NPM3D - TP4

Aubin TCHOÏ

February 2023

Question 1

The octree depth controls how smooth the discretisation of the surface will be. A deeper octree allows for smoother approximations, however the RAM consumption also greatly increases with the depth, and unfortunately CloudCompare crashes when trying to allocate more memory than available. In my case I only got up to a depth of 7.

The value of parameter `samples per node` specifies the minimum number of sample points that should fall within each octree node. We might want to increase the value of this parameter for noisy point clouds in order to provide a smoother and noise-reduced reconstruction. This is not the case here, and a value of 2.0 was sufficient.

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

A "better" reconstruction is one with more geometric details, less vertices and less unwanted holes.

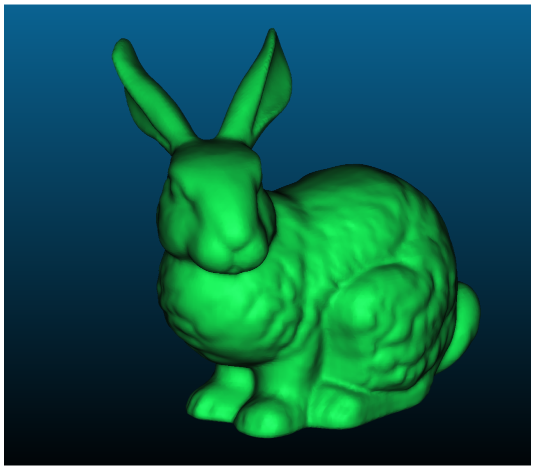


Figure 1: Poisson reconstruction using CloudCompare

The mesh displayed above contains 93458 triangles.

Question 2

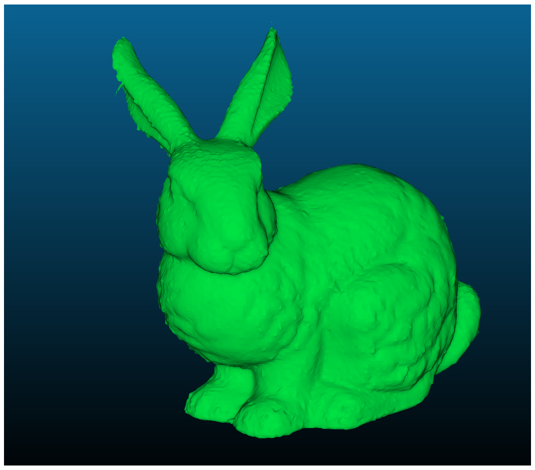
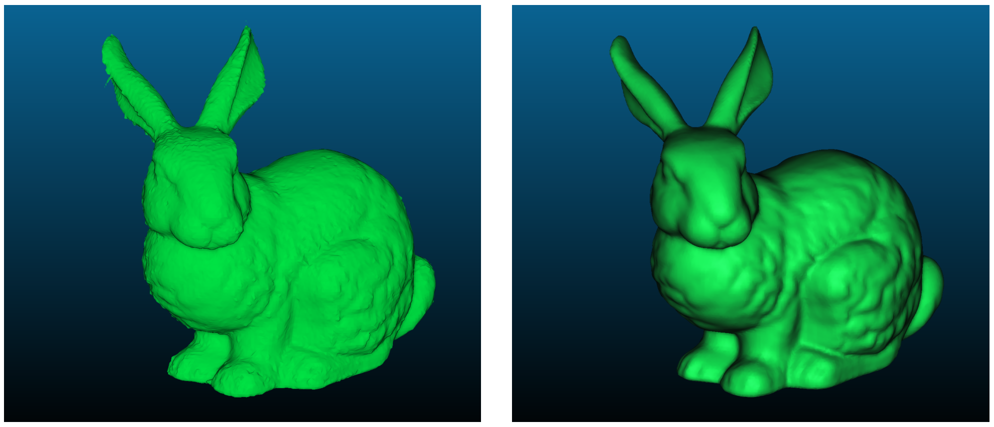


Figure 2: Reconstruction with the Hoppe function on a 128x128x128 voxel grid

Question 3



a) b)

Figure 3: Surface reconstruction using a) Hoppe implicit function
b) CloudCompare's PoissonRecon plugin

The reconstruction obtained with the Hoppe implicit function ran in 19.47 s (averaged on 10 runs), which is significantly slower than the CloudCompare PoissonRecon plugin (always less than 2 s). It contains 98284 triangles, compared to 93458 for the Poisson reconstruction. We can also observe many artefacts, there are some small portions of the surface that are not continuous as shown by the zoom in below (top of the right ear), which comes from the fact that both the Hoppe function and the surface obtained from it are not continuous. The quality of the reconstruction is thus not ideal, we can recognize each part of the bunny but many portions of the fur are not smooth even though it should be just like in the Poisson reconstruction.

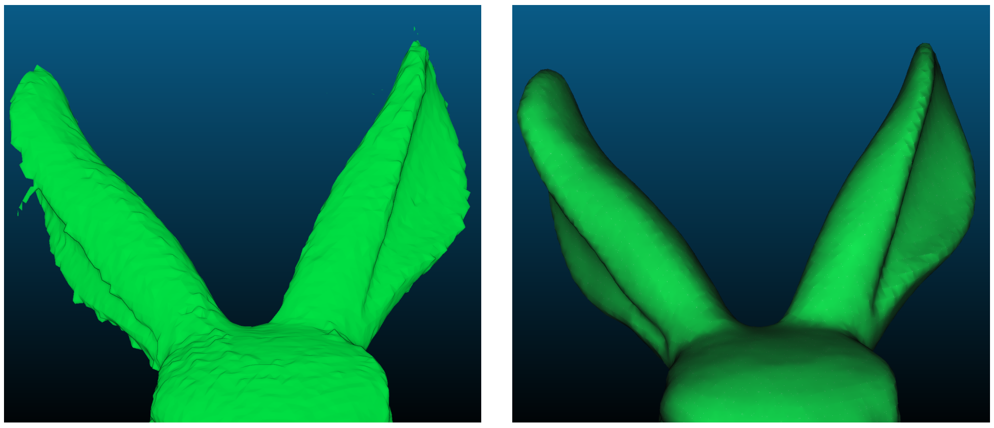


Figure 4: Zoom in on the bunny's ears a) Hoppe implicit function
b) CloudCompare's PoissonRecon plugin

Question 4

The reconstruction obtained with the IMLS implicit function ran in 38.75 s (averaged on 10 runs), which is the slowest among all three methods. Indeed, the computation of the IMLS implicit function requires an additional step of averaging compared to the Hoppe function. One thing to account for is that the IMLS method might be slower than the Poisson reconstruction because of the way it was implemented (Python with only some vectorization, no further optimization). In terms of memory consumption, the IMLS method consumes significantly less memory than the Poisson reconstruction on this example with the parameters used. It contains 92034 triangles, which is the lowest value among all three, and the surface obtained is particularily smooth as shown below.

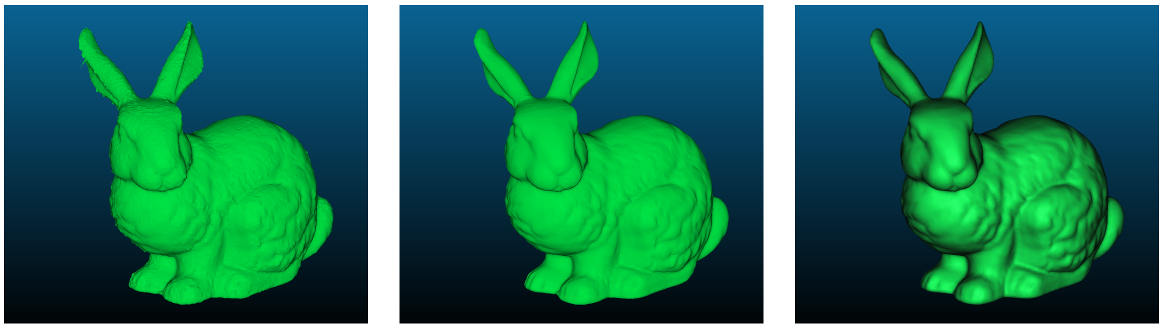


Figure 5: Reconstruction using a) Hoppe b) IMLS c) CloudCompare's PoissonRecon plugin

There are far less artefacts than in the previous reconstruction using the Hoppe implicit function. For instance, there is no noticeable hole in the bunny's ears.

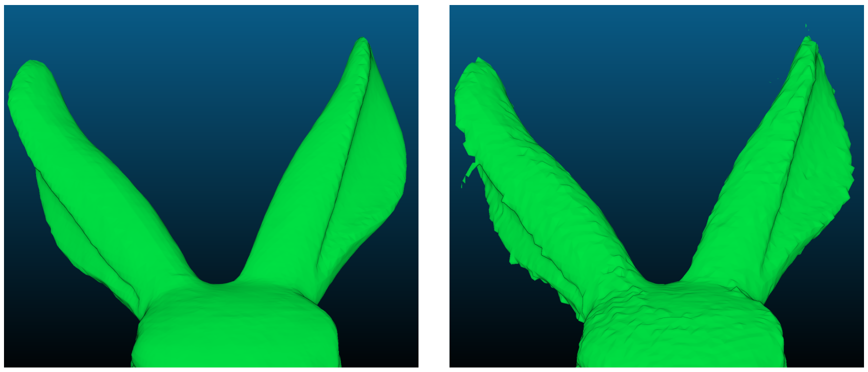


Figure 6: Zoom in on the bunny's ears a) IMLS implicit function
b) Hoppe implicit function

There is still one small visible issue with the IMLS reconstruction that can be found at the tip of the bunny's left foot.

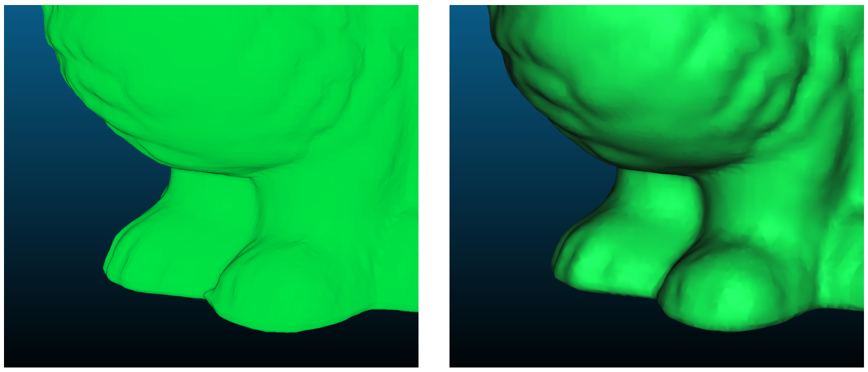


Figure 7: Zoom in on the bunny's feet a) IMLS implicit function
b) CloudCompare's PoissonRecon plugin

The IMLS method seems to expand the surface locally on some areas. The overall quality of the reconstruction is good, but in the case of a smooth and somewhat regular shape like the bunny, a global method like the Poisson reconstruction might be better suited than a local method like IMLS.