NPM3D - TP 4: Surface Reconstruction

Adrien Golebiewski

IASD 2022-23

1 Question 1

Figure 1: Poisson surface reconstruction on bunny point cloud

We use the Poisson surface reconstruction of CloudCompare on the bunny point cloud with the following parameters:

- Octree depyh : 8
- Neuman boundary
- Samples per node : 1.5
- Point weight : 2

We obtain a mesh grid of 373800 triangles and 186902 vertices (fig. 1).

The value of the parameter "**samples per node**" depicts the minimum number of sample points that should fall within each octree node. The idea is to increase the value of this parameter for noisy point clouds to provide a noise-reduced and smoother reconstruction : a value of 1.5 is sufficient in our case.

The parameter "**point weight**" controls the value of alpha in the energy minimized by the Laplace-Poisson equation. A value of 2.0 yielded good results.

2 Question 2

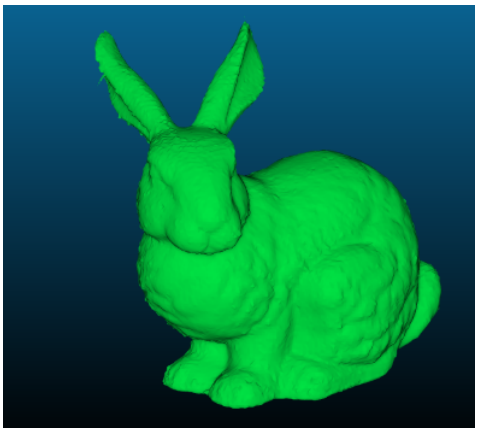
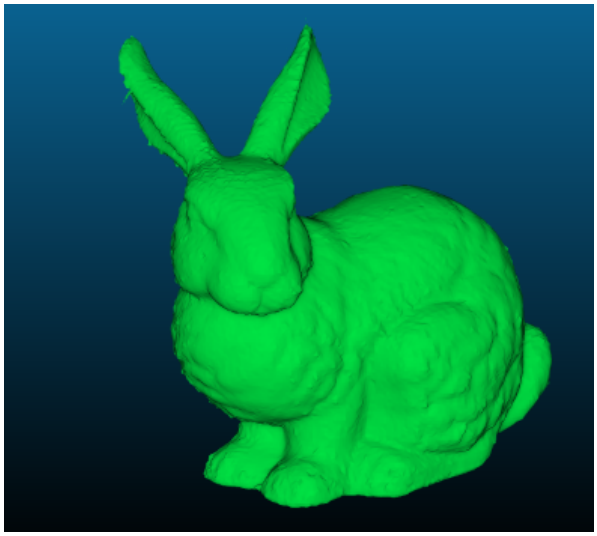


Figure 2: Surface Reconstruction with the Hoppe function on a 128x128x128 voxel grid

3 Question 3

(a) Surface reconstruction using Hoppe implicit function



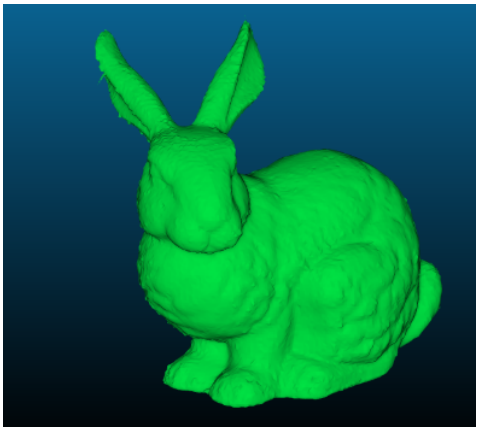
(b) Surface reconstruction using CloudCompare's PoissonRecon plugin

The reconstruction obtained with the Hoppe implicit function ran in 20.37 s, which is slower than the CloudCompare PoissonRecon plugin (1 second in average). It contains 97191 triangles, compared to 92429 for the Poisson reconstruction. We can also observe some small portions of the surface that are not continuous at all : the ears for example. We can recognize each part of the bunny but some portions of the fur are not smooth even though it should be just like in the Poisson reconstruction.

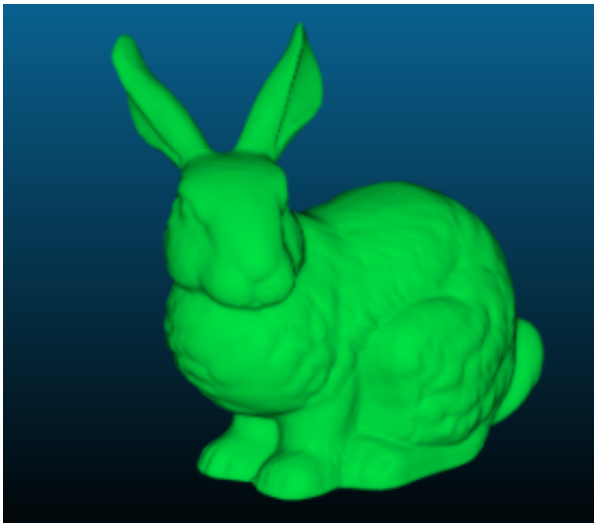
Overall, the quality of the reconstruction is thus not ideal.

4 Question 4

(a) Surface reconstruction using Hoppe implicit function



(b) Surface reconstruction using CloudCompare's PoissonRecon plugin



(c) Surface reconstruction using IMLS

The slowest method among all the three methods is the reconstruction obtained with the **IMLS implicit function** : ran in 37.55 seconds (10 runs). We can explain that is because of the way it was **implemented in Python with only some vectorization, no further optimization addeed in the code**. Indeed, the computation of the IMLS implicit function requires an additional step of averaging compared to the Hoppe function.

In a memory consumption point of view, the IMLS method consumes significantly less memory than the Poisson reconstruction with these parameters. It contains 91918 triangles, which is the lowest value among all three methods tested, and the surface obtained is particularily smooth as shown in the previous question.

5 Bonus

The IMLS function is a smooth version of the Hoppe function.

- We replace the scalar product with an averaged scaler product. This gaussian smoothing produces a more continuous surface representation.
- This will reduce the noise on the surface as shown in fig. 4.

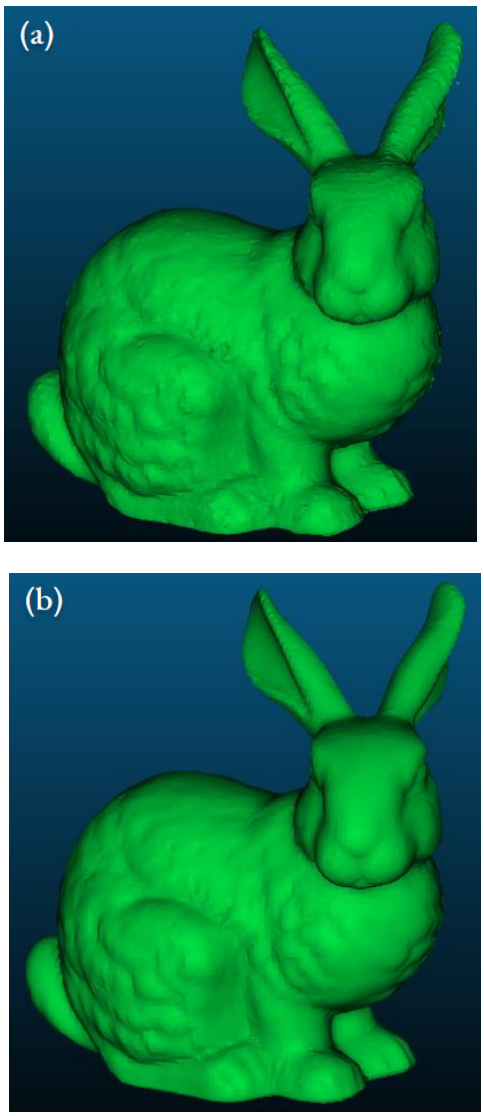


Figure 5: Iso-zero surface of bunny point cloud with a grid resolution of and 100 using (a) the IMLS function, (b) Hoppe function

The quality of this mesh is much better, the fur for example feels more well rendered. The whole rabbit is smoother (right above the bunny's leg, for example, the imperfection is very visible with Hoppe function but not at all with IMLS reconstruction).