Balthazar Neveu — Jamy Lafenetre
balthazarneveu@gmail.com —
jamy.lafenetre@ens-paris-saclay.fr

Lab session # 3
NPM 2024

01/28/24

Question 1

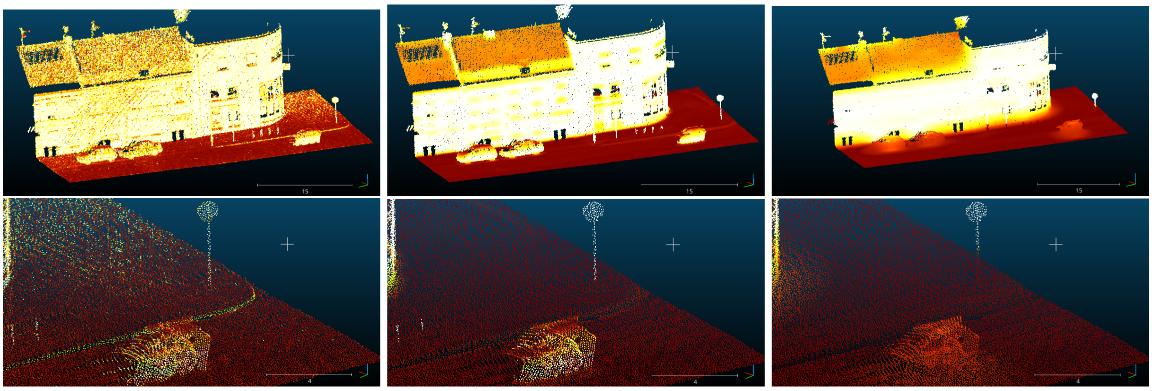

Figure 1: From left to right: 10cm, 50cm, 2m radiuses used to compute normals

If the neighborhood radius is too large, the observed geometry is no longer local and the normals are therefore oversmoothed. Figure 1 highlights this phenomenon: the normals of the car and of the stop sign pole cannot be recovered accurately when the radius is too large. If the neighborhood radius is too small, the estimated normals are inaccurate because of the noise, but also because of the local variance separating samples. Independantly of the measurement noise, samples are indeed scattered randomly and separated by a certain distance; estimations based on a narrow neighborhood can therefore be prone to aliasing, which can appear as structured artefacts in the normal estimations. As highlighted by Robert Cook's "Stochastic Sampling in Computer Graphics ", aliasing in such a stochastic sampling setting can also translate as a white noise in the normal estimation. This is visible in Fig 1 where both structured pattern and noise appear in the normals where the radius is too short to dealias fast local variations (textures such as the tiled roof or the cars are good examples).

Question 2

The radius choice depends on the application's constraints. If the goal is performance, you may trade a bit of larger radiuses for a faster computation. If the goal is quality (subjective term), you may consider larger radiuses. The optimal radius must satisfy a tradeoff between noisy normals and smoothing. There's at least a hard constraint where the radius must at least allow to get 3 points in each neighborhood to compute the normal.

Question 3

See Figure 2. When comparing side by side with the normals of CloudCompare, the results were similar (*this merely acts as a unitary test*)...

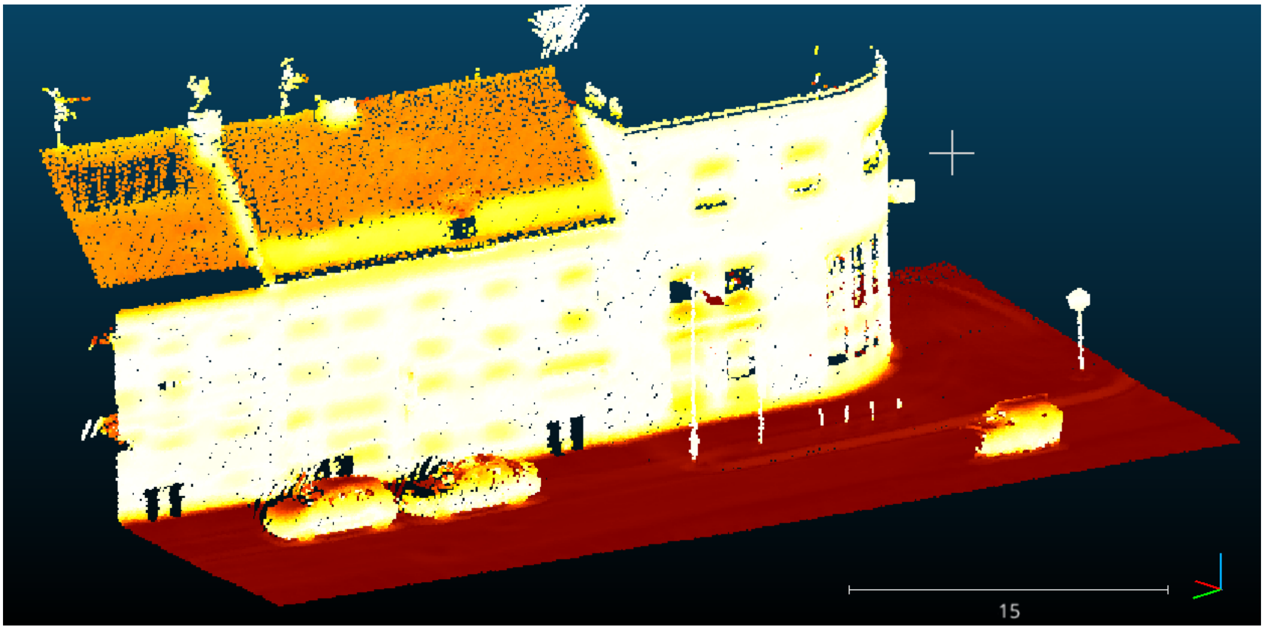

Figure 2: 50cm radiuses local PCA used to compute normals

Question 4

We compare our results for a radius of 50cm or for $k = 30$ neighbours in Figure 3. The $k - nn$ methods produced more accurate results, and is able to recover the step of the sidewalk as well as the building corner. We posit that this is due to the localy-variable density of points: samples can be more sparse in flat areas than on textures. Therefore, $k - nn$ **acts like an adaptive radius**, and can naturally integrate a narrower area around details, where samples are abundant). The non uniformity of the sample density can also be explained by the sampling process of the sensor.
In Figure 4, we reduce the radius to 10cm and the number of neighbours to 6. In this setting, $k - nn$ is less aliased because it is able integrate a sufficient amount of points, contrary to the radius method that cannot properly fit a plane.

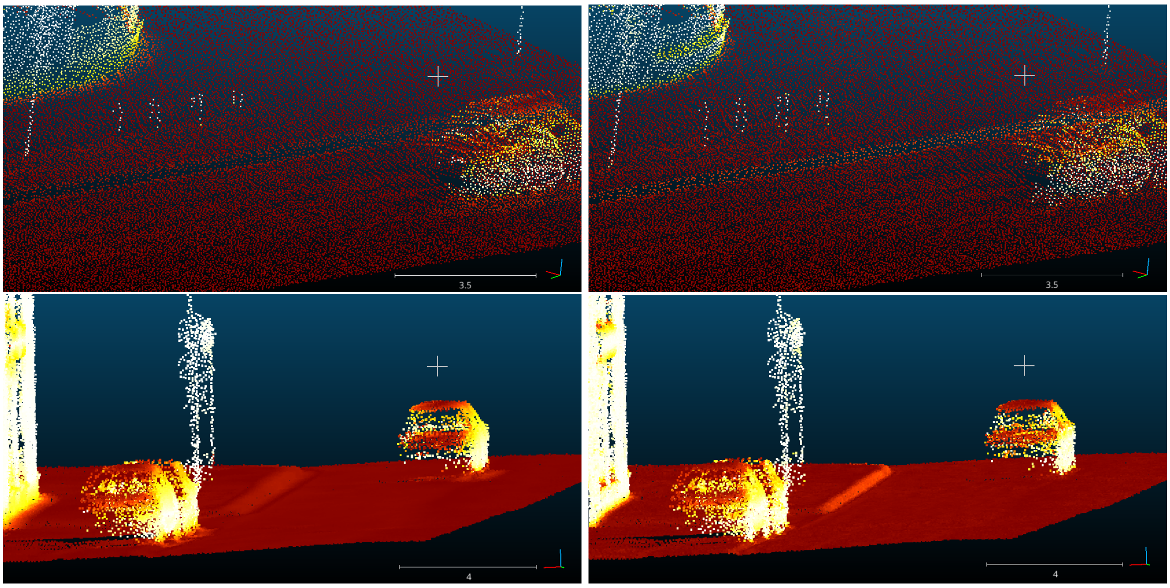

Figure 3: Left: 50cm radius local PCA used to compute normals. Right: 30 nearest neighbors used to compute normals.

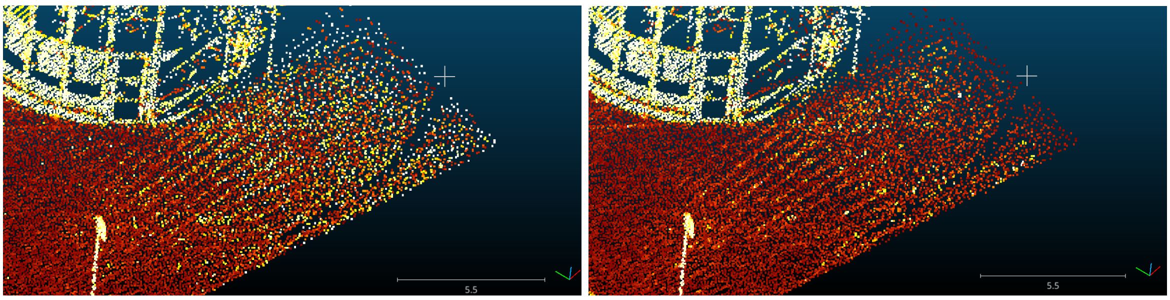

Figure 4: Left: 10cm radius local PCA used to compute normals. Right: 6 nearest neighbors used to compute normals.

Question Bonus

We represent on Figure 5 the verticality which describes, as suggested by the name, the verticality of the predicted normals. On Figure 6, we simultaneously represent the linearity, the planarity and the sphericity. The linearity is maximal when a single eigenvalue numericaly dominates the 2 others, therefore indicating the presence of an elongated 1D vectorial space (also commonly called line). The planarity is maximal when 2 eigenvalues are equivalently scaled, while numerically dominating the third, indicating a plane. The sphericity is maximal when all eigenvalues are numerically comparable, indicating that there is no favorised axis. This is typically the case for corners, textured areas, and objects of very small size. Figure 6 highlighs how these descriptors do indeed describe the aforementioned features.

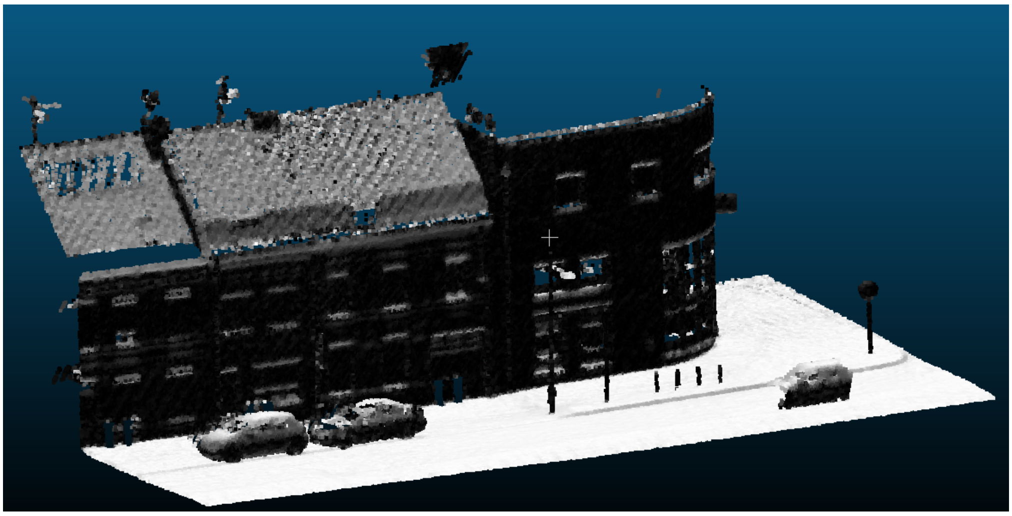

Figure 5: ”verticality” descriptor computed for a radius of 20cm. Dark shades indicate a verticality of 0 and white shades a verticality of 1. The descriptor accurately indicates surfaces orthogonal to the vertical axis.

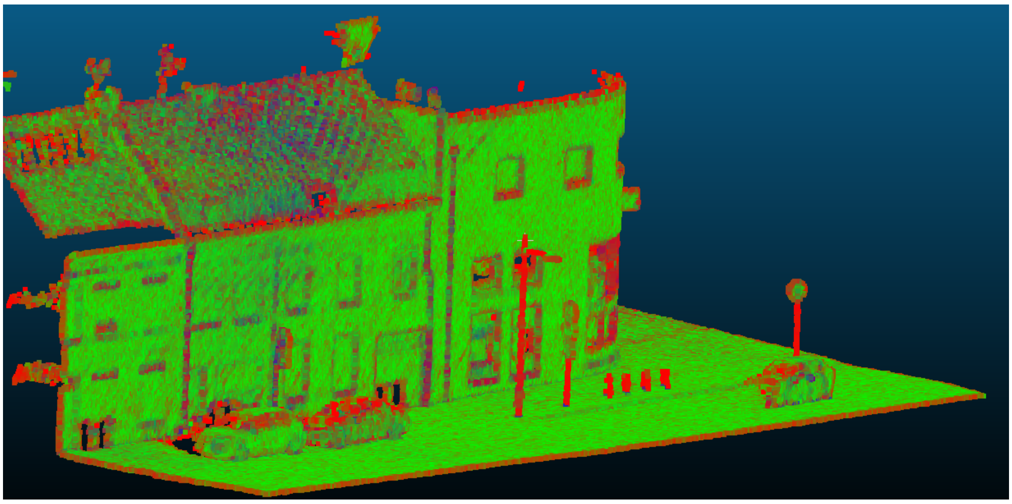

Figure 6: Local image descriptors computed for a radius of 20cm. The linearity (represented in red) accurately describes elongated objects such as poles. The planarity (in green) accurately represents flat areas. The sphericity (in blue) represents corners, textures and small objects such as a tiled roof, the base of pole and car mirrors. Note that the 3 descriptors never overlap, by design.