Balthazar Neveu — Jamy Lafenetre
balthazarneveu@gmail.com —
jamy.lafenetre@ens-paris-saclay.fr

Question 1

The octree depth is the parameter that has the most impact on the result, but also on the computation time. We highlight this impact in figure 1. Bigger values lead to a deeper tree and more scale for the SPR solving. It can also increase drastically the number of voxels present on the finest scale, which can lead the linear system solving to become non-tractable. If the number of scales is too low (we show the result for $n = 6$), the voxel grid is too coarse to represent the geometry and we can therefore notice blocky artifacts. If the number of scales is too high ($n = 10$), then the grid may become too fine for the sample density. The implicit discretization of integrals done in the Galerkine solver no longer holds, as there are too few samples within each voxel. This results in high-frequency artifacts.

Figure 1: From left to right: screened Poisson reconstruction with 6, 8, and 10 scales.

We can also tweak the number of samples per node to ensure that each node has enough samples. The built-in plugin claims that higher values such as 15 are preferable for noisy clouds. This is indeed plausible, as the low-frequency Bezier kernel used in the solver will integrate many samples within a single voxel to attenuate noise. For clouds without noise, using such high values will prevent the voxel hierarchy from populating the finest level, therefore resulting in some local blocky artifacts. Additional artifacts can be visible around structures featuring very high frequencies, because the leaf voxels are too coarse for the local geometry. This can be seen in figure 2.

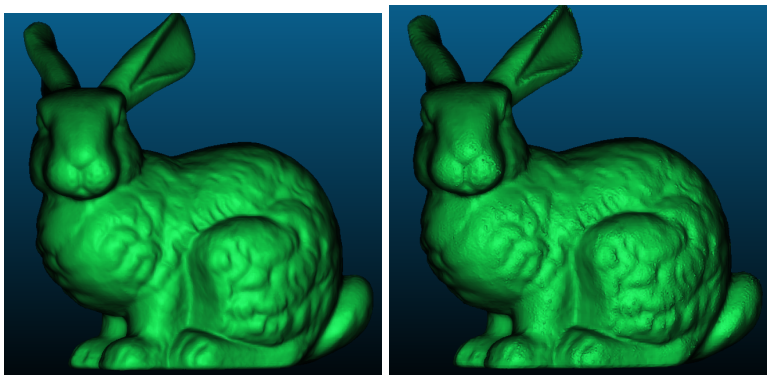

Figure 2: From left to right: screened Poisson reconstruction on 8 scales, for 1.5 and 15 maximum samples per node.

The regularization weight (dubbed ”point weight” in CloudCompare) drives the screening importance of the SPR. Lower values give a smooth surface, whereas very high values will ”overfit” the cloud and the noise. We compare different values in figure 3.
The given point cloud has very little noise, and the bunny is a very smooth closed structure. With a maximum tree depth of 8, a maximum sample-per-node value of 1.5, and a point weight value of 2, we were able to obtain the most satisfactory reconstruction. In our case, a satisfactory reconstruction is a mesh that preserves

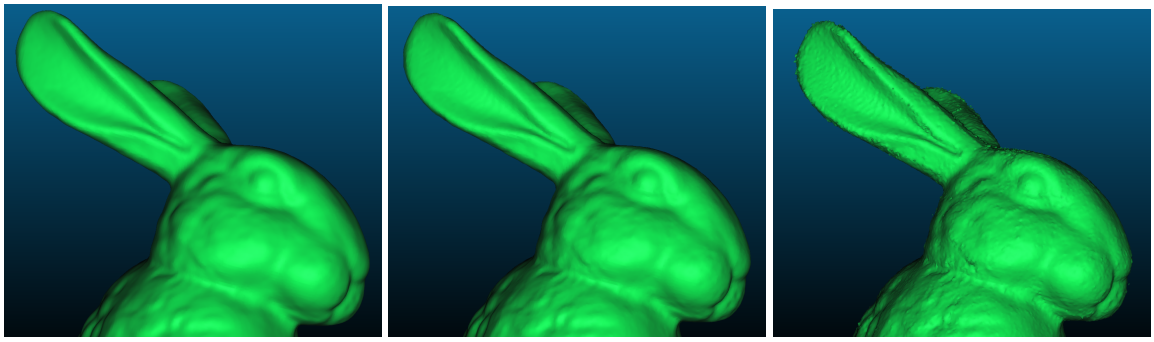

Figure 3: From left to right: screened Poisson reconstruction on 8 scales, for a point weight of 0, 2, and 20.

	SPR	Hoppe
Computation time	1s	10ms
Number of triangles	373 800	32 168
Quality of reconstruction	High	Poor

Table 1: Comparison between our Hoppe implementation and SPR.

the global smoothness of the bunny, without losing the few sharper details and without displaying grid artifacts or spiky artifacts.
The resulting mesh is composed of 373,800 triangles.

Questions 2-3

We compare our handcrafted Hoppe implementation with the SPR plugin of CloudCompare. Qualitative results can be seen in figure 4, and the quantitative results are reported in table 1. Our implementation takes 22 seconds to run on the bunny cloud, which is mostly due to the single-threaded kd-tree implementation we use. If we neglect the bottleneck interaction with the tree and do the processing on a GPU using PyTorch, our reconstruction takes around 17ms. A better optimization could realistically lower this overhead to a few milliseconds. SPR takes between 1 and 2 seconds to compute a surface in CloudCompare which is compiled code (usually leading to better performances than interpreted code like python.).

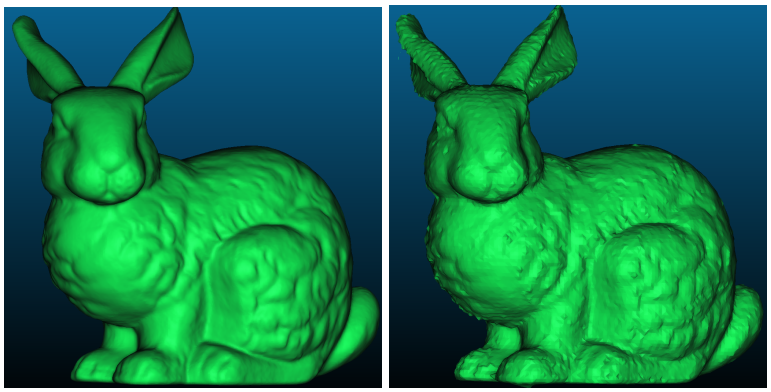

Figure 4: Left: SPR reconstruction. Right: our Hoppe reconstruction.

Question 4

We run the same comparison as before with our handcrafted implementation of IMLS, and compare the result in figure 5 and table 2. We use the same strategy as before to approximate the execution time. The result is smoother and more satisfactory than with Hoppe, at the cost of a few more operations. We can observe that IMLS is able to recover smooth structures such as the ears of the bunny accurately, contrary to Hoppe. However, only SPR can recover small details such as the nose.
Notice that the IMLS surface aboves exhibits a ”lonely” surface in the top right hand corner. This is is due to our careless first implementation for query point that are away from the cloud. If all neighbour samples

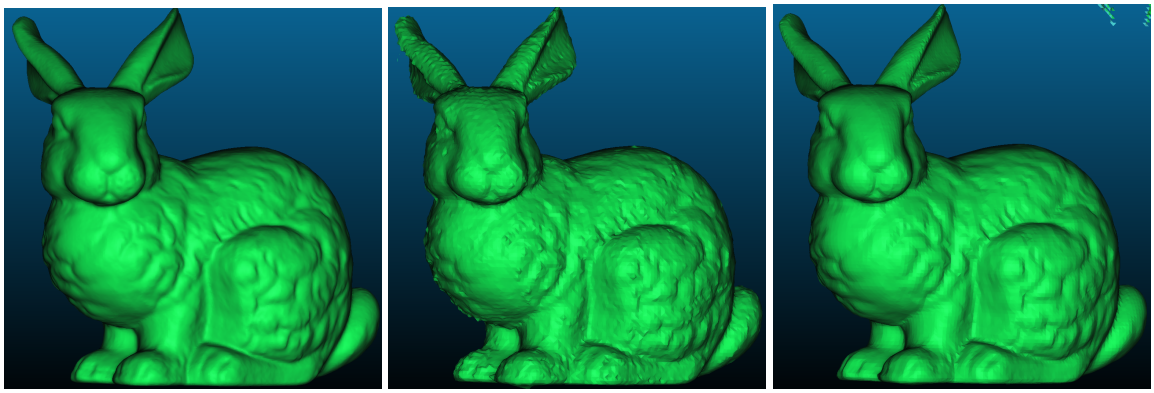

Figure 5: Left: SPR reconstruction. Middle: our Hoppe reconstruction. Right: our IMLS reconstruction.

	SPR	Hoppe	IMLS
Computation time	1s	10ms	100ms
Number of triangles	373 800	32 168	77 164
Quality of reconstruction	High	Poor	Medium

Table 2: Comparison between our implementation of Hoppe, IMLS and SPR.

are very far away, the IMLS estimation can take a value of 0 because of numerical precision. This is very undesirable because a sdf of 0 marks a surface. To counter that, we force the sdf to $+\infty$ when we detect that the sum of the neighbour kernel values is close to neglectable.