SOFTWARE REQUIREMENTS SPECIFICATION

for

Mesh Smoothing Tool

Release 1.0

Version 1.0 approved

Prepared by Jan Jose Hurtado Jauregui
Pontifícia Universidade Católica do Rio de Janeiro

Contents

Revision History

Date	Version	Description	Author
18/04/2017	1.0	Software Requirements Specification 1.0	Jan Hurtado

SOFTWARE REQUIREMENTS SPECIFICATION: Mesh Smoothing Tool

1 Introduction

1.1 Purpose

The purpose of this document is to collect and analyze all features which define the software behaviour of our mesh processing tool. We describe how we expect the software works regarding the final user, considering constraints and non-functional requirements.

1.2 Scope

3D shapes and surfaces obtained from real world data usually present undesired noise. This kind of problems are treated using smoothing techniques whose key aspects are denoising and fairing. Denoising is related to high-frequency noise removal, and fairing is related to obtain the smoothest version of the input regarding an energy function. This smoothing step is very important in a typical geometry processing pipeline.

Surfaces are commonly represented as triangular meshes due to its simplicity and easy processing. The smoothing task over meshes is called mesh smoothing, and is related to the modification of the geometric properties of the mesh (e.g. vertex positions). There are some important considerations in mesh smoothing algorithms such as detail-preserving, low normal variation, volume preservation, etc. Depending on the application these features determine robustness.

For example, in the case of medical data, several techniques are used to obtain a volume which represents the anatomy of the patient (e.g. X-ray radiography, medical ultrasound, Magnetic Resonance Imaging (MRI), etc.). Using a subset of this volume, a surface model can be reconstructed to represent the target region. Each step of the reconstruction process can introduce noise to the final surface.

To remove this kind of noise, several mesh processing frameworks introduce a mesh smoothing tool for this purpose.

1.3 Definitions and abbreviations

term	description
user	any user which will use our mesh smoothing tool
mesh	3D triangular mesh
noisy mesh	input mesh of mesh processing tool
smooth mesh	output mesh of mesh processing tool
surface	mathematical definition of a mesh
vertex	vertex of a triangular mesh
face	triangle of a triangular mesh

1.4 Overview

This document is structured as follows. In Section 2 we introduce the persepective, functions, user characteristics, constraints, assumptions and dependencies of the software. In Section 3 we start explaining the user interface and then specifying the functional and non functional requirements.

2 Overall Description

2.1 Product Perspective

The software consists of a single interface for execution of mesh smoothing algorithms and visualization of the input and resultant meshes. The user will manipulate these algorithms without knowledge of the parameters and corresponding behaviour. We will develop a friendly interface with an intuitive interaction.

The algorithms should support the features presented in a previous section and should be useful for the user. So, we have to consider execution time and automatic computation of implicit values regarding the input mesh.

The visualization should show rendered images in real time including interaction for navigation in the 3D space. Also, we will show different types of mesh visualization such as: facet, wireframe and point cloud.

2.2 Software functions

The software should allow the user to:

- Load input mesh
- Save output mesh (smooth mesh)
- Visualize and navigate in the 3D space of the mesh
- Smooth the input mesh using a global scheme
- Smooth the input mesh using a focalized scheme

2.3 User characteristics

The software is intended for any user who needs the processing of a noisy mesh to obtain a cleaner one. So, the user does not need any knowledge about mesh processing or mathematical definitions. In that sense, the user should have to interact with our software as simple as possible.

2.4 Constraints, assumptions and dependencies

The software will only process triangular meshes in any of the supported mesh formats of OpenMesh library. We will not support the processing of huge meshes (more than 10 million vertices), in this case the mesh should be previously partitioned and each partition should be processed individually. We assume that the target computer should have these minimum requirements: 2GB Ram and a modern graphics processing unit (later than 2010).

The software will be developed entirely in C++ programming language with the possibility to compile it in Windows and Linux operating systems. The software will use the following external libraries:

- OpenGL 4.0
- The OpenGL Extension Wrangler Library (GLEW)
- OpenGL Mathematics (GLM)
- OpenMesh 6.3
- Qt 5.7

To define the priority in the specification of requirements we will use 3 different levels, which are the following:

level	description
3	The requirement must be satisfied as soon as possible because it is important for the main software functionality (mesh smoothing)
2	The requirement is important and should be implemented to improve user's interaction
1	The requirement is a complement and not to much necessary for the user described above

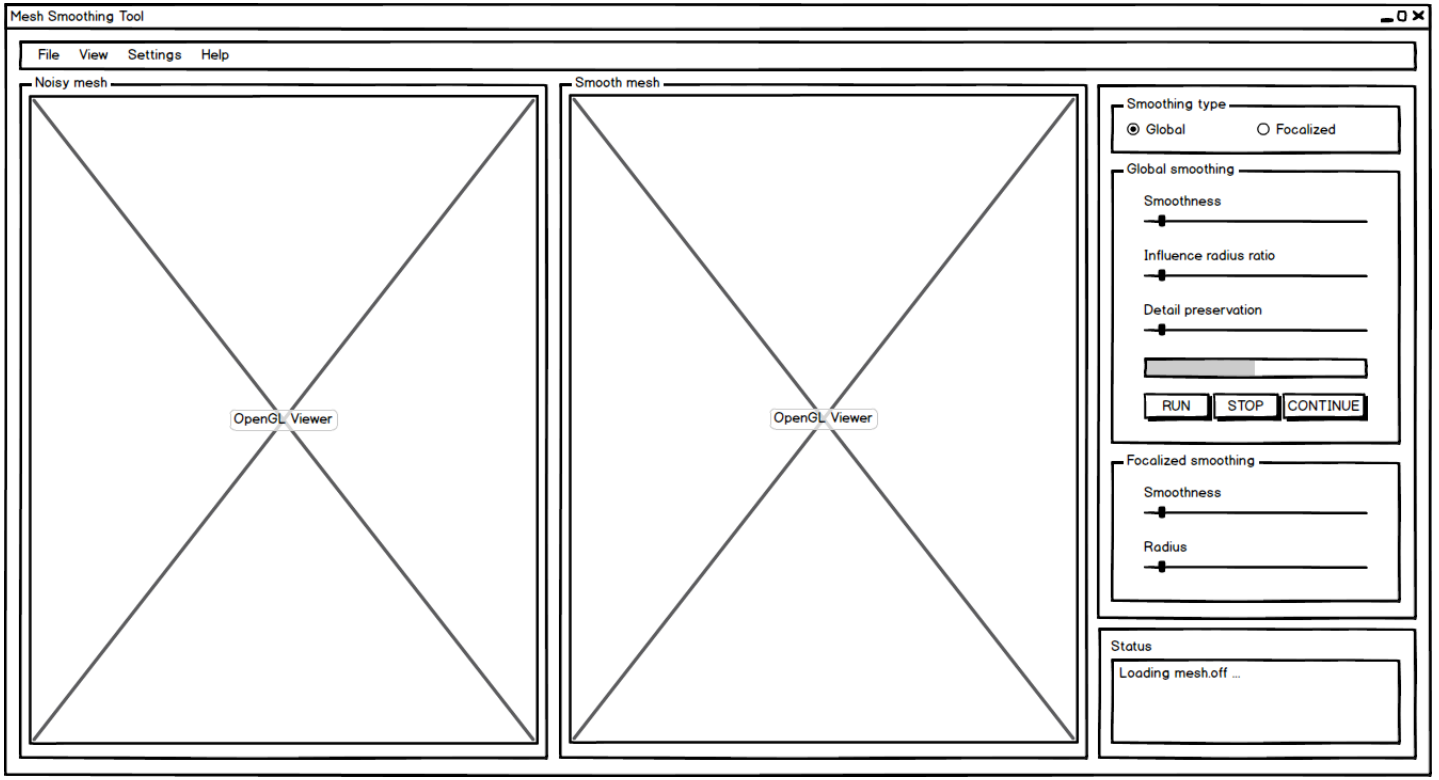

Figure 1: Mockup

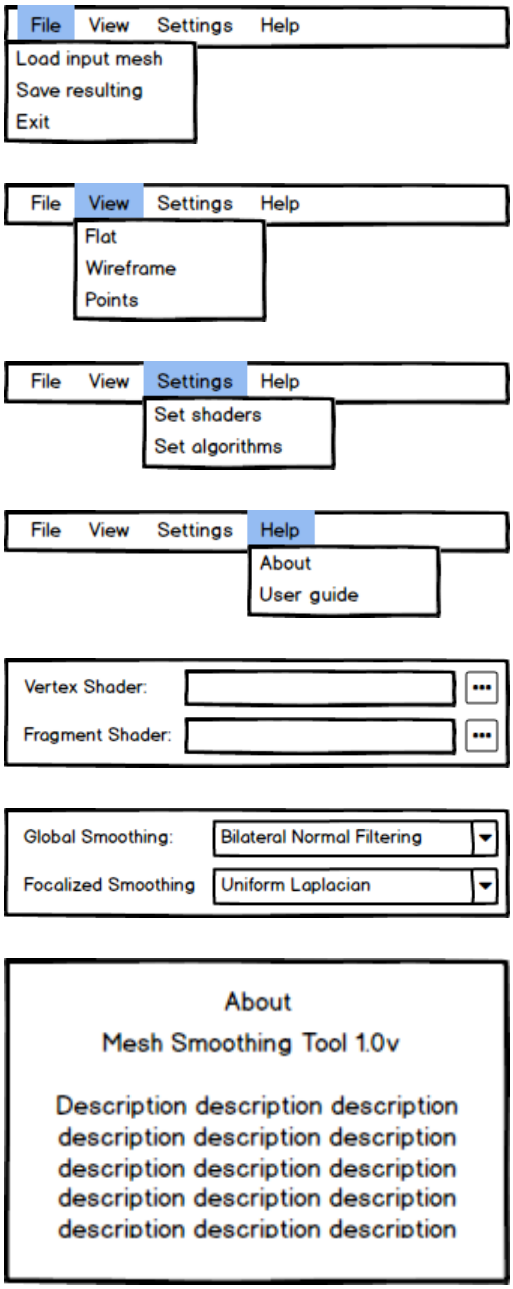

Figure 2: Mockup details. From top to bottom: File menu, View menu, Settings menu, Help menu, Set Shaders window, Set Algorithms window and About window.

3 Specific Requirements

3.1 User interface

As we introduced before, the software consists of a single interface composed of a main window and auxiliary windows to set up the software behaviour.

The main window will have a menu bar, a status block, two visualization blocks for 3D navigation and interaction, and a control block to manipulate the smoothing features (including global and focalized). In Figure 1 we show how will look like the structure of this main window.

The menu bar consists of four items: File, View, Settings and Help. These items are described as follows:

- File: This item will allow the user to load an input mesh file, save the output mesh file and close the program.
- View: This item will allow the user to choose a mode of rendering of the two visualizers. The render modes are flat, wireframe and points.
- Settings: This item will allow the user to choose shaders for visualization and change default algorithms for mesh smoothing.
- Help: This item will allow the user to show the "About" window and open a link for the user guide.

See Figure 2 for more details.

In the case of visualization blocks, we have two. The first one will be used to visualize the input mesh, allowing 3D navigation only (translation, rotation and zoom in/out). The second one, in addition to 3D navigation, will allow the user to interact with the mesh for focalized smoothing.

The control block will allow the user to choose between global or focalized smoothing, and manipulate the intuitive parameters of smoothing process. In the case of global smoothing the parameters are: smoothness, influence radius and detail preservation. Also, it will be possible to stop the algorithm when the mesh is sufficiently smooth or perform more smoothing steps (continue button). For the focalized smoothing the user will define the smoothness and brush size.

The final block will show the current status description of the program.

3.2 Functional requirements

3.2.1 Load input mesh

Description: The user should be able to select an input file in one of the supported mesh file formats (.obj, .off, .ply and .stl), and then set the corresponding data as the input mesh.

Rational: In order to read mesh files.

Priority level: 3.

3.2.2 Save output mesh (smooth mesh)

Description: The user should be able to export the resultant mesh (smooth mesh) in one of the supported mesh file formats (.obj, .off, .ply and .stl).

Rational: In order to write resultant meshes.

Priority level: 3.

3.2.3 Close program

Description: The user should be able to close the program.

Rational: In order to finish the program.

Priority level: 3.

3.2.4 Set flat rendering for visualizers

Description: The user should be able to visualize both meshes using a flat rendering mode, drawing mesh triangles.

Rational: In order to visualize the shape and compare it.

Priority level: 2.

3.2.5 Set wireframe rendering for visualizers

Description: The user should be able to visualize both meshes using a wireframe rendering mode, drawing mesh edges.

Rational: In order to visualize the topology and compare it.

Priority level: 2.

3.2.6 Set point rendering for visualizers

Description: The user should be able to visualize both meshes using a points rendering mode, drawing mesh vertices.

Rational: In order to visualize vertex distribution and compare it.

Priority level: 2.

3.2.7 Set vertex shader for visualizers

Description: The user should be able to select its own vertex shader code and use it for visualization. We will include some default vertex shaders.

Rational: In order to customize the visualization.

Priority level: 1.

3.2.8 Set fragment shader for visualizers

Description: The user should be able to select its own fragment shader code and use it for visualization. We will include some default fragment shaders.

Rational: In order to customize the visualization.

Priority level: 1.

3.2.9 Set global smoothing algorithm

Description: The user should be able to select the global smoothing algorithm

Rational: In order to obtain different results depending on the algorithm.

Scenarios: The user can choose between the following algorithms, "Bilateral Normal Filtering for Mesh Denoising" and "Guided Mesh Normal Filtering".

Priority level: 2.

3.2.10 Set focalized smoothing algorithm

Description: The user should be able to select the focalized smoothing algorithm.

Rational: In order to obtain different behaviours of the focalized smoothing.

Scenarios: The user can choose between the following algorithms, "Uniform Laplacian" and "HC Laplacian".

Priority level: 2.

3.2.11 Open "About" window

Description: The user should be able to see a brief description of the software.

Rational: In order to show software details.

Priority level: 1.

3.2.12 Open user guide link

Description: The user should be able to open the user guide.

Rational: In order to show the user guide.

Priority level: 1.

3.2.13 Camera movement in both visualizers

Description: The user should be able to navigate in the 3D scene (forward, back, left and right).

Rational: In order to visualize the meshes from different point of views.

Priority level: 3.

3.2.14 Object rotation in both visualizers

Description: The user should be able to rotate the meshes.

Rational: In order to visualize the meshes from different point of views.

Priority level: 3.

3.2.15 Area selection and corresponding focalized smoothing in smooth mesh visualizer

Description: The user should be able to select a region in the mesh and smooth it (depending on the mouse position and the smoothing radius).

Rational: In order to perform a focalized smoothing.

Priority level: 3.

3.2.16 Selection of mesh smoothing type (Global/Focalized)

Description: The user should be able to select the type of smoothing.

Rational: In order to perform the smoothing type he want.

Scenarios: The user can choose between global smoothing and focalized smoothing.

Priority level: 3.

3.2.17 Increment/Decrement smoothness for global smoothing

Description: For global smoothing, the user should be able to manipulate the smoothness parameter (increment or decrement).

Rational: In order to obtain different results. More or less smoothness.

Priority level: 3.

3.2.18 Increment/Decrement influence radius ratio for global smoothing

Description: For global smoothing, the user should be able to manipulate the radius ratio parameter (increment or decrement).

Rational: In order to obtain different results. More or less smoothness.

Priority level: 3.

3.2.19 Increment/Decrement detail preservation for global smoothing

Description: For global smoothing, the user should be able to manipulate the detail preservation parameter (increment or decrement).

Rational: In order to obtain different results. More or less detail preservation.

Priority level: 3.

3.2.20 Start global smoothing

Description: The user should be able to start the global smoothing algorithm.

Rational: In order to execute global smoothing.

Priority level: 3.

3.2.21 Stop global smoothing

Description: The user should be able to stop the global smoothing algorithm.

Rational: In order to obtain a partial result.

Priority level: 3.

3.2.22 Continue global smoothing

Description: The user should be able to continue performing global smoothing over the current output mesh.

Rational: In order to smooth more the current result.

Priority level: 3.

3.2.23 Increment/Decrement smoothness for focalized smoothing

Description: For focalized smoothing, the user should be able to manipulate the smoothness parameter (increment and decrement).

Rational: In order to obtain different results. More or less smoothness.

Priority level: 3.

3.2.24 Increment/Decrement radius for focalized smoothing

Description: For focalized smoothing, the user should be able to manipulate the radius parameter (increment and decrement).

Rational: In order to increment or decrement the area to be smoothed.

Priority level: 3.

3.3 Non-functional requirements

3.3.1 Multithreading execution of smoothing algorithms

Description: The smoothing algorithms should be executed using multithreading.

Rational: In order to maintain visualization and navigation while executing the algorithm. Also, to show partial results.

Priority level: 3.

3.3.2 Management of software state

Description: Manipulate states of the software.

Rational: In order to allow and avoid user interactions according to the current state.

Priority level: 3.

3.3.3 Visualization using GPU

Description: The rendering pipeline should be based on GPU usage.

Rational: In order to give the user a better visualization regarding frames per second and large data processing.

Priority level: 3.

3.3.4 Mesh management

Description: The software should manage meshes for mesh processing (smoothing), visualization and interaction.

Rational: In order to obtain better performance (execution time and "quality").

Priority level: 3.