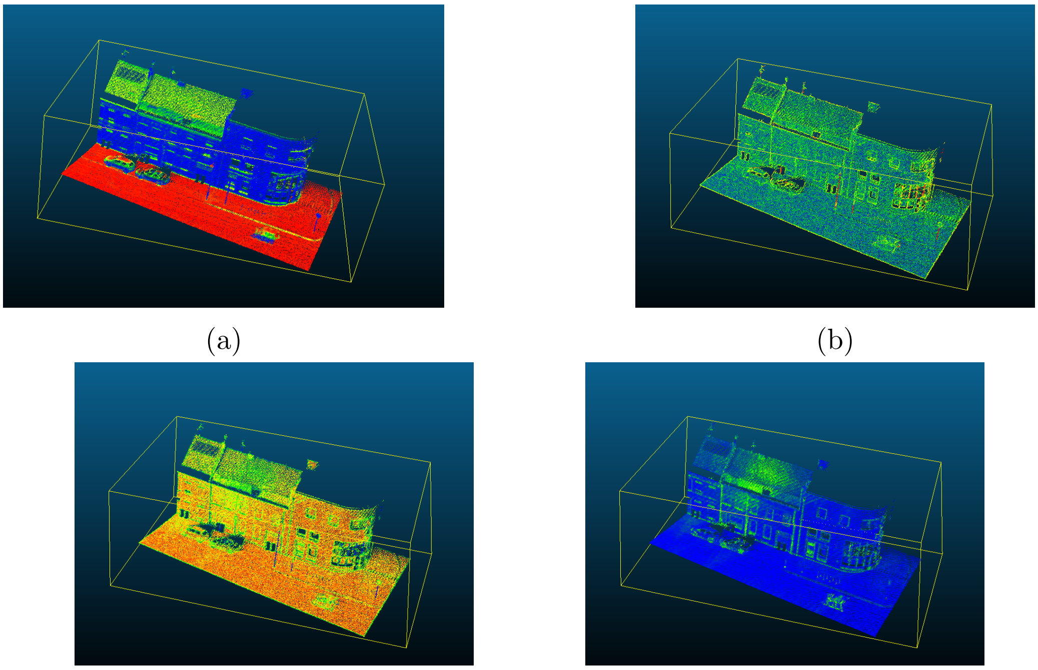


Figure 3: Normals computed by calculating local PCA using 30 knn

Question Bonus: Show screenshots of the 4 features as scalar fields of the point cloud. Can you explain briefly the names of the 3 last features considering their definition with eigenvalues?

Linearity measures how much the neighborhood resembles a line, so if 1 is much bigger than 2, it will become 1. On the other hand, **Planarity** quantifies how well the neighborhood fits a plane. It will be high if 2 is much bigger than 3. Finally **Sphericity** indicates isotropic distribution in all directions. It will be high if 3 is approximately equal to 1.



(a) (b)

(c) (d)

Figure 4: Scalar fields of point cloud of each feature: (a) verticality, (b) linearity, (c) planarity and (d) sphericity .