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 Screened Poisson Reconstruction (SPR) 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: Modify the parameters (octree depth, boundary, samples per node, point weight) to get the "best" reconstruction of the surface from PoissonRecon. Take a screenshot of the result. How many triangles does your final mesh have? Give the parameters that allowed you to have that result. Explain what "better" reconstruction means for this object.
Tips: to improve the quality of your screenshots, you can used EDL in CloudCompare

B. Surface Reconstruction in Python

a. 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.

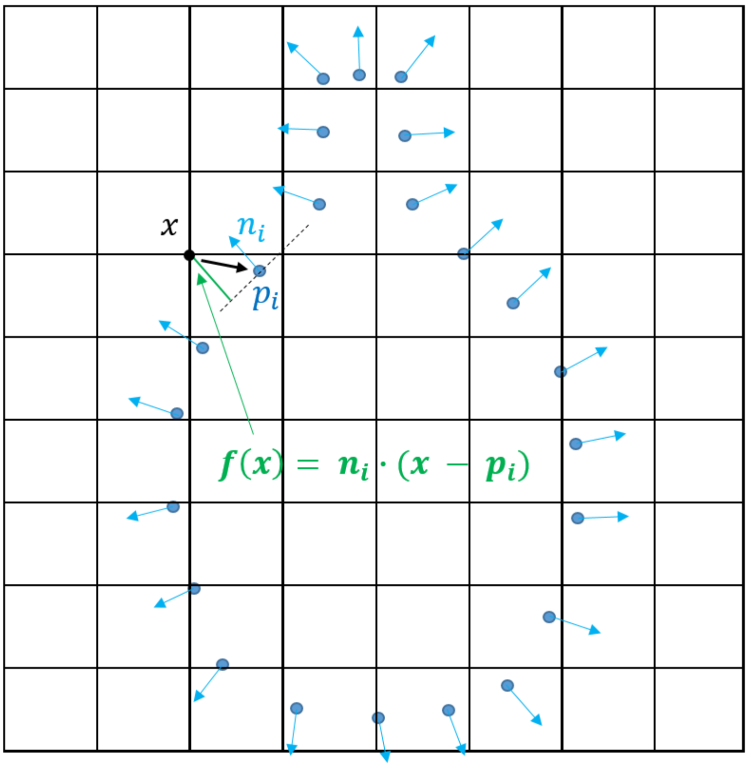

Figure 1 Hoppe implicit function

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.

Question 2: Make a surface reconstruction of the Bunny with the Hoppe function on a 128x128x128 voxel grid. Take a screenshot of the final mesh.
Tips: start the implementation on a 16x16x16 voxel grid and when it is working, change the grid size.

Question 3: Make a comparison between Hoppe and the reconstruction obtained with PoissonRecon in CloudCompare (visual comparisons with side-by-side screenshots, computation time, number of triangles, quality of reconstruction).

b. Implement the IMLS implicit function

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:

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

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

The parameter h is proportional to the noise of the point cloud.
$h = 0.001$ 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 fine for many point clouds). In reconstruction.py, implement the function compute_imls to fill the scalar_field with IMLS implicit function value.

<u>Question 4:</u> Make a surface reconstruction of the Bunny with the IMLS function on a 128x128x128 voxel grid. Take a screenshot of the final mesh. Make a comparison between Hoppe, IMLS and the reconstruction obtained with PoissonRecon in CloudCompare (visual comparisons with side-by-side screenshots, computation time, number of triangles, quality of reconstruction).

Tips: the computation for a 128x128x128 grid will take around 20 min on a laptop for the IMLS mesh (you can improve the code by vectorizing calculations or computing the function only on intersecting voxels).

C. Going further (BONUS)

As a bonus, you will test a recent surface reconstruction method using neural networks, Points2Surf.
Paper on arXiv: https://arxiv.org/pdf/2007.10453.pdf
Github: https://github.com/ErlerPhilipp/points2surf

<u>Question Bonus:</u> Make a reconstruction of the Bunny with Points2Surf using 128x128x128 voxel grid as a fair comparison with previous methods. Take a screenshot of the final mesh. Detail the model used, the training set and the parameters used for your result.
Make a comparison between Points2Surf, IMLS and PoissonRecon from CloudCompare (visual comparisons with side-by-side screenshots, computation time, number of triangles, quality of reconstruction).