TP 5 : Surface Reconstruction

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

The report should be a pdf containing the answers to the **Questions** and named “TPX\_LASTNAME\_Firstname.pdf”. Send your code along with the report in a zip file following the same naming rule.

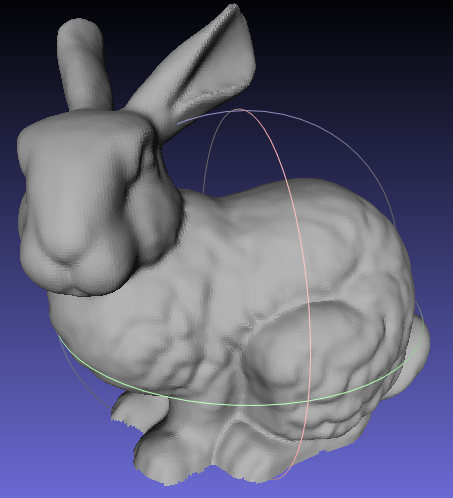
# A. 3D Reconstruction on Meshlab

The goal is to test and understand the two main surface reconstruction methods (RIMLS and Poisson) on two point clouds : “bunny.ply” and “dragon.ply”.

1. You need to install Meshlab v2016 (or newer)
2. First, you compute the normals with “Filters->Point Set->Compute normals for point sets” (the parameter “Neighbour sum” is the k of the k nearest neighbors for the PCA computation, 10 is good for dense point clouds with little noise)  
   Hint: you can render the normals with “Render->Show Normal”. To get a mesh with an orientation toward exterior, you need normals with orientation toward exterior. You can invert the orientation of the normals with “Filters->Normals, Curvature and Orientation->Per Vertex Normal Function”
3. Then, you can test the two reconstruction methods seen in the course : RIMLS and Poisson :
   1. RIMLS : “Filters->Remeshing, Simplification and Reconstruction->Marching Cubes (RIMLS)”. The two main parameters are the MLS filter scale and the grid resolution
   2. Poisson : “ Filters->Remeshing, Simplification and Reconstruction->Screened Poisson Surface Reconstruction”. The main parameter is the reconstruction depth.

**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.**

**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)**

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

The goal is to compute the hoppe implicit function on a regular grid. Then, you will be able to extract the surface as a mesh from the iso-zero using the software Paraview.

You need as input, point clouds with normals. You can find “sphere\_normals.ply” and “bunny\_normals.ply” in the data folder (normals have been computed with Meshlab).

1. To compute the hoppe function, you need to create a regular grid of the space around your input point cloud.
2. Then, on every node of the grid, the hoppe function is when is the closest point of the point cloud to (and the associated normal of point ).

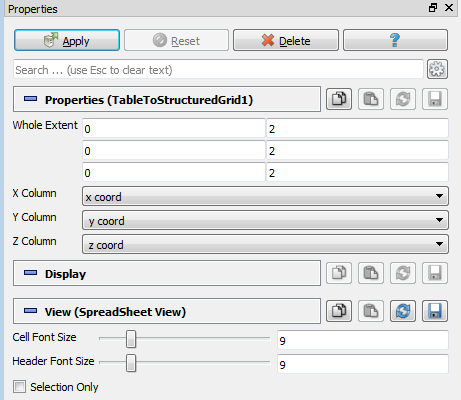
The result is a scalar field on a regular grid. To visualize your result and extract the iso-zero, you can use the software Paraview, a data analysis and visualization application. You can download the software here : <https://www.paraview.org/download/>

Paraview is able to read a csv file when you put your scalar field data by row with “x coord, y coord, z coord, scalar”. An example of a csv file is given in the data folder.

To visualize the scalar field and extract the iso-zero surface on Paraview :

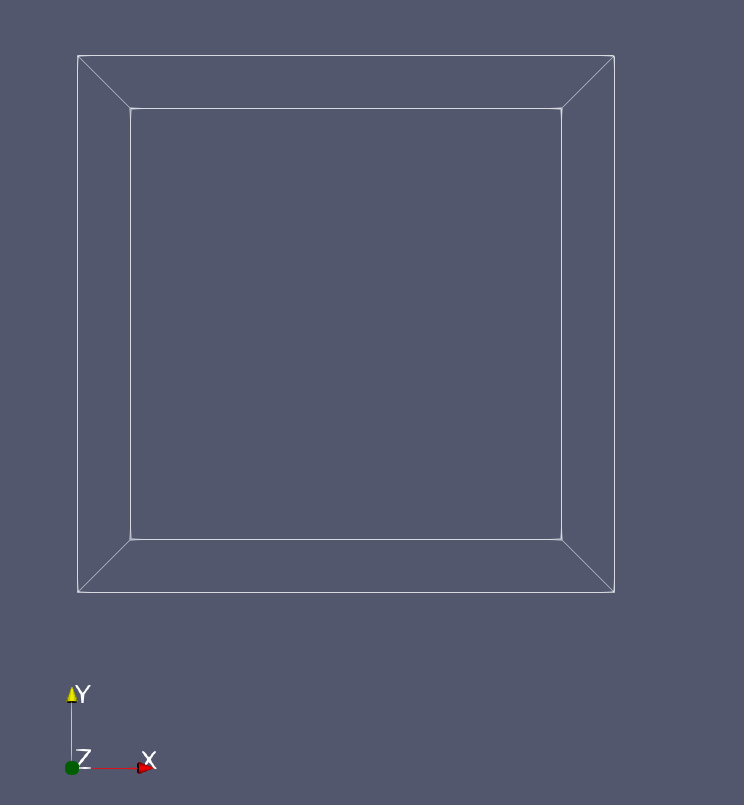
1. Open your file “grid.csv” and click on “Apply” (the software should find the correct delimiters).
2. Transform your data in a regular volume with “Filters->Alphabetical->Table To Structured Grid”. In the fields “Whole Extent”, you need to write the number of cells you have in each direction (in your csv example, you have “0” to “2”). In the field X Column, you choose “x coord”. Idem in Y Column and Z Column.

The filled fields should be like the screen below :



1. Click on “Apply”. To see the data, right-click inside the render widow and then on the little eye next to “TableToStructuredGrid1”

You should now see a white box like below :



1. To see the scalar field and the volume, you need to change :



to :

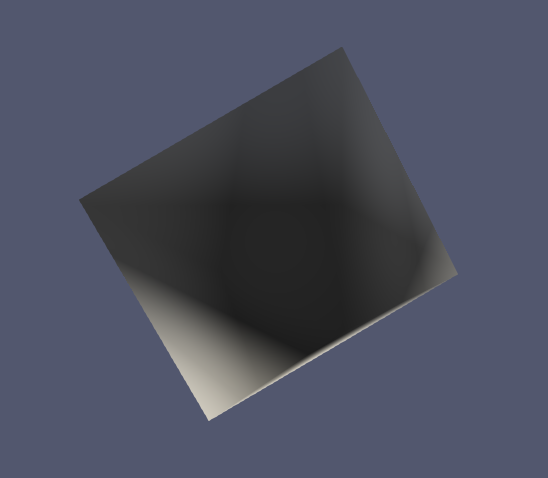


1. Extract the iso-zero of your scalar-field with “Contour” button :



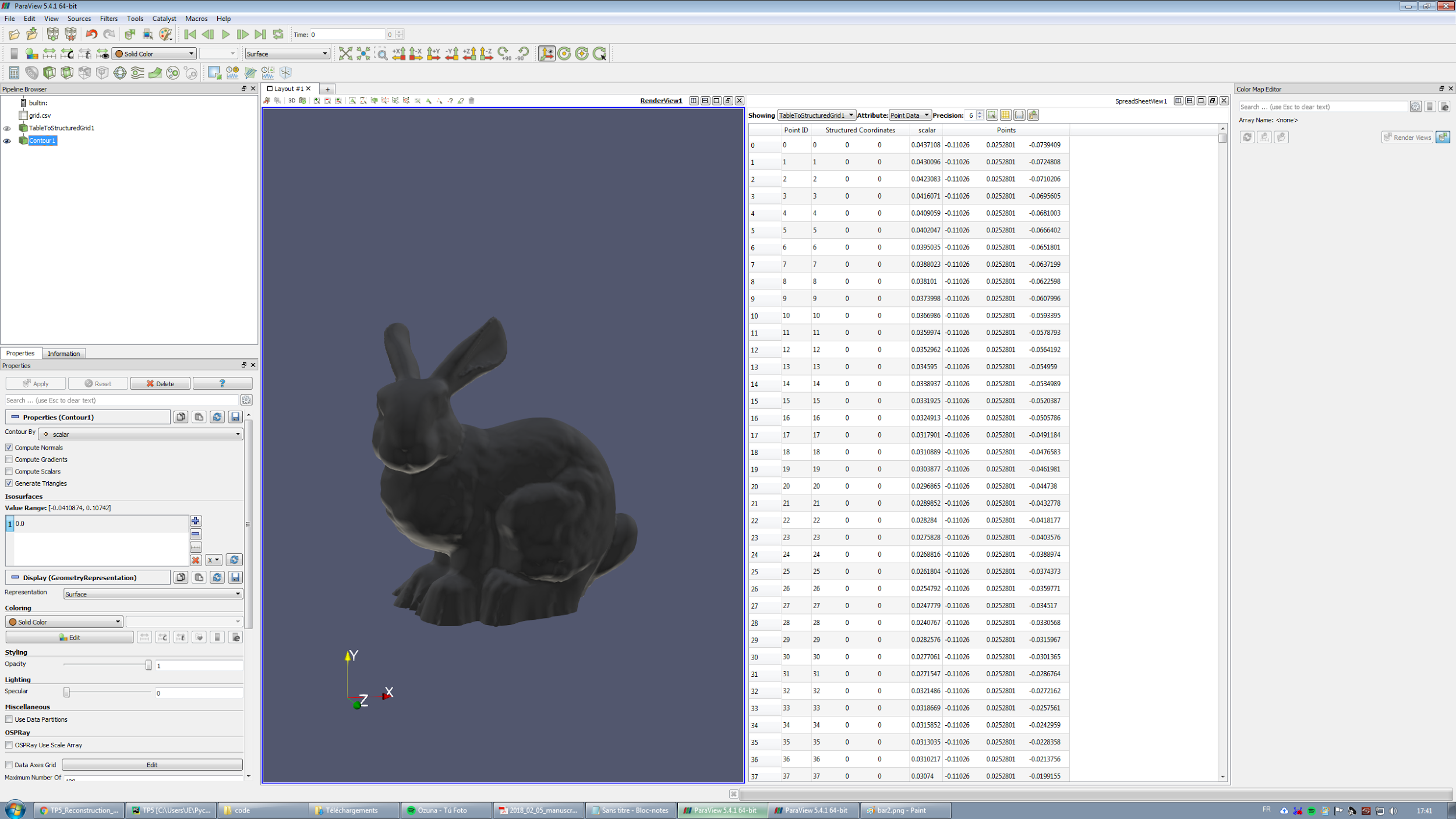
1. Click on the eye next to “Contour1” and remove the eye next to “TableToStructuredGrid1” to see you iso-zero surface. You have to put “0.0” in the Value Range of the iso-surface to extract the iso-zero and click on “Apply”.

The result mesh should be like this :



This does not look like a sphere. It is because of the number of the cells used to compute the scalar field. In that example, we used 2\*2\*2 cells. For the sphere, a good number of cells is 10\*10\*10.

For the bunny, with 100\*100\*100 cells, you will get the iso-zero surface like below :



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

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

# C. Implement the IMLS function (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 good approximation of the distance of any point to the surface when the point is close enough to the surface.

For any point , the IMLS function is defined by :

with

The parameter is proportional to the noise of the point cloud. is a good trade-off for the bunny point cloud.

You can compute the function using all points of the point cloud but you can see that the weights decrease quickly when are far from the point . It is more efficient to compute the function using only the k nearest points around ( is enough for the bunny point cloud).

You will have an issue when the point is far from the surface: the exponential weights will become null because of numerical limits of floats representations on computers.

To deal with that, we define the Extended Implicit Moving Least Square (EIMLS), the parameter will vary following (with the closest point of the point cloud to ):

For more details about the EIMLS, you can search for Hassan Bouchiba Thesis.

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