SPECTRAL LAPLACE REPRESENTATION

IMPLEMENTATION IN OPENGL

SOFTWARE ENGINEERING PROJECT

PRAMITA WINATA

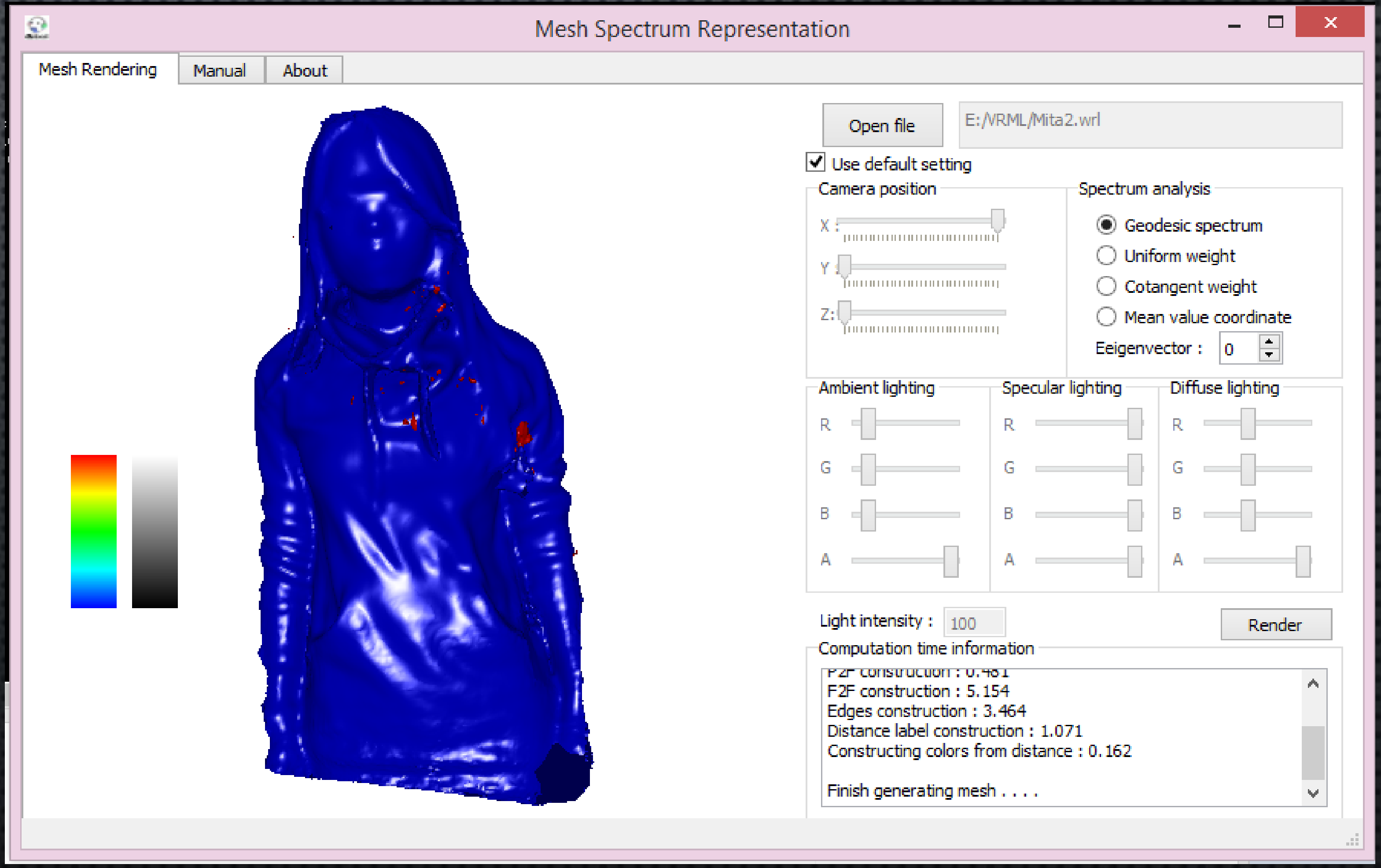
DRAGUTIN

AGENDA

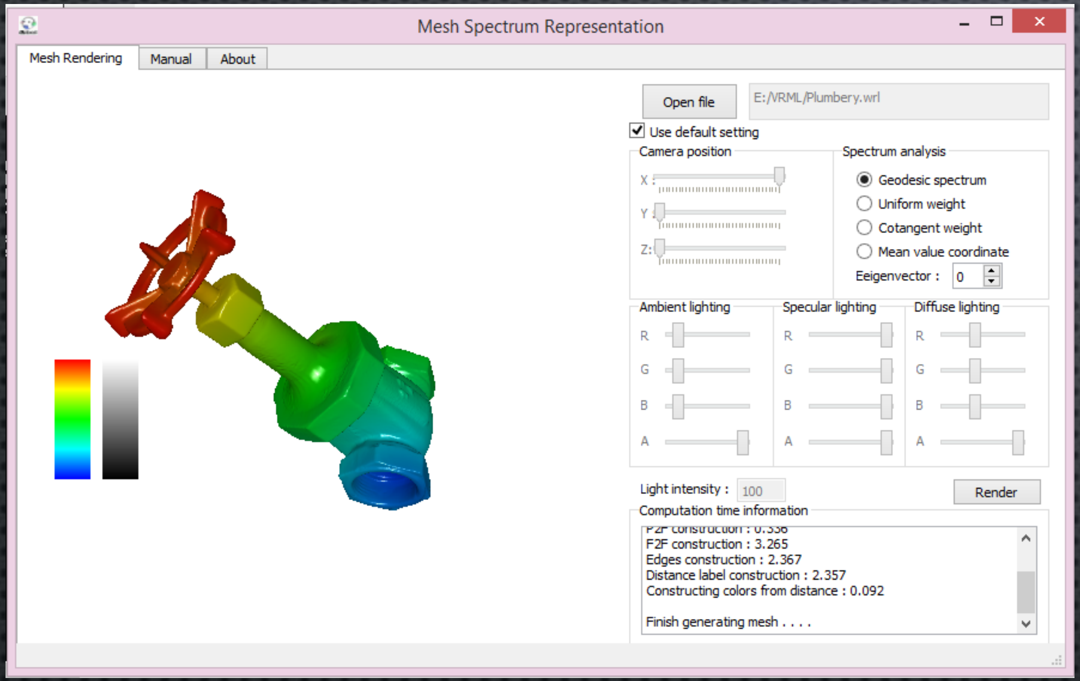
- INTRODUCTION
- APPLICATION – OPENGL WIDGET APPLICATION
- BASIC LAPLACE MESH REPRESENTATION
- DEMO

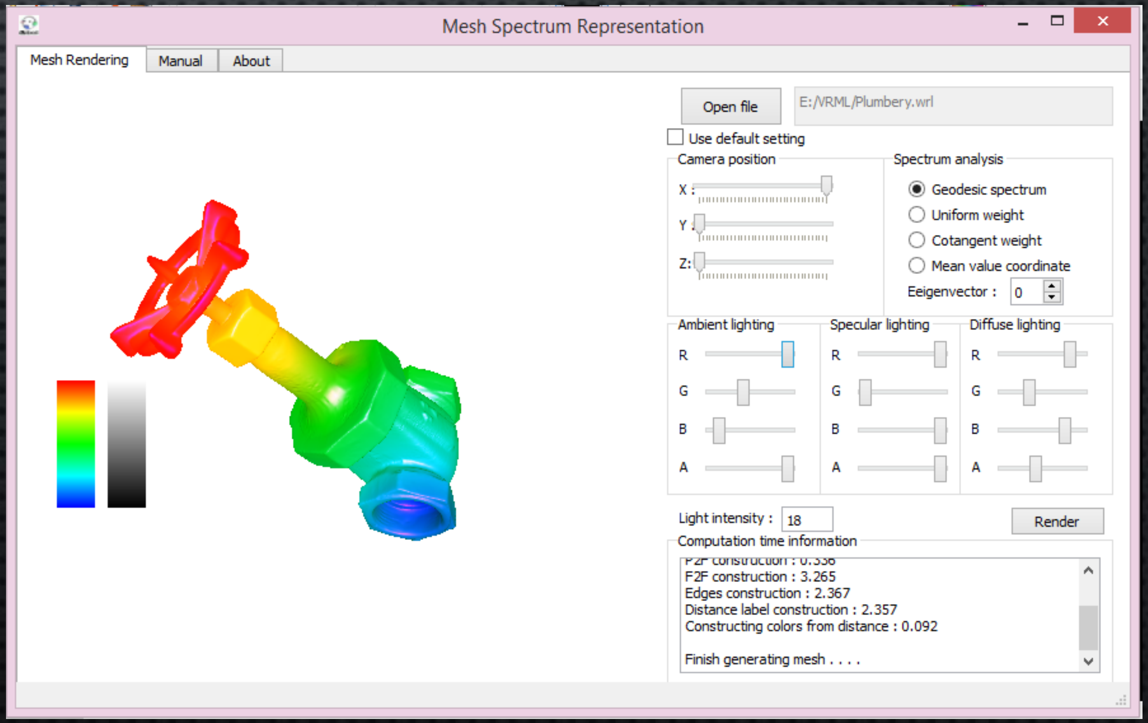
APPLICATION – QT APPLICATION

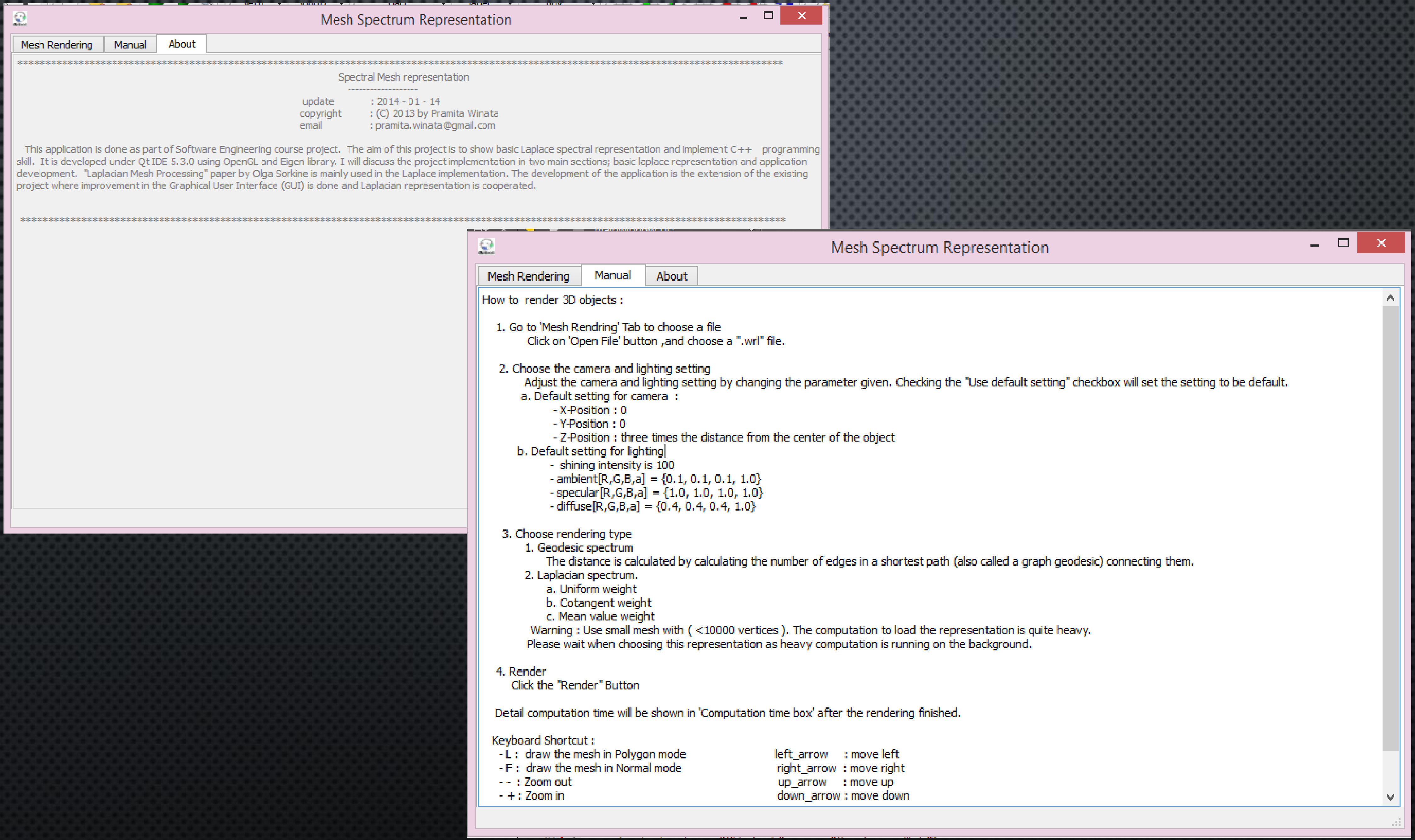
- QT SDK 5.30, MINGW 32 BIT
- OPENGL
- EIGEN LIBRARY



DEFAULT SETTING VS USER CONFIGURATION






Mesh Spectrum Representation
Mesh Rendering
Manual
About
Spectral Mesh representation
update : 2014 - 01 - 14
copyright : (C) 2013 by Pramita Winata
email : pramita.winata@gmail.com
This application is done as part of Software Engineering course project. The aim of this project is to show basic Laplace spectral representation and implement C++ programming skill. It is developed under Qt IDE 5.3.0 using OpenGL and Eigen library. I will discuss the project implementation in two main sections; basic laplace representation and application development. "Laplacian Mesh Processing" paper by Olga Sorkine is mainly used in the Laplace implementation. The development of the application is the extension of the existing project where improvement in the Graphical User Interface (GUI) is done and Laplacian representation is cooperated.
Mesh Spectrum Representation
Mesh Rendering
Manual
About
How to render 3D objects :
1. Go to 'Mesh Rendring' Tab to choose a file
Click on 'Open File' button ,and choose a ".wrl" file.
2. Choose the camera and lighting setting
Adjust the camera and lighting setting by changing the parameter given. Checking the "Use default setting" checkbox will set the setting to be default.
a. Default setting for camera :
- X-Position : 0
- Y-Position : 0
- Z-Position : three times the distance from the center of the object
b. Default setting for lighting
- shining intensity is 100
- ambient[R,G,B,a] = {0.1, 0.1, 0.1, 1.0}
- specular[R,G,B,a] = {1.0, 1.0, 1.0, 1.0}
- diffuse[R,G,B,a] = {0.4, 0.4, 0.4, 1.0}
3. Choose rendering type
1. Geodesic spectrum
The distance is calculated by calculating the number of edges in a shortest path (also called a graph geodesic) connecting them.
2. Laplacian spectrum.
a. Uniform weight
b. Cotangent weight
c. Mean value weight
Warning : Use small mesh with (<10000 vertices). The computation to load the representation is quite heavy.
Please wait when choosing this representation as heavy computation is running on the background.
4. Render
Click the "Render" Button
Detail computation time will be shown in 'Computation time box' after the rendering finished.
Keyboard Shortcut :
- L : draw the mesh in Polygon mode
- F : draw the mesh in Normal mode
- - : Zoom out
- + : Zoom in
left_arrow : move left
right_arrow : move right
up_arrow : move up
down_arrow : move down

Basic Laplace Representation

- In Laplacian framework, each vertex is represented with consideration to its neighbors. Laplacian meshes store the location of a vertex relative to its neighboring vertices, *delta coordinate*. This is useful when the information of object's shape are important for example when deforming the object itself. With Laplacian mesh representation, we can use manipulate one vertex, as known as 'anchor' based on, to have the overall objects deformed in a correct way.
- Since a delta coordinate is a linear combination of a vertex and its neighbors, to obtain delta coordinates for all vertices can be represented as a matrix, called the **Laplacian Matrix** .
- It is important to understand that each vertex will have weight correspond to each of its neighbor. In this project, I will implement three method to get the weight of the vertex. These method are uniform weight, cotangent weight, and mean-value-coordinate.

Uniform Weight

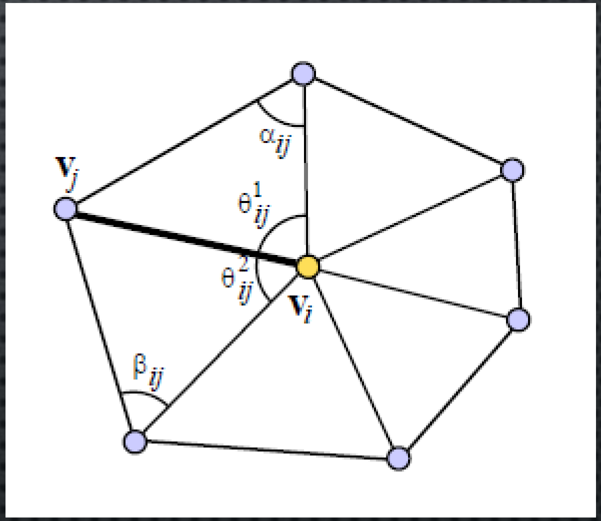
- Based on the size of its immediate neighbor.
- The Laplacian Matrix is constructed as follows:
    - If i = j, the **ijth** entry is equal to the size of its immediate neighbour
    - If i ≠ j then the **ijth** entry is -1
    - 0, otherwise

COTANGENT WEIGHT

- Based on the size of its immediate neighbor.

$$w_{i,j} = \frac{1}{2}(\cot \alpha_{i,j} + \cot \beta_{i,j})$$

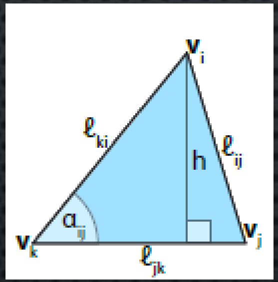
- The Laplacian Matrix is constructed as follows:
    - If $i \neq j$ and (i, j) are neighbor the ij^{th} entry is equal to weight
    - If $i = j$ then the ij^{th} entry is 1
    - 0, otherwise



Cotangent weight illustration
Obtained from Olga Sorkine
STAR report, EUROGRAPHICS 2005.

GETTING THE COTANGENT WEIGHT

- MEYER M., DESBRUN M., SCHRÖDER P., BARR A. H.: DISCRETE DIFFERENTIAL-GEOMETRY OPERATORS FOR TRIANGULATED 2-MANIFOLDS. IN \TEXTIT{"VISUALIZATION AND MATHEMATICS III}, HEGE H.-C., POLTHIER K., (EDS.). SPRINGER-VERLAG, HEIDELBERG, 2003, PP. 35–57.



$$A_{ijk} = \sqrt{r(r - l_{ij})(r - l_{jk})(r - l_{ki})}$$

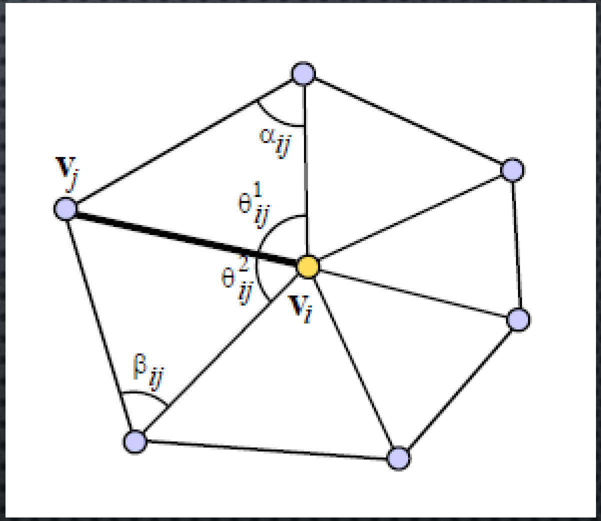
$$A_{ijk} = \frac{1}{2} l_{jk} l_{ki} \sin \alpha_{ij} \;\rightarrow\; \sin \alpha_{ij} = \frac{2A_{ijk}}{l_{jk} l_{ki}}.$$

$$l_{ij}^2 = l_{jk}^2 + l_{ki}^2 - 2 l_{jk} l_{ki} \cos \alpha_{ij} \;\rightarrow\; \cos \alpha_{ij} = \frac{-l_{ij}^2 + l_{jk}^2 + l_{ki}^2}{2 l_{jk} l_{ki}}.$$

$$\cot \alpha_{ij} = \frac{\cos \alpha_{ij}}{\sin \alpha_{ij}} = \frac{-l_{ij}^2 + l_{jk}^2 + l_{ki}^2}{2 l_{jk} l_{ki}} \frac{l_{jk} l_{ki}}{2A_{ijk}} = \frac{-l_{ij}^2 + l_{jk}^2 + l_{ki}^2}{4A_{ijk}}.$$

MEAN VALUE COORDINATE

- BASED ON THE SIZE OF ITS IMMEDIATE NEIGHBOR.
- $$w_{i,j} = \frac{1}{\|v_i - v_j\|}(tan\theta_{ij}^{1}\frac{}{2} + tan\theta_{ij}^{2}/2)$$
- THE LAPLACIAN MATRIX IS CONSTRUCTED AS FOLLOWS:
  - IF I ≠ J AND (I ,J) ARE NEIGHBOR THE IJTH ENTRY IS EQUAL TO WEIGHT
  - IF I = J THEN THE IJTH ENTRY IS 1
  - 0, OTHERWISE



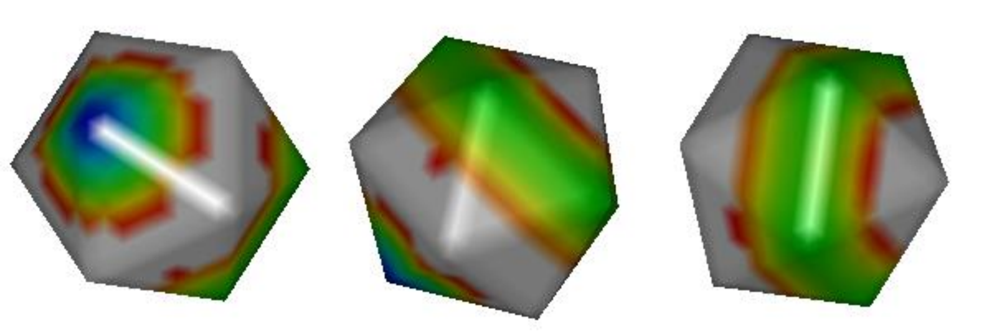
Mean value coordinate illustration

Obtained from Olga Sorkine
STAR report, EUROGRAPHICS 2005.

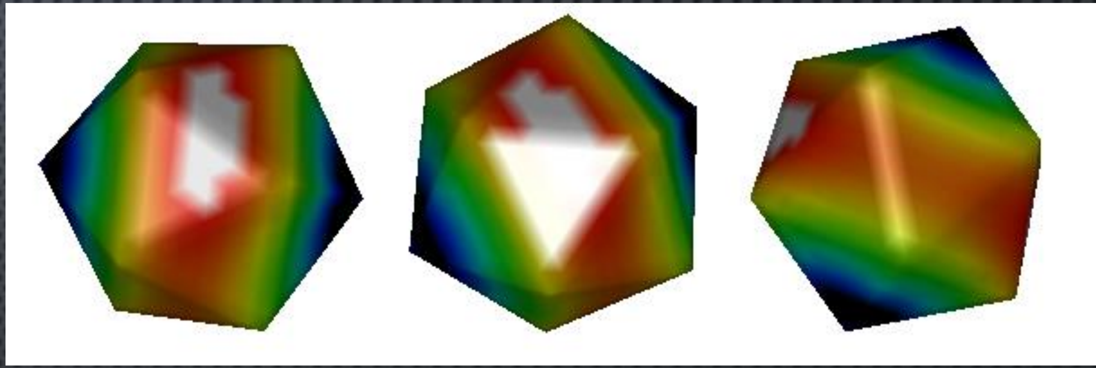
EIGEN DECOMPOSITION – COLOR MAP

- OPENGL – EIGEN SOLVER
- CHOOSE ONE EIGENVECTOR FROM THE GENERATED BY THE SOLVER
- BUILD A COLOR MAP – 6 AREAS OF COLORS
- RESULT

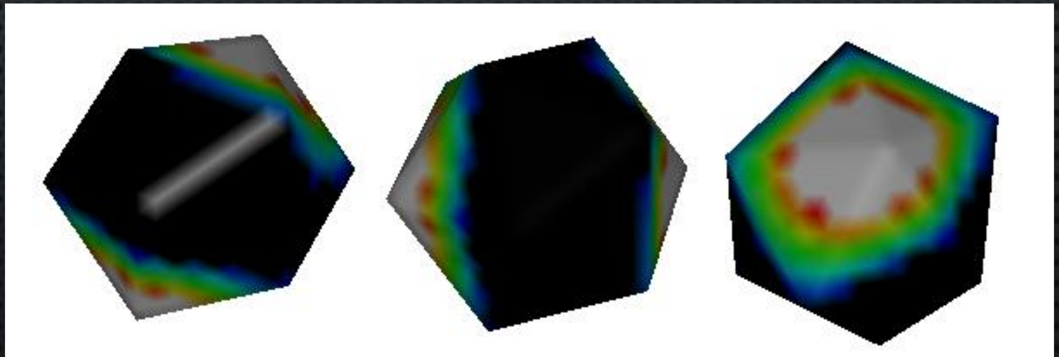
RESULT



Uniform weight



Cotangent weight



Mean value coordinate

DEMO

REFERENCES

[1] Olga Sorkine, *Laplacian Mesh Processing*. STAR report, Eurographics, 2005.

[2] Mesh Processing, G. Peyr, *https://www.ceremade.dauphine.fr/ peyre/teaching/manifold-sci/*

[3] Eigen library, *http://eigen.tuxfamily.org/index.php?title=Main Page*

[4] The C++ Standard library, see *http://www.yolinux*.

[5] MEYER M., DESBRUN M., SCHRDER P., BARR A. H.: Discrete dierential-geometry operators for triangulated 2-manifolds. In *Visualization and Mathematics III*, Hege H.-C., Polthier K., (Eds.). Springer-Verlag, Heidelberg, 2003, pp. 3557.

[6] HERON. 60. *Metrica*. Alexandria, Roman Egypt.

[7] FIEDLER M.: Algebraic connectivity of graphs. *Czech. Math. Journal* 23 (1973), 298305.

[8] LOATER M. S.: *Mean value coordinates*. CAGD 20, 1 (2003), 1927.