GigaMesh

Manual

Susanne Krömker
August 31, 2011

Chapters 1 and 2 revised by

Hubert Mara
August 7, 2013

Interdisciplinary Center for Scientific Computing
Visualization and Numerical Geometry Group
Heidelberg University, Germany

Universität Heidelberg
Zukunft. Seit 1386.

IWR
Interdisciplinary Center
for Scientific Computing

Contents

Preface

GigaMesh is a software framework for processing large meshes. It is mainly running on linux operating systems, and a windows version is preparation. *GigaMesh* is able to read established mesh formats like `Stanford PLY` or `Wavefront OBJ`. It has a graphical user interface and features the possibility of a fast and easy display of large 3D models. The mesh can be polished to result in a perfect 2D manifold, true to scale screenshots and image sequences for video output can be achieved easily, and profile lines can be exported as vector graphics. Command line tools are provided for intense computing tasks like the application of 3D filters for pattern recognition.

GigaMesh is developed by Hubert Mara during his Ph.D. thesis in collaboration with the *Heidelberg Academy of Sciences and Humanities*. Its first intension was the visualization of cuneiform script but soon developed into a tool for a wide range of objects in Cultural Heritage. Contributions concerning the fill holes algorithm (using the library `libpsalm`) and the unwrapping algorithm are made by Bastian Rieck. Other contributions are due to Andreas Beyer, Robert Kreuzer, and Philipp Stekl. The first version of this manual was written by Anja Schäfer. Everyone who is working with the program and its manual is welcome to make suggestions for improvements.

The software name *GigaMesh* derives from the Epic of Gilgamesh (see the recent translation [Mau05]), which is among the earliest known works of the literature, written in cuneiform script. Shorten the first syllable in *GigaMesh* by the letter "l" associates the SI prefix G = giga = 10^9 (a billion), and thereby describes the capability of dealing with large meshes. Intentionally it is developed to improve the reading of cuneiform tablets, so the name is already program. The *GigaMesh* logo is derived from a cuneiform known as *kaskal* = *path* or *way*, combined with a red highlighted ball in the style of the golden ball of our group logo.

We thank Willi Jäger, Stefan M. Maul and Stefan Jakob for their support and granting access to the collections of the *Assyriologie, Seminar für Sprachen und Kulturen des Vorderen Orients* of the Heidelberg University. The 3D scanners used for the acquisition of the presented data were provided by the *Heidelberg Academy of Sciences and Humanities* (HAW) and the *Heidelberg Graduate School of Mathematical and Computational Methods for the Sciences* (HGS MathComp). This work is part of the *IWR Pioneering Projects* (IPP) and partially funded by the HGS MathComp – DFG Graduate School 220.

Susanne Krömker

Heidelberg, August 2011

Chapter 1

Introduction

GigaMesh is a software framework for displaying, processing and visualizing large meshes representing 2D surfaces typically acquired with 3D scanning techniques.

Its core piece mainly base on the *Multi Scale Integral Invariant* (MSII) filtering technique [Mar12]. Integral invariants [PWHY09, HFG$^+$06] using volume and surface descriptors can be computed for feature detection in large triangular meshes. Results of the computations are stored in the so-called feature vectors. Visualizations of these feature vectors use standard metrics known from image processing like auto-correlation or distance measures. These metrics can be visualized by false-coloring the 3D models using various colorramps.

GigaMesh can be operated with or without a graphical user interface (GUI). It is based on OpenGL and Trolltech Qt, and features the possibility of a fast and easy display of large 3D models. The mesh can be polished removing measurement errors like solo vertices, singularities and holes, resulting in an oriented filled and perfect mesh of the quality of a 2D manifold. The resulting mesh can be saved and afterwards imported by other software tools. Identified features can be extracted as points or polygonal lines. *GigaMesh* offers the possibilities to create orthographic or perspective normalized viewings, true to scale screenshots, to prepare image sequences for videos and interactive Adobe Flash and Apple QuickTime output, and to export polygonal lines as vector graphics in `*.svg` [Eis02] format. There exist several editors for this vector format such as the *Inkscape*[1], which is highly recommended as it supports `*.svg` natively and it is freely available for all major operating systems.

[1] `http://inkscape.org` – last visited on 7.August 2013.

1.1 Operating system

GigaMesh is currently running mainly on Linux. The version for windows is expected to be released, when Qt 5.1 becomes available. An unstable Macintosh binary exists missing some of the functions.

1.2 Mesh formats

GigaMesh supports the by now most established 3D formats for mesh data used in metrology. They can be opened within the graphical user interface. A file browser and appropriate filter masks then show the following data due to their filename extensions:

- Stanford PLY – ASCII and binary with big and little endianness[2]
- Wavefront OBJ – ASCII
- Pointclouds written in ASCII having file extension `*.txt` or `*.xyz`

Reading and writing of these formats is due to filename extension. Stanford PLY file format is highly recommended. Meta information stored when saving PLY-files with *GigaMesh* are displayed when using these files later on with *GigaMesh* . Other programs skip these headers.

There are also two types of proprietary formats (ASCII): It is possible to (re-)import texture maps as `*.tex` and feature vectors as `*.mat` files which have been generated in previous runs of the program.

Pitfall: due to new Operating system libraries reading and writing **ASCII** files is prone to errors due to different characters for floating point values. Many regional settings use a dot (”`.`”) for the **decimal point**, while Operating system having e.g. a German localization use a comma (”`,`”). This affects also import/export from/to software packages

[2] Byte order.

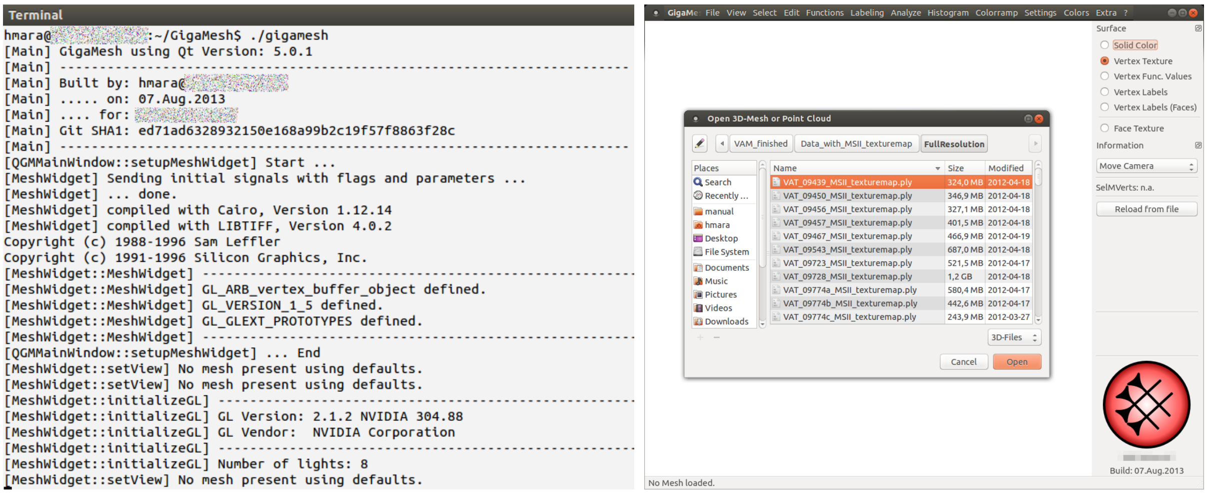
Chapter 2

Displaying a Mesh

This chapter is on getting started with *GigaMesh*. You may also go to chapter 6 for brief examples in case you are already familiar with the program. Chapter 5 gives a survey on the pulldown menu. Keyboard shortcuts are summarized in the appendix A.

2.1 First steps

Open a console or a terminal and change into your *GigaMesh* folder (with the `cd GigaMesh` command). Then type `./gigamesh` and hit ↵. *GigaMesh* can be started using a file browser or from a menu entry, but it is recommended to be exectued within a terminal due to the output of additional information.



```
Terminal
hmara@[redacted]:~/GigaMesh$ ./gigamesh
[Main] GigaMesh using Qt Version: 5.0.1
[Main] -----------------------------------------------------------------------
[Main] Built by: hmara@[redacted]
[Main] ..... on: 07.Aug.2013
[Main] .... for: [redacted]
[Main] Git SHA1: ed71ad6328932150e168a99b2c19f57f8863f28c
[Main] -----------------------------------------------------------------------
[QGMMainWindow::setupMeshWidget] Start ...
[MeshWidget] Sending initial signals with flags and parameters ...
[MeshWidget] ... done.
[MeshWidget] compiled with Cairo, Version 1.12.14
[MeshWidget] compiled with LIBTIFF, Version 4.0.2
Copyright (c) 1988-1996 Sam Leffler
Copyright (c) 1991-1996 Silicon Graphics, Inc.
[MeshWidget::MeshWidget] ----------------------------------------------------
[MeshWidget::MeshWidget] GL_ARB_vertex_buffer_object defined.
[MeshWidget::MeshWidget] GL_VERSION_1_5 defined.
[MeshWidget::MeshWidget] GL_GLEXT_PROTOTYPES defined.
[MeshWidget::MeshWidget] ----------------------------------------------------
[QGMMainWindow::setupMeshWidget] ... End
[MeshWidget::setView] No mesh present using defaults.
[MeshWidget::setView] No mesh present using defaults.
[MeshWidget::initializeGL] --------------------------------------------------
[MeshWidget::initializeGL] GL Version: 2.1.2 NVIDIA 304.88
[MeshWidget::initializeGL] GL Vendor:  NVIDIA Corporation
[MeshWidget::initializeGL] --------------------------------------------------
[MeshWidget::initializeGL] Number of lights: 8
[MeshWidget::setView] No mesh present using defaults.
```

Name	Size	Modified
VAT_09439_MSII_texturemap.ply	324,0 MB	2012-04-18
VAT_09450_MSII_texturemap.ply	346,9 MB	2012-04-18
VAT_09456_MSII_texturemap.ply	327,1 MB	2012-04-18
VAT_09457_MSII_texturemap.ply	401,5 MB	2012-04-18
VAT_09467_MSII_texturemap.ply	466,9 MB	2012-04-19
VAT_09543_MSII_texturemap.ply	687,0 MB	2012-04-18
VAT_09723_MSII_texturemap.ply	521,5 MB	2012-04-17
VAT_09728_MSII_texturemap.ply	1,2 GB	2012-04-18
VAT_09774a_MSII_texturemap.ply	580,4 MB	2012-04-17
VAT_09774b_MSII_texturemap.ply	442,6 MB	2012-04-17
VAT_09774c_MSII_texturemap.ply	243,9 MB	2012-03-27

SelMVerts: n.a.

Build: 07.Aug.2013

No Mesh loaded.

Figure 2.1: Terminal in which you started *GigaMesh* (left) and the graphical interface with the open file dialog (right).

As long as *GigaMesh* is running it writes to the standard output device, i.e. the terminal from which you started the program. So do not close this terminal, since all information concerning the current status of the program are displayed there. The program starts with the open file dialog window (see figure 2.1), where you can select the file you want to display and modify. A file browser and appropriate filter masks only show folders and 3D data with filename extensions PLY, OBJ, TXT, or XYZ.

After having opened a data file, the browser window disappears and the 3D object is displayed in the graphics window. Due to usually large data files of several hundreds of megabytes even reading the data may take some time (up to minutes). However, transformations work in realtime.

The graphics window has pulldown menu in the top bar and a sidebar with some buttons in top and the *GigaMesh* logo at the bottom. Sketches to guide the user and a progress bar also appear in the sidebar. With the buttons in top the vertices change their colors from those given in the data file to no color at all or to the values given by some function value computed during the scope of the program and visualized with a certain color map. For any task that might take some time, there is a progress bar above the logo and the *GigaMesh* logo pops up in green (like a traffic light) for some seconds to signalize that the program is ready for the next commands.

Hint: *GigaMesh* can be started from the console with one given filename (no wildcards):

```
./gigamesh [filename]
```

2.2 Viewing an Object

Generally the view can be changed using the move, the keyboard and by entering rotation angles. Figure 2.2 show the keys used to navigate the OpenGL camera about an object. The upper and middle row of keys change one of the three axis of the camera orientation by 1°. With the lower keys the view can be changed by 90°, which allow a quick inspection of all side, top and bottom views. The above Figure can also be (de-)activated within *GigaMesh* using

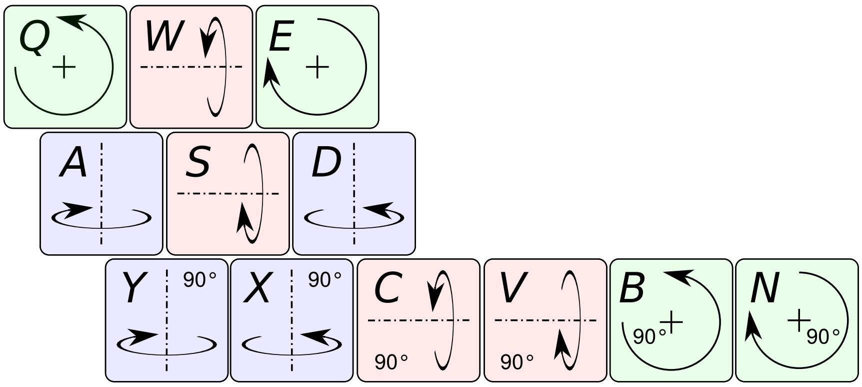


Figure 2.2: Keyboard layout for navigating the OpenGL camera.

? → `Keyboard Layout`.

Additional shortcuts are shown with

? → `Keyboard Shortcuts`.

In case a rotation about a specific angle e.g. less than 1° is required, the menus

View

→ `Rotate Yaw (U/D)`,

→ `Rotate Pitch (L/R)` and

→ `Rotate Roll`

will provide a dialog to input an arbitrary value. The terms *yaw*, *pitch* and *roll* refer to three critical flight dynamics parameters, which are the angles of rotation about a vehicle's center of mass i.e. the OpenGL camera's center.

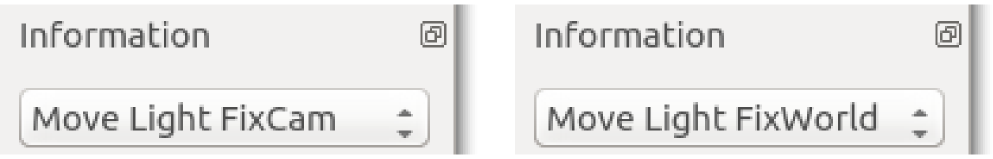
2.3 Lighting

For lighting the object there are two lights available, one is enabled by default and can be switched on and off with 1. This light position and orientation is fixed to the view position and orientation – in OpenGL terms: it is fixed to the camera. The second light can be toggled 2 and is fixed to the object and therefore does not move, when the view direction is changed. The ambient light is enabled by default and can be toggled with 9. The amount of ambient light can be set by choosing the pulldown-menu

`Settings` → `Light Ambient - Set Amount`.

You can turn on and off all lights together with the cutout switch for lighting which is assigned to 0.

Both virtual light sources project ideal parallel light. Therefore only there orientation can be changed. To change the orientation selected the according item from the Combo-Box on the right hand side:



(a) (b)

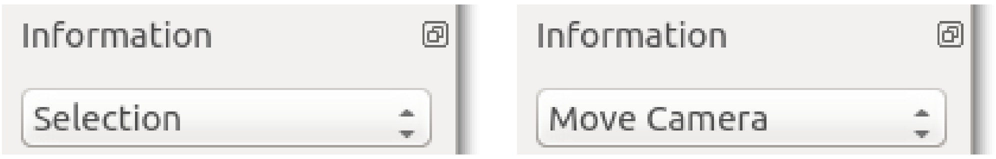
Figure 2.3: Combo-Box choices for using the mouse to move the two lights with (a) fixed orientation to the camera and with (b) fixed orientation to the object.

Hint: The mode for manipulating the light orientation can be activated by short pressing[1] the Alt key for the first light. To change the orientation of the second light, the AltGr key has to be pressed while the mouse is moved. To return to the default mode – moving the view with the mouse – the ␣ bar has to be pressed. This will also change the Combo-Box to `Move Camera` as shown left in Figure 2.4a.

[1] Holding Alt and moving the mouse will move the window in *Ubuntu Unity*.

2.4 Selection and Visualizing distances

The previous section showed the manipulation of the view and the light sources using mouse movement. For selection there is an additional mode, which can either be selected from the Combo-Box or activated by holding the Ctrl key. In both cases the Combo-Box will show the term `Selection` as shown in Figure 2.4a. Using the □ will return the operating mode of the mouse to manipulation of the view shown in Figure 2.4b.



(a) (b)

Figure 2.4: Combo-Box choices for using the mouse for (a) interactive selection and (b) to move the view direction (default).

Distance to a single vertex (`SelVert`): The default selection allows to chose a single vertex. It can also be activated by

`Select` → `Primitive - Vertex SelVert/SelPrim`.

Therefore holding Ctrl and single clicking with the left mouse button will select an appropriate vertex of the object. Then select `Functions` → `Distance to SelPrim`.

`Functions` → `Distance to SelPrim (COG)`.

All vertices get colored according to the default color map which is called `Hot` and goes from black at the minimum over red to orange and finally white for the maximum. An overview of all possible color maps can be found in section 5.9.

Distance to a plane: You may also measure the distances to a selected plane. Therefore change the selection mode to

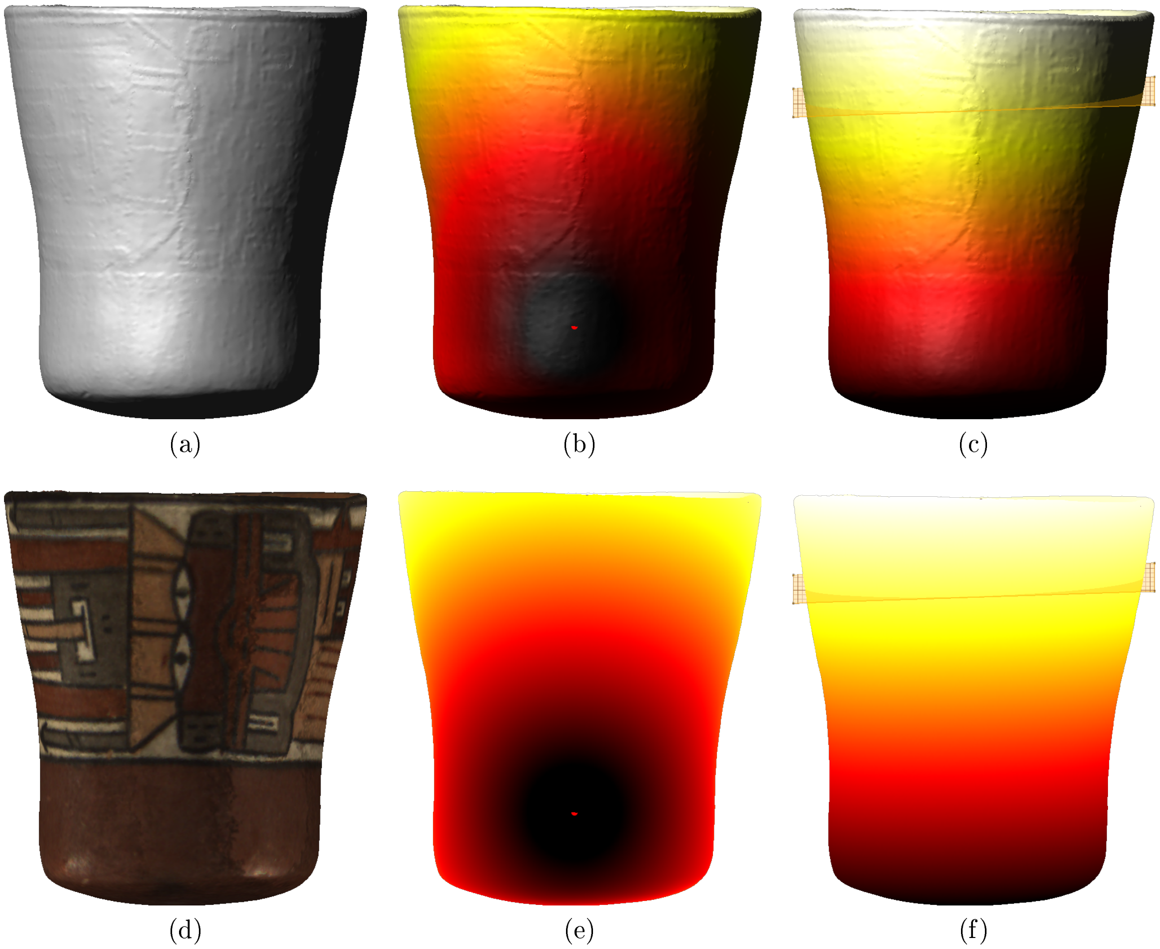
`Select` → `Plane - 3 Points`

and note that a user guiding sketch of a triangle appears in the sidebar, when Ctrl is pressed. Select the three face points of the desired plane by single clicking on three vertices of the object. As long as this selection is active, you can change the position of the plane by clicking on other points of the object. You iteratively change the positions of the three defining face points. When you are satisfied with the selected plane chose

`Functions` → `Distance to Plane`

to displays the distance in shades of the current color map.

Figure 2.5 shows examples for distance to a single vertex and a plane, which are shown with and without lighting.



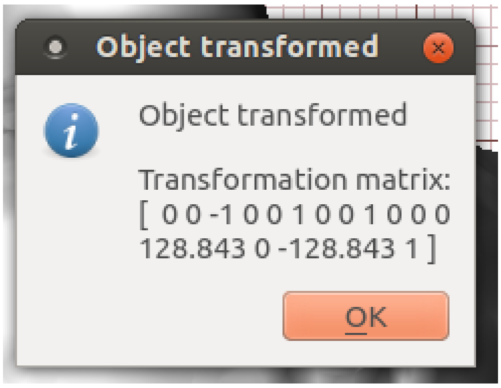
(a) (b) (c)

(d) (e) (f)

Figure 2.5: Example for distance visualization on a replica of a (a,d) Nasca vessel. Distances to (b,e) a single vertex (`SelVert`) shown as red dot and (c,f) to a plane shown in light orange color. Light colors mean high values and dark colors mean low values.

2.5 Storing and Restoring the Default View

The key F6 saves the current view of a 3D model to the default view. This view and the default light positions can be retrieved by F12. Pressing Shift ⇑ together with F12 also resets the zoom factor.



Object transformed

Transformation matrix:
[0 0 -1 0 0 1 0 0 1 0 0 0 128.843 0 -128.843 1]

Figure 2.6: Information window after setting the current view as default view.

Note: As changing the default view is actually a change of position of the 3D model in world coordinates, the 3D model has to be saved to make the new default view permanent.

Tip: The transformation matrix shown in Figure 2.6 can be selected, copied and then saved as text-file. This allows to (re-)apply the same transformation later or to other 3D-models e.g. when a new version of 3D-model was created from raw data. The 16 elements of the 4 × 4 matrix can be entered using the pulldown-menu

`Edit` → `Apply 4x4 Matrix - All Vertices`.

2.6 Meta-data

The model ID and its material get saved in the comment section of the header of the output file (e.g. `*.ply`). Note that this info gets lost when saving the file with external programs like e.g. `Meshlab`. *GigaMesh* asks to enter the meta-data, when a 3D-model is saved and no data is present. It can also be edited selecting

`Extra` → `Cuneiform Figure Latex Info`

from the pulldown-menu. As the name indicates this functionality was introduced, when large amounts of cuneiform tablets were processed in the early days of *GigaMesh* . Therefore an additional LaTeX template exists to easily prepare a corpus of 3D-models visualized with virtual illumination and MSII-based texture maps.

Chapter 3

Compute Feature Vectors (MSII)

In order to use the full functionality of *GigaMesh* a batch program has to be run for generating feature vectors. These vectors contain additional information per vertex concerning surface and volume of a set of spheres intersecting the mesh. See [MKJB10] for more background information on the *Multi Scale Integral Invariant* (MSII) filtering technique. This operation is rather time consuming (it takes hours or even days of computing time) and therefore better runs without graphical user interface. Although generating feature vectors is quite robust against solo vertices, singularities, non-manifolds and holes, you should first clean up your mesh data to get a proper result. So switch to the advanced task of polishing your mesh in section 4.1 and return to this section when you have got a cleaned mesh. If you do not want to manipulate your mesh you may continue directly.

Open a terminal and type and change to the mesh-folder by typing `cd GigaMesh/mesh` (note that `GigaMesh` stands for the *GigaMesh* installation folder). Then start the program

```
nohup ./meshgeneratorfeaturevectors25d_threads -f <path-to-filename> [-r 2] &
```

and use `nohup` at the beginning of the command and `&` at the end to ensure that the job runs in the background. This is because this step can take several hours and you do not want to block the terminal.

For `path-to-filename` the (relative or absolute) path to the folder of the data file and the name of the data file to be visualized has to be given. The parameter given behind the option `-r` displays the radius of the maximal sphere for the MSII filter. The default value has been set to the unit 1.0. It can be changed into some other appropriate number depending on the size of the features to be detected. An educated guess is the maximum size of the feature width. Smaller values will only detect noise and bigger values lose the fine tuning when operating with the feature vectors in *GigaMesh*.

Note again that depending on the filesize of the object this step can take several hours. Every time when having processed 5000 vertices the program approximates the estimated time of finishing. This is written to the standard output device which is the terminal. Multithreading is active. The estimation gets better and better when each thread has processed several 5000 vertices.

Finally there should have appeared six additional files adjacent to the data file. In the listing below the gray font indicates that some of these files might be empty or even do not exist at all depending on the version of the program.

```
<filename>_r2.00_n4_v256.info.txt
<filename>_r2.00_n4_v256.normal.mat
<filename>_r2.00_n4_v256.ply
<filename>_r2.00_n4_v256.surface.mat
<filename>_r2.00_n4_v256.volume.mat
<filename>_r2.00_n4_v256.vs.mat
```

The naming convention added the radius `_r2.0`, the number of scales in power of two e.g. `_r2.0` means $2^4 = 16$ equidistant radii in the interval $[r, 0]$, and a rasterization of the volume with e.g. `_v256` means 256 voxels along the diameter of the biggest sphere. One of the files is a short text file containing information on the process itself (with the suffix `info.txt`), four of them store the feature vectors separated in only volume or only surface or both, or store the normals. They have the suffix `*.mat`, where the wildcard * stands for `surface, volume, vs` or `normal`. These files can be used to re-import feature vectors if they have been discarded when saving other information important for a certain visualization. The sixth data file is a `*.ply`-file which also contains the feature vectors and can be processed by the graphical user interface of *GigaMesh* for visualization purposes. In case that some of these files are missing something went wrong during the computation (see B for further information).

Chapter 4

Advanced Tasks

This chapter is a selection of advanced tasks that can be done with *GigaMesh*. Most of those tasks require several steps. In chapter 6 the single steps will be accomplished in brief examples, so skip the detailed description if you are already familiar with the program. The survey of the pulldown menu entries is given in chapter 5 in great detail.

Start the GUI-version of *GigaMesh* and open a 3D data file.

4.1 Polishing the mesh

Usually measured data is very nasty due to errors in the meshing of pointclouds. You have outliers, holes and twisted areas. In order to get rid of all these dirt in your mesh you start by clicking on `Edit` → `Remove Unclean/Small`. You will be asked for the amount of small areas to be removed, the default is 10% of the total surface. Now the object gets cleaned prior to the fill holes algorithm. `Remove Unclean/Small` is a compilation of several steps needed to optimize the mesh combined in one task.

- remove *sticky faces*
- remove *non-manifold faces*
- remove *small areas*
- remove *solo vertices*
- remove *singular vertices*

Sticky faces are faces using the same vertices and therefore stick together. Non-manifold faces occur when at least three faces share a common edge (see figure 4.1). A small area is characterized by being not connected to other parts of the mesh and containing less than about 10% of the total surface. A double cone is a face, that combines two parts of a surface at a single vertex. Solo vertices are just single points without any edges connecting them to the mesh. Singular vertices happen when the borders of at least two holes meet in a vertex.

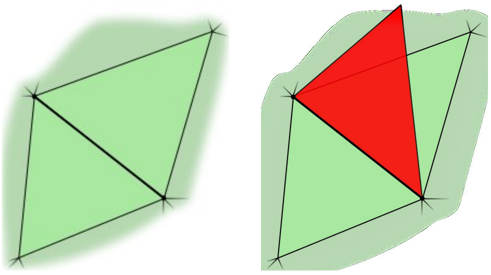


Figure 4.1: a) manifold edge, b) non-manifold edge

The combined steps are to select e.g. the solo vertices by `Select` → `Vertices` → `Solo` and then `Edit` → `Remove SelVerts`. In the terminal the progress bar on each of the steps is reported.

Next go to `Analyze` → `Mesh Borders to Polylines` where all borders are selected as polygonal lines (short: polylines). The union of these lines form the boundary of the mesh. As result the boundary namely all these polylines are colored in light blue. In some cases these are only borders of tiny holes that can be immediately filled with `Edit` → `Fill Holes (PSALM)`. There is no outer border of the mesh since the mesh is homeomorphic to a sphere e.g. the head of a sculpture, or homeomorphic to a torus like e.g. teapot. But in other cases the mesh has an outer border like just the face of a head or the front of a relief. Then the outer border does not belong to the category of holes to be filled. So you have to `Select` → `Polyline Longest` which hopefully is this outer border and remove it from the list of polygonal lines (so-called polylines) to be filled.

Then the next step is to click on `Edit` → `Fill Holes (PSALM)`. Even a large number of holes get filled rather quickly and an info window informs about the number of holes which could be filled and those which are kept due to less information given or other errors reported in the terminal. After finishing the filled areas are colored in red to indicate where you do not have measured data and reliable color information. Finally you should get rid of all polylines.

Note that eventually this process has to be repeated several times since any time when removing vertices and when holes getting filled, new nasty areas might appear. So here again is the cycle to be repeated until the mesh is perfect:

1. `Edit` → `Remove Unclean/Small`
2. `Analyze` → `Mesh Borders to Polylines`
3. `Select` → `Polyline Longest`
4. `Edit` → `Remove SelMPolylines`
5. `Edit` → `Fill Holes (PSALM)`
6. `Edit` → `Remove All Polylines`

The mesh really is perfect, when no hole was filled and none was kept and the `Edit` → `Remove Unclean/Small` returns 0 Faces and 0 Vertices. Due to our naming convention you may save this file as an *GigaMesh* **GM** **O**riented **C**leaned **F**illed **P**erfect mesh and name it `<filename>_GMOCFP.ply`. If you miss out any of these steps, just leave out the according letter. You can just accomplish the specific task later.

Extend borders If holes cannot be filled as they are too large or too small they will be kept and the number of theses holes is shown in the finally popping up window. To make a hole easier to be filled or worked with it is sometimes necessary to extend its border and thereby making the hole larger. To do this go to `Select` → `Vertices - Border` and then click on `Edit` → `Remove - SelMVerts`.

4.2 Visualize feature vectors

Open the `<filename>_r2.00_n4_v256.ply` file. Select the deepest cuneiform with a double click with the left mouse button. Correlated vertices are then visualized by clicking on `Functions` → `Feature Correlation to SelVert`. This may take some time. The default color map is `Hot` but can be changed to anything you like best by selecting another map from the `Colorramp` e.g. `HSV` or `Grayscale`. For other possibilities see section 5.9.

4.3 Create a cross section with a cutting plane

GigaMesh features the possibility to create cutting planes. In order to get the proper outline of, e.g. a vessel, this possibility sets a plane inside the vessel and "cuts" the object thereby showing its outline.

The first step is to `Select` → `Plane - 3 Points)` where the user will define a plane by three points: mark three points on the plane with (). The plane can be shown with `View` → `Mesh Plane`. In theory this plane is movable. Then calculate the distance to the plane. By selecting `Analyze` → `Iso Value - Set` the value for iso value should be set to 0. Then with `File` → `Export Screenshots (SVG + PNG)` → `Screenshot (SVG + PNG)` of the current view. This projects the polygonal lines to screenshots and calculates the outline.

4.4 Unwrapping objects

Objects being being approximately rotational symmetric can be unwrapped by specifying a sphere or a cone as auxiliary surface and projecting the object onto this surface prior to developing it to a plane. Choose `Select` → `Cone` and a user guiding sketch appears in the sidebar. It resembles a conical frustum and shows a red symmetry axis with a point a that first has to be chosen by the user. To do so the object has to be positioned in such a way that you look down this symmetry axis of your object. Double click with the left mouse button to the position of this axis. The user guide sketch now indicates the second point p_{r1} on a gray line perpendicular to the red axis and at the mantle of the cone. Select this point on the object by double click at a vertex of the surface. Now the user guide sketch indicates that the third point p_{r2} has to be selected at the bottom of the cone. Selecting these three points results in a semitransparent red cone displayed together with the object. The red axis shows the symmetry axis and a green line indicates the split meridian. You may edit this prime meridian with `Edit` → `Set Prime Meridian for Rollouts` and then move it around the cone with the left mouse button. You may also `Edit` → `Cone - Set Cone Data` and change the radii manually.

As long as `Select` → `Cone` is still active, a double click will iterate the procedure of selecting those three points necessary for a cone description.

Finally `Edit` → `Cone - Unroll Mesh` will execute the unwrapping. ATTENTION! This procedure is NOT reversible. To restart the unwrapping with another cone or sphere you have to reload the mesh from the data file.

4.5 Fat cross or flat hexahedron

In many cases you need a classic six-side view of the (visualized) model that is an orthogonal projection of the object onto the six sides of the bounding box. This can be done easily with this software. The first step to do this is to orient the 3D object either with the mouse or with the keyboard. Then hit F6 (this view will be saved as standard view) and save the file. Use the model ID you want. Next set the resolution with `Settings` → `Ortho Scale (Set DPI)`. After that go to `File` → `Export Screenshots` → `Screenshot Views`. Since the main subject *GigaMesh* has been developed for is visualization of cuneiform tablets, the popup menu entry for exporting everything into a LaTeX-file is called `Extra` → `Cuneiform Figure Latex Info`. In step 7 of section 6.1 is a screenshot of the final view.

4.6 Generating image stacks for videos or flashes

GigaMesh proposes the opportunity to create images that can be used for a flash video. The user can choose to create an image stack for a video moving the camera on a circular orbit around the object. The central axis of the orbit can be the horizontal axis of the view, the vertical axis of the view, a normal of a selected primitive, or the normal of a predefined plane. This can be selected by e.g `File` → `Export Image Stack` → `360° Circular Orbit Horizontal Axis`.

A flash video is similar to a video and additionally provides two interactions, firstly a rotation in two directions and secondly different states of the visualization can be shown, e.g. the captured surface colors and the results of an MSII filtering. There exist several possibilities to create such image stacks. The image stack for a flash moving the lightsource on a spherical orbit around the object can be chosen by `File` → `Export Image Stack` → `Spherical Stack Lightsource Orbiting`. Alternatively the camera is orbiting around the object. Therefore choose `File` → `Export Image Stack` → `Spherical Stack Camera Orbiting`. By default, `Longitudinal` is selected, but you may change to

`Altitudinal`. In order to separate the different visualizations, a state number can be given as a suffix for the filenames of the image stacks. Then these stacks can be processed into a flash object e.g. with the *Garden Gnome Software* **Object2VR**.

4.7 Usage of Object2VR with an image stack

ATTENTION: This section describes how to generate a flash file using the software package **Object2VR**.

1. Click on `select input` and choose `image sequence` as type
2. Set the path for the pictures in `file location`
3. Set the states right in `rows` and `columns` (e.g. spherical images requires 36 columns and 18 rows, states depends on how many have been done with GigaMesh. Attention: GigaMesh starts counting at 0).
4. Now the search string has to be given:

   ```
   'filename'+fill(column,5,0)+'_'+fill(row,5,0)+'_'+fill(state,2,0)+'.tiff'
   ```

   This command is for TIFF-files; for PNG-files just replace `.tiff` by `.png`.
5. Click on `test pattern`: here it is checked whether all files are found or not. If `all files found` then confirm with `ok`.
6. In `Viewing Parameters` select `modify` and set the initial position (this is optional). Hint: If `vertical images` has been selected in GigaMesh, the `horizontal` box has to be checked.
7. In output choose `Flash`, reduce the speed under `Autoplay` (appropriate: 0.05) and choose a skin (this can be the default or a self-made one).
8. If also needed for HTML, choose the tab `HTML` and check the box `enable html file`
9. Finally click on `ok` and answer the upcoming question with `yes`.

Chapter 5

Detailed Dialog

The following sections give an overview of the possibilities you have to edit, analyze and visualize the surface of your 3D object. Since the program is still work in progress, the names may slightly differ, new entries might occur or gain in functionality and other might disappear. Therefore each section starts with a brief introduction what can be found within this pull-down menu item. Menu items in gray (in this manual, black in the program menu) are disabled as long as necessary data is missing. Separators given in the program do not appear in this brief description. A dot • in front of an item shows the selected option of multiple possible options, a check mark ✓(hooklet or cross) indicates the activation of a toggling option.

The following abbreviations are used to shorten lengthy texts:

`Polyline`	*Polygonal line*
`FuncVal`	*Value of the active function in a primitive, e.g. the distance of a vertex to a plane or the area of a face*
`SelPrim`	*Selected primitive*
`SelVert`	*Selected primitive, which has to be a vertex*
`SelFace`	*Selected primitive, which has to be a face*
`SelMVerts`	*Selection of multiple vertices*
`SelMFaces`	*Selection of multiple faces*
`SelPolyline`	*Selected polygonal line*

Remark: The `FuncVal` is saved as 'quality' entry/field per vertex within a PLY file. It is also read from a PLY file, if present. The 'quality' entry/field is also used for multiple purposes by software packages like *Meshlab* or *OptoCAT*.

5.1 The file dialog

This is the usual file dialog with additional possibilities to unload and (re-)import feature vectors once computed with the program `meshgeneratorfeaturevectors25d_threads` (see chapter 3). The user can as well create image stacks for a video moving the camera on a circular orbit around the object. Or he can create spherical stacks with rotating the camera or the lightsources. Or even more simple the display can be exported as screenshot.

```
File

→ Open        Ctrl + O
→ Reload      Ctrl + R
→ Save As     Ctrl + S
→ Import Texture Maps
→ Import Feature Vectors
→ Export Polylines (ASCII)
→ Export Polylines Projected (ASCII)
→ Export FunctVals
→ Export Face Normal Angles
→ Export Image Stack
    → 360° Circular Orbit Horizontal Axis
    → 360° Circular Orbit Vertical Axis
    → 360° Circular Orbit Normal of SelPrim
    → 360° Circular Orbit Normal of Plane
    → Spherical Stack Lightsource Orbiting
    → Spherical Stack Camera Orbiting
    → ● Camera on Longitudinal Orbit
    → ○ Camera on Altitudinal Orbit
    → Spherical Stack State No.
→ Export Screenshots
    → Screenshot (TIFF/PNG)
    → Screenshot (SVG + PNG)
    → Screenshot Views
    → Screenshot Ruler
    → ✓ Screenshot Cropping
    → Video Frame Size (Set)
    → Video Frame Size (Fix)
→ Unload Feature Vectors
→ Unload 3D
→ Quit       Ctrl + Q
```

5.2 View on the displayed object

The object can be displayed in various manner, as mesh, with or without faces, monochrome or textured, with additional bounding box and camera position. The view on the object can be transformed as well with keys (see appendix A) and follow the convention of roll, pitch and yaw, known from (airplane) navigation.[1]

View

- → 3D Data
- → Show Camera Position
- → Set Camera Position
- → Bounding Box F1
- → Graticules F2
- → Vertices
  - → Texture
  - → Texture Inverted
  - → Monochrome
  - → Solo
  - → Border
  - → Singular
  - → Non-Manifold
  - → Local Minima
  - → Local Maxima
- → ✓ Faces
- → Edges
- → Mesh Plane
- → Mesh Plane Flip
- → ✓ Mesh Plane as Clipping Plane F3
- → Clipping thru SelPrim Shift + F3
- → Normals - Vertex
- → Normals - Face
- → Normals - Polyline
- → Normals - Polyline Mean
- → Polylines
- → Polylines w. Curvature
- → Polyline - Set Curvature Scale

[1] http://en.wikipedia.org/wiki/File:Rollpitchyawplain.png

→ `Spheres`

→ `Boxes`

→ `Default View and Light` F12

→ `Default View, Light and Zoom` Shift + F12

→ `Rotate - Yaw (L/R)`

→ `Rotate - Pitch (U/D)`

→ `Rotate - Roll`

→ `Rotate - Ortho to Plane` F8

→ `SelPrim to View Reference` F4

→ `Current View to Default` F6

5.3 Selection of primitives

With this menu the user is able to select primitives and auxiliary surfaces like for instance a plane to be set with three points and which then is given in Hessian Normla Form (HNF). Once these primitives or surfaces have been selected, they can be edited or analyzed within the next menu items (see in the sections 5.4 and 5.7).

`Select`

→ `Vertices - SelMVerts - Show Indices`

→ `Vertices - SelMVerts - Enter Indices`

→ ○ `Vertices - SelMVerts - GUI`

→ ● `Primitive - Vertex - SelVert/SelPrim`

→ ○ `Primitive - Face - SelVert/SelFace`

→ ○ `Cone`

→ ○ `Sphere`

→ ○ `Plane - 3 Points`

→ `Plane - Set VPos`

→ `Plane - Set HNF`

→ `Plane - Show VPos`

→ `Plane - Show HNF`

→ `Vertices - None`

→ `Vertices - FuncVal <`

→ `Vertices - FuncVal >`

→ `Vertices - FuncVal Local Minimum`

→ `Vertices - FuncVal Local Maximum`

→ `Vertices - Solo`

→ `Vertices - Non-manifold`

→ `Vertices - Singular`

→ `Vertices - Labelarea < Abs`

→ `Vertices - Labelarea < %`

→ `Vertices - Border`

→ `Vertices - Face Minimum Angle <`

→ `Vertices - Face Maximum Angle >`

→ `Vertices - Label No.`

→ `Faces - None`

→ `Faces - Sticky`

→ `Faces - Non-manifold`

→ `Faces - Zero Area`

→ `Faces - In Sphere`

→ `Faces - Random`

→ `Polylines - Not Labeled`

→ `Polylines - Run Length >`

→ `Polylines - Run Length <`

→ `Polyline - Longest`

5.4 Editing the mesh

Any kind of editing the mesh or the function value, or the cone or sphere for unrolling, can be done within this menu. An often used item is the removing of unclean parts of the mesh as measurements often have outliers, small disjointed areas, or the like. These parts of the mesh may cause poor images or time consuming computations at unnecessary, complicated long borders of the mesh. The item `Remove - Unclean/Small` therefore combines repeated steps of the previous section 5.3 with its selecting possibilities, and this editing section. Nevertheless, the skilled user can select only those vertices or faces disturbing the mesh and remove them step by step (see section 4.1). The algorithm for a simple and therefore fast filling of small holes can also be found here. It invokes the external libraby `libpsalm` written by Bastian Rieck.

`Edit`

→ `Remove - Unclean/Small`

→ `Remove - SelMVerts`

→ `Remove - SelMFaces`

→ `Remove - SelMPolylines`

→ `Remove - All Polylines`

→ `Fill Holes (PSALM)`

→ `Feature Elements cut-off SelVert`

→ `Feature Elements cut-off SelFace`

→ `FuncVal - Set`

→ `FuncVal - Cut-off`

→ `FuncVal - Normalize`

→ `FuncVal - Absolute`

→ `FuncVal - Add`

→ `Split by Plane`

→ `Split by Iso Value`

→ `Cone - Set Cone Data`

→ `Cone - Center Mesh around`

→ `Cone - Unroll Mesh`

→ `Sphere - Center Mesh around`

→ `Sphere - Unroll Mesh`

→ `Set Prime Meridian for Rollouts`

→ `Apply 4x4 Matrix - All Vertices`

→ `Apply 4x4 Matrix - SelMVerts`

→ `Melting Sphere - All Vertices`

5.5 Visualize certain properties along the surface

A property of the surface like a distance (Euklidian or Manhattan distance) to a selected vertex or plane, or the center of gravity (COG) of a primitive can be computed as a function of each of the vertices or faces. This menu invokes the computation and then visualizes the values by coloring the surface accordingly. The user first has to select and sometimes edit his selection, and second go to one of the following options. Some of these options require feature vectors to be present, others just work on geometric data.

`Functions`

→ `Feature Vector Length (Euklid)`

→ `Feature Vector Length (Man)`

→ `Feature Distance to SelVert (Euklid)`

→ `Feature Distance to SelVert (Man)`

→ `Feature Correlation to SelVert`

→ `Feature Auto Correlation`

→ `Feature Auto + Correlation to SelVert`

→ `Feature Element`

→ `Distance to Plane`

→ `Distance to Line (Pos, Dir)`

→ `Distance to SelPrim (COG)`

→ `Distance to Cone`

→ `Vertex - Indices`

→ `Vertex - 1-Ring Area`

→ `Vertex - 1-Ring Sum Angles`

→ `Vertex - Octree`

→ `Vertex - Sphere Angle Maximum to Faces`

→ `Vertex - Sphere MeanNorm Angle Max`

→ `Faces - Sort Index`

→ `Faces - Sphere Marching Front Indices`

→ `Color - Hue to FuncVal`

5.6 Labeling

This pull-down menu provides methods to compute local neighborhoods and labeled areas of interest.

`Labeling`

→ `Label - All Vertices / SelMVerts`

→ `Label - SelMVerts to Seeds`

→ `Label - SelMVerts`

→ `Label - Vertices w. Equal FuncVal`

→ `Label + Remove Small Areas`

→ `Label Borders to Polyline(s)`

→ `Label Polylines - Advance to Threshold`

→ `Label FuncVal Minima`

→ `Label FuncVal Maxima`

5.7 Analyze the surface

This pull-down menu provides methods to analyze the surface, e.g. the compution of geodesic distances to selected points of interest.

`Analyze`

→ `MSII Feature Vectors - SelVert`

→ `Mesh Borders to Polylines`

→ `SelMVerts to Polylines`

→ `Iso Lines to Polylines`

→ `Iso Value - Set`

→ `Polyline Extrema`

→ `Polylines w. Absolute Curvature`

→ `Set Length for Smoothing`

→ `Geodesic Distance to SelVert`

→ `Geodesic Distance to SelMVerts`

→ `Sphere Intersect`

→ `Fit Ellipse`

→ `Compute Volume (dx, dy, dz)`

→ `Compute Volume (Plane)`

→ `Estimate Bounding Box`

→ `Feature Vector Wavelet decomp.`

5.8 2D plots of statistics

In case of the presence of `libqwt`, some statistics can be computed and then are shown as 2D plots in separate windows. This helps to understand the properties and qualities of the meshes.

`Histogram`

→ `FuncVal`

→ `Label (Vertex) Areas`

→ `Faces Minimum Angle`

→ `Faces Maximum Angle`

→ `Edge Length`

→ `Polyline Length (Absolute)`

→ `Vertex Feature Element`

→ `Face Feature Element`

→ `Vertex Feature Auto-Correlation`

→ `Logarithmic`

→ `Feature Vector - SelVert`

5.9 Coloring the surface

The Colorramp menu assigns different color maps to the vertices according to their function values. From minimum to maximum value `Hot` means a map from black over red and orange to white. This is the default color map. `HSV` means a usual color circle from red (R), yellow (Y), green (G), cyan (C), blue (B), magenta (M) and back to red (`HSV (full)`) or not (`HSV (part)`). Grayscale are shades of gray from black to white, `Traffic light` is from R over Y to G, `Hypsometric tint` is inspired by geography[2] and goes from dark green (low areas) over yellow to brown and finally white (top of the icy mountains). `RdGy` stands for red over white to black, the `Spectral` color map goes from R over Y to B, and the `RdYeGn` is from R over Y to G. Of course, all these maps can be inverted which means, that the maximum value gets assigned to the beginning of the ramp. See for example [MK03] and [BHH03].

The `Waves` color map is a bit different. It is a repeated sinusoidal black and white change, and a `Wavelength` has to be set to get proper results (default value is 1.0 in terms of the selected function). It reveals its evidence when assigned to a function value like `Distance to vertex (selected)`. Then it results in a target pattern centered in the selected vertex.

`Colorramp`

→ `Hot`

→ `HSV (full)`

→ `HSV (part)`

→ `Grayscale`

→ `Traffic Light`

→ `Hypsometric Tint`

→ `RdGy`

→ `Spectral`

→ `RdYlGn`

→ `Waves`

→ `Auto Min/Max`

[2] http://colorbrewer2.org/

→ `Quantile`

→ `Quantile Min`

→ `Quantile Max`

→ `Fixed Min/Max`

→ `Fixed Min/Max Settings`

→ `Invert`

→ `Logarithmic`

→ `Equalized`

→ `Show Labels Only`

→ `Overlay Label Color`

→ `Wavelength`

5.10 Texturing the surface

Within this menu the storing of the current visualization (that is its current function value at each vertex expressed as an RGB value per vertex) can be done. These RGB values then get saved when saving the mesh to a file again, This enables to use this visualization within other software packages (e.g. `blender`).

`Texture Map`

→ `Store Current per Vertex`

→ `Stretch Contrast per Vertex`

→ `Traffic Light`

→ `Black White Black`

→ `White Black White`

→ `Black - White`

→ `White - Black`

5.11 Some settings

General settings concerning not only the view but more general concepts like switching between perspective or orthographic projection or the general settings of light in combination with material can be changed in this menu.

`Settings`

→ `Perspective - Set Field of View`

→ ○ `Perspective View`

→ ● `Orthographic View`

→ ✓`Ortho Grid (Show Metric)` F7

→ `Ortho Scale (Set DPI)` Shift + F7

→ ✓`Lighting` 0

→ `Light 1 (Headlight)` 1

→ `Light 2 (Object, Spot)` 2

→ `Light 3 (Object, Parallel)` 3

→ `Light 4 (Camera, Spot)` 4

→ `Light 5 (Camera, Parallel)` 5

→ `Ambient Amount` 9

→ `Ambient - Set Amount`

→ `Light Position - Set w. Mouse` P

→ `Material - Shininess`

→ `Material - Specular`

→ ✓`Smooth Shading`

→ ✓`Faces - Backface Culling` F5

→ `Fog`

→ ✓`Screen Info`

→ `Datum - Sphere Transp.`

5.12 Coloring the appearance

Some elements of the appearance of *GigaMesh* have default colors. The user can change the default coloring e.g. the color of the background or the color of the backfaces of a mesh (which has some kind of brown as default value).

`Colors`

→ `Background Color`

→ `Solid Color`

→ `Backface Color`

→ `Vertex Color`

→ `Edge Color`

→ `Polyline Color`

→ `No Label Color`

→ `Labels Mono Color`

→ `Labels Mono`

→ `Transparency SelMVerts`

5.13 Some extras

In order to transform the six side view into a PDF-file the model ID and the model material have to be set. Then a window will open with LaTeX-code ready to be copied from the clipboard. Finally an *About* box pops up showing contacts and references.

`Extra`

→ `Cuneiform Figure LaTeX Info`

→ `SelMFaces Normals Matlab`

`?`

→ `About`

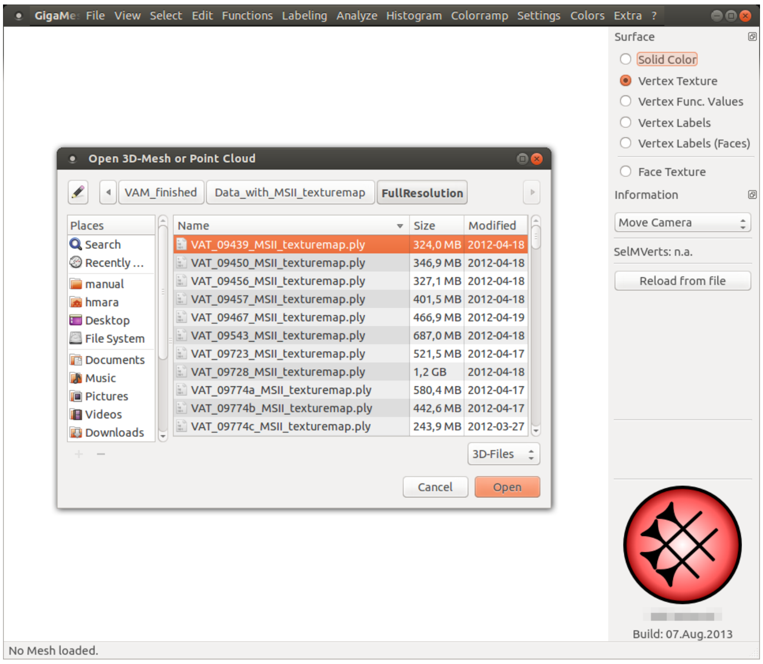
Chapter 6

Examples

6.1 Cuneiform tablets

Here is first and detailed example what can be done with *GigaMesh* to get a feeling on how to use the program. However, some of the pictures of the illustration may be outdated due to the ongoing development of the program. As the main focus lies on the visualization of cuneiform script this section will show how to visualize the 3D model of a cuneiform tablet and how to do a fat-cross view.

Step 1 Start *GigaMesh* and browse to open the 3D model to be visualized.



Name	Size	Modified
VAT_09439_MSII_texturemap.ply	324,0 MB	2012-04-18
VAT_09450_MSII_texturemap.ply	346,9 MB	2012-04-18
VAT_09456_MSII_texturemap.ply	327,1 MB	2012-04-18
VAT_09457_MSII_texturemap.ply	401,5 MB	2012-04-18
VAT_09467_MSII_texturemap.ply	466,9 MB	2012-04-19
VAT_09543_MSII_texturemap.ply	687,0 MB	2012-04-18
VAT_09723_MSII_texturemap.ply	521,5 MB	2012-04-17
VAT_09728_MSII_texturemap.ply	1,2 GB	2012-04-18
VAT_09774a_MSII_texturemap.ply	580,4 MB	2012-04-17
VAT_09774b_MSII_texturemap.ply	442,6 MB	2012-04-17
VAT_09774c_MSII_texturemap.ply	243,9 MB	2012-03-27

○ Solid Color
● Vertex Texture
○ Vertex Func. Values
○ Vertex Labels
○ Vertex Labels (Faces)
○ Face Texture

SelMVerts: n.a.

Build: 07.Aug.2013

No Mesh loaded.

Figure 6.1: Start up of *GigaMesh*

Step 2 Orient the model with the help of the mouse or the keyboard (Q / E for roll (counterclockwise (CCW) or clockwise (CW) rotation about z-axis) by 1°; W / S for yaw (CCW/CW rotation about x-axis) by 1°; A / D for pitch (CCW/CW rotation about y-axis) by 1°). This step is needed later on for a proper six side view.

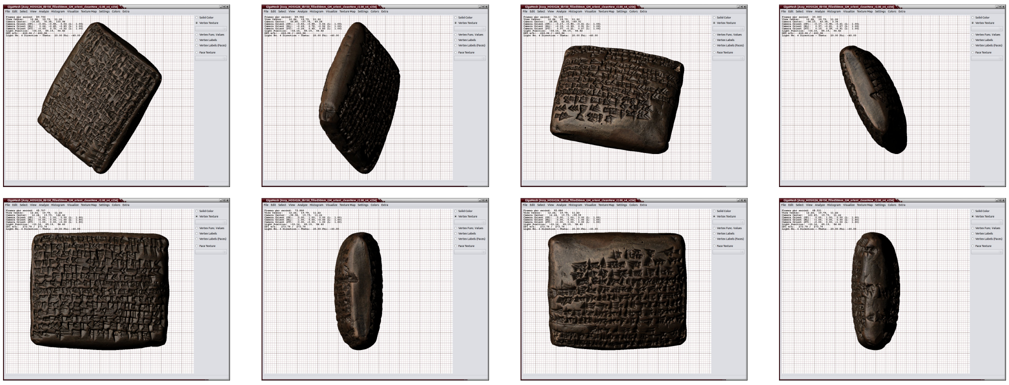


Figure 6.2: not oriented (upper row) and oriented (lower row) version

With the help of Y / X (pitch by 90 °), C / V (yaw by 90 °) and B / N (roll by 90 °) it is possible to check if all sides of the object are oriented correctly. When finished and happy with the result, press F6 to save the current orientation matrix. If you do additional transformation you are now able to recover this saved orientation by pressing F12 or Shift + F12, respectively. Now save the whole file with `File` → `Save As`. As a naming convention extend the filename by `_GMO` for *GigaMesh* Oriented.

Step 3 Next the feature vectors have to be calculated. These vectors contain additional information per vertex concerning surface and volume of a set of spheres intersecting the mesh. See [MKJB10] for more background information on the *Multi Scale Integral Invariant* (MSII) filtering technique. This operation is rather time consuming (it takes hours or even days of computing time) and therefore better runs without graphical user interface. Open a terminal and change into the `GigaMesh/mesh/` folder (with the help of the `cd` command). There should be an executable called:

```
featurevectorthreads_25d
```

If the folder contains no executables then they still have to be compiled (see B).

Step 4 Type

```
./meshgeneratorfeaturevectors25d_threads -f <path-to-filename> [-r 2]
```

For `path-to-filename` the (relative or absolute) path to the folder of the data file and the name of the data file to be visualized has to be given. The parameter given behind the option `-r` displays the radius of the maximal sphere for the MSII filter. The default value has been set to the unit 1.0. It can be changed into some other appropriate number depending on the size of the features to be detected. An educated guess is the maximum size of the feature width. Smaller values will only detect noise and bigger values lose the fine tuning when operating with the feature vectors in *GigaMesh*. Note again that depending on the filesize of the object this step can take several hours. Finally there should have appeared a data file named

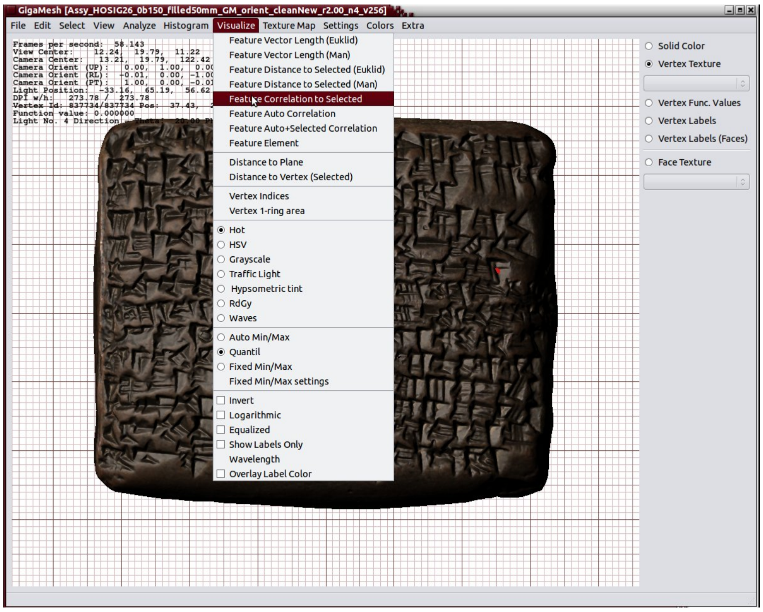
`filename_r2.00_n4_v256.ply`

due to naming conventions which contains the feature vectors and can be processed by *GigaMesh* in the next steps. For more information see chapter 4.

Step 5 To proceed with the visualization switch to *GigaMesh*'s GUI and load

`filename_r2.00_n4_v256.ply`

This is the object already containing the feature vector. Now select the deepest wedge on the surface of the object by double clicking it with the left mouse button. Next go to `Functions` → `Feature Correlation to SelVert`.



```
Frames per second:  58.143
View Center:     12.24,  19.79,  11.22
Camera Center:   13.21,  19.79,  122.42
Camera Orient (UP):   0.00,  1.00,  0.00
Camera Orient (RL):  -0.01,  0.00, -1.00
Camera Orient (PT):   1.00,  0.00, -0.01
Light Position:  -33.16,  65.19,  56.62
DPI w/h:   273.78 /  273.78
Vertex Id: 837734/837734 Pos:  37.43,
Function value: 0.000000
Light No. 4 Direction - Theta:  20.00
```

Figure 6.3: Selected Vertex (red dot) and dropdown menu for feature detection

Please note that this step may take several minutes before it is completed. The default colormap is the so-called `Hot`, displaying the a ramp from dark red to orange to light yellow.

Step 6 Colors can be modified to `Colorramp` → `Grayscale`. Next click `Colorramp` → `Invert` which inverts all the colors or shades of gray. Switch off the light with 0. If necessary double click somewhere empty for deselection and refresh. What you see is the coloring of the surface according to the curvature and without any lighting active. Of course you may switch on the light again with $\boxed{0}$ and also manipulate the lights with $\boxed{1}$ to $\boxed{5}$ and pressing the ctrl-key while moving the mouse. This enhances the realistic impression.

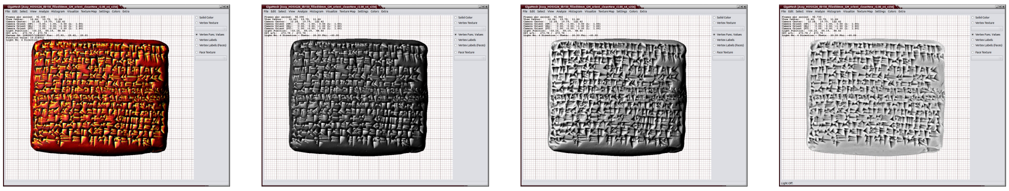
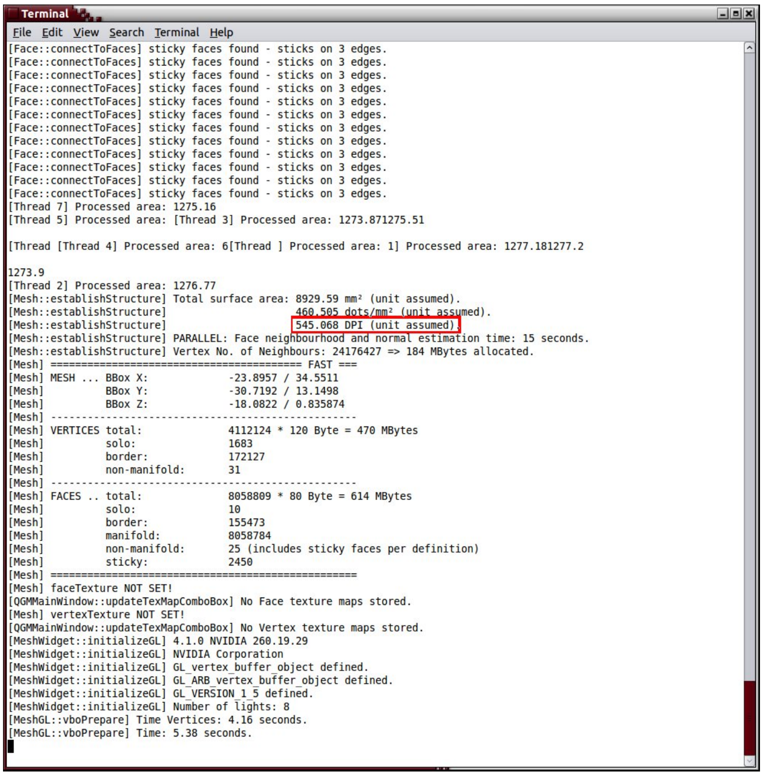


Figure 6.4: From left to right: Object with feature vector in hot correlation display, then in grayscale, in inverted grayscale and in inverted grayscale without light

Step 7 To do a six side view of the above visualized object go to `Settings` → `Ortho Scale (Set DPI)` and set the resolution. A good help for the right resolution is to check the terminal (see marked part in picture 3.5 below) and set the value a little bit above the one in the terminal. E.g. in this example the terminal says 545.068 DPI, so it is good to set it to about 600 DPI.



```
[Face::connectToFaces] sticky faces found - sticks on 3 edges.
[Face::connectToFaces] sticky faces found - sticks on 3 edges.
[Face::connectToFaces] sticky faces found - sticks on 3 edges.
[Face::connectToFaces] sticky faces found - sticks on 3 edges.
[Face::connectToFaces] sticky faces found - sticks on 3 edges.
[Face::connectToFaces] sticky faces found - sticks on 3 edges.
[Face::connectToFaces] sticky faces found - sticks on 3 edges.
[Face::connectToFaces] sticky faces found - sticks on 3 edges.
[Face::connectToFaces] sticky faces found - sticks on 3 edges.
[Face::connectToFaces] sticky faces found - sticks on 3 edges.
[Face::connectToFaces] sticky faces found - sticks on 3 edges.
[Face::connectToFaces] sticky faces found - sticks on 3 edges.
[Thread 7] Processed area: 1275.16
[Thread 5] Processed area: [Thread 3] Processed area: 1273.871275.51

[Thread [Thread 4] Processed area: 6[Thread ] Processed area: 1] Processed area: 1277.181277.2

1273.9
[Thread 2] Processed area: 1276.77
[Mesh::establishStructure] Total surface area: 8929.59 mm² (unit assumed).
[Mesh::establishStructure]                     460.505 dots/mm² (unit assumed).
[Mesh::establishStructure]                     545.068 DPI (unit assumed).
[Mesh::establishStructure] PARALLEL: Face neighbourhood and normal estimation time: 15 seconds.
[Mesh::establishStructure] Vertex No. of Neighbours: 24176427 => 184 MBytes allocated.
[Mesh] ======================================== FAST ===
[Mesh] MESH ... BBox X:            -23.8957 / 34.5511
[Mesh]          BBox Y:            -30.7192 / 13.1498
[Mesh]          BBox Z:            -18.0822 / 0.835874
[Mesh] ------------------------------------------------
[Mesh] VERTICES total:             4112124 * 120 Byte = 470 MBytes
[Mesh]          solo:              1683
[Mesh]          border:            172127
[Mesh]          non-manifold:      31
[Mesh] ------------------------------------------------
[Mesh] FACES .. total:             8058809 * 80 Byte = 614 MBytes
[Mesh]          solo:              10
[Mesh]          border:            155473
[Mesh]          manifold:          8058784
[Mesh]          non-manifold:      25 (includes sticky faces per definition)
[Mesh]          sticky:            2450
[Mesh] ================================================
[Mesh] faceTexture NOT SET!
[QGMMainWindow::updateTexMapComboBox] No Face texture maps stored.
[Mesh] vertexTexture NOT SET!
[QGMMainWindow::updateTexMapComboBox] No Vertex texture maps stored.
[MeshWidget::initializeGL] 4.1.0 NVIDIA 260.19.29
[MeshWidget::initializeGL] NVIDIA Corporation
[MeshWidget::initializeGL] GL_vertex_buffer_object defined.
[MeshWidget::initializeGL] GL_ARB_vertex_buffer_object defined.
[MeshWidget::initializeGL] GL_VERSION_1_5 defined.
[MeshWidget::initializeGL] Number of lights: 8
[MeshGL::vboPrepare] Time Vertices: 4.16 seconds.
[MeshGL::vboPrepare] Time: 5.38 seconds.
```

Figure 6.5: The DPI of the file marked with red.

Then hit `File` → `Export Screenshots` → `Screenshot Views` and answer the question for tiled rendering with yes. For the optional export of the information into a LaTeXfile go to `Extra` → `Cuneiform Figure Latex Info`.

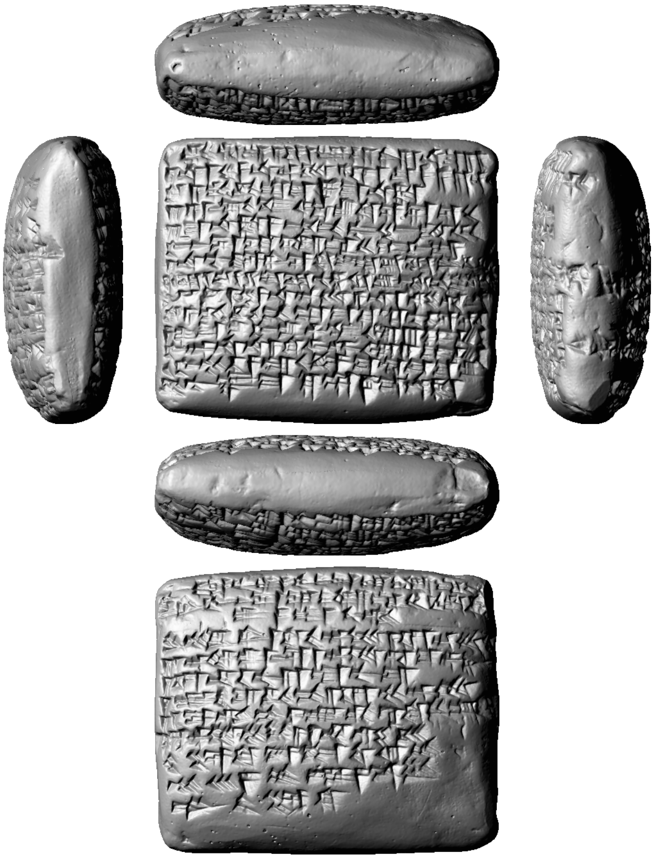


Figure 6.6: HOSIG 26 – 6 views, scale 1 : 1.5

Max. dim.	$6{,}8 \times 5{,}7 \times 2{,}4\,cm$	Surface, acq.	$10\,765\,mm^2$
Vertices	1 628 456	Res., avg	$151\,mm^{-2}$
Faces	3 255 868		$312\,DPI$
Material		Volume, est.	$61\ \pm 1\,cm^3$

6.2 Ceramics

GigaMesh features the possibility to create cutting planes. In order to get the proper outline of, e.g. a vessel, this possibility sets a plane inside the vessel and “cuts” the object thereby showing its outline.

The first step is to `Select` → `Plane - 3 Points)` where the user will define a plane by three points: mark three points on the plane with three doubleclicks. The plane can be shown with `View` → `Mesh Plane`. In theory this plane is movable. Then calculate the distance to the plane. By selecting `Analyze` → `Iso Lines to Polylines` the value for Isovalue should be set to 0. Then with `File` → `Export Screenshots` → `Screenshot (SVG + PNG)` of the current view. This projects the polygonal lines to screenshots and calculates the outline.

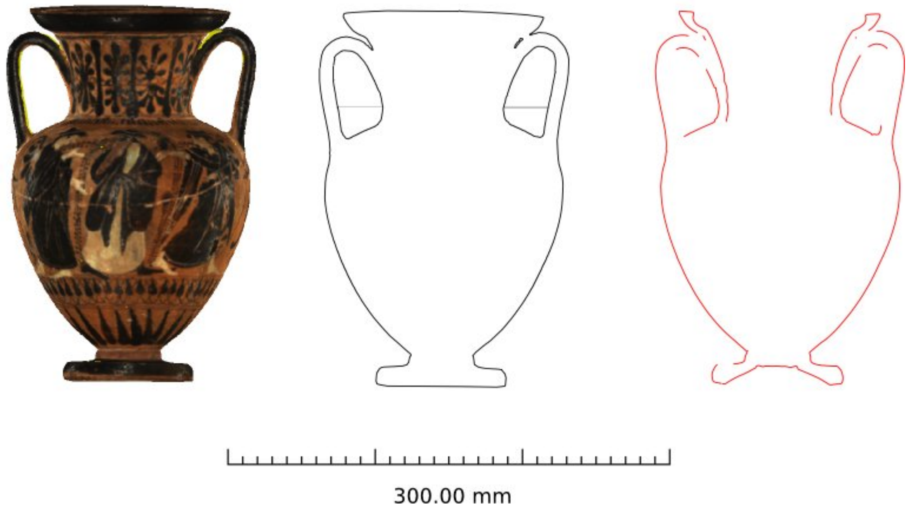


Figure 6.7: Vessel and its cutting planes

Appendix A

Keyboard Usage

The following keys are used for often recureing tasks. They are not necessarily shortcuts of the items in the pull-down menu. They are meant for convenience and precise transformations.

Transformation of the view

A / D Rotate CCW/CW about y-axis by 1° (yaw)

W / S Rotate CCW/CW about x-axis by 1° (pitch)

Q / E Rotate CCW/CW about z-axis by 1° (roll)

Y / X Rotate CCW/CW about the y-axis by 90° (yaw)

C / V Rotate CCW/CW about the x-axis by 90° (pitch)

B / N Rotate CCW/CW abbout the z-axis by 90° (roll)

← ↑ ↓ → Shift the viewport in orthographic projection mode

Lighting

P Toggles between moving the lights and moving the object in relation to the camera

0 Light (generic) on/off

1 Switch on/off light no. 1

2 Switch on/off light no. 2

3 Switch on/off light no. 3

4 Switch on/off light no. 4

5 Switch on/off light no. 5

9 Switch on/off ambient light

Ctrl + Left mouse button Move the lights (all together)

Additional shortcuts

F1 Toggle bounding box (on/off)

F2 Toggle graticules (on/off)

F3 Toggle clip through selection (on/off)

F4 Set SelPrim to reference point for the view of the camera

F5 Toggle backface-culling (on/off)

F6 Make current view to default view

F7 Toggle grid in orthogonal mode (on/off)

Shift + F7 Set DPI-value for grid in orthogonal mode

F8 Rotate orthogonal to plane

F12 Reset view and lights to the default, ignoring the zoom factor

Shift + F12 Reset view and zoom factor and lights to the default (A-side)

Shift + Left mouse button Rotate mesh plane

Shift + Right mouse button Translate mesh plane

Ctrl + Q Quit the program

Appendix B

FAQ

Question: What do I do if there is only one new file after the computation of the feature vector?

Answer: Something went wrong during the computation. Since the program is robust against unclean meshes you should try to run it again and hope for a proper result in the second run. Therefore you have to delete the `*info.txt` file and restart the computation.

Question: There are NO executables in the `mesh` folder.

Answer: If the folder contains no executables then they still have to be compiled. Just type make and wait until you get prompted again. If there are no error messages the compiling was successful.

Question: After selecting `Visualize` → `Feature Correlation to Selected` a window poped up saying that the function is not available, ALGLIB library missing.

Answer: Make sure that the library has been compiled or do it now by changing into the folder `external` in your *GigaMesh* folder. Then execute the shellscript by typing `make_alglib.sh`. NOTE: If the library is available, the ERROR message indicates that there are NO feature vectors available in your `*.ply` file. Make sure that you have opened a file with precomputed feature vectores as described in paragraph 3.

Bibliography

[BHH03] Cynthia A. Brewer, Geoffrey W. Hatchard, and Mark A. Harrower. ColorBrewer in Print: A Catalog of Color Schemes for Maps. *Cartography and Geographic Information Science*, 30:5–32, 2003. URL `http://colorbrewer2.org/`. cited p. 29

[Eis02] J. David Eisenberg. *SVG Essentials*. O'Reilly, 1 edition, 2002. URL `http://books.google.de/books?id=AhXV40mCC94C`. cited p. 3

[HFG$^+$06] Qi-Xing Huang, Simon Flöry, Natasha Gelfand, Michael Hofer, and Helmut Pottmann. Reassembling Fractured Objects by Geometric Matching. In *Proc. of Int. Conference on Computer Graphics and Interactive Technique (SIGGRAPH)*, pages 569–578. ACM, Boston, MA, USA, 2006. ISBN 1-59593-364-6. URL `http://portal.acm.org/citation.cfm?id=1179352.1141925&coll=GUIDE&dl=GUIDE&CFID=88441359&CFTOKEN=11938920`. cited p. 3

[Mar12] Hubert Mara. *Multi-Scale Integral Invariants for Robust Character Extraction from Irregular Polygon Mesh Data*. PhD thesis, Ruperto Carola, Heidelberg University, Interdisciplinary Center for Scientific Computing (IWR), September 2012. URL `http://www.ub.uni-heidelberg.de/archiv/13890`. Supervisors: Willi Jäger and Hans Georg Bock. cited p. 3

[Mau05] Stefan M. Maul. *Das Gilgamesch-Epos*. C.H.Beck, Munich, Germany, 3 edition, 2005. ISBN 3406528708, 9783406528705. 191 pp. URL `http://books.google.at/books?id=H99u2oexd7YC`. cited p. 1

[MK03] Hubert Mara and Martin Kampel. Automated Extraction of Profiles from 3D Models of Archaeological Fragments. In *Proc. of the XIX CIPA Int. Symposium: New Perspectives to Save Cultural Heritage,*, pages 87–93. Antalya, Turkey, 2003. URL `http://cipa.icomos.org/text\%20files/antalya/20.pdf`. cited p. 29

[MKJB10] Hubert Mara, Susanne Krömker, Stefan Jakob, and Bernd Breuckmann. GigaMesh and Gilgamesh - 3D Multiscale Integral Invariant Cuneiform Character Extraction. In *Proc. VAST Int. Symposium on Virtual Reality, Archaeology and Cultural Heritage*, pages 131–138. Palais du Louvre, Paris, France, 2010. cited p. 13, 34

[PWHY09] Helmut Pottmann, Johannes Wallner, Qi-Xing Huang, and Yong-Liang Yang. Integral Invariants for Robust Geometry Processing. *Computer Aided Geometric Design*, 26(1):37–60, Jannuary 2009. URL `http://portal.acm.org/citation.cfm?id=1464795`. cited p. 3