GMP 2015

Poisson-driven seamless completion of triangular meshes

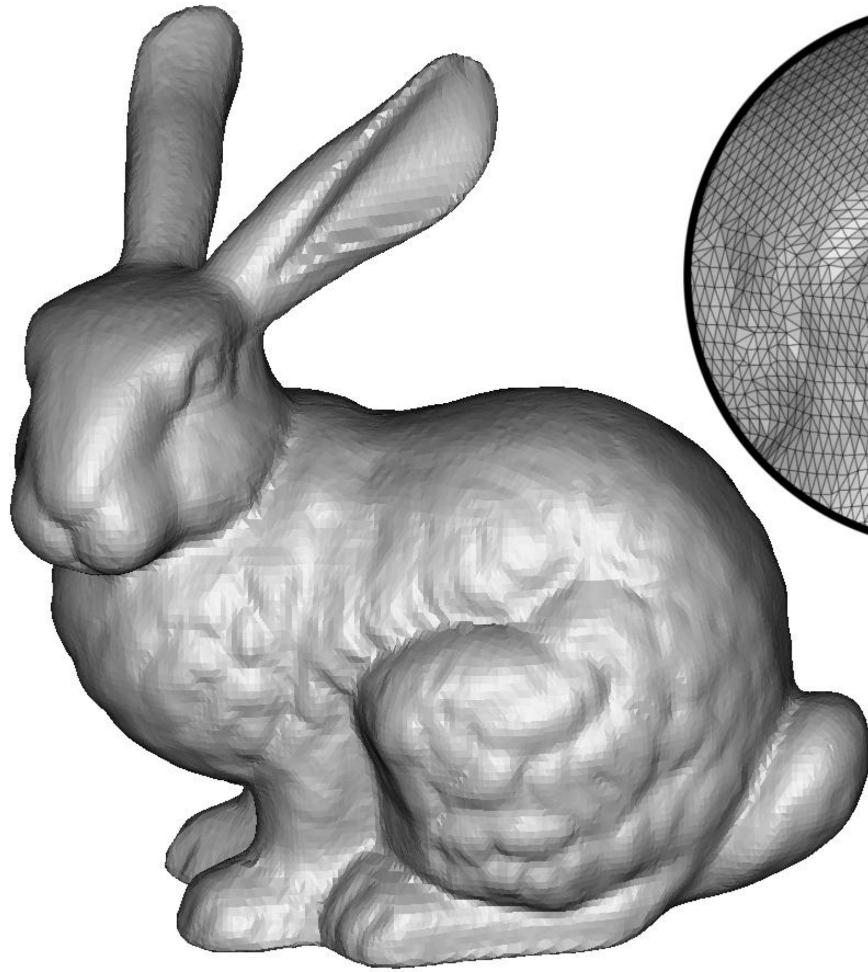
<u>Marco Centin</u>

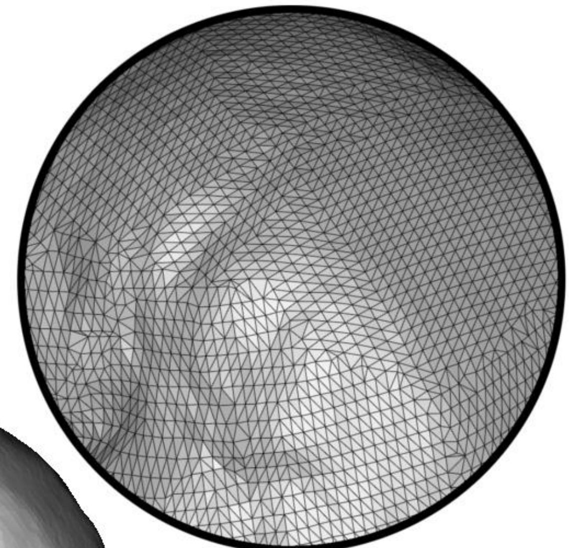
Nicola Pezzotti

Alberto Signoroni

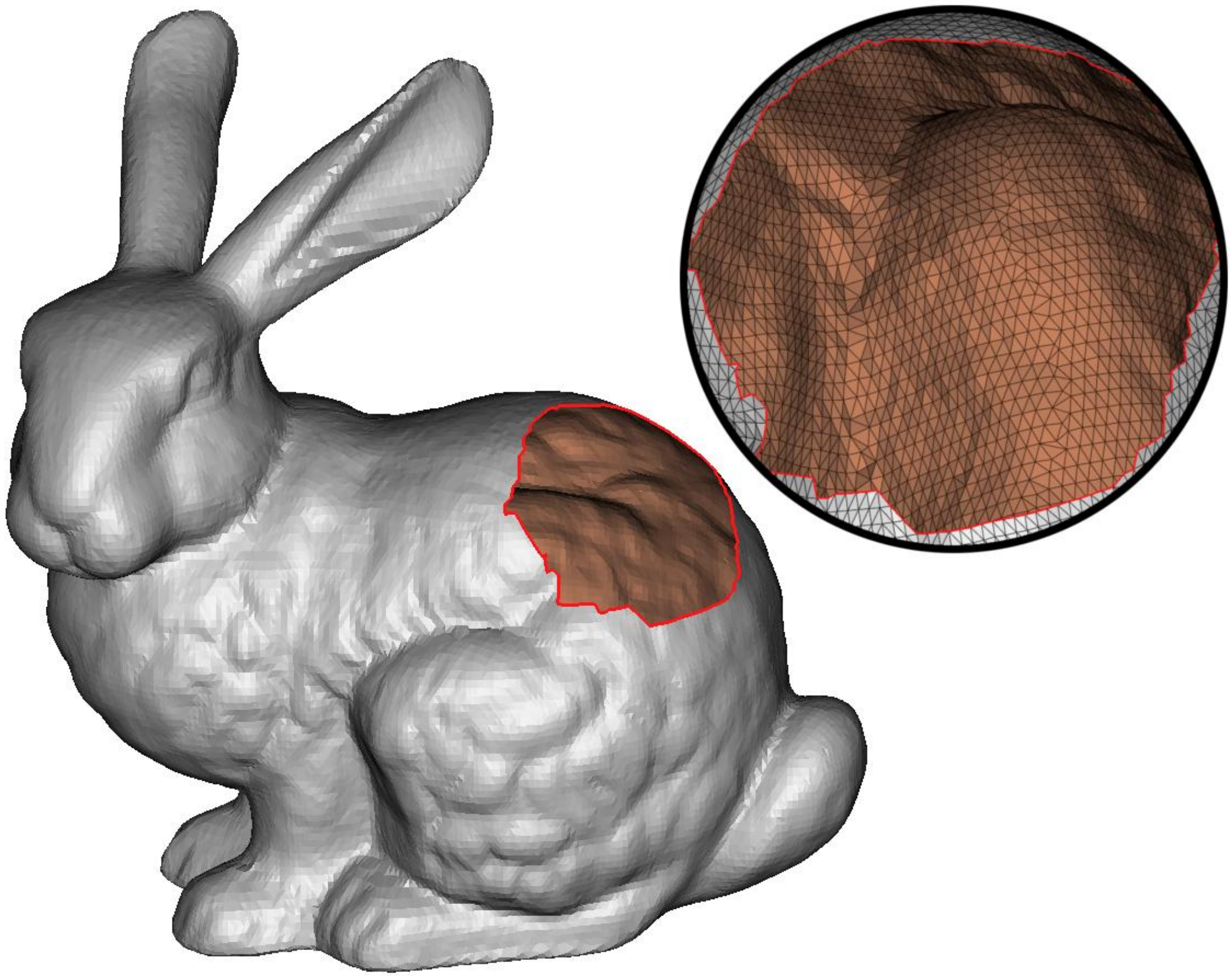
University of Brescia, Italy

A triangular mesh

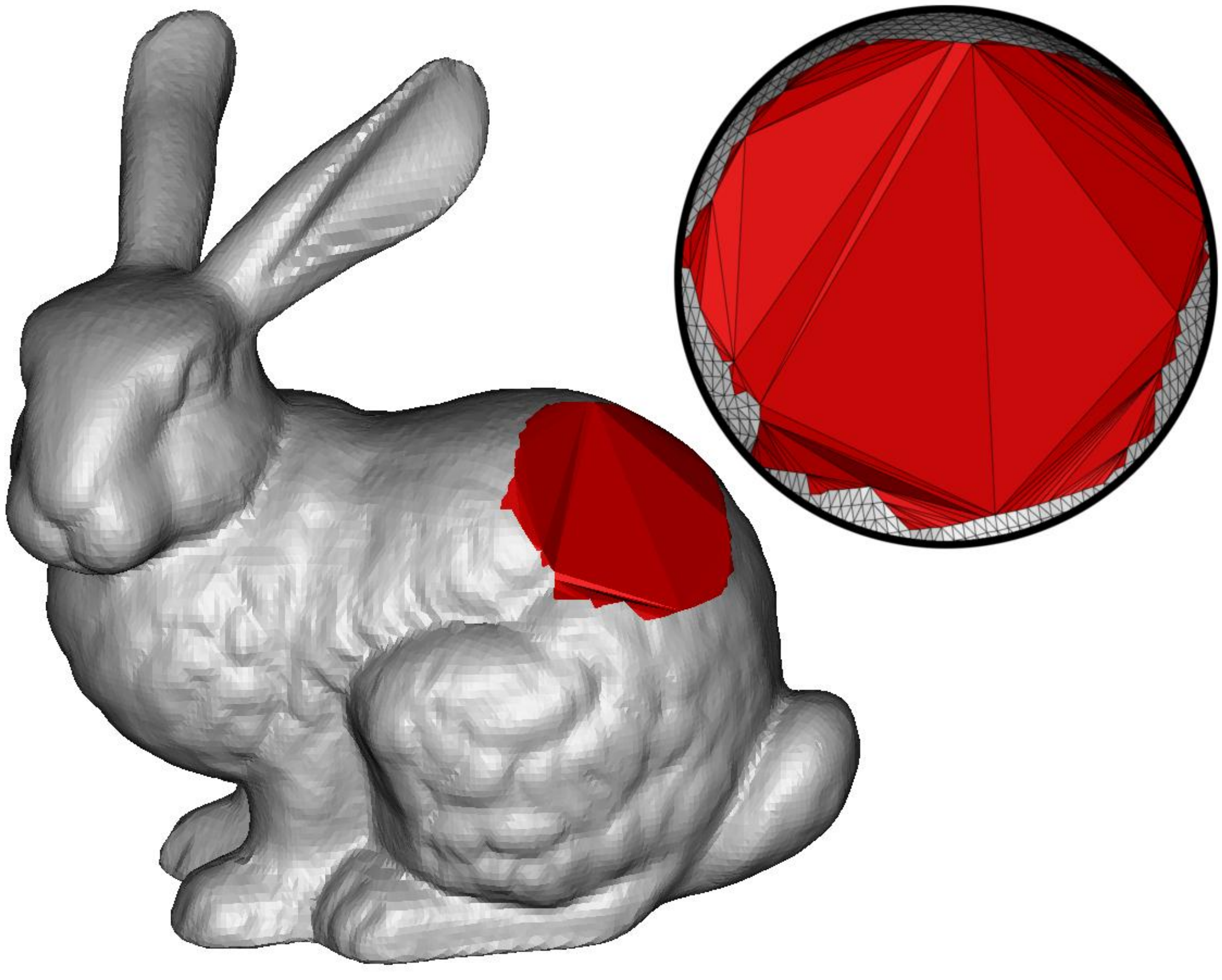




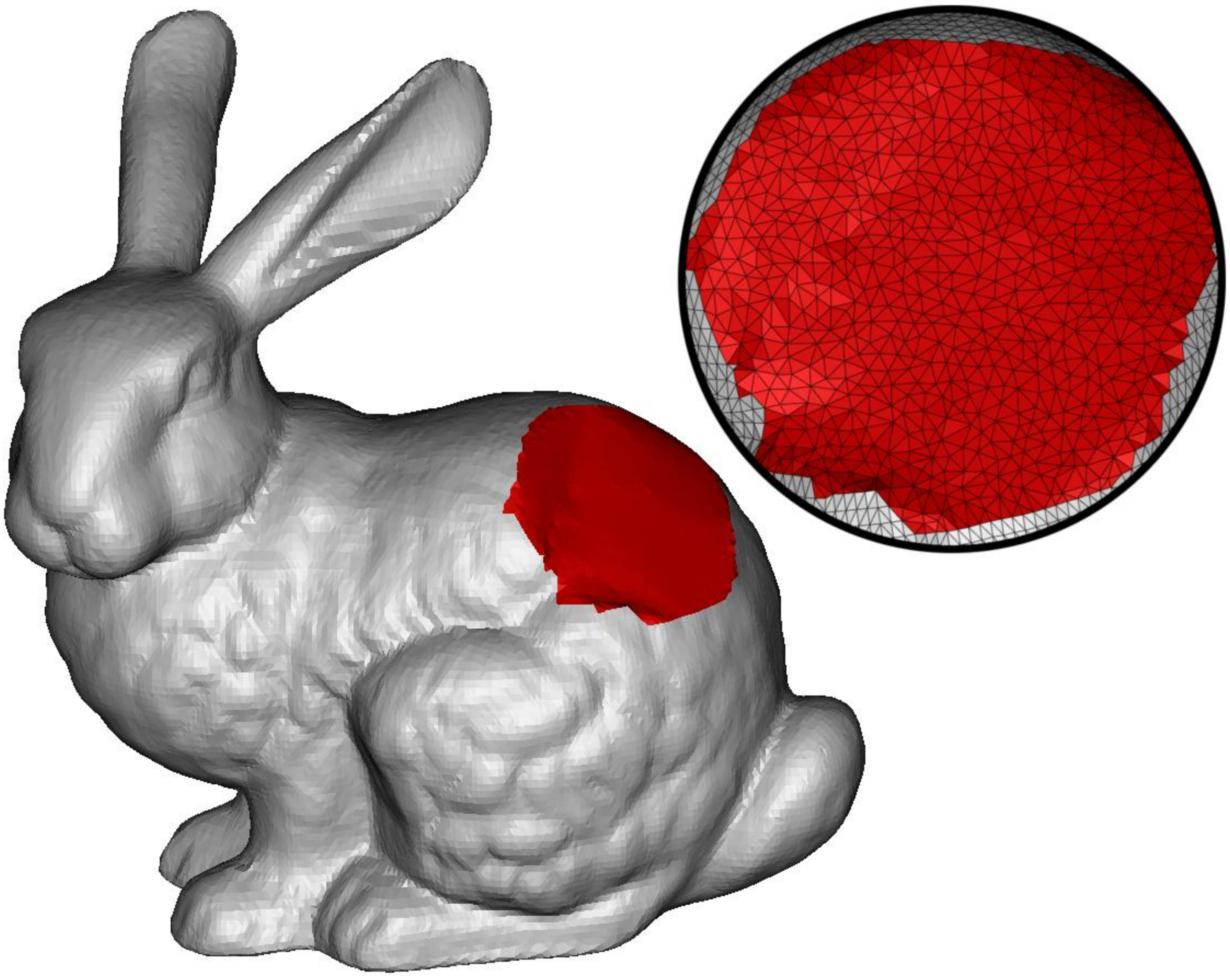
A simple "hole"



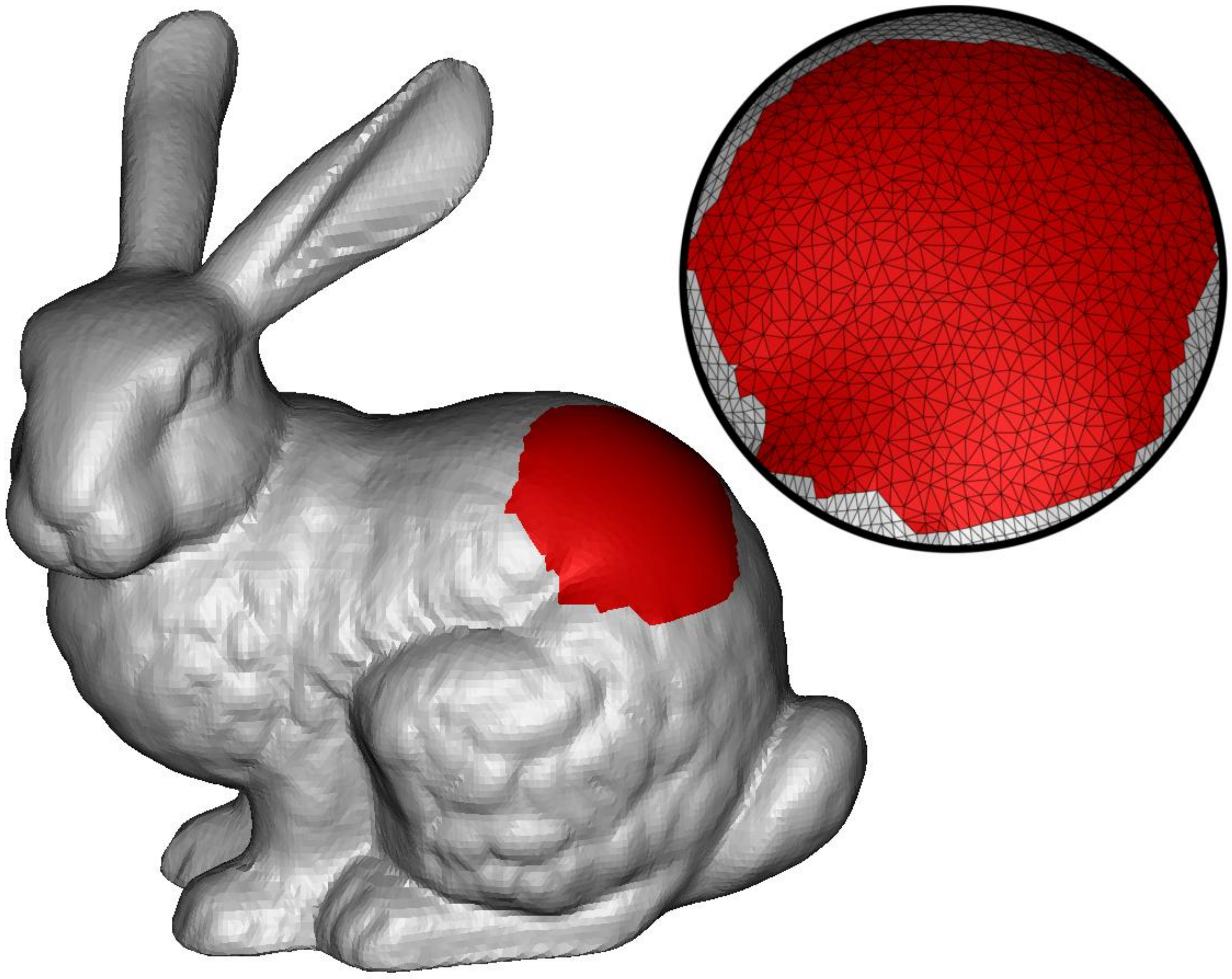
Triangulating the boundary loop



Remeshing of the patch

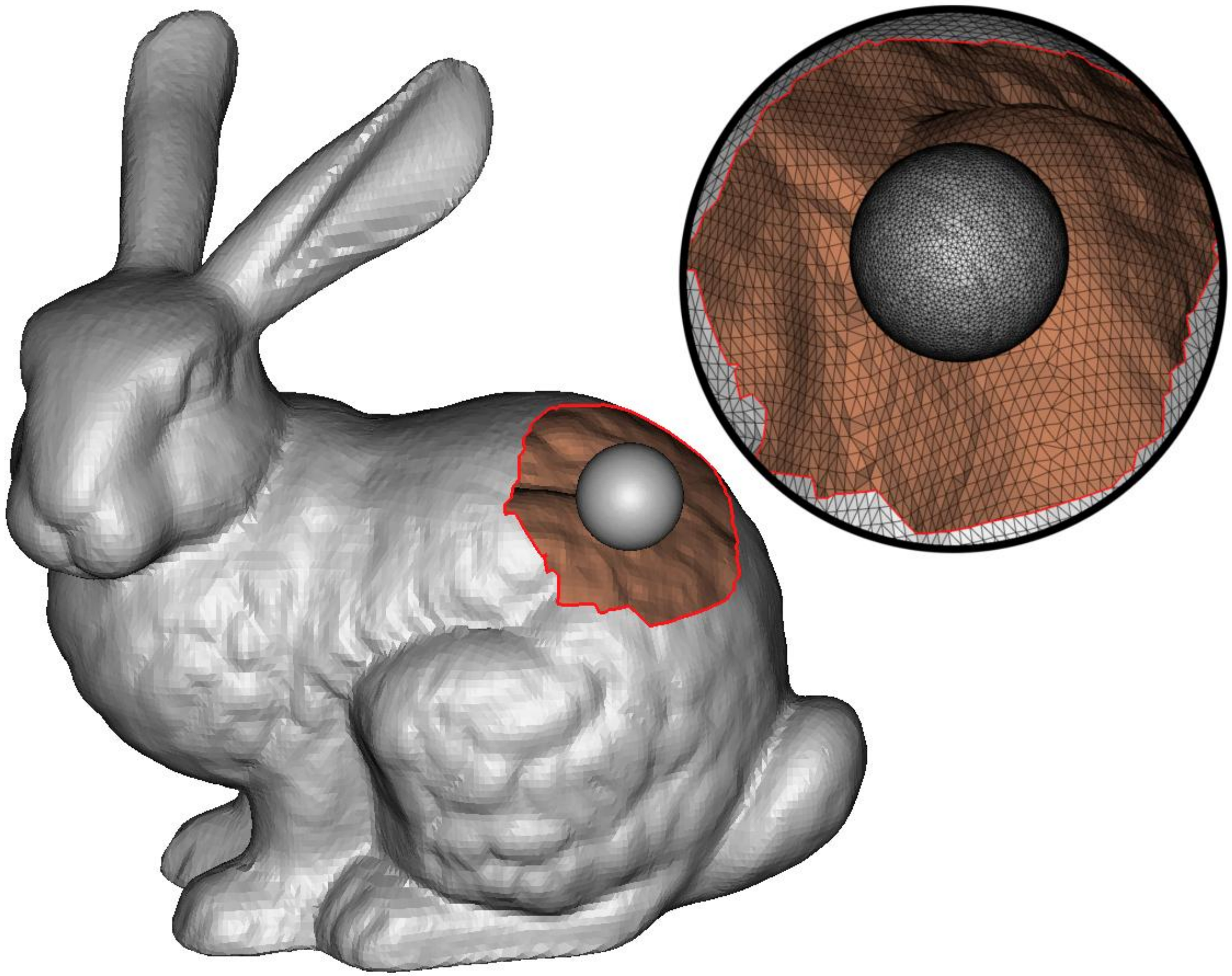


Fairing, laplacian editing

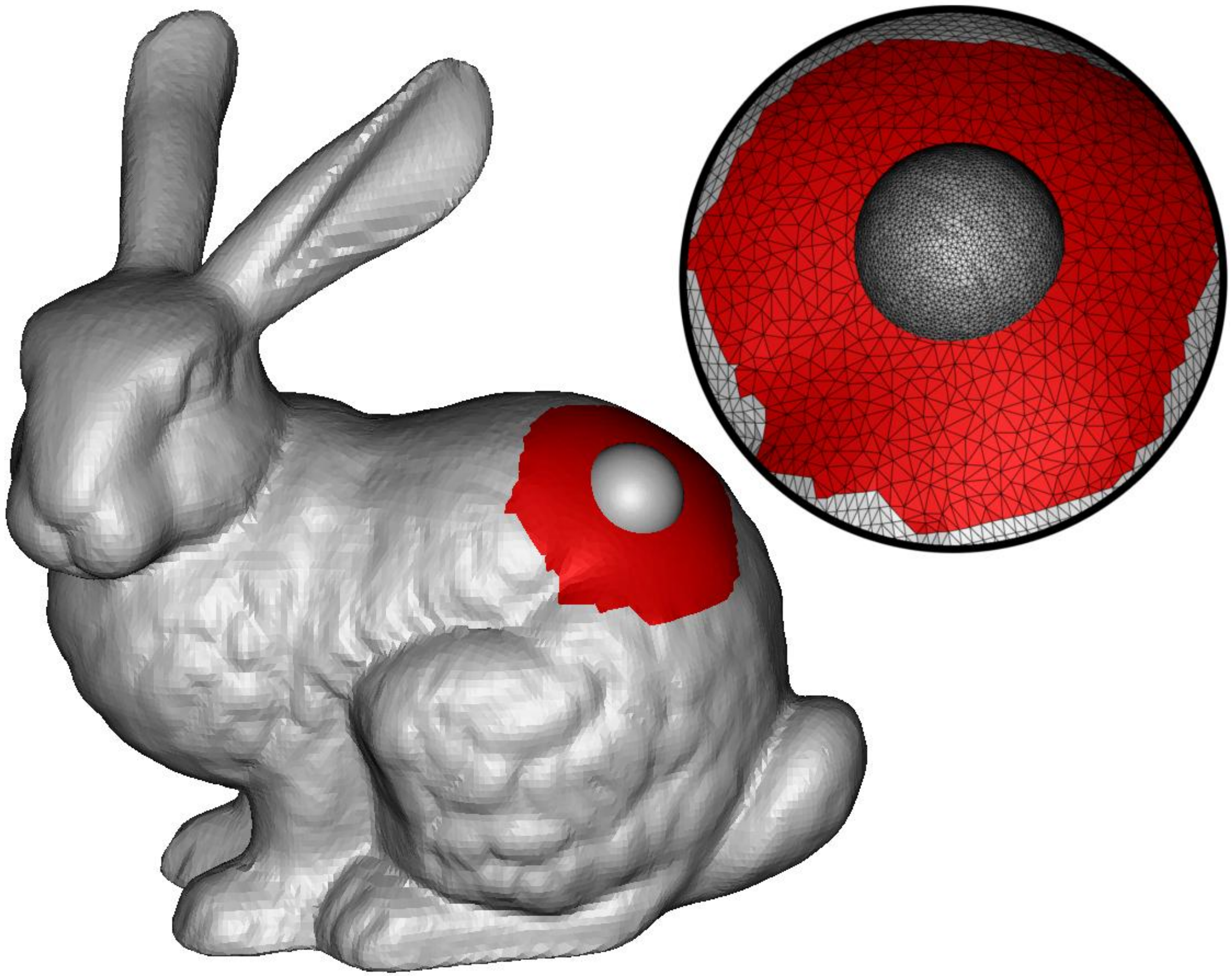


Thank you for your attention!

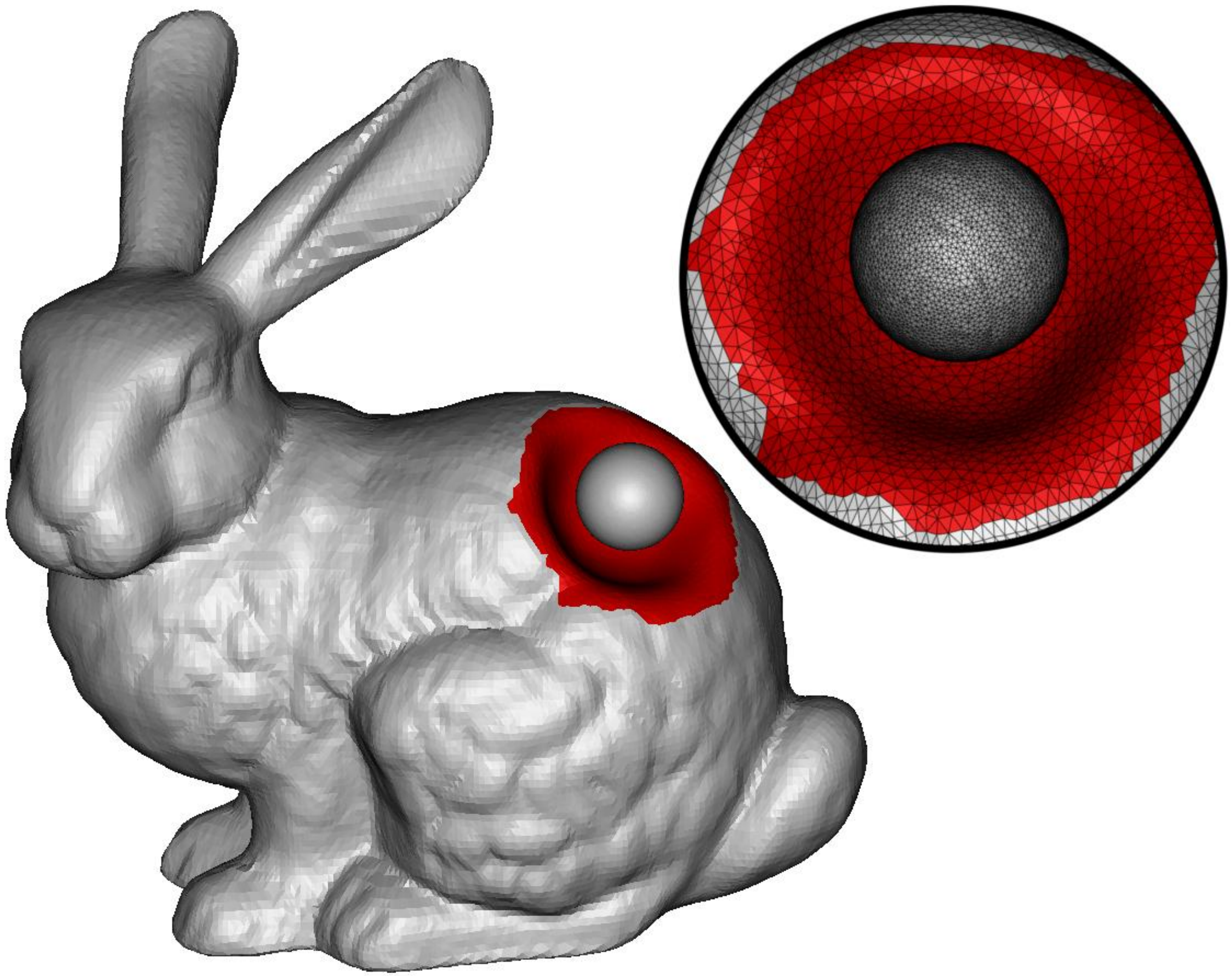
What if the hole is not... empty?



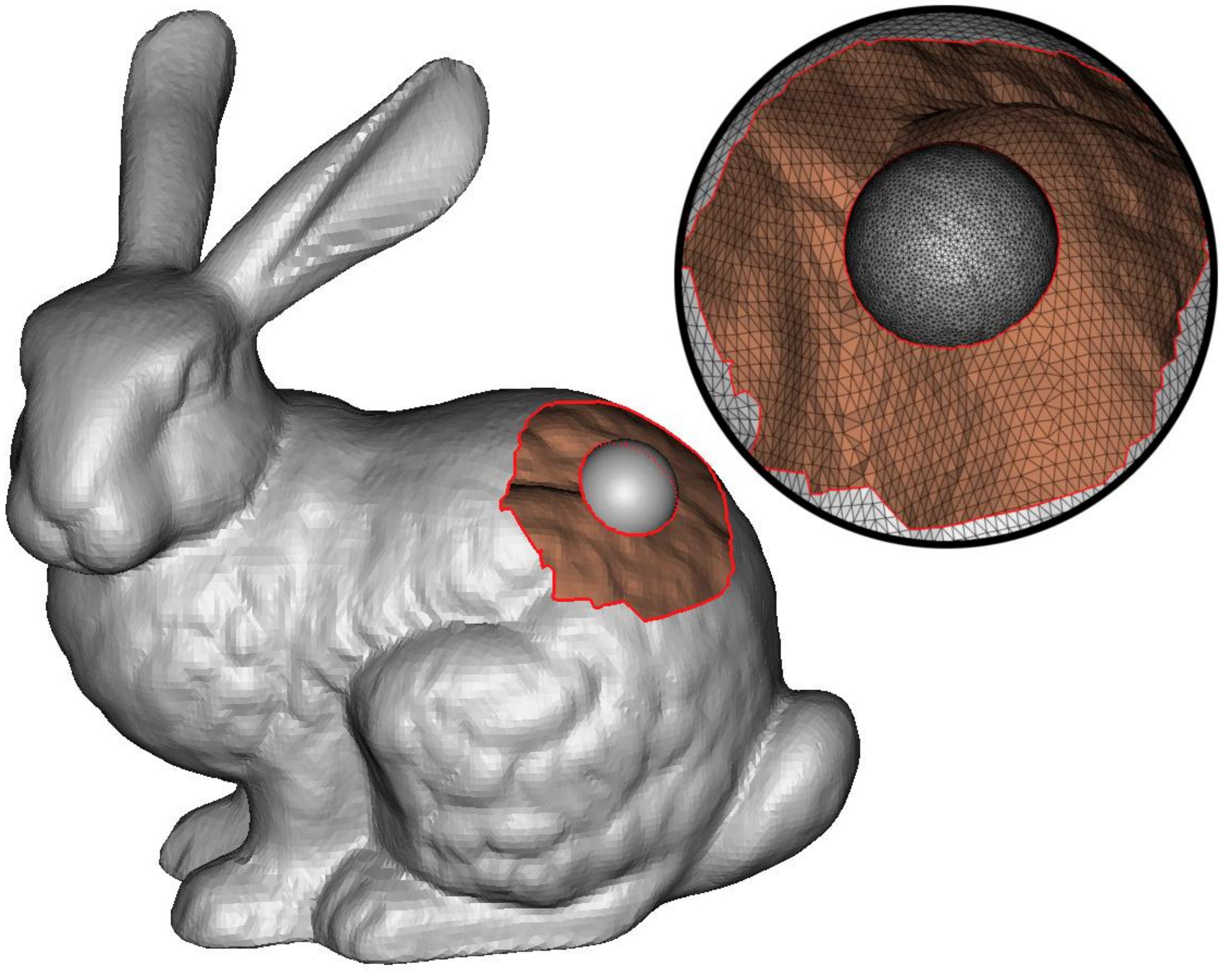
Self intersections



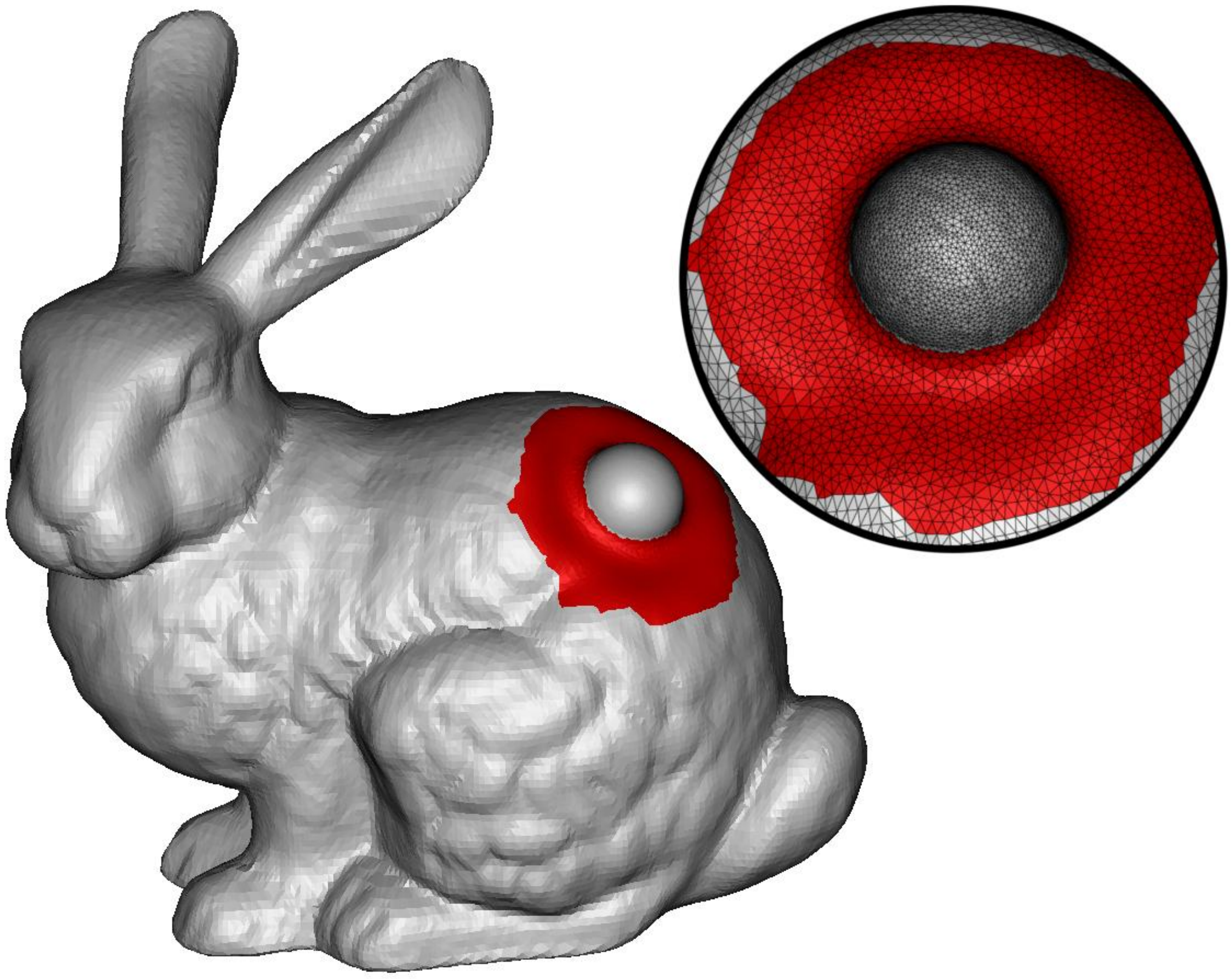
Hole Filling vs Mesh Completion



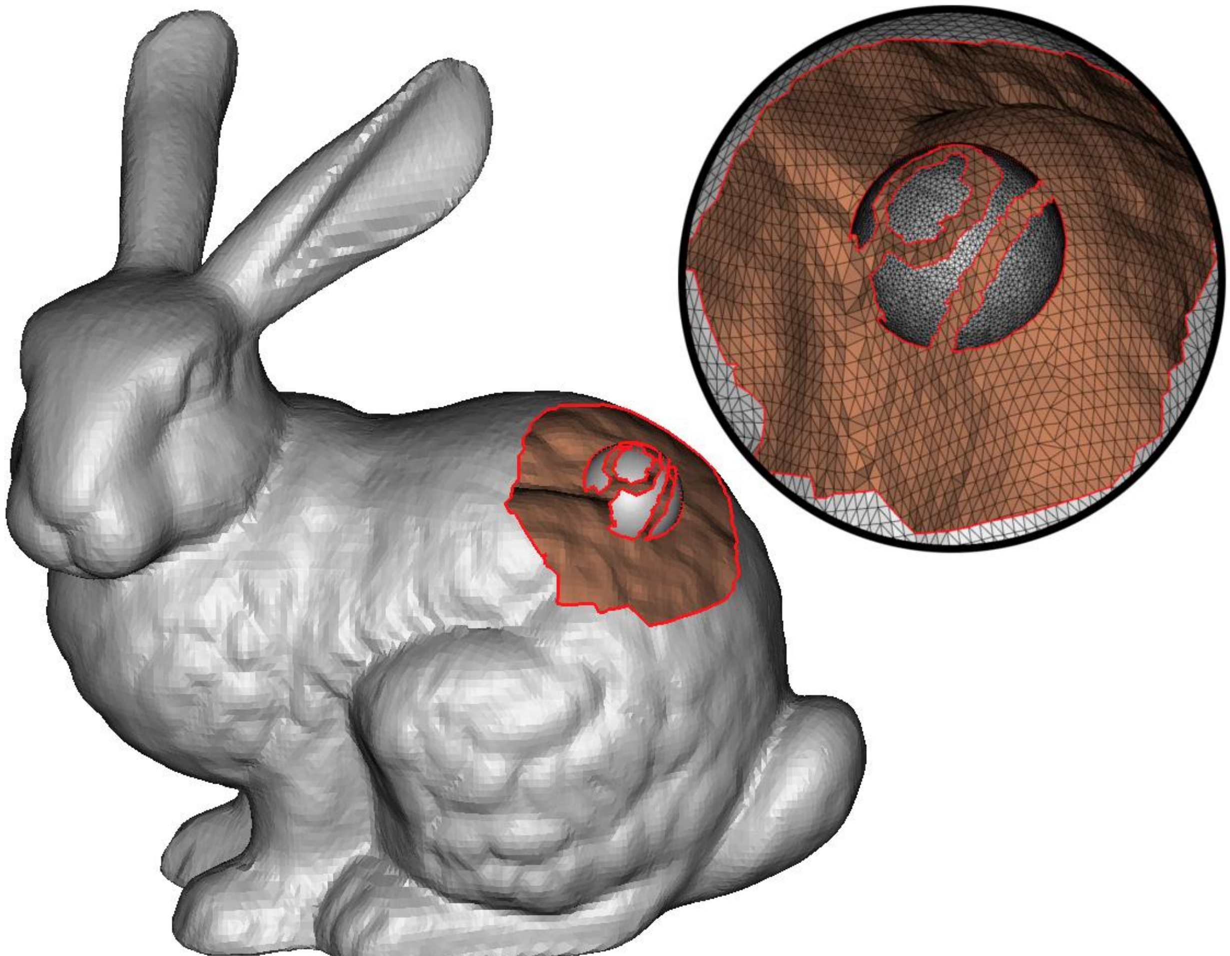
Hole Filling vs Mesh Completion



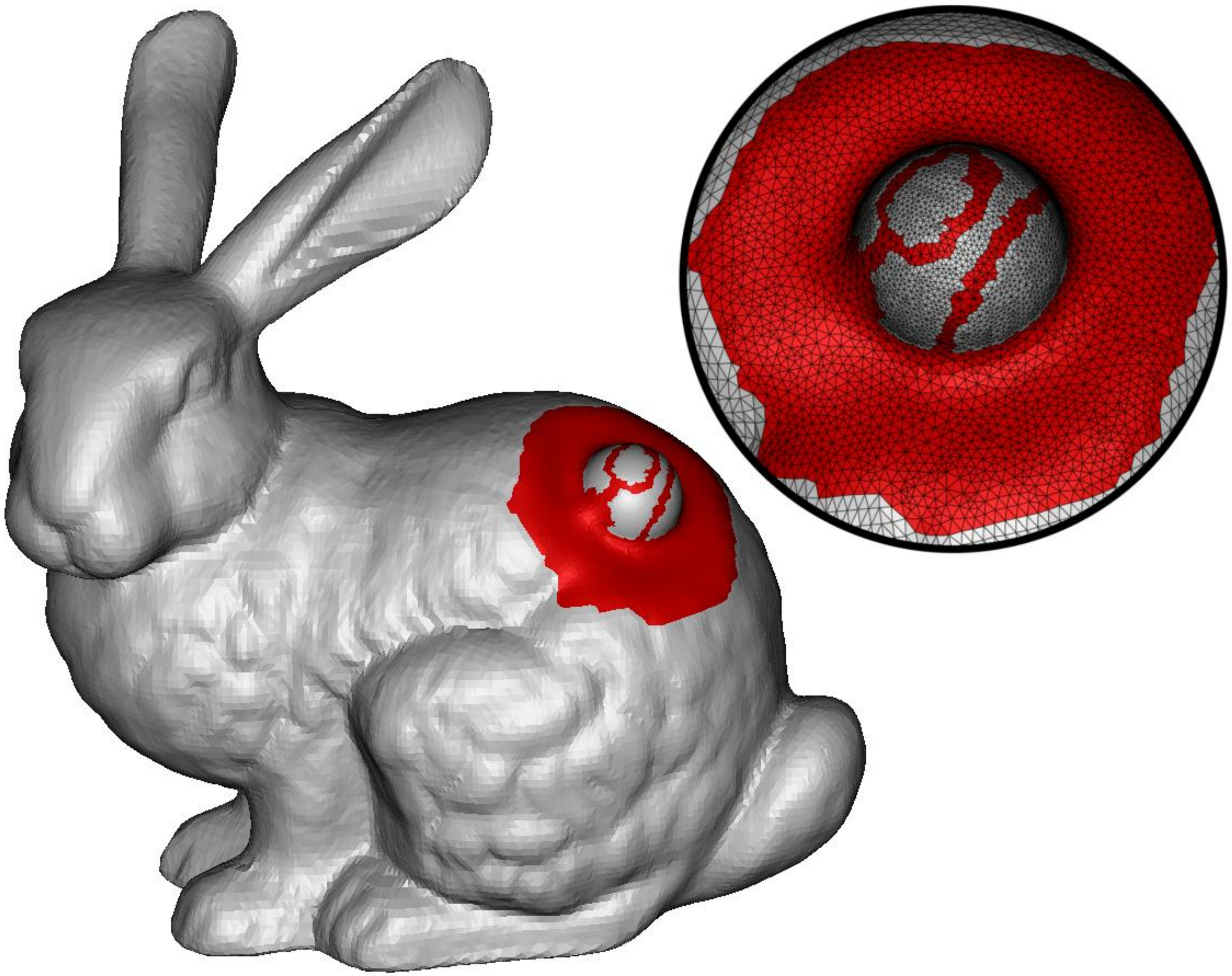
Hole Filling vs Mesh Completion



Is the boundary the key?

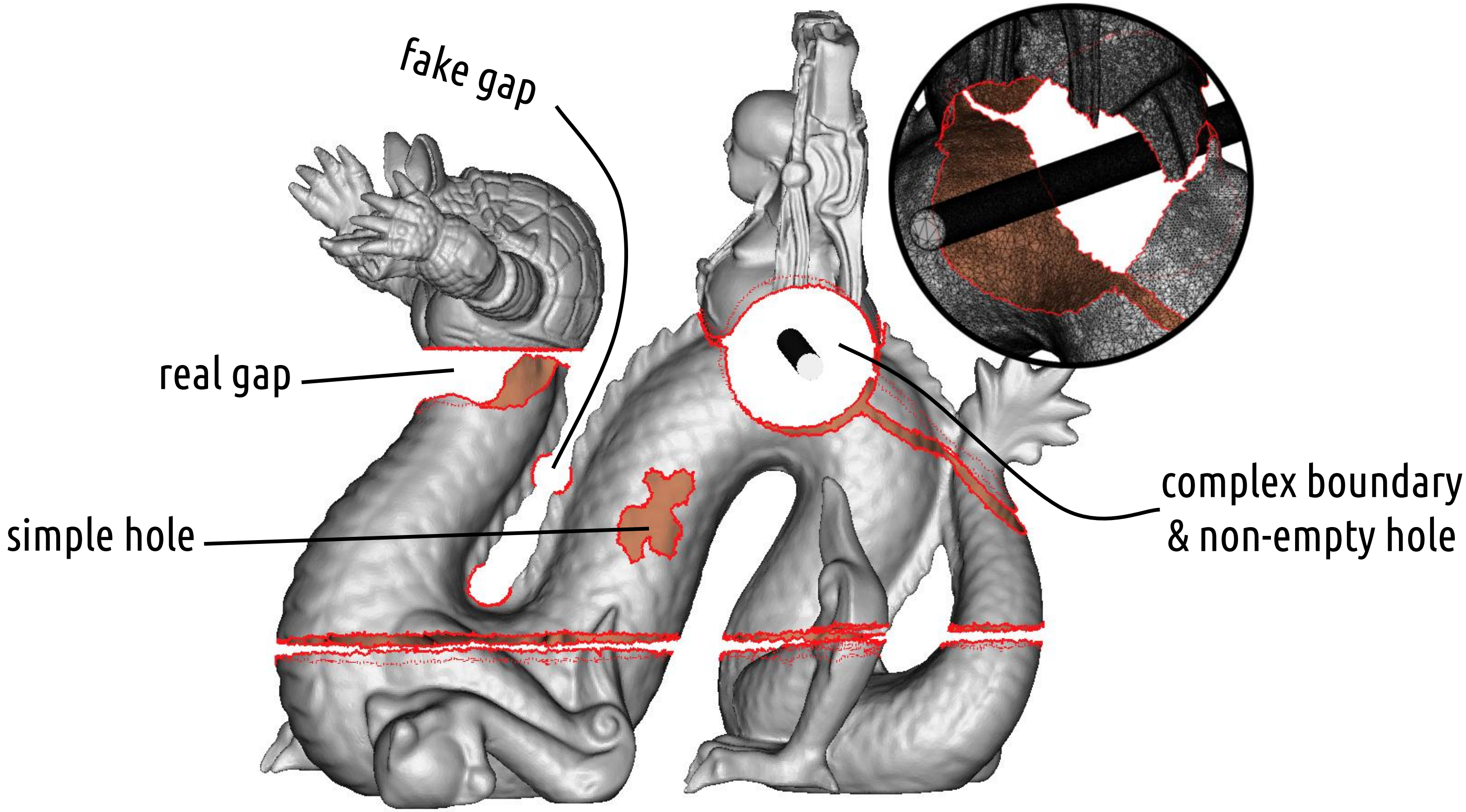


Is the boundary the key?



The proposed method

Input and requirements



Input (example)

Step 1: reconstruction

We compute a **Poisson Reconstruction, but...**

- We are interested in the missing parts only!
- Coarse reconstruction in known zones
- We do not extract the mesh in this step

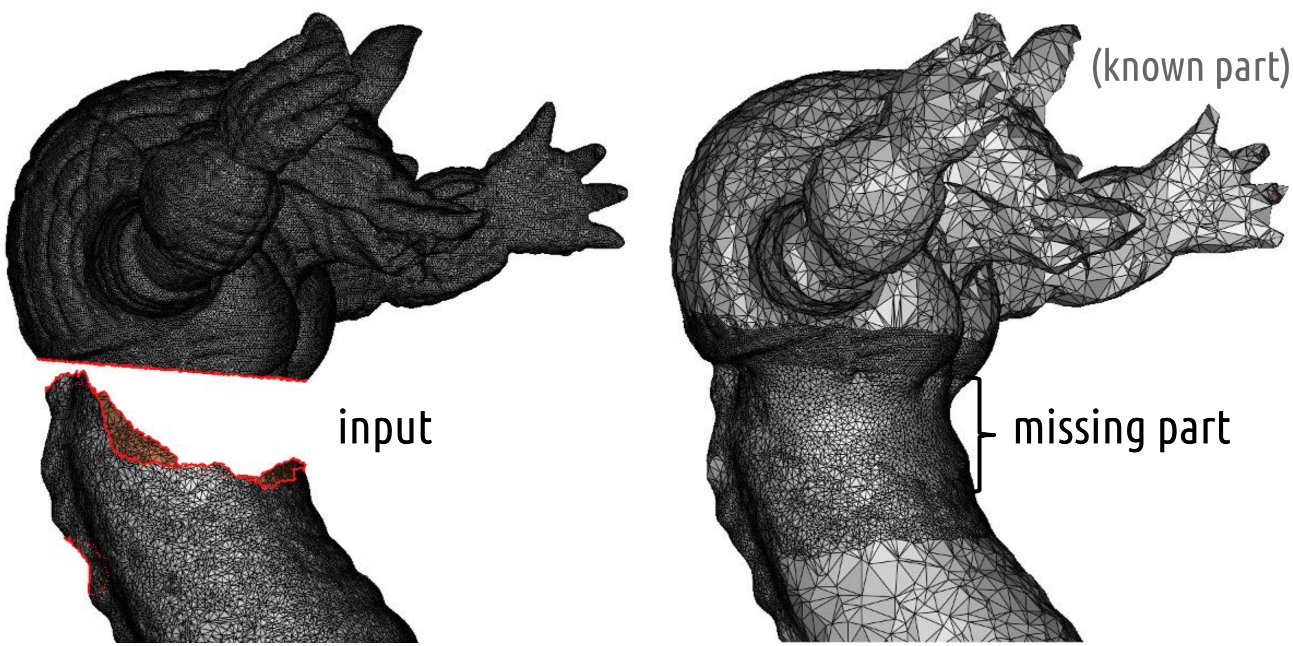
Output of this step:

the characteristic function X(p) of the object

Step 2: generate completion mesh

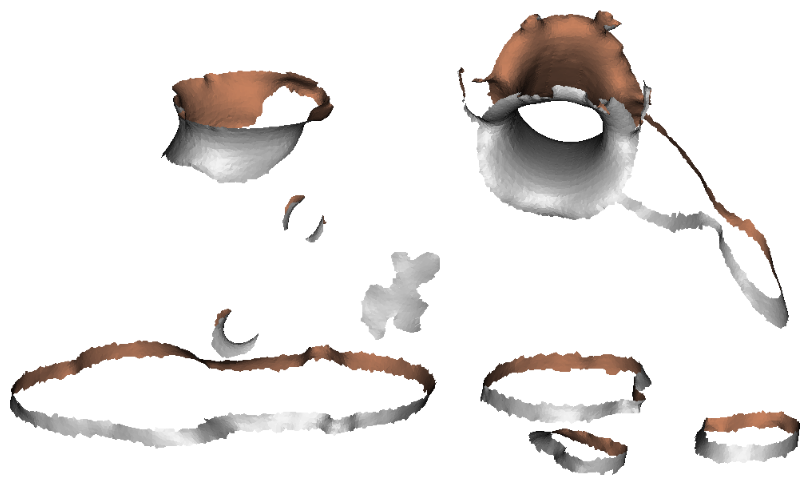
We adopt an Iterative **Delaunay refinement** method

- Known parts of the mesh remain coarse
- Boundaries need to be well represented



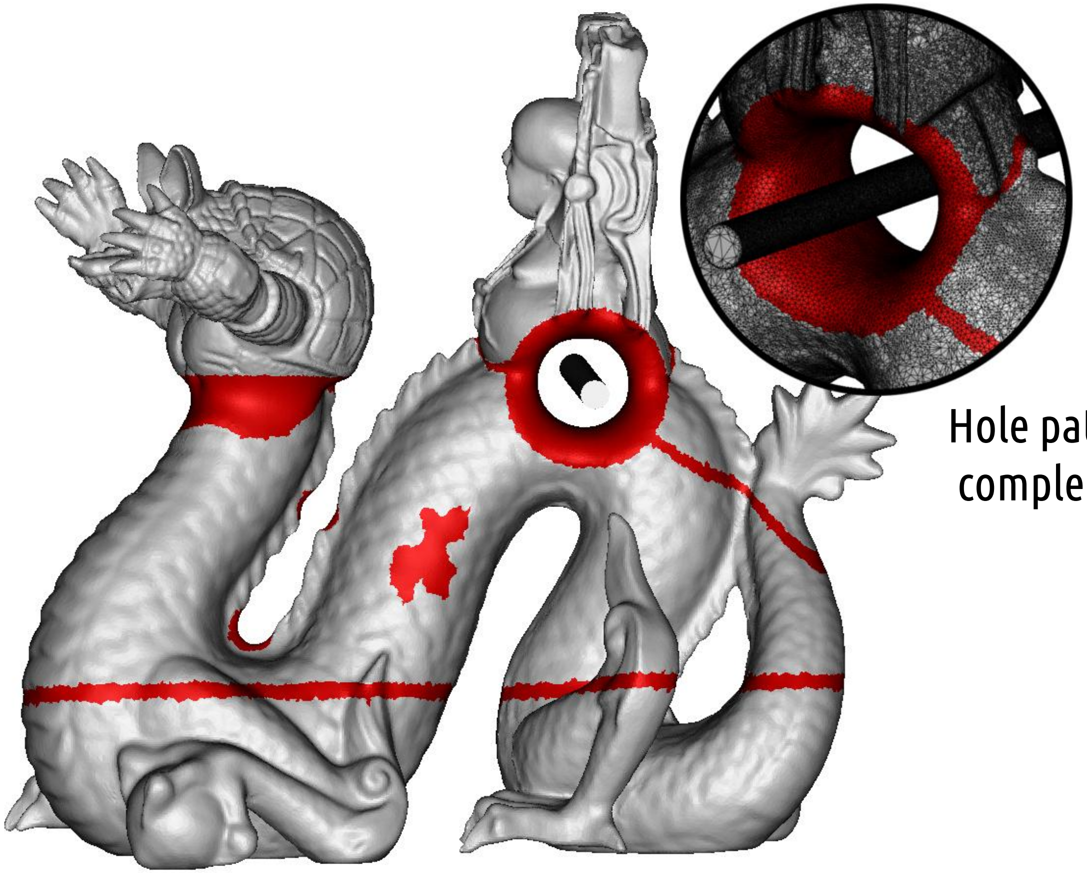
Step 3: tailoring the hole patches

1. Match the boundary vertices (and their 1-ring)
2. Cut the completion mesh and extract hole patches:



3. Merge the patches with input mesh
4. Eventually close small gaps along the boundaries

Output



Hole patches from completion mesh

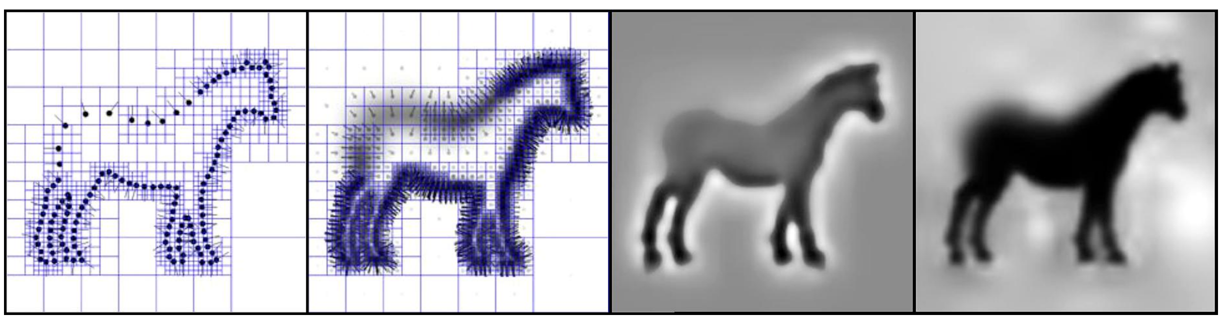
Few technical details

Review of Poisson Reconstruction

Normal field V as the gradient of the characteristic function X:

$$\min_{\chi} \int \|\nabla\chi(\mathbf{p}) - \mathbf{V}(\mathbf{p})\| d\mathbf{p} \qquad \Delta\chi = \nabla \cdot \mathbf{V} = \mathrm{div}(\mathbf{V})$$

Problem is discretized on a set of RBFs $\{B_1, \ldots, B_n\}$ centered in the nodes of an adaptive **octree**, constructed from the samples. A multigrid solver computes coefficients $\{x_1, \ldots, x_n\}$ s.t. $X(p) = x_1 B_1 + \ldots + x_n B_n$.



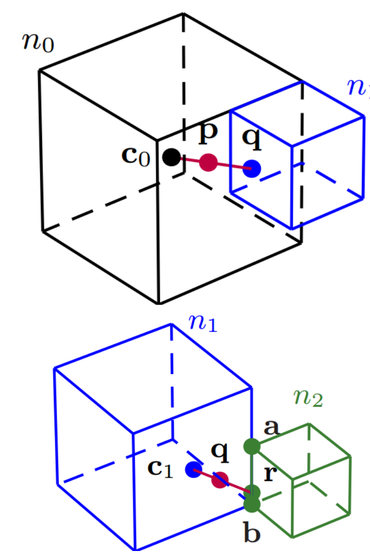
Note: direct computation of X(p) is expensive.

Efficient interpolation of X

We pre-compute the values of X at the corners of the octree.

Given a point p in space, we want to interpolate X(p) from them.

1. Locate the leaf node n_0 containing p;
2. Face projection: cast the ray from c_0 to p, get q;
3. Set n_1 to be the smallest node across that face;
4. Edge projection: cast ray from c_1 to q, get r;
5. Set n_2 to be the smalles cube on that edge;



Linearly interpolate inside the tetrahedron with vertices (c_0, c_1, a, b), so that *continuity* is guaranteed.

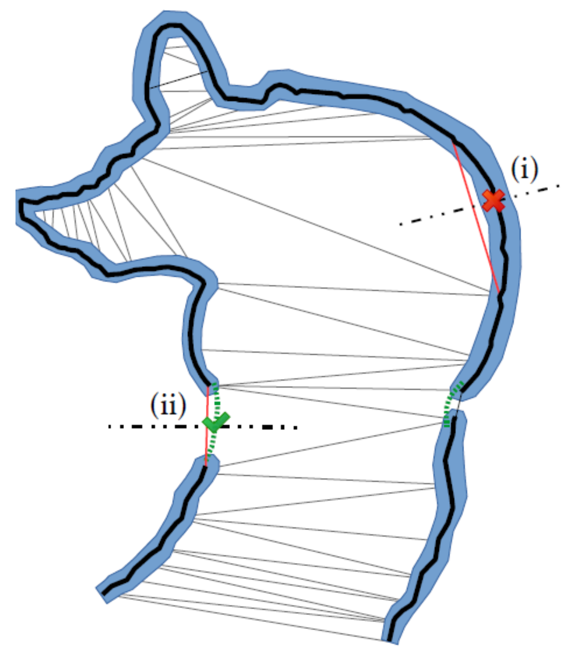
Intersect a segment with surface: bisection method.

Completion mesh generation

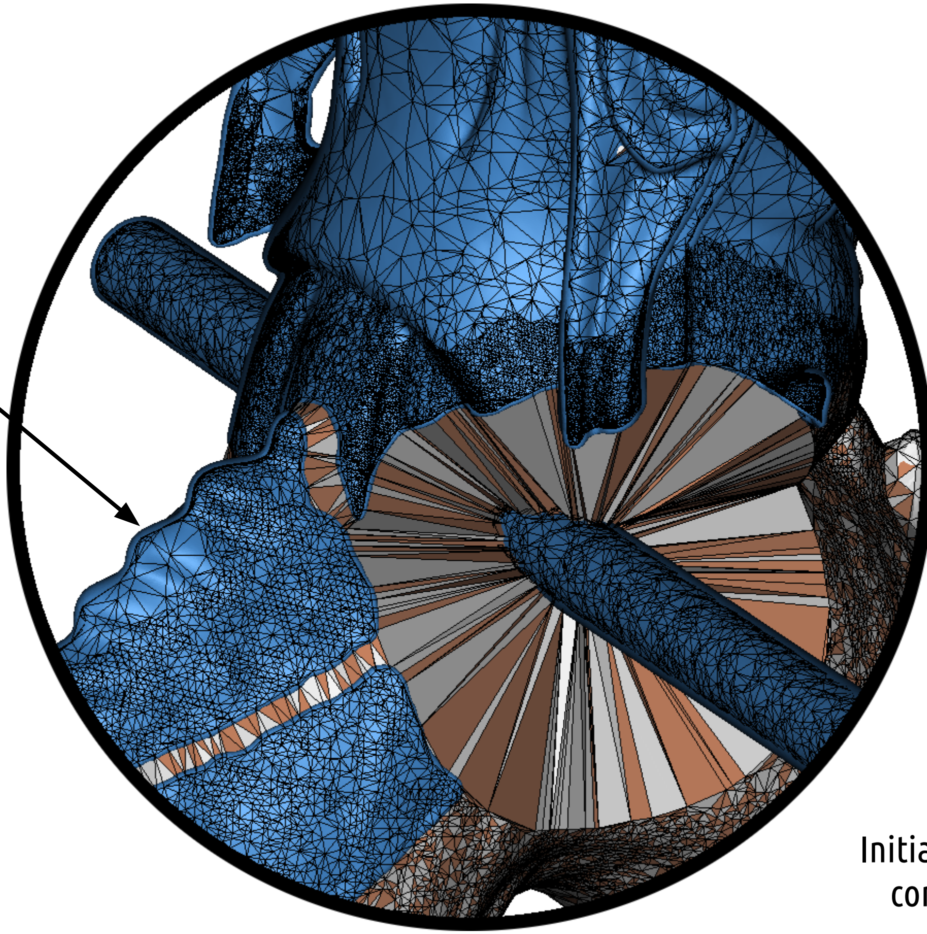
We block refinement inside a cushon of the input

Algorithm 1 Modified Delaunay Refinement

function $\mathfrak{T} = \mathfrak{D}(\mathfrak{S}, \mathfrak{Q}, \mathfrak{P}, E_0)$
    $\mathfrak{T} \leftarrow$ initial tessellation of E_0
    $\mathfrak{B} \leftarrow$ bad elements of $\mathfrak{T}$ with respect to $\mathfrak{Q}$
    while $\mathfrak{B} \neq \emptyset$ **do**
        $t \leftarrow$ pull worst element from $\mathfrak{B}$
        $s \leftarrow$ dual segment of t
        $p \leftarrow$ intersection of s with $\mathfrak{S}$
        if p is outside $\mathfrak{P}$ **then**
            refine $\mathfrak{T}$ by adding p
            update $\mathfrak{B}$ with new bad elements
        end if
    end while
end function

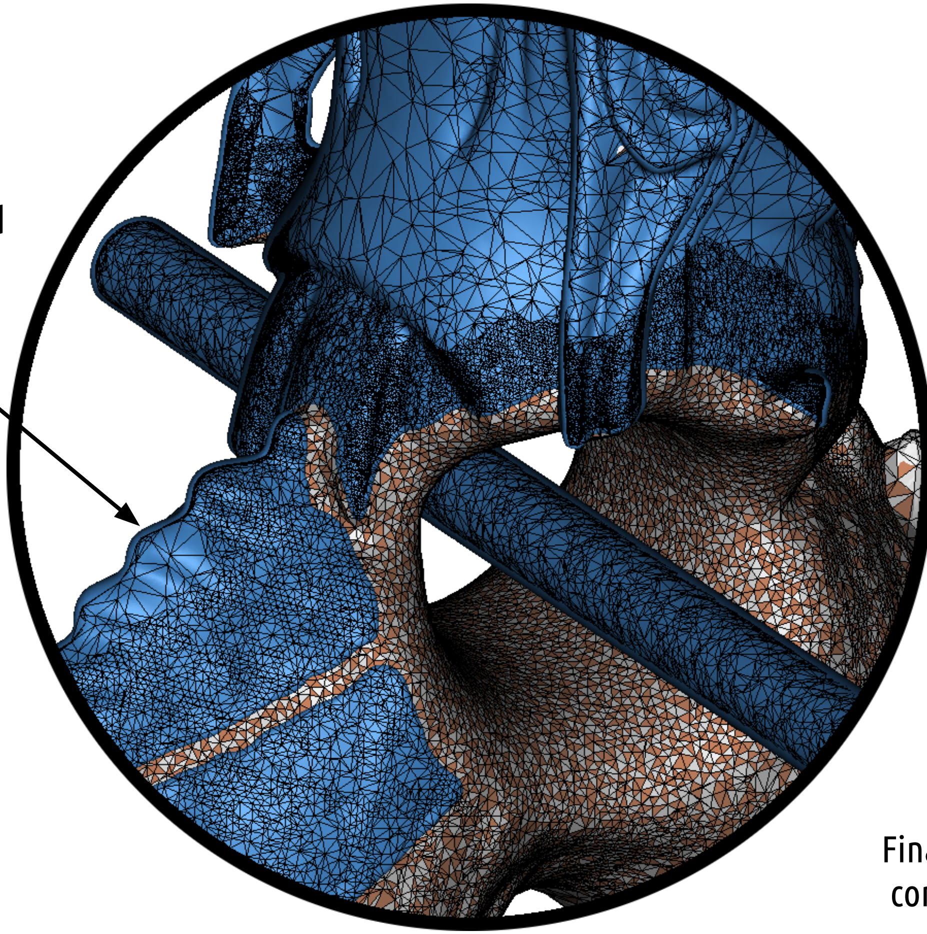


Completion mesh generation



Completion mesh generation

Refinement is blocked
inside this cushon!

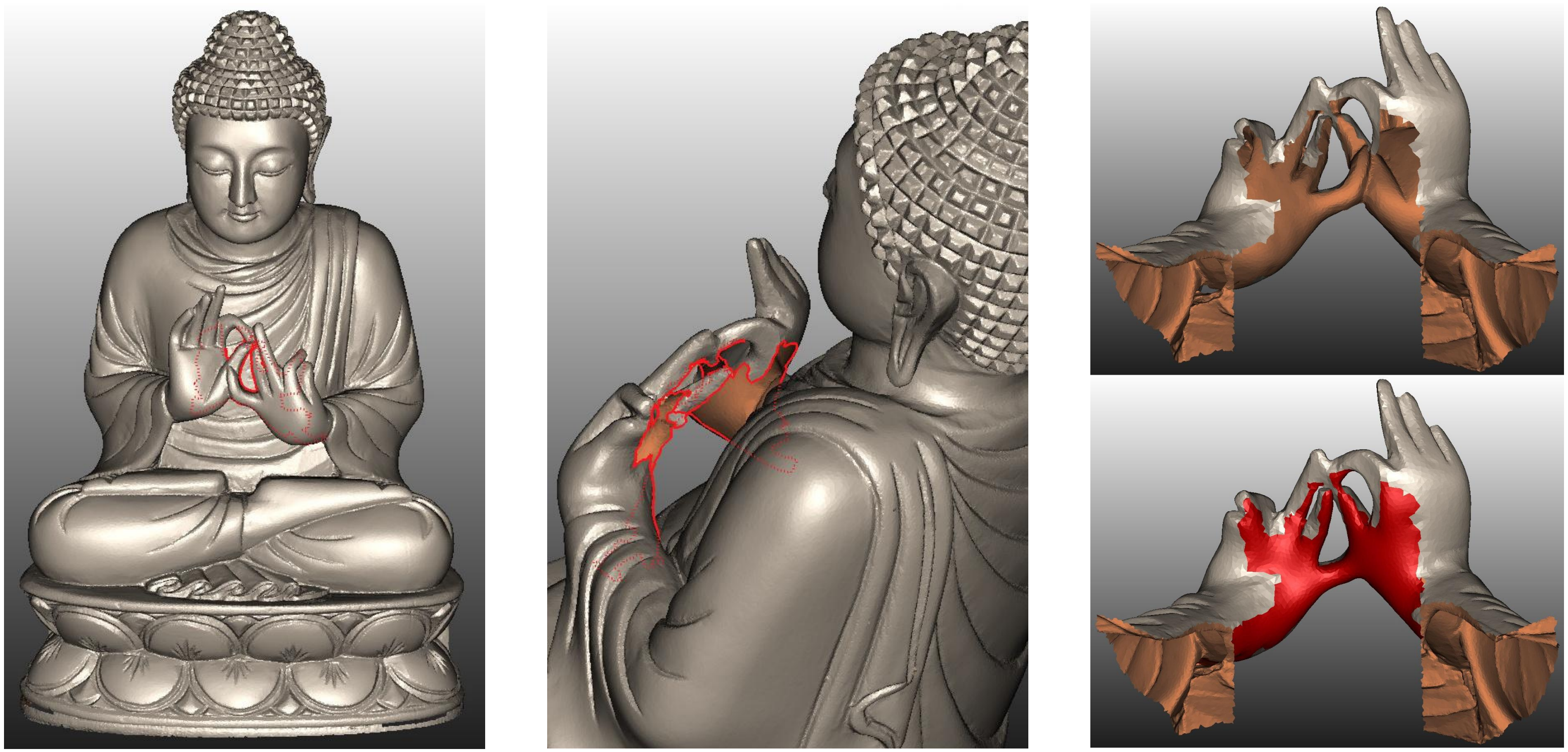


Final restricted
complex faces

Applications and results

Hole filling

Filling complex holes that require a volumetric reconstruction



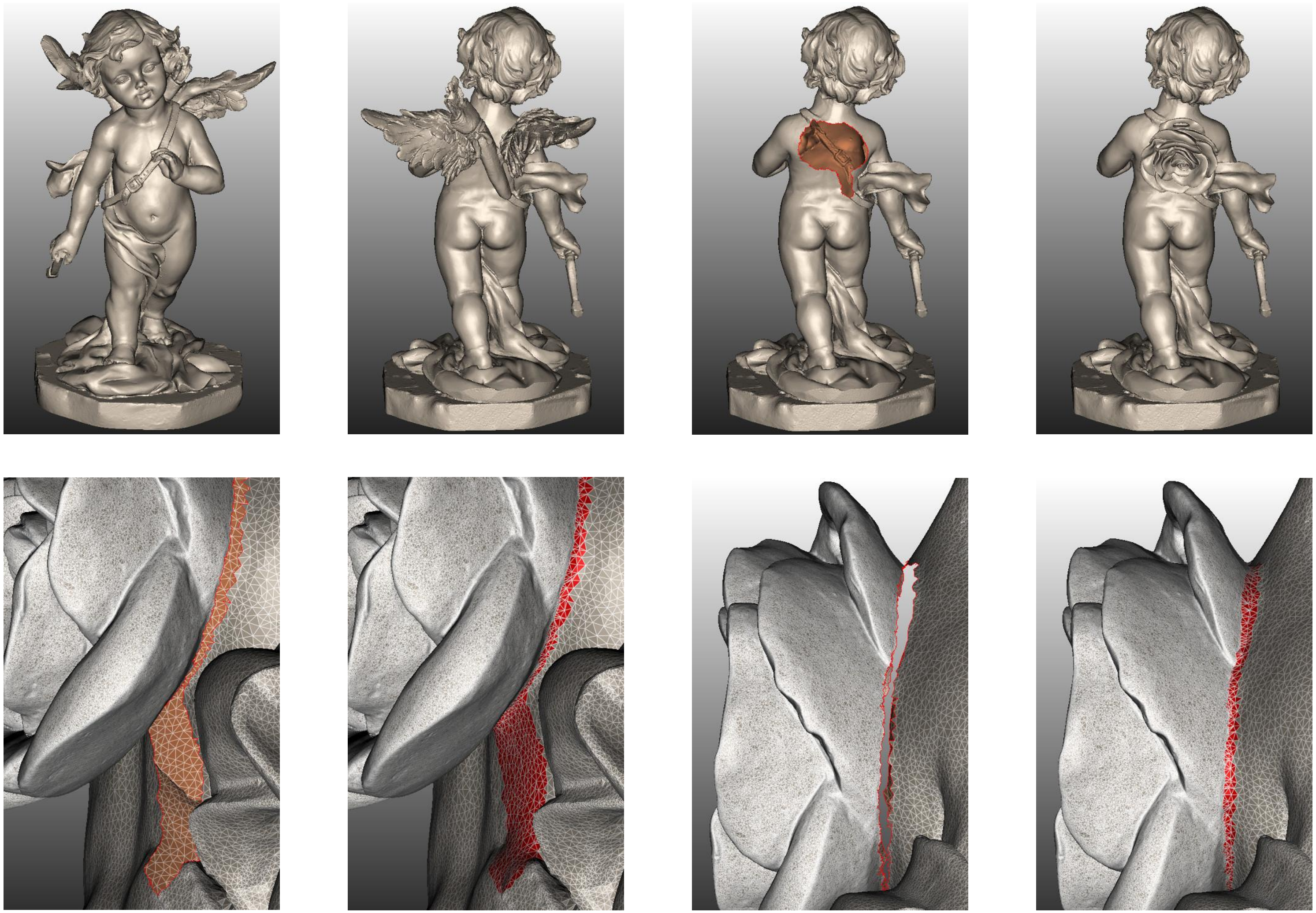
Input　　Detail　　Completion

Hole filling

Finalizing a model by reconstructing missing parts

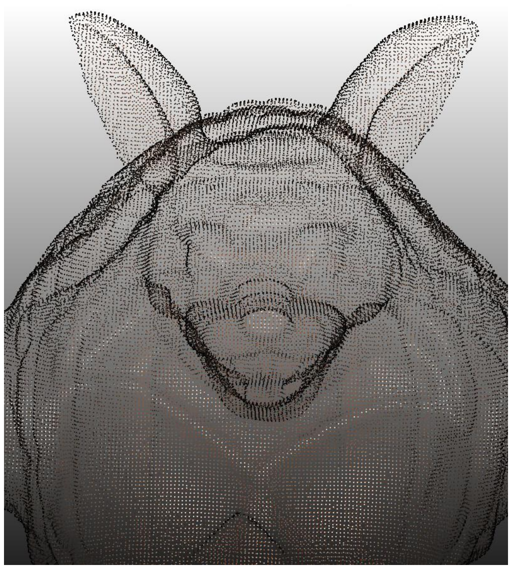



Creative editing

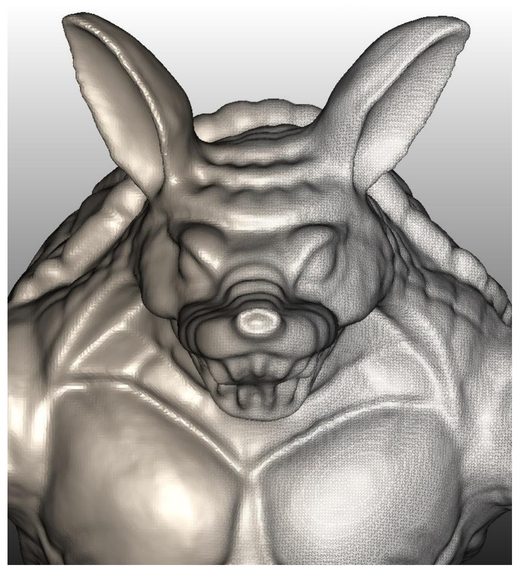


Remeshing

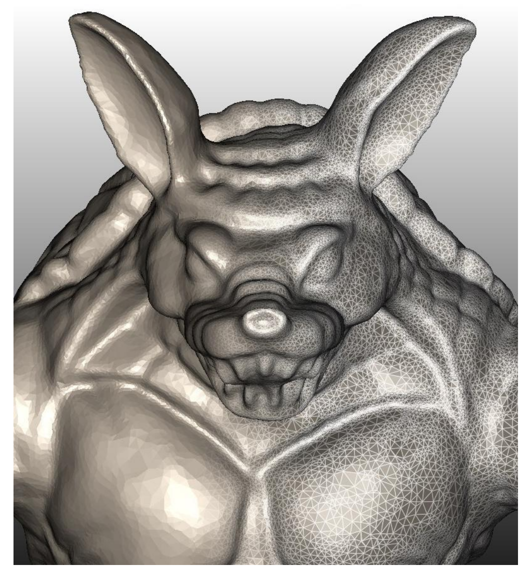
Improve the quality of the mesh of a Poisson reconstruction
by adopting a Delaunay-refinement with our interpolated oracle



Input



Original Poisson



Delaunay-Poisson

Thank you for your attention!