Interface manual

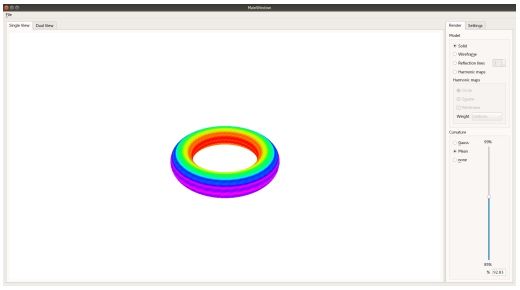
January 14, 2019

1 Introduction

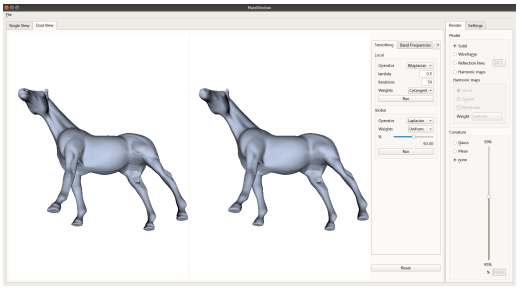
The interface is divided into two views: single view and dual view.

- Single view contains two tabs: rendering and settings. These two are still accessible in the dual view. The operations they offer are explained in section 2.
- Dual view contains three tabs, and its main characteristic is that it provides two viewers. Both viewers start with the same mesh and only the mesh in the right viewer is modified. This is useful for comparing the result of applying the operations accessible through the 3 different tabs in this view. These operations are explained in section 3.

All viewers (in both the single and dual view) allow the user to rotate the mesh and zoom in and out.



(a) Single view.



(b) Dual view.

The full project (code and documentation) can be found via this link:

`https://github.com/lluisalemanypuig/geometry-processing/`

The full details on what each option does can be found in the doxygen documentation of the library, in the directory ***geoproc-docs***. The code of the library can be found in the directory ***geoproc***, and a command-line application to apply the algorithms implemented can be found in the directory ***command-line***. The code of the interface can be found in the directory ***interface***.

See section 5 to see the state of functionality of every algorithm (what works at a 100%, what is not implemented, ...). See the project's **README.md** file for instructions on how to compile the project.

2 Single View

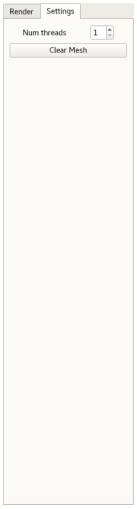
The operations accessible in this view are divided into two tabs. The most simple one is the Settings tab. This contains the option to increase the number of threads that the algorithms can use to speed-up the software (see section 4 to see what features benefit from increasing the number of threads). There is also the option to clean up the loaded meshes.

The operations in the Render tab are meant to only change the way a mesh is rendered. One can visaulise the Gaussian and Mean curvature of a mesh, in the Curvature box. Since the curvature values can range from a really small value to a really high one, most of the values being clustered to one end of the spectrum, a slider has been provided to let the user choose how many of these curvature values should be taken into account when coloring the vertices of the mesh using them.

Also, the renderisation of the mesh can be modified in the Model box:

- We can render it either in solid mode (1), or wireframe (2).
- Reflection lines (3) on the loaded mehs can be visualised. The amount of reflection lines to consider can be modified.
- We can render the mesh with a procedurally generated texture on it using the Harmonic Maps parametrisation (4). This parametrisation can only be done in meshes with a single boundary. This parametrisation starts distributing the vertices on this boundary on either a circle (5) or on a square (6). Computing the coordinates of the texture can be done either using Uniform or Cotangent weights (8). Finally, this parametrisation can be rendered to look like a remeshing (7).

(a) Render tab.



(b) Settings tab.

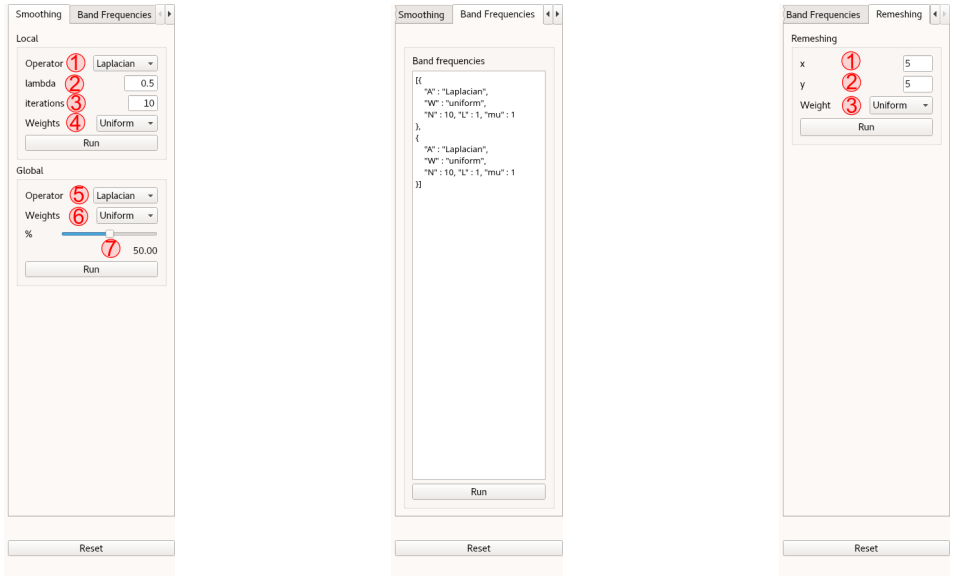
Figure 2: Tabs of single view.

3 Dual View

Dual view offers several operations. As mentioned in the introduction, this view is split into two viewers and these operations only affect the mesh in the right viewer. The tabs, to the right of the viewers and to the left of the operations available for both views, classify the operations into three categories:

- Smoothing (see section 3.1).
- Band frequencies (see section 3.2).
- Remesing (see section 3.3).

One common option to all these categories is to reset the mesh in the right viewer: it takes the unmodified mesh in the left viewer and sets it to the one to the right. In order to have these operations applied, the user has to click on the corresponding *Run* button.



(a) Smoothing. (b) Band frequencies. (c) Remeshing.

Figure 3: Tabs of dual view.

3.1 Smoothing

In this tab the user has access to two categories of smoothing algorithms: local and global (see figure 3a).

3.1.1 Local

Here the user can choose between three operators: Laplacian, BiLaplacian, Taubin-$\lambda\mu$ (1). The value of the λ coefficient can be set by the user (2), also the amount of

iterations (3) that the local algorithm should perform, and the type of weight: Uniform or Cotangent (4).

3.1.2 Global

The global smoothing algorithm is only implemented for the Laplacian opearator (5). The user, however, is able to choose the type of weight: Uniform or Cotangent (6). Additionally, the user can select the amount of fixed vertices that the algorithm needs to perform the smoothing. This amount is given using the slider that specifies the percentage of vertices to be fixed (7).

3.2 Band frequencies

This tab allows the user to apply several smoothing options to the mesh in the right viewer (see figure 3b). These operations have to be described by writing a list of JSON-formatted operations.

Formally, the operations applied to the mesh are:

$$M' = M + \sum_{i=1}^{n} \mu_i \cdot D_i$$

where the band frequencies D_i are defined as:

$$D_i = S_i(M) - S_{i+1}(M)$$

In these expressions, M and M' denote the input and output meshes, respectively. The coefficient μ_i is the weight of each band frequency and $S_i(M)$ is the result of applying a smoothing algorithm on the input mesh M.

3.2.1 Format of the text

The description of the operations is a JSON list of at least two band frequencies. Each band frequency is defined using a brace-enclosed list of options. In the following example we can see the options that need to be given:

```
{
    "A" : "Laplacian",                  1
    "W" : "Uniform",                    2
    "N" : 10,                           3
    "L" : 1,                            4
    "mu" : 1                            5
}
```

1. Specify smoothing operator. Must be a string.

   Allowed values: Laplacian, BiLaplacian, TaubinLM.

2. Specify type of weight. Must be a string.

   Allowed values: Uniform, Cotangent.

3. Specify a number of iterations. Must be an integer.

4. Specify value of λ, the coefficient for the smoothing algorithm. Must be a floating-point value.

5. Specify value of μ, the weight of this band frequency. Must be a floating-point value. Although it will be ignored, the last band frequency can also have this option.

Needles to say that the numbers to the right-most part of the example should not be included in the interface.

3.3 Remeshing

See figure 3c for a screenshot of this tab. Remeshing does two things. First, it computes the Harmonic Maps parametrisation of the input mesh **using the "Square" shape** (see bullet list of section 2). Then, applies the remeshing. This remeshing is guided by the amount of points the regular grid will have in the x- and y- axis (1 and 2 respectively), and the weight type (3).

4 Parallelised algorithms

Here are listed the options that can take advantage of the use of several threads:

- The computation of the curvature values. Their visualtion, however, does not use multithreading.
- All the local smoothing algorithms (that is, for all the operators available).
- Since band frequencies use local smoothing algorithms, this option can benefit from using multithreading too.
- Part of the remeshing algorithm has been parallelised. Since the parametrisation using Harmonic Maps is not parallel, only the construction of the new mesh is actually parallel.

5 Status of functionalities

All features provided in the interface are fully functional, except for the following:

- Reflection lines rendering mode. Reflection lines are perfectly rendered for those models in which the normals needed for this purpose (see next paragraph) can be computed as the normal of every triangle (if a triangle has vertices v_i, v_j and v_k then its normal is the normalisation of $(v_j - v_i) \times (v_k - v_i)$). A model of a completely flat surface, for example, is perfectly suitable for this rendering mode.

  Ideally, the normals to be used in order to render these lines properly should be computed differently (not done). The normals of all triangles should be computed as the average of **some** of their neighbouring faces. In this project, the normals are taken as the cross product, as explained before.

Figure 4: Visualisation of 14 reflection lines on a flat surface.

- Global smoothing. Only the Laplacian operator is implemented for this algorithm.
- Remeshing. Remeshing is not implemented for the shape "circle".