Assignment Two: Harmonic Map for Topological Disk

David Gu

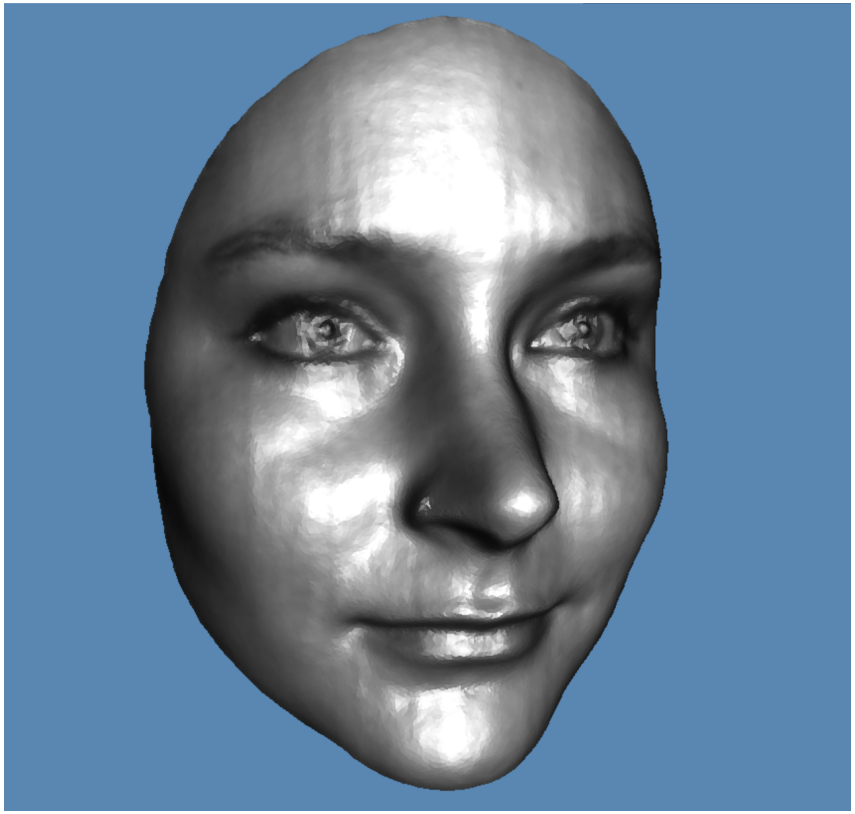
Yau Mathematics Science Center
Tsinghua University
Computer Science Department
Stony Brook University

gu@cs.stonybrook.edu

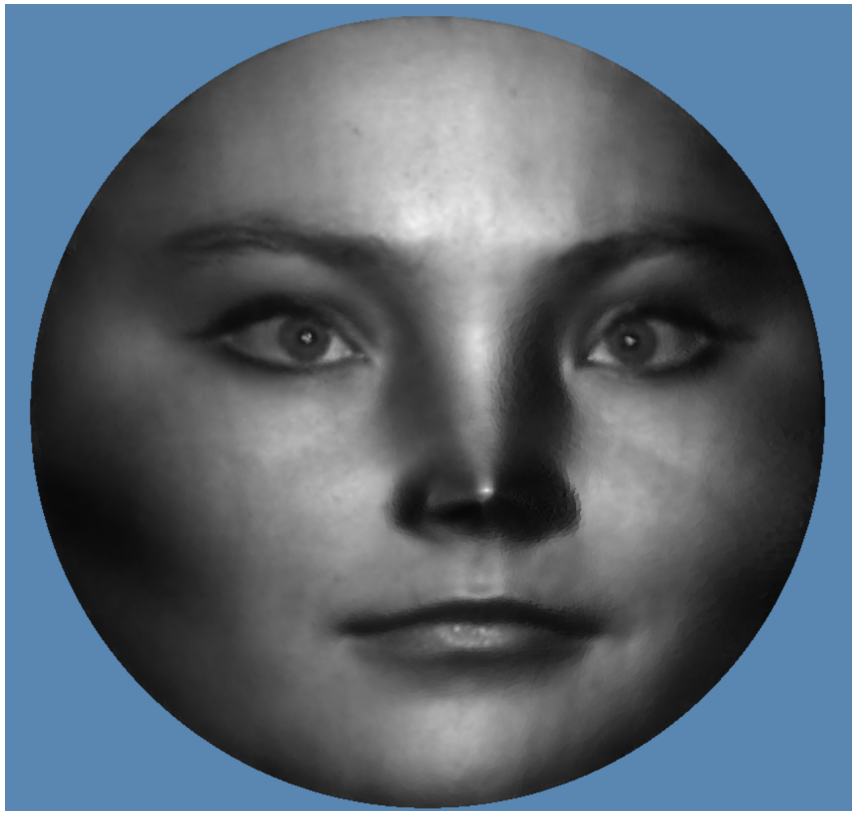
July 16, 2020

Triangle Mesh: Discrete Harmonic Maps

Surface Harmonic Map

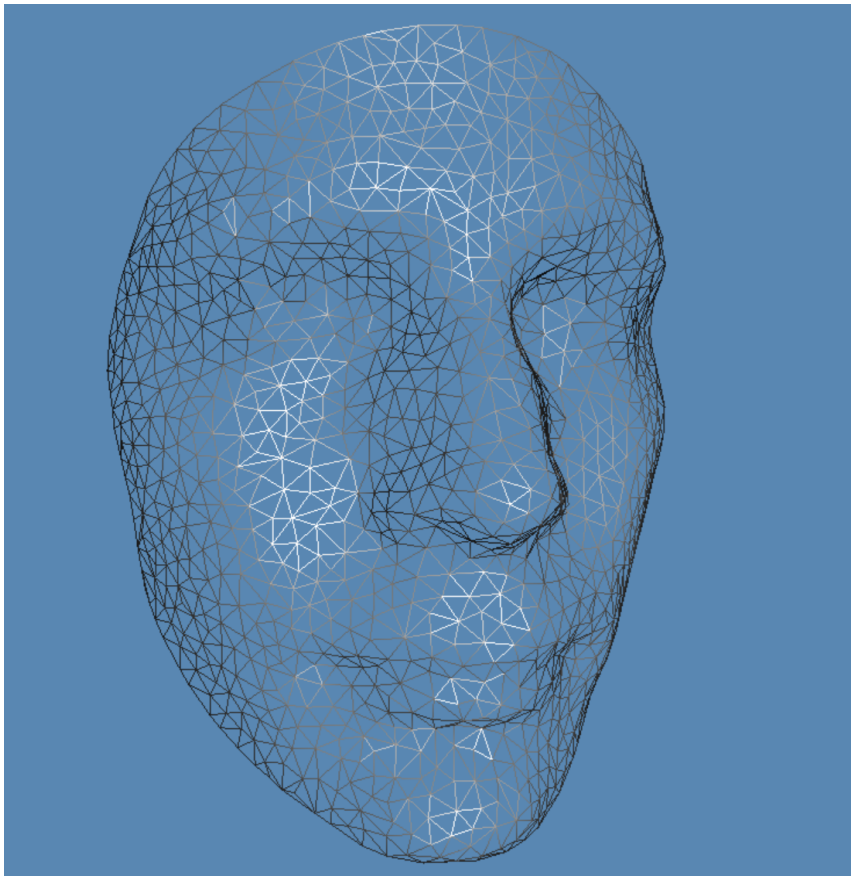


input mesh

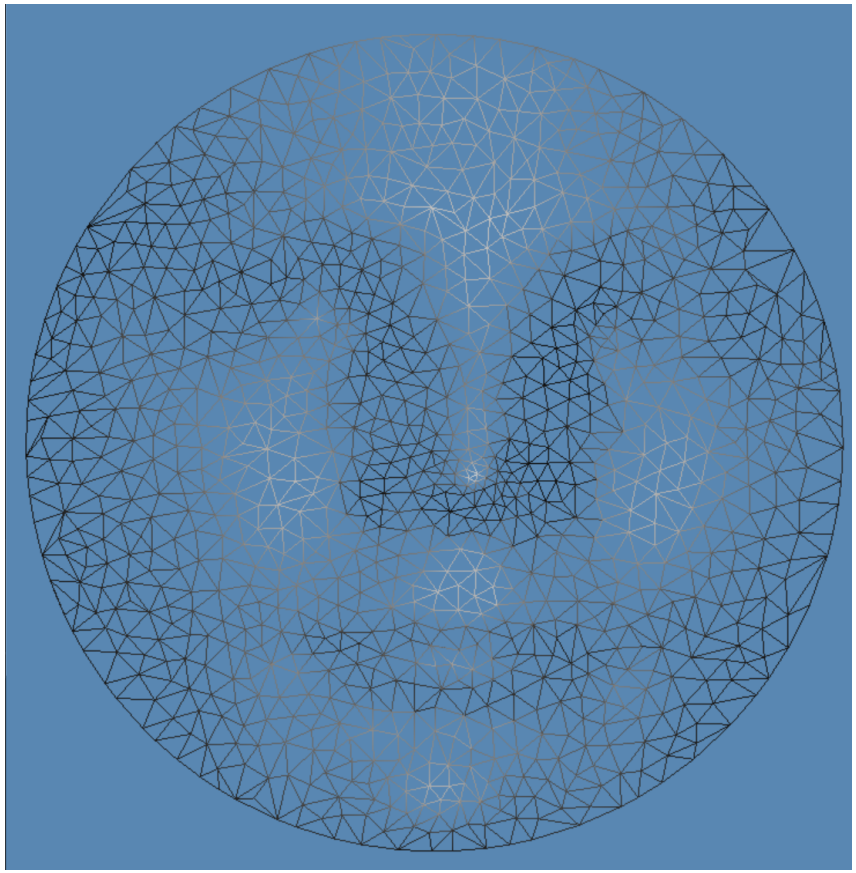


harmonic map image

Surface Harmonic Map



input mesh



harmonic map image

Task

This C++ project aims at helping students to implement geometric algorithms: harmonic maps for topological disks.

System

The code has been tested on Windows, Linux and Mac. If there is any problem on the platforms, please let the instructor know.

Dependencies

1. 'MeshLib', a mesh library based on halfedge data structure.
2. 'freeglut', a free-software/open-source alternative to the OpenGL Utility Toolkit (GLUT) library.

Directory Structure

- harmonic_map/include, The header files of cut graph
- harmonic_map/src, The source files of cut graph algorithm.
- data,Some models.
- CMakeLists.txt, CMake configuration file.
- resources, Some resources needed.
- 3rdparty, MeshLib and freeglut libraries.

Configuration

Before you start, read README.md carefully, then go three the following procedures, step by step.

1. Install [CMake](https://cmake.org/download/).
2. Download the source code of the C++ framework.
3. Configure and generate the project for Visual Studio.
4. Open the .sln using Visual Studio, and complie the solution.
5. Finish your code in your IDE.
6. Run the executable program.

3. Configure and generate the project

1. open a command window
2. cd ccg_homework_skeleton
3. mkdir build
4. cd build
5. cmake ..
6. open HarmonicMap.sln inside the build directory.

5. Finish your code in your IDE

- You only need to modify one file: HarmonicMap.cpp
- search for comments

  $$//insertyourcodehere$$

  and insert your code
- Modify

  CHarmonicMap::set_mesh(CHarmonicMapMesh * pMesh)

  move the image of all the vertices to the origin.
- Modify

  CHarmonicMap::step_one()

  move the image of each vertex to the weighted center of the images of its neighbors;
- Modify

  CHarmonicMap::map()

  Constuct each element of the Matrix A and B;

5. Finish your code in your IDE

- Modify

  CHarmonicMap::set_calculate_edge_weight()

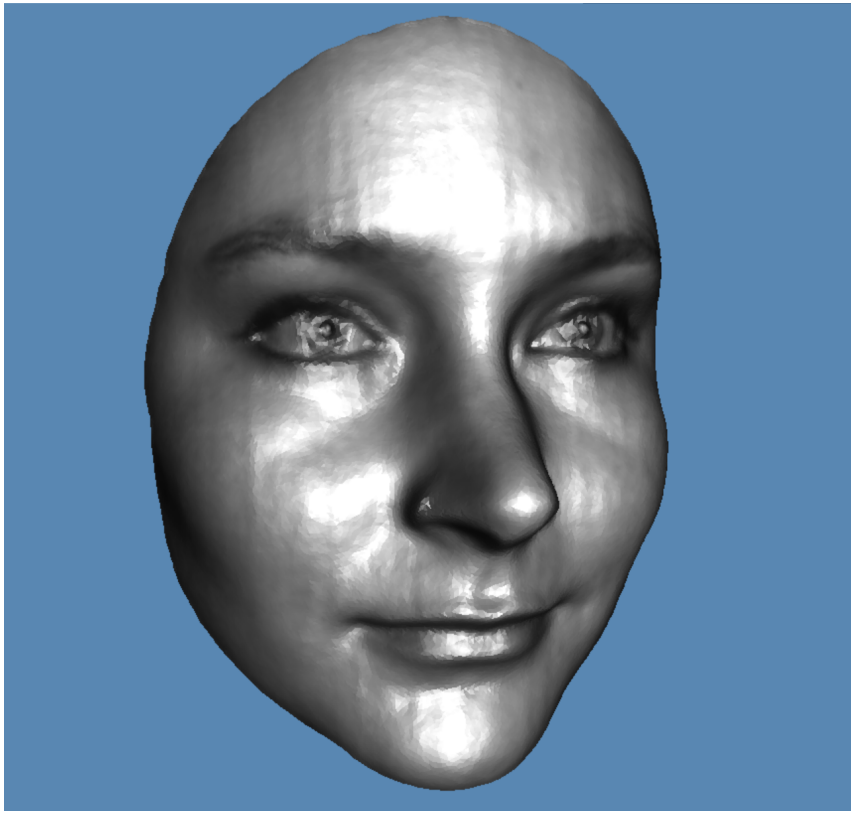
  Compute the corner angles using cosine law.
- Modify

  CHarmonicMap::set_calculate_edge_weight()

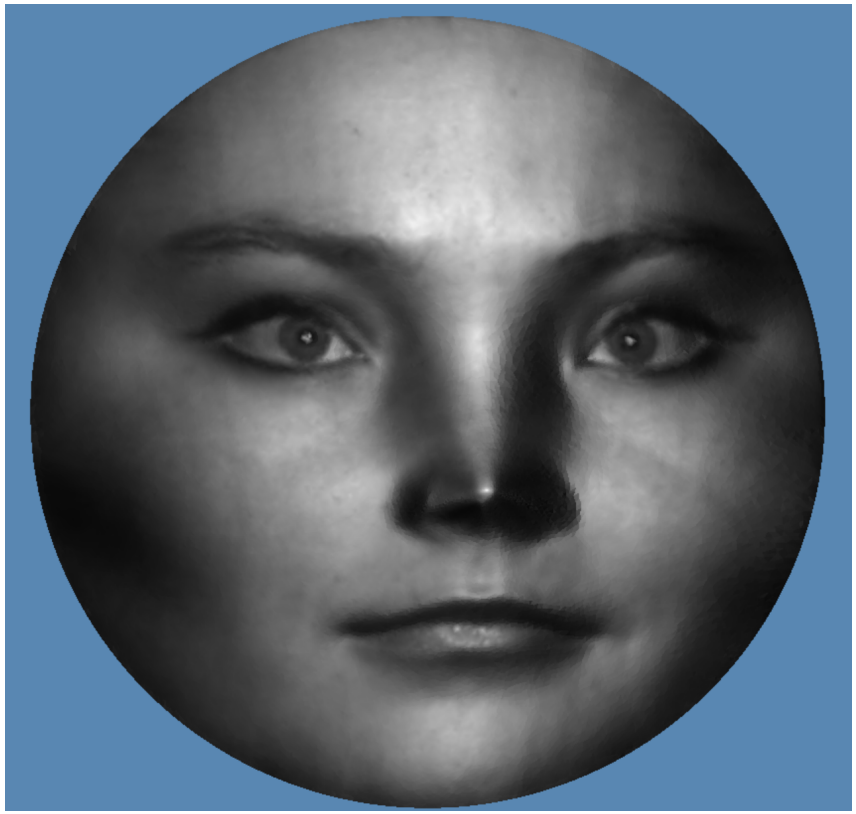
  Compute the cotangent edge weight;

6 Run the executable program

Command: HarmonicMap.exe ../../data/girl.m, press 'h', then press '2'



input mesh



harmonic map image

Iterative Algorithm

Input: A topological disk mesh M;
Output: Harmonic map result, stored at the vertex uv coordinates;

1. Trace the boundary of M counter clockwisely, set the uv to be on the unit circle, the angle for each vertex is proportional to the arc length;
2. Set all the interior vertices uv to be at the original $(0, 0)$;
3. Compute all the corner angles;
4. Compute edge cotangent edge weight, w_{ij} for edge $[v_i, v_j]$;
5. For each vertex v_i, move it uv to the weighted center of its neighbors,

$$uv(v_i) \leftarrow \frac{\sum_j w_{ij}\ uv(v_j)}{\sum_j w_{ij}}$$

6. Repeat step 5, until it converges.

Direct Algorithm

Input: A topological disk mesh M;
Output: Harmonic map, $\varphi : V \to \mathbb{R}^2$;

1. Trace the boundary of M counter clockwisely, set the uv to be on the unit circle, the angle for each vertex is proportional to the arc length;
2. Compute all the corner angles;
3. Compute edge cotangent edge weight, w_{ij} for edge $[v_i, v_j]$;
4. For each interior vertex v_i, construct a linear equation

$$\sum_{v_j \sim v_i} w_{ij}(\varphi(v_j) - \varphi(v_i)) = 0.$$

with Dirichlet boundary condition.

Cotangent Edge Weight

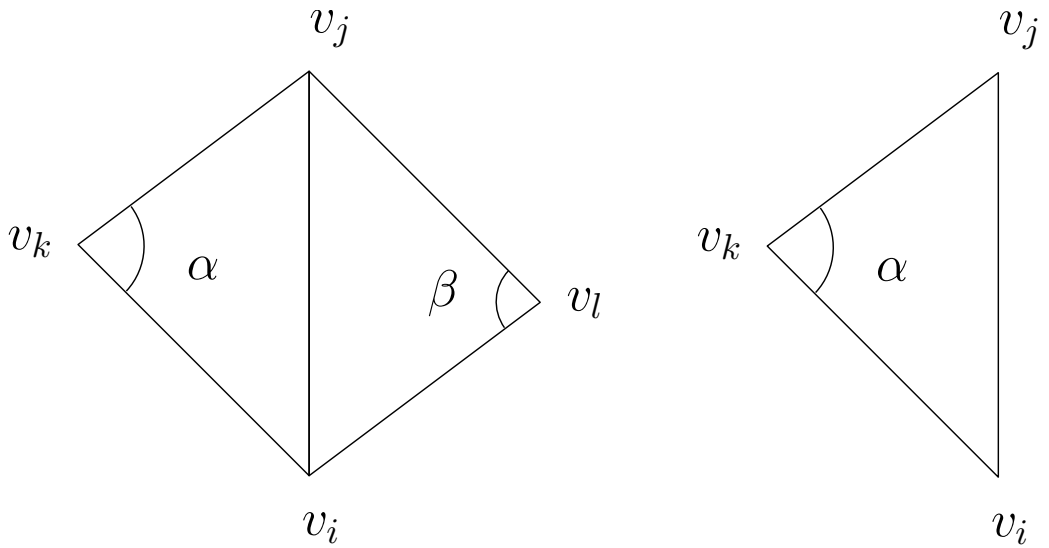
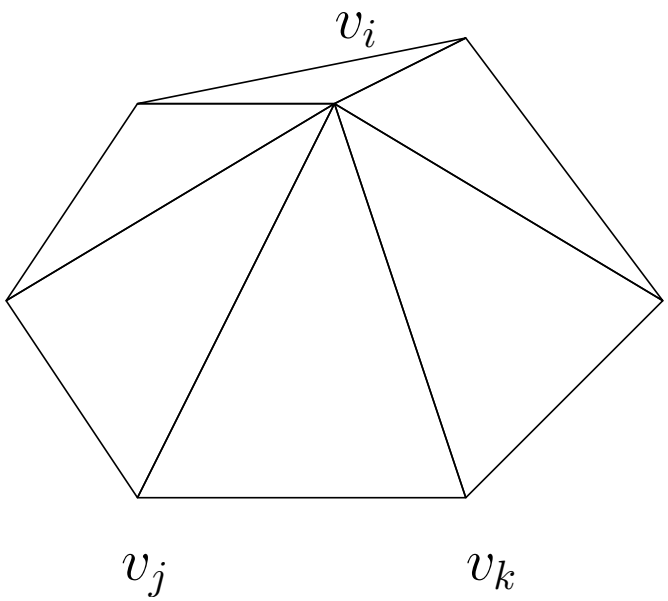


Figure: Cotangnet edge weight.

$$w_{ij} = \begin{cases} \cot\alpha + \cot\beta & [v_i, v_j] \notin \partial M \\ \cot\alpha & [v_i, v_j] \in \partial M \end{cases}$$

Discrete Harmonic Map



Given a discrete map $\varphi : V \to \mathbb{R}^2$,

$$\Delta\varphi(v_i) = \sum_j w_{ij}(\varphi(v_j) - \varphi(v_i)) = 0, \quad \forall v_i \notin \partial M,$$

with Dirichlet boundary condition $\varphi(v_k) = f(v_k)$, for all $v_k \in \partial M$.