TP 5 : Surface Reconstruction

Leman FENG

NPM3D - 12/02/2018

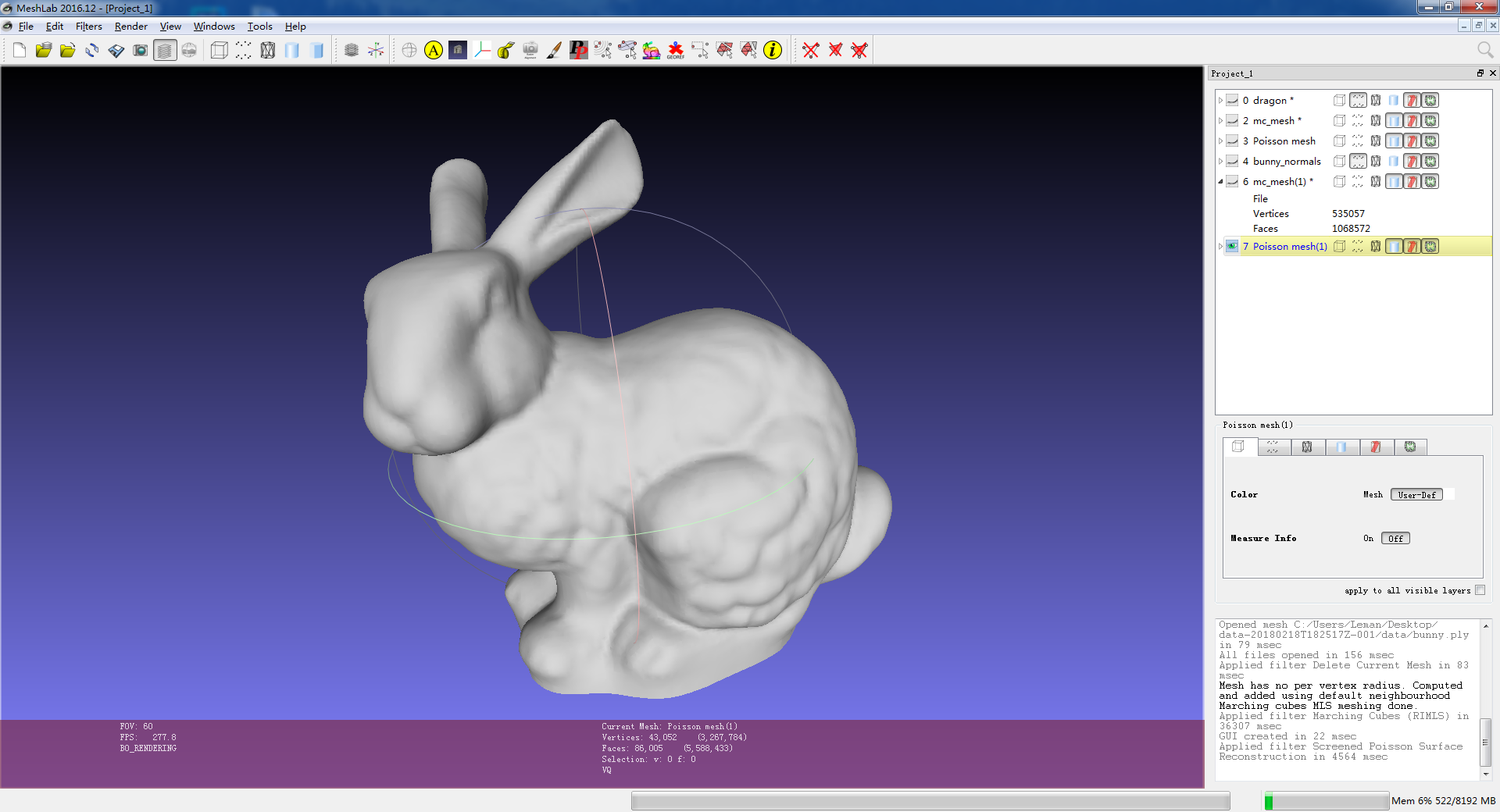
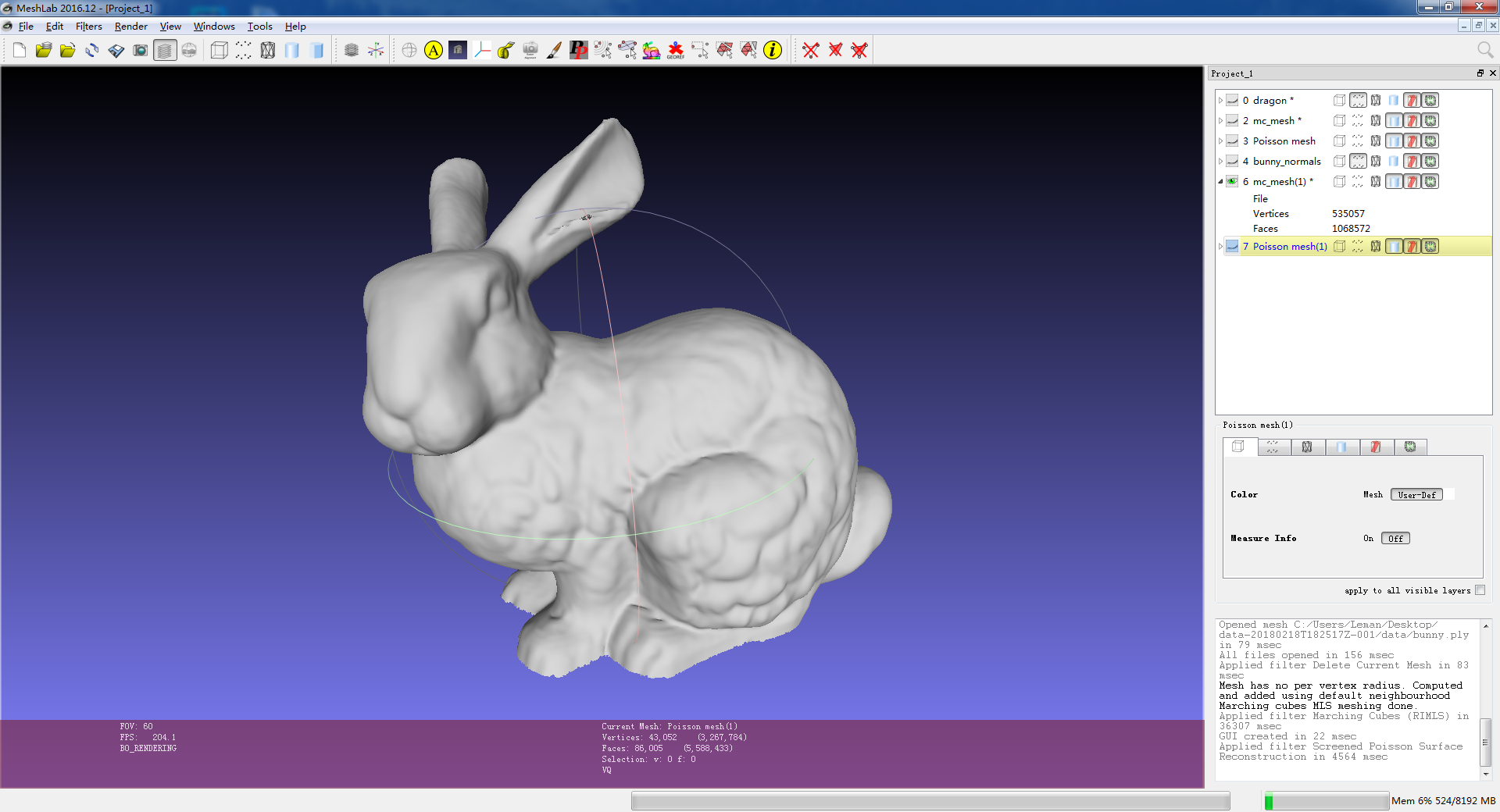
[jean-emmanuel.deschaud@mines-paristech.fr](mailto:jean-emmanuel.deschaud@mines-paristech.fr)

# Objectives

* Test 3D Surface Reconstruction on Meshlab
* Surface Reconstruction on Python : implement the Hoppe implicit function

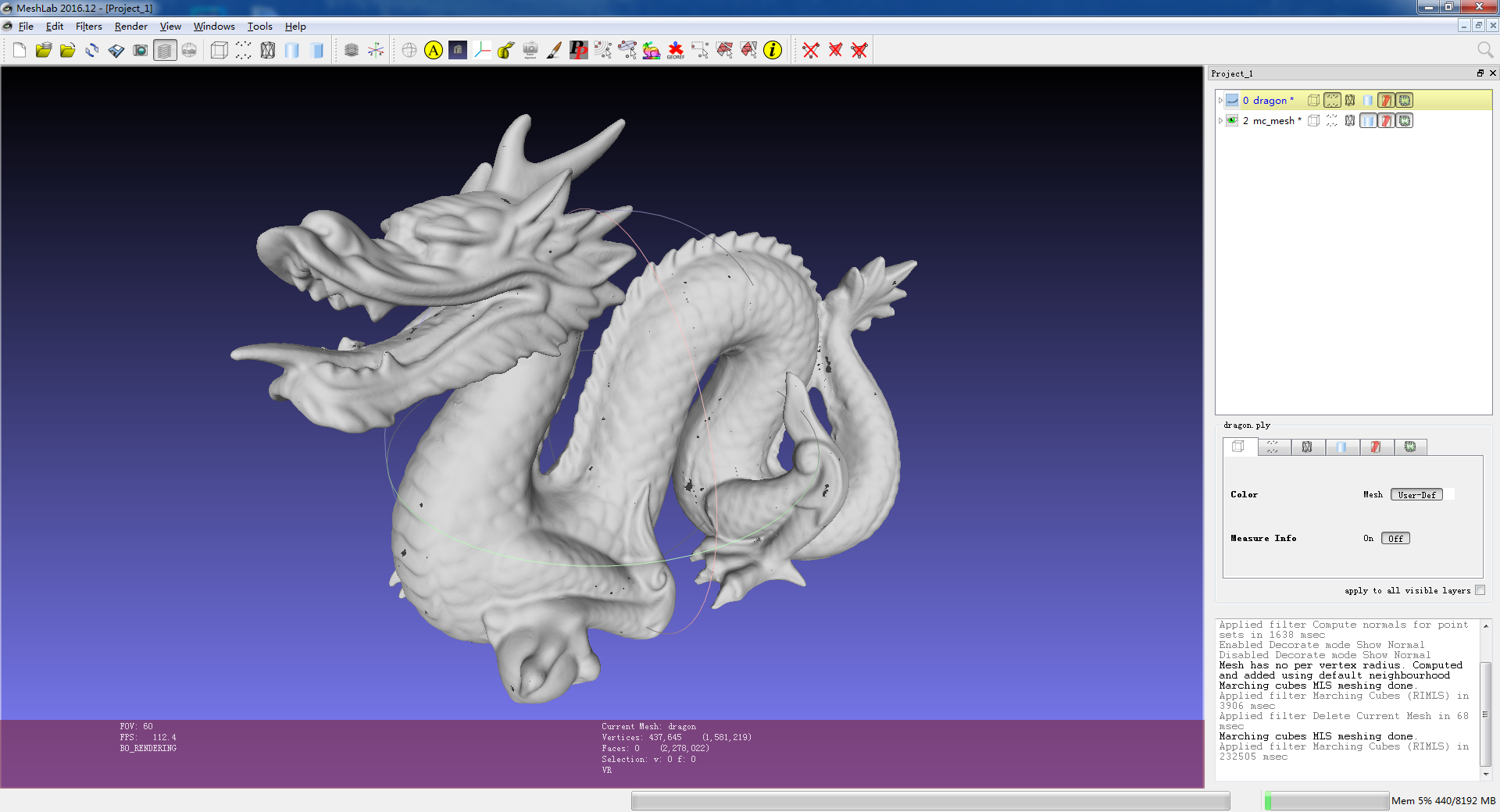
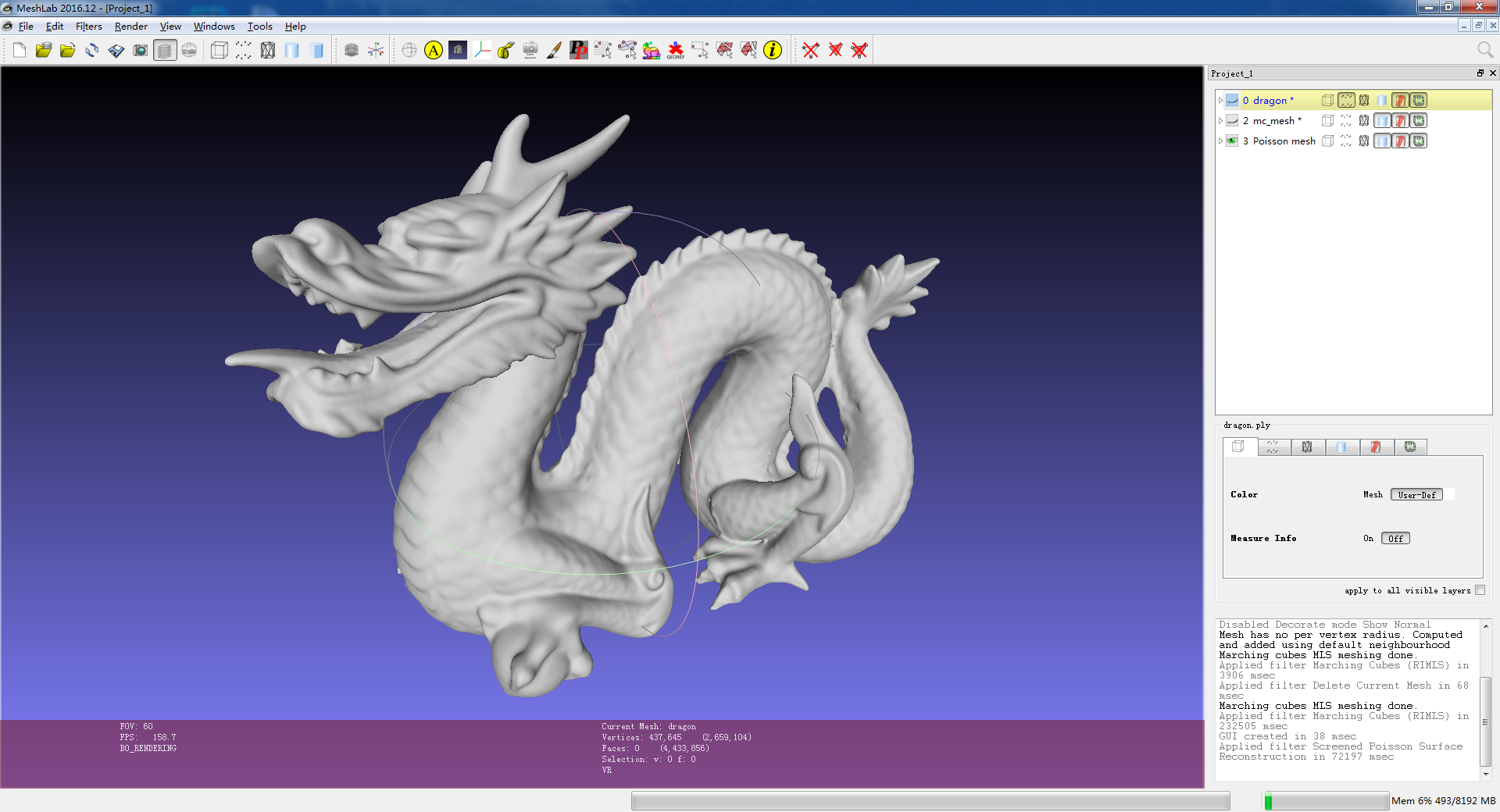
# A. 3D Reconstruction on Meshlab

**Question 1 : Take screenshots of the two reconstructed meshes for the two point clouds (with best possible parameters). In total, your report should have 4 screenshots.**



Left: RIMLS with grid = 500. (535k vertices, 1.1m faces)

Right: Screen Poisson with depth = 9 (43k vertices, 86k faces)

Left: RIMLS with grid = 800 (1.1m vertices, 2.2m faces).

Right: Screen Poisson with depth = 10 (1.0m vertices, 2.2m faces)

**Hint : To find best parameters, you need to define what best is for a surface reconstruction method: usually it is a good trade-off between geometric details of the surface, minimum unwanted holes and minimum vertices of the mesh**

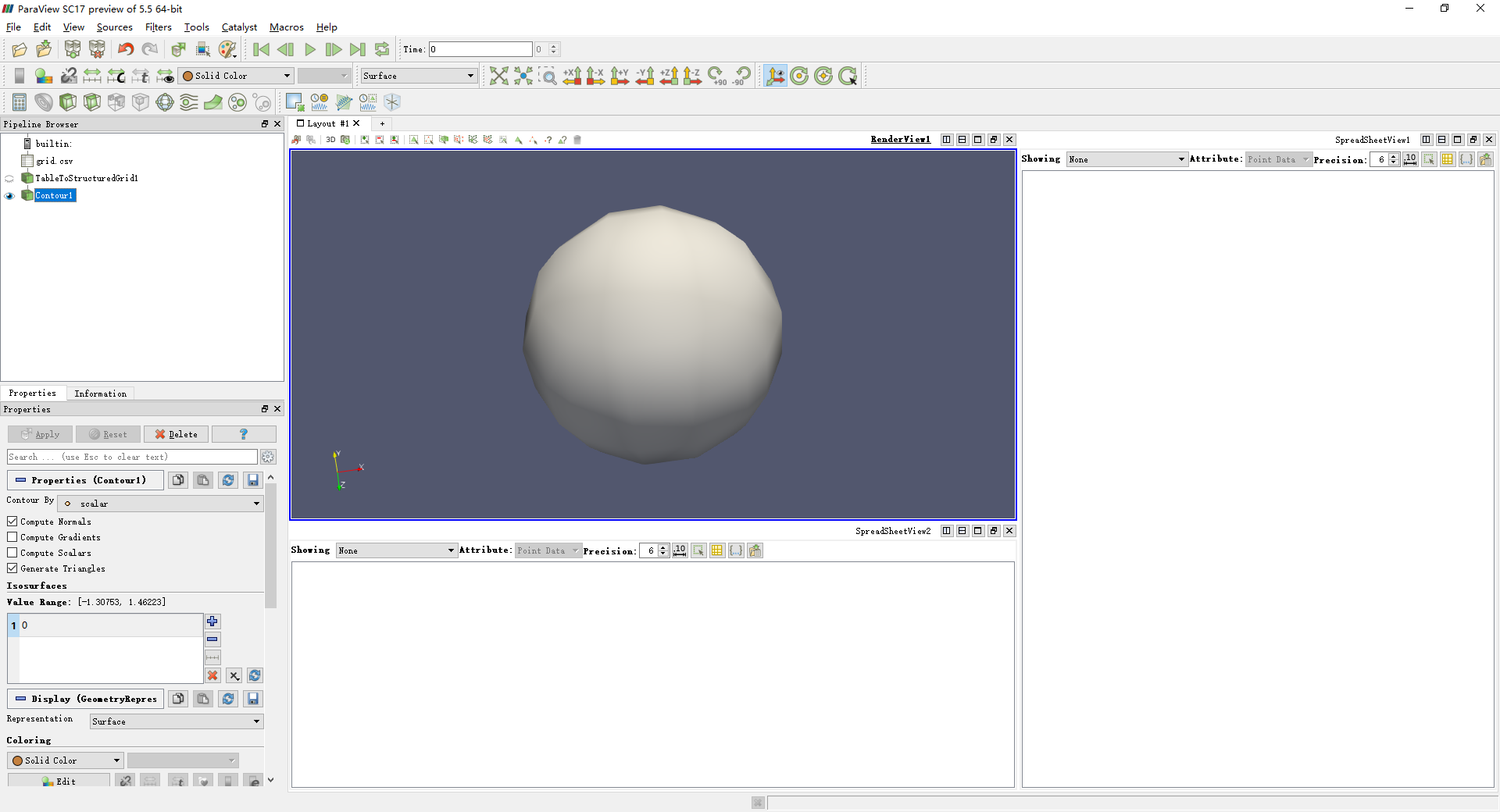
**Question 2 : For each point cloud, what is the “best” surface reconstruction method? (give the parameters used for that reconstruction, the final number of vertices and faces)**  
For the bunny point cloud, since the point density is very low, Screened Poisson need a minimum number of samples in each octree cell, so the result is more smoothed out than RIMLS. With RIMLS, if the grid size is more than 800, I observed strong noises. So I choose RIMLS with grid size = 500. Final number of vertices is 535k, and 1.1m for faces.

For the dragon point cloud, the point density is highly non-uniform. While RIMLS use a uniform grid, we must choose a very dense grid to assure that all details are captured. In practice, if the sizing grid is lower than 800, there will be a lot of holes in mesh. Screened Poisson can always produce a water-tight closed surface, and octree can capture non-uniform point cloud easily. By Screened Poisson with depth=10, final result is 1.0m vertices, 2.2m faces.

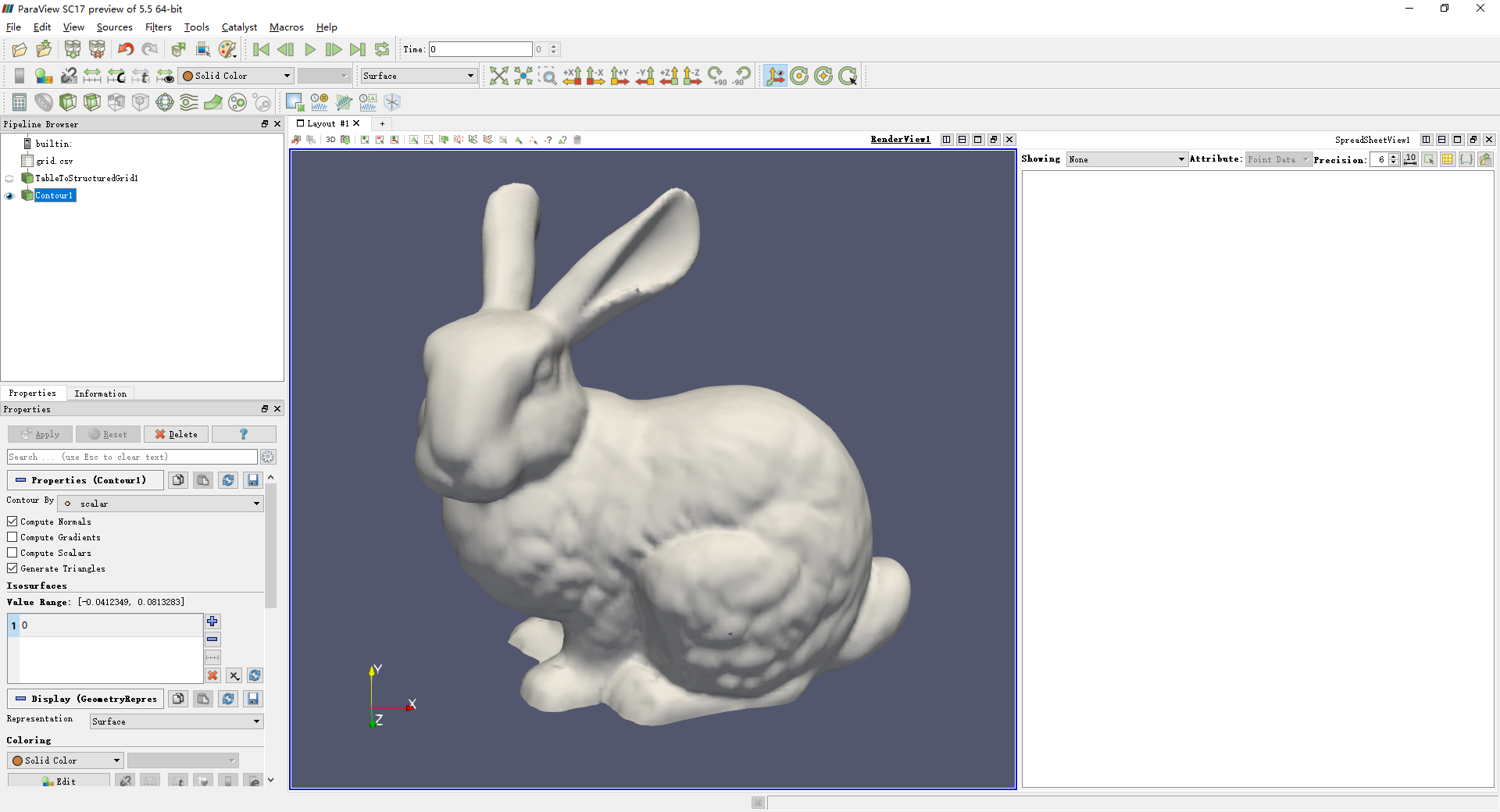
# B. Surface Reconstruction on Python: implement the Hoppe implicit function

**Question 3 : Take a screenshot of your iso-zero surface of the sphere point cloud**

With a grid of 5x5x5:

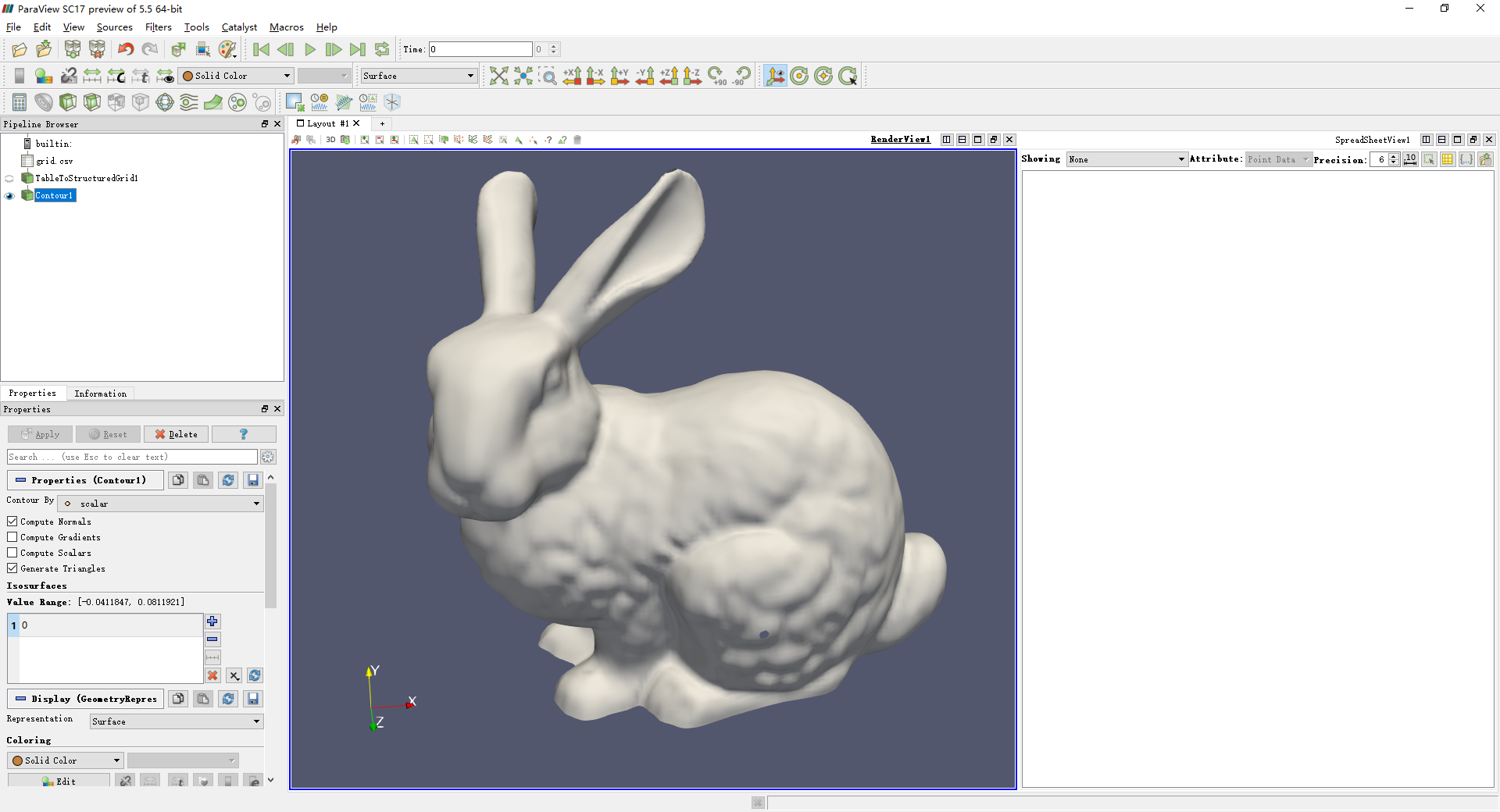


**Question 4 : Take a screenshot of your iso-zero surface of the bunny point cloud**

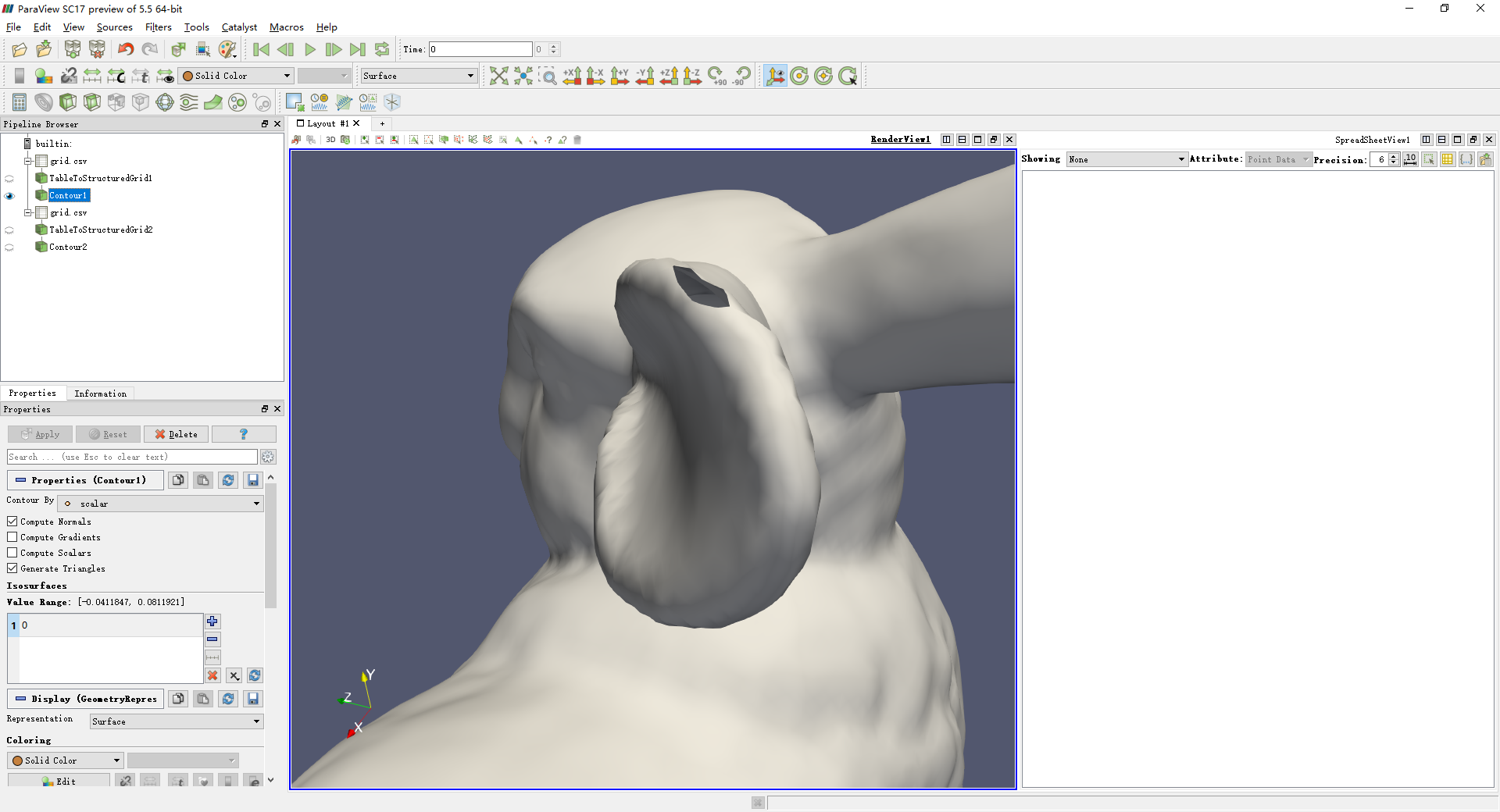
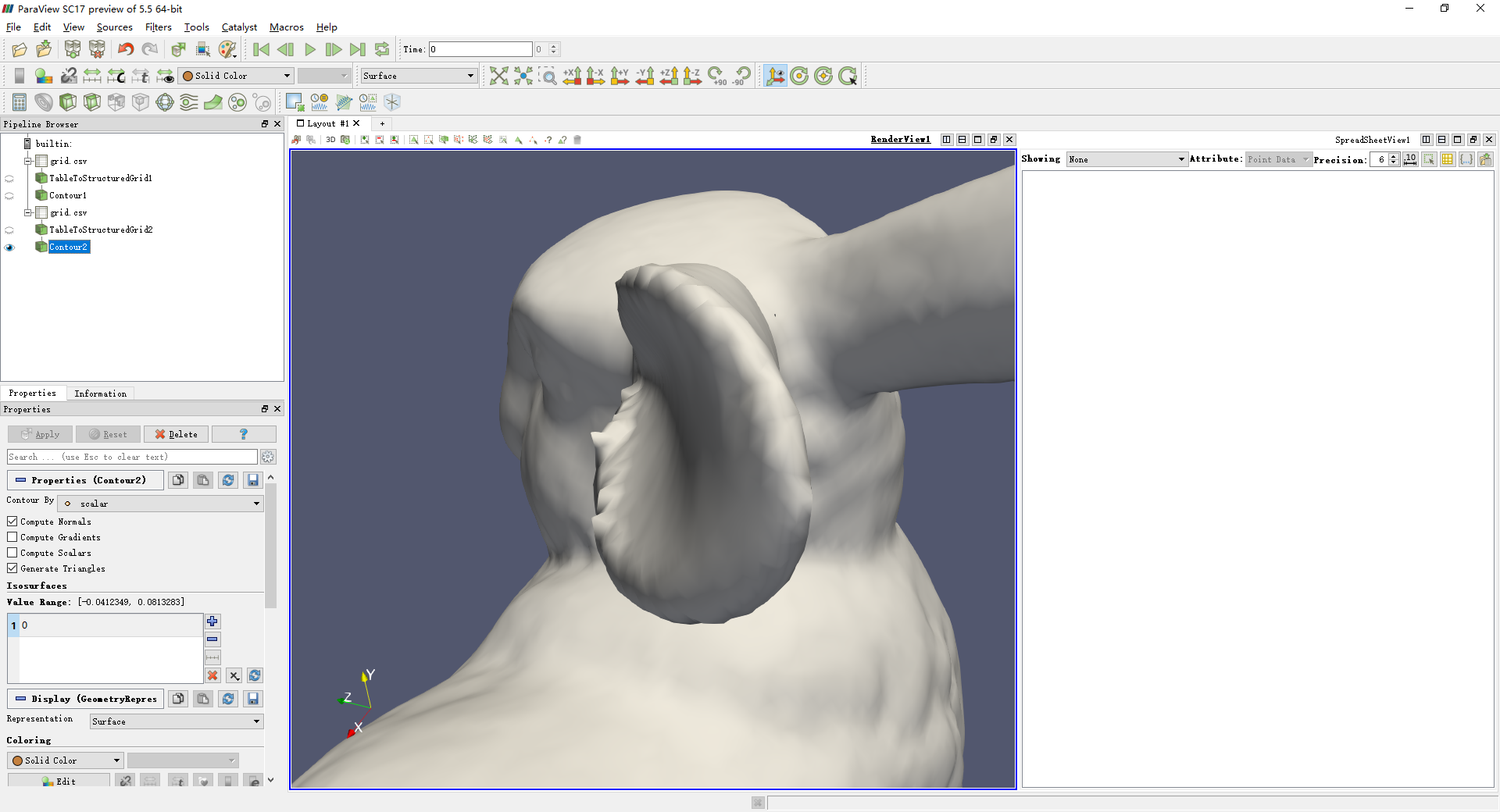


# C. Implement the IMLS function (BONUS)

**Question 5 : Show with screenshots the differences between the Hoppe surface and the IMLS surface of the Bunny? Where does it comes from?**



Bunny by EIMLS



Comparison: left: Hoppe, right: EIMLS

The EIMLS is smoother thanks to the weighted smoothing term