TP 4: Surface Reconstruction

Objectives

- Test Poisson Surface Reconstruction method in CloudCompare
- Surface Reconstruction in Python : implement the Hoppe/IMLS implicit function

The report should be a pdf containing the answers to the **Questions** and named "TPX_LASTNAME.pdf". Your code should be in a zip file named "TPX_LASTNAME.zip". You can do the report as a pair, just state both your names inside the report and in the pdf and zip filenames, like "TPX_LASTNAME1_LASTNAME2.pdf"
Send your code along with the report to the email mva.npm3d@gmail.com. The object of the mail must be "[NPM3D] TPX LASTNAME" or "[NPM3D] TPX LASTNAME1 LASTNAME2" if you are a pair working on the report.

A. 3D Reconstruction in CloudCompare

The goal is to test and understand the well-known surface reconstruction method named Poisson on the point cloud "bunny_normals.ply".

1) Open "bunny_normals.ply" in CloudCompare and test the Poisson Reconstruction algorithm with Plugins -> PoissonRecon

2) You can change some parameters of the method with tab "Advanced"

In the file "bunny_normals.ply", the normals have been computing using k nearest neighbors (k=30) and have been oriented to point away from the surface.

Question 1: Take a screenshot of the bunny mesh using the Poisson surface reconstruction

Question 2: Find another point cloud (you can use a point cloud from a previous TP or on Internet) and apply the Poisson reconstruction in CloudCompare. Show a screenshot. Comment on the result: is the reconstruction working?

B. Surface Reconstruction in Python: implement the Hoppe implicit function

The goal is to implement in Python a classical surface reconstruction algorithm based on the Hoppe implicit function.

You need input point clouds with normal: you can use “bunny_normals.ply” with pre-computed normals.

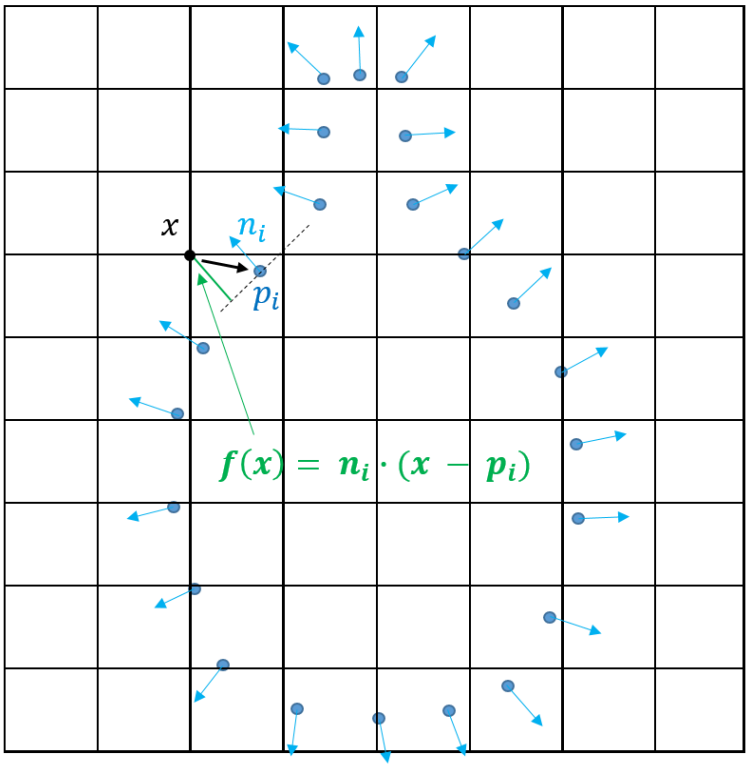

Figure 1 Hoppe implicit function

1) To compute the Hoppe function, we create first a regular grid of the volume space around the input point cloud.

2) Then, on every node x of the grid, the hoppe function is $f(x) = n_i \cdot (x - p_i)$ when p_i is the closest point of the point cloud to x (and n_i the associated normal of p_i).

3) The function f is a scalar field on a regular grid. To build of mesh, we need to extract the iso-zero of f with Marching Cubes. The Python package “scikit-image” has an efficient implementation of Marching Cubes.

4) We can finally export the mesh with the library trimesh and see the result in CloudCompare.

In reconstruction.py, implement the function compute_hoppe to fill the scalar_field with the Hoppe implicit function value.

<u>Question 3:</u> Take a screenshot of the bunny point cloud mesh with the hoppe implicit function (with grid_resolution = 30)

<u>Question 4:</u> Take a screenshot of the bunny point cloud mesh with the hoppe implicit function (with grid_resolution = 100). Comment on the differences with the previous result.

C. Going further (BONUS)

We have seen in the course that Hoppe implicit function is not a continuous surface representation. The Implicit Moving Least Square (IMLS) function is a better implicit function: it is a continuous surface representation.

For any node x, the IMLS function $f(x)$ is defined by :

$$f(x) = \frac{\sum_i \left(n_i \cdot (x - p_i)\right) \theta(x - p_i)}{\sum_i \theta(x - p_i)}$$

With: $\theta(x - p_i) = e^{-\frac{||x-p_i||^2}{h^2}}$

The parameter h is proportional to the noise of the point cloud.
$h = 0.001m$ is a good trade-off for the bunny point cloud.

Instead of computing the function $f(x)$ using every point of the point cloud, you can take only the k nearest neighbors (k = 30 is enough for many point clouds). In reconstruction.py, implement the function compute_imls to fill the scalar_field with IMLS implicit function value.

Compute the Hoppe mesh and IMLS mesh for the Bunny point cloud with grid_resolution = 100. The computation will take around 10 min on a laptop for the IMLS mesh (you can improve the code by vectorizing calculations or computing the function only on intersecting voxels).

<u>Question Bonus:</u> Show with screenshots the differences between the Hoppe surface and the IMLS surface of the Bunny? (with grid_resolution = 100)
Explain the differences between the two meshes.