Assignment Three: Hodge Decomposition and Riemann Mapping

David Gu

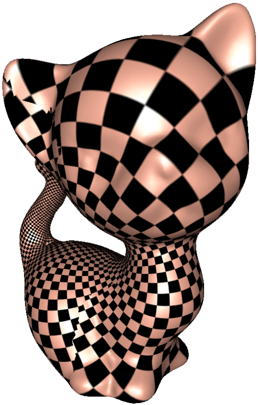
Yau Mathematics Science Center
Tsinghua University
Computer Science Department
Stony Brook University

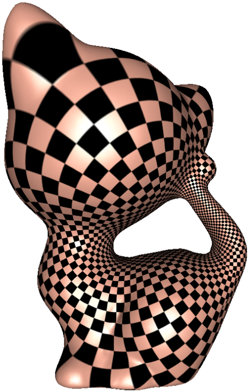
gu@cs.stonybrook.edu

July 27, 2020

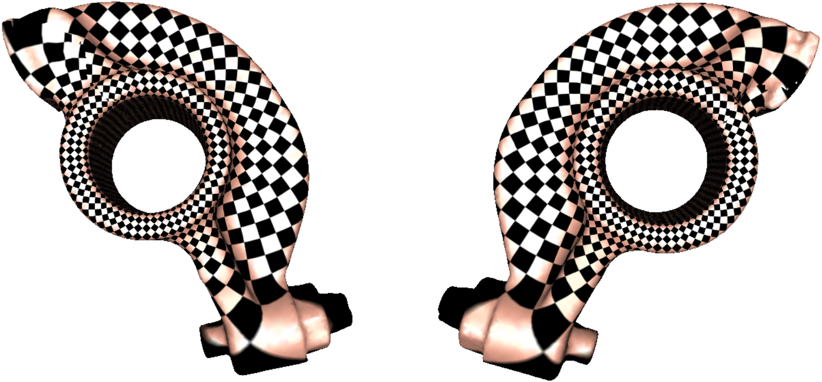
Hodge Decomposition

Holomorphic One-form

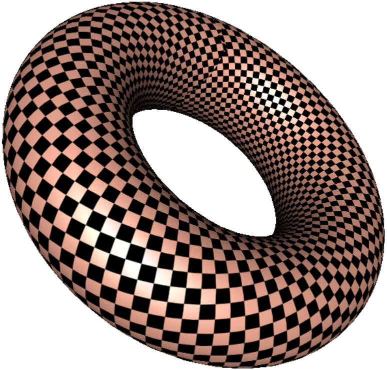


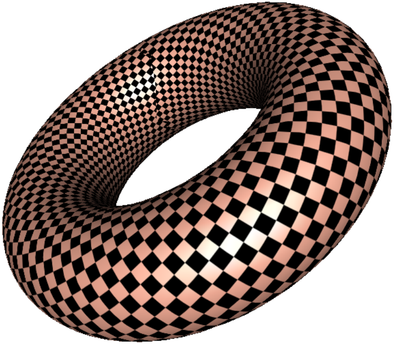


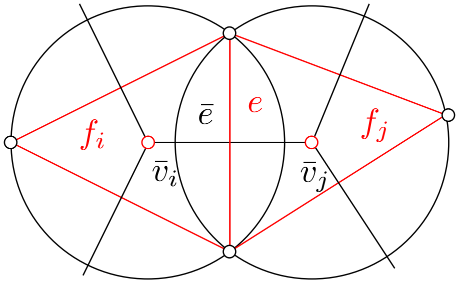
Holomorphic One-form



Holomorphic One-form



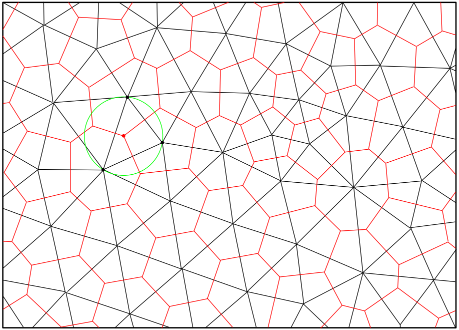




Cotangent edge weight:

$$w_{ij} = \frac{1}{2}(\cot\alpha + \cot\beta)\omega(e). \quad (1)$$

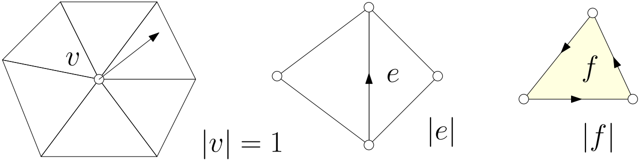
Dual Mesh



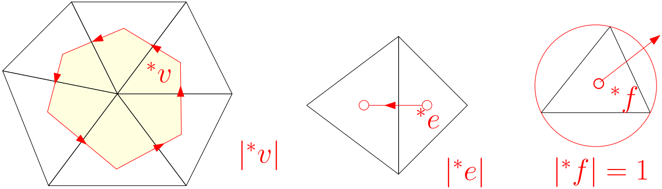
Poincaré's duality, equivalent to Delaunay triangulation and Voronoi diagram. The Delaunay triangulation is the primal mesh, the Voronoi diagram is the dual mesh.

Duality

$0-form\ \eta$ $\quad$ $1-form\ \omega$ $\quad$ $2-form\ \Omega$



$$\frac{\eta(v)}{|v|} = \frac{{}^*\eta({}^*v)}{|{}^*v|} \qquad \frac{\omega(e)}{|e|} = \frac{{}^*\omega({}^*e)}{|{}^*e|} \qquad \frac{\Omega(f)}{|f|} = \frac{{}^*\Omega({}^*f)}{|{}^*f|}$$



Discrete Operator

Discrte Codifferential Operator

The codifferential operator $\delta : \Omega^p \to \Omega^{p-1}$ on an n-dimensional manifold,

$$\delta := (-1)^{n(p+1)+1} \, {}^* d^*.$$

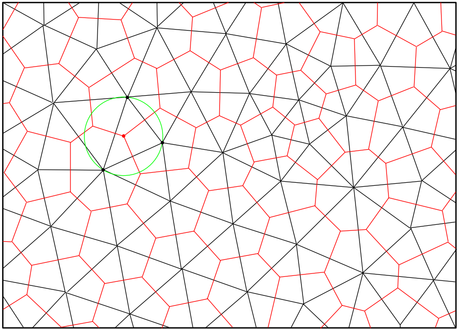
Discrte Hodge star operator

$** : \Omega^p \to \Omega^p$,

$$** := (-1)^{(n-p)p}$$

$$^*(^*\omega)(e) = (^*\omega)(^*e)\frac{|e|}{|^*e|}(-1) = \omega(e)\frac{|^*e|}{|e|}\frac{|e|}{|^*e|}(-1).$$

Dual Mesh

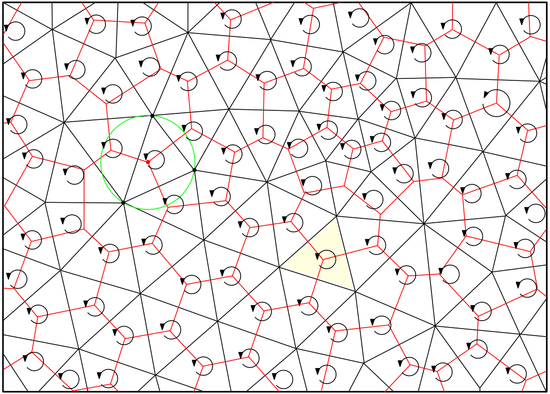


Generate a random one-form ω on the prime mesh, by Hodge decomposition theorem:

$$\omega = d\eta + \delta\Omega + h$$

where η is a 0-form, Ω a 2-form and h a harmonic one-form.

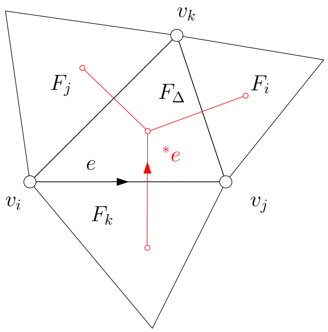
Discrete Harmonic One-form



compute $d\omega$,

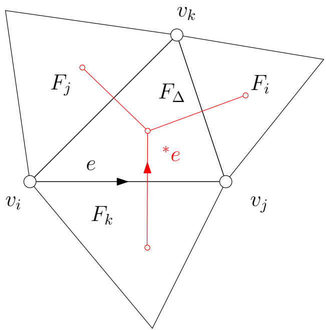
$$d\omega = d^2\eta + d\delta\Omega + dh = d\delta\Omega, \quad \Omega = (d\delta)^{-1}(d\omega).$$

Discrete Harmonic One-form



$$
\begin{aligned}
& \delta\Omega([v_i, v_j]) \\
=& (-1)(^*d^*)\Omega([v_i, v_j]) \\
=& (-1)(d^*\Omega)(^*[v_i, v_j])\frac{1}{w_{ij}}(-1) \\
=& \frac{1}{w_{ij}}(d^*\Omega)([^*f_k, {}^*f_\Delta]) \\
=& \frac{1}{w_{ij}}(^*\Omega)(\partial[^*f_k, {}^*f_\Delta]) \\
=& \frac{1}{w_{ij}}\left\{^*\Omega(^*f_\Delta) - {}^*\Omega(^*f_k)\right\} \\
=& \frac{1}{w_{ij}}\left\{\frac{\Omega(f_\Delta)}{|f_\Delta|} - \frac{\Omega(f_k)}{|f_k|}\right\}
\end{aligned}
$$

Discrete Harmonic One-form



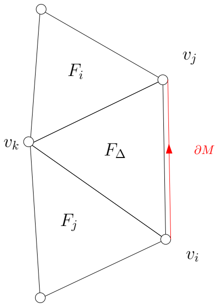
$$\delta\Omega([v_i, v_j]) = \frac{1}{w_{ij}} \left\{ \frac{\Omega(f_\Delta)}{|f_\Delta|} - \frac{\Omega(f_k)}{|f_k|} \right\}$$

For each face Δ, we have the equation $d\omega(\Delta) = \omega(\partial\Delta) = d\delta\Omega(\Delta)$,

$$\omega(\partial\Delta) = \frac{F_i - F_\Delta}{w_{jk}} + \frac{F_j - F_\Delta}{w_{ki}} + \frac{F_k - F_\Delta}{w_{ij}} \quad (2)$$

where $F_i = -\frac{\Omega(f_i)}{|f_i|}$'s are 2-forms, ω is 1-form, w_{ij}'s are cotangent edge weights.

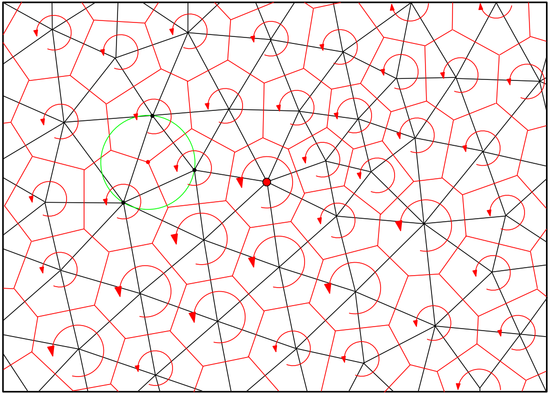
Discrete Harmonic One-form



For each boundary face Δ, we have the equation

$$d\omega(\Delta) = \omega(\partial\Delta) = \frac{F_i - F_\Delta}{w_{jk}} + \frac{F_j - F_\Delta}{w_{ki}} + \boxed{\frac{0 - F_\Delta}{w_{ij}}} \quad (3)$$

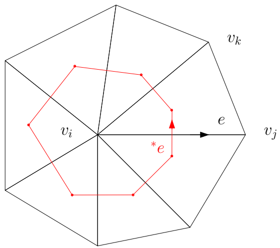
Discrete Harmonic One-form



compute $\delta\omega$,

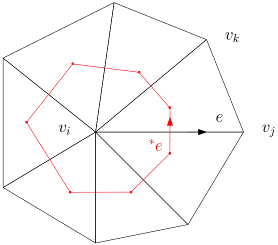
$$\delta\omega = \delta d\eta + \delta^2\Omega + \delta h = \delta d\eta, \quad \eta = (\delta d)^{-1}(\delta\omega).$$

Discrete Harmonic One-form



$$
\begin{aligned}
&\delta\omega(v_i)\\
&= (-1)({}^*d^*)\omega(v_i)\\
&= (-1)(d^*\omega)({}^*v_i)\frac{1}{|{}^*v_i|}\\
&= (-1)({}^*\omega)(\partial\ {}^*v_i)\frac{1}{|{}^*v_i|}\\
&= (-1)\sum_j({}^*\omega)({}^*e_{ij})\frac{1}{|{}^*v_i|}\\
&= (-1)\frac{1}{|{}^*v_i|}\sum_j w_{ij}\ \omega(e_{ij})
\end{aligned}
$$

Discrete Harmonic One-form



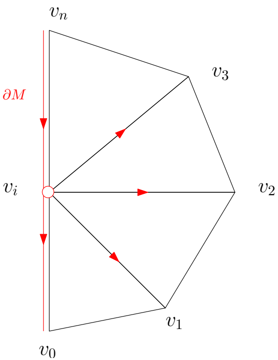
$$\delta\omega(v_i) = (-1)\frac{1}{|^*v_i|}\sum_j w_{ij}\ \omega(e_{ij})$$

For each vertex v_i, we obtain an equation $\delta\omega(v_i) = \delta d\eta(v_i)$,

$$\sum_{v_i \sim v_j} w_{ij}\ \omega([v_i, v_j]) = \sum_{v_i \sim v_j} w_{ij}(\eta_j - \eta_i). \quad (4)$$

where η_i's are 0-forms, w_{ij}'s are cotangent edge weights.

Discrete Harmonic One-form



for each boundary vertex v_i, we obtain an equation:

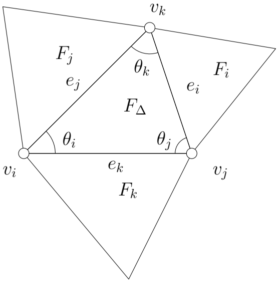
$$\sum_{j=0}^{n-1} w_{ij}\ \omega([v_i, v_j]) \boxed{-w_{i,n}\ \omega([v_n, v_i])} = \sum_{j=0}^{n} w_{ij}(\eta_j - \eta_i). \quad (5)$$

Algorithm for Random Harmonic One-form

Input:A closed genus one mesh M;
output: A basis of harmonic one-form group;

1. Generate a random one form ω, assign each $\omega(e)$ a random number;
2. Compute cotangent edge weight using Eqn. (1);
3. Compute the coexact form δF using Eqn. (2);
4. Compute the exact form df using Eqn. (4);
5. Harmonic 1-form is obtained by $h = \omega - d\eta - \delta\Omega$;

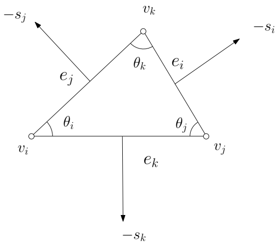
Wedge Product



Given two one-forms ω_1 and ω_2 on a triangle mesh M, then the 2-form $\omega_1 \wedge \omega_2$ on each face $\Delta = [v_i, v_j, v_k]$ is evaluated as

$$\omega_1 \wedge \omega_2(\Delta) = \frac{1}{6} \begin{vmatrix} \omega_1(e_i) & \omega_1(e_j) & \omega_1(e_k) \\ \omega_2(e_i) & \omega_2(e_j) & \omega_2(e_k) \\ 1 & 1 & 1 \end{vmatrix} \tag{6}$$

Wedge Product Formula



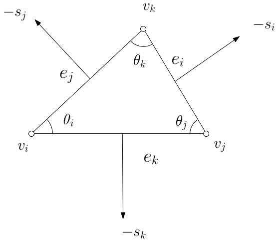
Set $f : \Delta \to \mathbb{R}$,

$$\begin{cases} f(v_i) = 0 \\ f(v_j) = \omega(e_k) \\ f(v_k) = -\omega(e_j) \end{cases}$$

$$\nabla f(p) = \frac{1}{2A}(f(v_i)\mathbf{s}_i + f(v_j)\mathbf{s}_j + f(v_k)\mathbf{s}_k)$$

$$\begin{aligned} \mathbf{w} &= \frac{1}{2A}[\omega(e_k)\mathbf{s}_j - \omega(e_j)\mathbf{s}_k] \\ &= \frac{\mathbf{n}}{2A} \times [\omega(e_k)(\mathbf{v}_i - \mathbf{v}_k) - \omega(e_j)(\mathbf{v}_j - \mathbf{v}_i)] \\ &= -\frac{\mathbf{n}}{2A} \times [\omega(e_k)\mathbf{v}_k + \omega(e_j)\mathbf{v}_j + \omega(e_i)\mathbf{v}_i] \end{aligned}$$

Wedge Product Formula



$$\int_\Delta \omega_1 \wedge \omega_2 = A|\mathbf{w}_1 \times \mathbf{w}_2|$$
$$= \frac{A}{4A^2}(\omega_k^1\omega_j^2 - \omega_j^1\omega_k^2)|\mathbf{s}_j \times \mathbf{s}_k|$$
$$= \frac{1}{2}\begin{vmatrix} \omega_k^1 & \omega_j^1 \\ \omega_k^2 & \omega_j^2 \end{vmatrix}$$

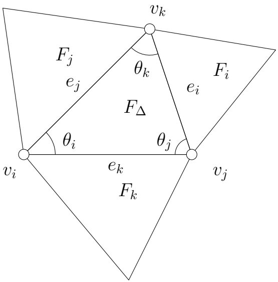
$$\mathbf{w} = \frac{1}{2A}(\omega_k \mathbf{s}_j - \omega_j \mathbf{s}_k)$$

since $\omega_i^\gamma + \omega_j^\gamma + \omega_k^\gamma = 0$, $\gamma = 1, 2$, we obtain

$$\mathbf{w} = \frac{-1}{6A}\begin{vmatrix} \omega_i & \omega_j & \omega_k \\ \mathbf{s}_i & \mathbf{s}_j & \mathbf{s}_k \\ 1 & 1 & 1 \end{vmatrix}$$

$$\boxed{\int_\Delta \omega_1 \wedge \omega_2 = \frac{1}{6}\begin{vmatrix} \omega_k^1 & \omega_j^1 & \omega_i^1 \\ \omega_k^2 & \omega_j^2 & \omega_i^2 \\ 1 & 1 & 1 \end{vmatrix}}$$

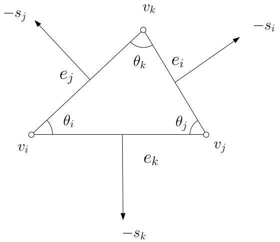
Wedge Product



Given two one-forms ω_1 and ω_2 on a triangle mesh M, then the 2-form $\omega_1 \wedge {}^*\omega_2$ on each face $\Delta = [v_i, v_j, v_k]$ is evaluated as

$$\omega_1 \wedge {}^*\omega_2(\Delta) = \frac{1}{2}[\cot\theta_i \omega_1(e_i)\omega_2(e_i) + \cot\theta_j \omega_1(e_j)\omega_2(e_j) + \cot\theta_k \omega_1(e_k)\omega_2(e_k)] \quad (7)$$

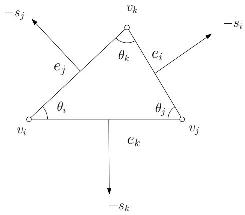
Wedge Product Formula



$$w_1 = \frac{1}{2A}(\omega_k^1 s_j - \omega_j^1 s_k)$$
$$w_2 = \frac{1}{2A}(\omega_k^2 s_j - \omega_j^2 s_k)$$

$$\int_\Delta \omega_1 \wedge {}^*\omega_2 = A\langle w_1, w_2\rangle$$
$$= \frac{1}{4A}\left\{\omega_k^1\omega_k^2\langle s_j, s_j\rangle + \omega_j^1\omega_j^2\langle s_k, s_k\rangle\right.$$
$$\left.-(\omega_k^1\omega_j^2 + \omega_j^1\omega_k^2)\langle s_j, s_k\rangle\right\}$$
$$= \frac{1}{4A}\left\{-\omega_k^1\omega_k^2\langle s_j, s_i + s_k\rangle\right.$$
$$- \omega_j^1\omega_j^2\langle s_k, s_i + s_j\rangle$$
$$\left.-(\omega_k^1\omega_j^2 + \omega_j^1\omega_k^2)\langle s_j, s_k\rangle\right\}$$

Wedge Product Formula



$$
\begin{aligned}
&= \frac{1}{4A}\left\{-\omega_k^1\omega_k^2\langle s_j, s_i\rangle - \omega_k^1\omega_k^2\langle s_j, s_k\rangle\right.\\
&- \omega_j^1\omega_j^2\langle s_k, s_i\rangle - \omega_j^1\omega_j^2\langle s_k, s_j\rangle\\
&\left.-(\omega_k^1\omega_j^2 + \omega_j^1\omega_k^2)\langle s_j, s_k\rangle\right\}\\
&= -\omega_k^1\omega_k^2\frac{\langle s_j, s_i\rangle}{4A} - \omega_j^1\omega_j^2\frac{\langle s_k, s_i\rangle}{4A}\\
&- \frac{\langle s_k, s_j\rangle}{4A}(\omega_k^1\omega_k^2 + \omega_j^1\omega_j^2 + \omega_k^1\omega_j^2 + \omega_j^1\omega_k^2)\\
&= -\omega_k^1\omega_k^2\frac{\langle s_j, s_i\rangle}{4A} - \omega_j^1\omega_j^2\frac{\langle s_k, s_i\rangle}{4A}\\
&- \frac{\langle s_k, s_j\rangle}{4A}(\omega_k^1 + \omega_j^1)(\omega_k^2 + \omega_j^2)\\
&= -\omega_k^1\omega_k^2\frac{\langle s_j, s_i\rangle}{4A} - \omega_j^1\omega_j^2\frac{\langle s_k, s_i\rangle}{4A} - \omega_i^1\omega_i^2\frac{\langle s_j, s_k\rangle}{4A}\\
&= \frac{1}{2}\left(\omega_i^1\omega_i^2\cot\theta_i + \omega_j^1\omega_j^2\cot\theta_j + \omega_k^1\omega_k^2\cot\theta_k\right)
\end{aligned}
$$

Holomorphic 1-form Basis

Given a set of harmonic 1-form basis $\omega_1, \omega_2, \dots, \omega_{2g}$; in smooth case, the conjugate 1-form ${}^*\omega_i$ is also harmonic, therefore

$$
{}^*\omega_i = \lambda_{i1}\omega_1 + \lambda_{i2}\omega_2 + \cdots + \lambda_{i,2g}\omega_{2g},
$$

We get linear equation group,

$$
\begin{pmatrix} \omega_1 \wedge {}^*\omega_i \\ \omega_2 \wedge {}^*\omega_i \\ \vdots \\ \omega_{2g} \wedge {}^*\omega_i \end{pmatrix} = \begin{pmatrix} \omega_1 \wedge \omega_1 & \omega_1 \wedge \omega_2 & \cdots & \omega_1 \wedge \omega_{2g} \\ \omega_2 \wedge \omega_1 & \omega_2 \wedge \omega_2 & \cdots & \omega_2 \wedge \omega_{2g} \\ \vdots & \vdots & & \vdots \\ \omega_{2g} \wedge \omega_1 & \omega_{2g} \wedge \omega_2 & \cdots & \omega_{2g} \wedge \omega_{2g} \end{pmatrix} \begin{pmatrix} \lambda_{i,1} \\ \lambda_{i,2} \\ \vdots \\ \lambda_{i,2g} \end{pmatrix} \tag{8}
$$

We take the integration of each element on both left and right side, and solve the λ_{ij}'s.

Holomorphic 1-form Basis

In order to reduce the random error, we integrate on the whole mesh,

$$\begin{pmatrix} \int_M \omega_1 \wedge {}^*\omega_i \\ \int_M \omega_2 \wedge {}^*\omega_i \\ \vdots \\ \int_M \omega_{2g} \wedge {}^*\omega_i \end{pmatrix} = \begin{pmatrix} \int_M \omega_1 \wedge \omega_1 & \cdots & \int_M \omega_1 \wedge \omega_{2g} \\ \int_M \omega_2 \wedge \omega_1 & \cdots & \int_M \omega_2 \wedge \omega_{2g} \\ \vdots & & \vdots \\ \int_M \omega_{2g} \wedge \omega_1 & \cdots & \int_M \omega_{2g} \wedge \omega_{2g} \end{pmatrix} \begin{pmatrix} \lambda_{i,1} \\ \lambda_{i,2} \\ \vdots \\ \lambda_{i,2g} \end{pmatrix} \tag{9}$$

and solve the linear system to obtain the coefficients.

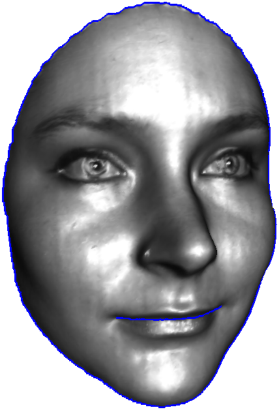
Algorithm for Holomorphic 1-form Basis

Input: A set of harmonic 1-form basis $\omega_1, \omega_2, \ldots, \omega_{2g}$;
Output: A set of holomorphic 1-form basis $\omega_1, \omega_2, \ldots, \omega_{2g}$;

1. Compute the integration of the wedge of ω_i and ω_j, $\int_M \omega \wedge \omega_j$, using Eqn. (6);
2. Compute the integration of the wedge of ω_i and ${}^*\omega_j$, $\int_M \omega \wedge {}^*\omega_j$, using Eqn. (7);
3. Solve linear equation group Eqn. (9), obtain the linear combination coefficients, get conjugate harmonic 1-forms, ${}^*\omega_i = \sum_{j=1}^{2g} \lambda_{ij}\omega_j$
4. Form the holomorphic 1-form basis $\{\omega_i + \sqrt{-1}{}^*\omega_i, \quad i = 1, 2, \ldots, 2g\}$.

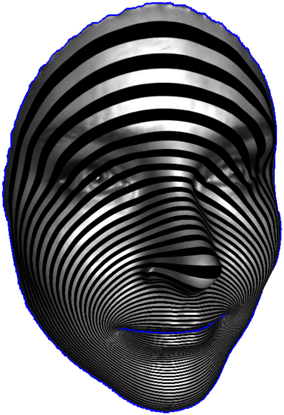
Riemann Mapping

Topological Annulus



Conformal mapping for topological annulus.

Topological Annulus



exact harmonic form

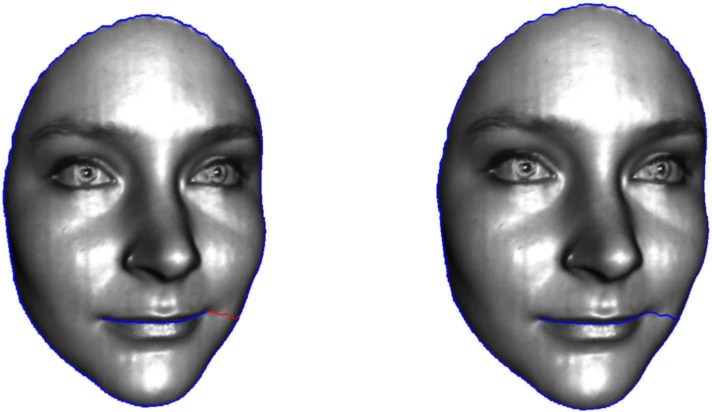
closed harmonic 1-form

Exact Harmonic One-form

Input: A topological annulus M;
Output: Exact harmonic one-form ω;

1. Trace the boundary of the mesh $\partial M = \gamma_0 - \gamma_1$;
2. Set boundary condition:
$$f|_{\gamma_0} = 0, \quad {}_{\gamma_1} = -1;$$
3. Compute cotangent edge weight;
4. Solve Laplace equation $\Delta f \equiv 0$ with Dirichlet boundary condition, for all interior vertex,
$$\sum_{v_i \sim v_j} w_{ij}(f_j - f_i) = 0;$$
5. $\omega = df$.

Topological Fundamental Domain



Find the shortest path τ connecting γ_0 and γ_1, slice the mesh along τ to get a topological disk $\bar{M}$.

Holomorphic One-Form

holomorphic 1-form

1. Use the algorithm for random harmonic One-form algorithm to compute a closed but non-exact harmonic one-form ω_1;
2. Use holomorphic 1-form basis algorithm with $\{\omega, \omega_1\}$ as input to compute a holomorphic 1-form $\omega + \sqrt{-1}^*\omega$.

Integration

Input: A topological disk $\bar{M}$, a holomorphic 1-form;
Output: Integration

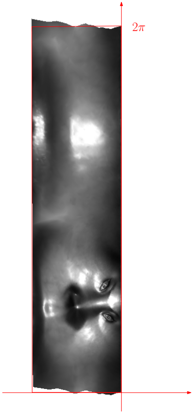
$$\varphi(q) := \int_p^q \omega + \sqrt{-1}^*\omega$$

1. Choose a base point p, set $\varphi(p) = (0,0)$. $p \to touched() = true$, put p to the queue Q;
2. while Q is non-empty, $v_i \leftarrow Q.pop()$;
3. for each adjacent vertex $v_j \sim v_i$, if v_j hasn't been touched, $v_j \to touched() = true$, enqueue v_j to Q;

$$\varphi(v_j) = \varphi(v_i) + (\omega, {}^*\omega)([v_i, v_j]);$$

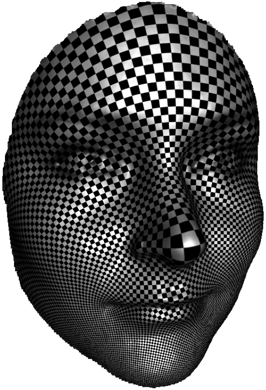
4. repeat step 3,4 until all vertices have been touched.

Integration



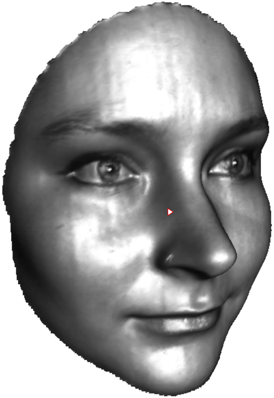
Integrating $\omega + \sqrt{-1}^*\omega$ on $\bar{M}$, normalize the rectangular image $\varphi(\bar{M})$, such that $\varphi(\gamma_0)$ is along the imaginary axis, the height is 2π, $\varphi(\gamma_1)$ is $x = -c$, $c > 0$ is a real number.

Integration



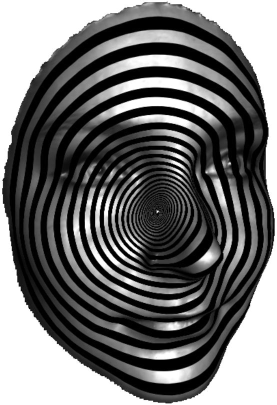
Compute the polar map e^{φ}, which maps $\varphi(\bar{M})$ to an annulus.

Riemann Mapping

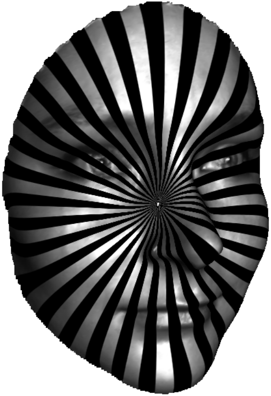


Riemann mapping can be obtained by puncturing a small hole on the surface, then use topological annulus conformal mapping algorithm.

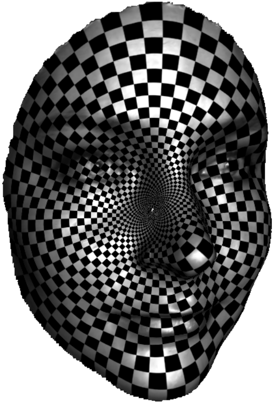
Riemann Mapping



ω



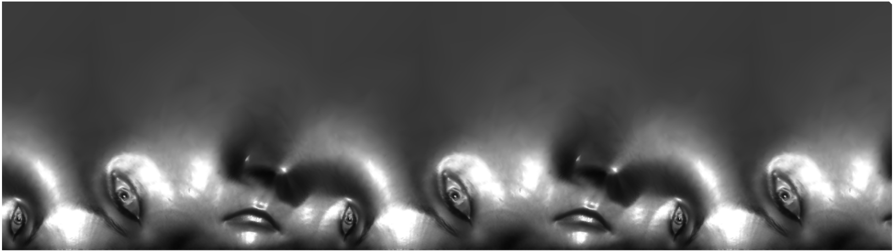
${}^*\omega$



$\omega + \sqrt{-1}{}^*\omega$

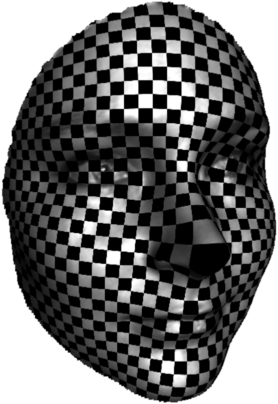
Exact harmonic 1-form and closed, non-exact harmonic 1-form.

Riemann Mapping



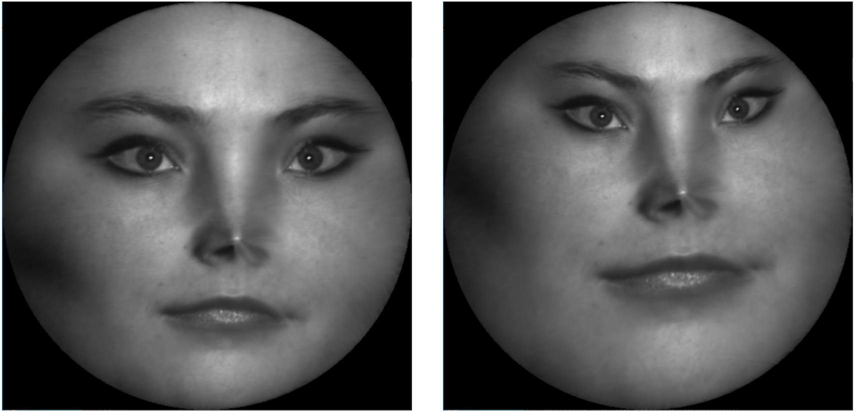
Periodic conformal mapping image $\varphi(M)$.

Riemann Mapping



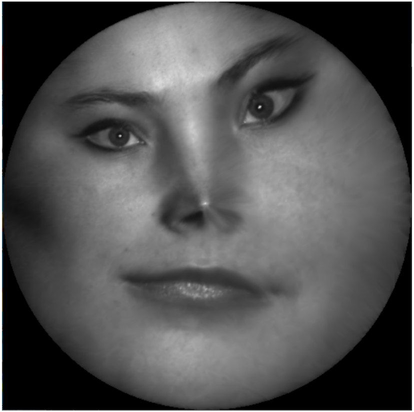
Polar map $e^{\varphi(p)}$ induces the Riemann mapping.

Riemann Mapping



The choice of the central puncture, and the rotation determine a Möbius transformation.

Riemann Mapping



The conformal automorphism of the unit disk is the Möbius transformation group.

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

- hodge_decomposition/include, the header files for Hodge decomposition;
- hodge_decomposition/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.

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 CCGHomework.sln inside the build directory.

5. Finish your code in your IDE

- You need to modify the file: HodgeDecomposition.cpp
- search for comments

$$//insertyourcodehere$$

  and insert your code
- Modify

  MeshLib::CHodgeDecomposition::_d(int dimension)

  MeshLib::CHodgeDecomposition::_delta(int dimension)

  MeshLib::CHodgeDecomposition::_remove_exact_form()

  MeshLib::CHodgeDecomposition::_compute_coexact_form()

  MeshLib::CHodgeDecomposition::_remove_coexact_form()

5. Finish your code in your IDE

- You need to modify the file: WedgeProduct.h
- search for comments

$$//insertyourcodehere$$

  and insert your code
- Modify

  double CWedgeOperator::wedge_product()

  double CWedgeOperator::wedge_star_product()

6. Run the executable program

Command line:

HodgeDecomposition.exe closed_mesh.m open_mesh.m texture_image.bmp

All the data files are in the data folder, all the texture images are in the textures folder.