OpenVDB Remeshing filter

This filter uses the OpenVDB Library to perform a remeshing of an input mesh.
The goal was to include a(n even) faster remeshing algorithm in VCGlib and meshlab.

Table of contents

1. Compilation Requirements

OpenVDB requires:

- C++17
- TBB
- Boost
- Eigen 3

2. VCGlib part

A `OpenVDBAdapter` has been defined in `vcglib/wrap/openvdb/`. This adapter exposes the remeshing function used in OpenVDB, handling the conversion between VCGlib's internal mesh representation and OpenVDB's.

The two main functions are `meshToLevelSet` and `volumeToMesh`, which invoke the respective functions in OpenVDB. The adapter also includes a `MeshTypeDataAdapter` that could be used in the future for other OpenVDB functions. `meshToLevelSet` did not unfortunately support it, but `meshToVolume` does.

In order to use the adapter, an instance needs to be created, templated with the correct mesh type. Then, the remeshing parameters (`isovalue`, `adaptivity`, and `voxelSize`) have to be set, along with the mesh to be processed. After this, `meshToLevelSet` can be called, which will convert the mesh into a level set representation, which is stored in the adapter. At this point, the reference to the input mesh is not needed anymore. Finally, `volumeToMesh` can be called, to obtain back a triangular mesh.

2.1. Notes on `volumeToMesh`

Due to how OpenVDB implements this procedure, the output is made of two arrays: one for tris, and one for quads. One of the processing steps that the adapter function does is then to split

quads into two tris. Another quirk of the OpenVDB implementation is that the triangles are oriented inwards. The adapter function flips them by reordering the vertices.

2.2. openvdb-remesher app

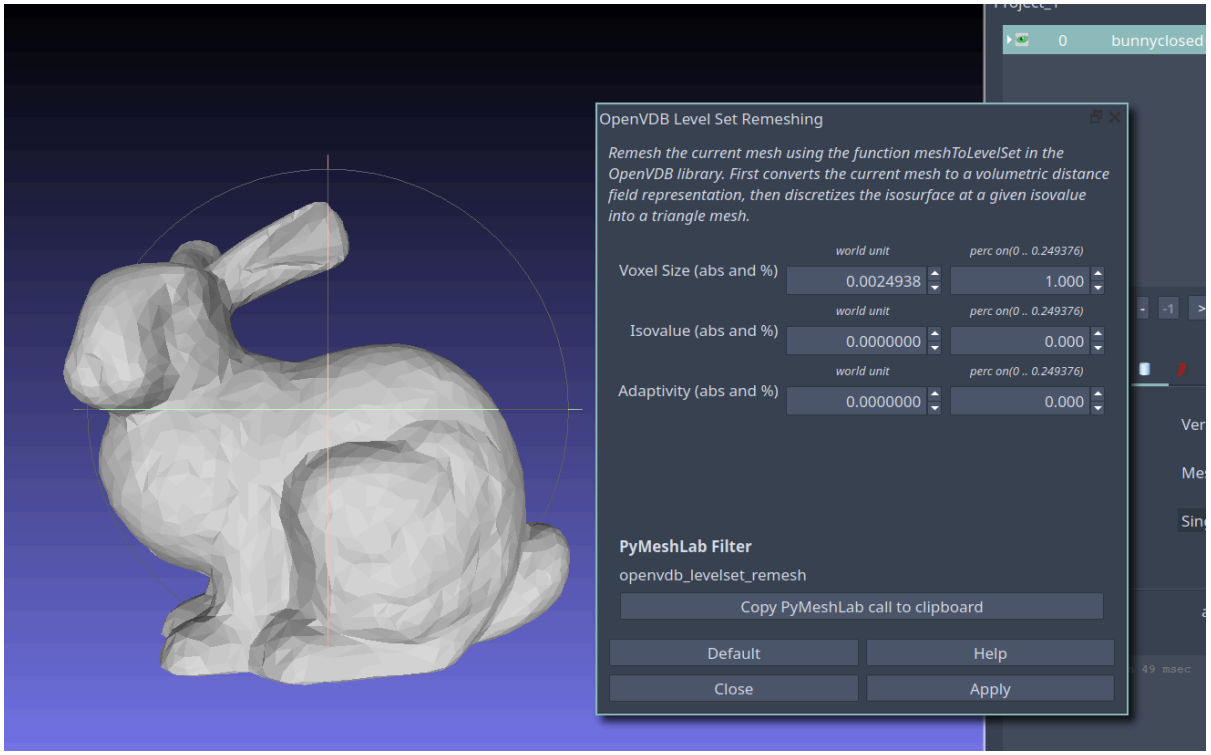
To exemplify the usage of the adapter, a simple program has been developed which takes a `.obj` mesh in input and remeshes it. This is to show the intended usage of the adapter and the compilation requirements. It can be found in `vcglib/apps/openvdb`.

3. Meshlab plugin

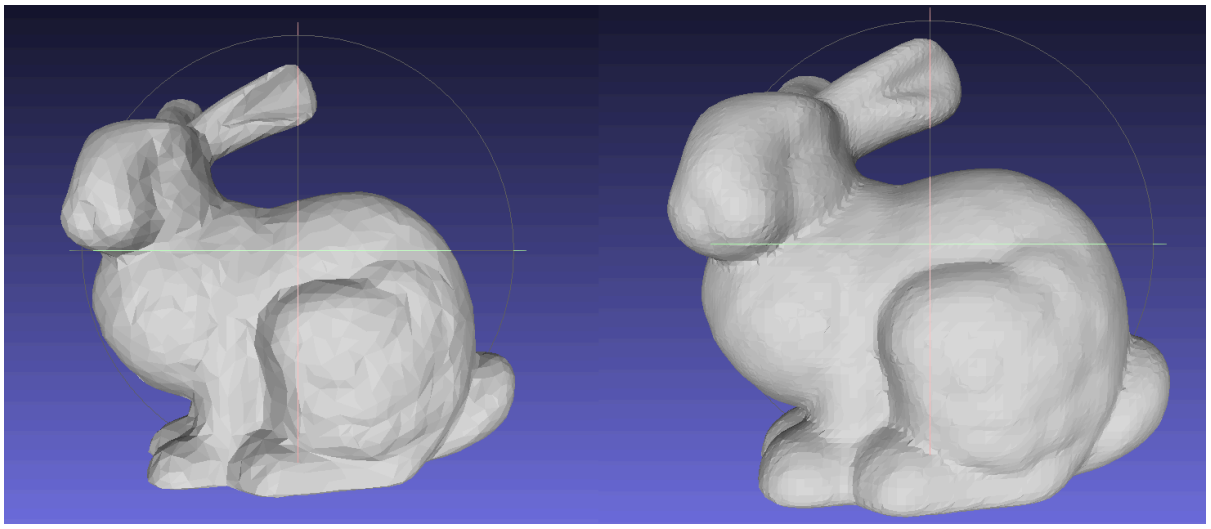
The part described above has been then used to make a meshlab filter that applies the remeshing.

Notable parts about this are the checks in place to make sure that the input mesh is watertight, which is required by `meshToLevelSet` to yield a satisfactory result. Passing a non-watertight mesh leads to a recoverable exception.

Some logging has also been added to provide a measure of how long each part of the process has taken.



Dialog of the plugin that lets the user select the parameters for the remeshing.



The Stanford Bunny, before and after being remeshed (voxel size: 1%, isovalue: 2%, adaptivity: 0).