Assignment Four: Spherical Harmonic Map

David Gu

Computer Science Department
Stony Brook University

gu@cs.stonybrook.edu

October 6, 2023

Harmonic Maps

Harmonic Map

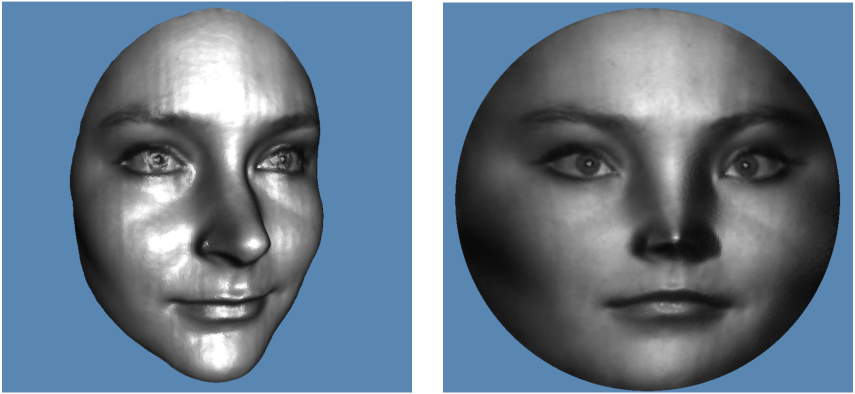

Figure: Harmonic map between topological disks.

Harmonic Map

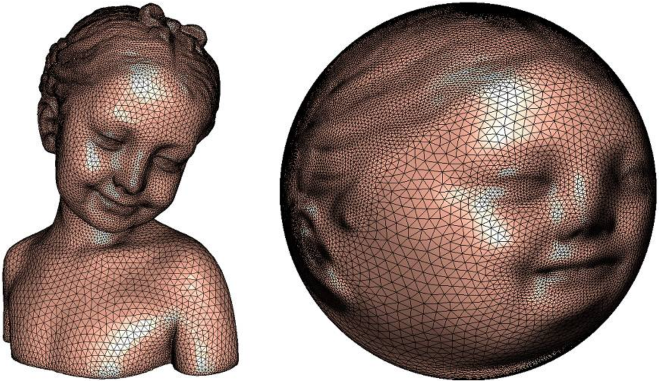

Figure: Harmonic map between topological spheres.

Surface Double Covering Algorithm

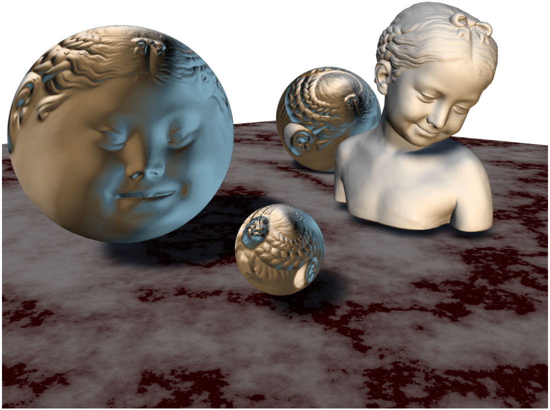

Figure: Spherical harmonic map.

Surface Double Covering Algorithm

Figure: Spherical harmonic map.

Computational Algorithm for Disk Harmonic Maps

Input:A topological disk M;
Output:A harmonic map $\varphi : M \to \mathbb{D}^2$

1. Construct boundary map to the unit circle, $g : \partial M \to \mathbb{S}^1$, g should be a homeomorphism;
2. Compute the cotangent edge weight;
3. for each interior vertex $v_i \in M$, compute Laplacian

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

4. Solve the linear system, to obtain φ.

Computational Algorithm for Spherical Harmonic Map

Input: A genus zero closed mesh M;step length $\lambda > 0$
Output: A spherical harmonic map $\varphi : M \to \mathbb{S}^2$;

1. Compute Gauss map $\varphi : M \to \mathbb{S}^2$, $\varphi(v) \leftarrow \mathbf{n}(v)$;
2. Compute the cotangent edge weight, compute Laplacian
$$\Delta\varphi(v_i) = \sum_{v_i \sim v_j} w_{ij}(\varphi(v_j) - \varphi(v_i)),$$
3. project the Laplacian to the tangent plane,
$$D\varphi(v_i) = \Delta\varphi(v_i) - \langle \Delta\varphi(v_i), \varphi(v_i)\rangle \varphi(v_i)$$
4. for each vertex, $\varphi(v_i) \leftarrow \varphi(v_i) + \lambda D\varphi(v_i)$;
5. compute the mass center $c = \sum A_i \varphi(v_i) / \sum_j A_j$; normalize $\varphi(v_i) \leftarrow (\varphi(v_i) - c)/|\varphi(v_i) - c|$;
6. Repeat step 2 through 5, until the Laplacian norm is less than ε.

Line Search

The step length can be calculated by line search. We minimize the harmonic energy,

$$E(\lambda) = \frac{1}{2}\sum_{[v_i,v_j]} w_{ij}|\varphi(v_i) + \lambda D\varphi(v_i) - \varphi(v_j) - \lambda D\varphi(v_j)|^2,$$

$$\frac{\partial E}{\partial \lambda} = \sum w_{ij}\langle(\varphi(v_i) - \varphi(v_j)) + \lambda(D\varphi(v_i) - D\varphi(v_j)), D\varphi(v_i) - D\varphi(v_j)\rangle$$

By $\partial E/\partial\lambda = 0$, we obtain

$$\lambda = -\frac{\sum w_{ij}\langle\varphi(v_i) - \varphi(v_j), D\varphi(v_i) - D\varphi(v_j)\rangle}{\sum w_{ij}|D\varphi(v_i) - D\varphi(v_j)|^2}$$

Computational Algorithm for a Random Walk

Input: A Topological Disk mesh M, a starting interior vertex v_0
Output: A random walk from v_0 to the boundary ∂M;

1. Compute cotangent edge weight w_{ij}, if $w_{ij} < 0$ replace it by an constant $\varepsilon > 0$;
2. at the current vertex v_i, generate a random number $r \in [0, 1]$,
3. sort all the neighboring vertex v_j counter-clock-wisely, find k, compute γ_k
$$\gamma_k := \frac{\sum_{j=0}^{k} w_{ij}}{\sum_{v_i \sim v_l} w_{il}}$$
4. find k, such that $r \in [\gamma_k, \gamma_{k+1})$, walk to v_k,
5. Repeat step 2 through 4, until reaches the boundary.

Computational Algorithm for Stochastic Harmonic Map

Input: A Topological Disk mesh M
Output: Harmonic map $\varphi : M \to \mathbb{D}$

1. Compute cotangent edge weight w_{ij}, if $w_{ij} < 0$ replace it by an constant $\varepsilon > 0$;
2. Set the boundary condition, maps each boundary vertex to the unit circle, $\varphi(v_k) = e^{i\theta_k}$, θ_k is proportional to the arc length parameter of ∂M;
3. for each interior vertex v_i, generate n random walks γ_i, the end vertex of γ_i is denoted as $e(\gamma_i)$, the start every $s(\gamma)$, estimate the expect value
$$\hat{E}_{s(\gamma)=v_i}(\varphi(e(\gamma))) = \frac{1}{n}\sum_{i=1}^{n}\varphi(e(\gamma_i)).$$
4. set $\varphi(v_i) \leftarrow \hat{E}_{s(\gamma)=v_i}(\varphi(e(\gamma)))$.

Instruction

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

- spherical_harmonic_map/include, the header files for Hodge decomposition;
- spherical_harmonic_map/src, the source files for Hodge decomposition algorithm.
- data,Some models.
- CMakeLists.txt, CMake configuration file.
- resources, Some resources needed.
- 3rdparty, MeshLib and freeglut libraries.

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 CCGHomework.sln inside the build directory.

5. Finish your code in your IDE

- Modify

  double CSphericalHarmonicMap::step_one(int steps, double step_length)

1. compute vertex laplacian
2. get the noraml component
3. get the tangent_component
4. update u
5. normalize the vertex u() to the unit sphere
6. normalize the mapping, such that mass center is at the origin
7. compute the harmonic energy

5. Finish your code in your IDE

- Modify

  double CSphericalHarmonicMap::_normalize()

1. compute the mass center of the image, using the vertex u() and vertex area();
2. move the mass center to the origin;
3. normalize vertex u() to be on the unit sphere.

Run the executable program

Command line:

spherical_harmonic_map.exe mesh.m

All the data files are in the data folder.