Geometry, Imaging and Computing
Volume 2, Number 2, 109–131, 2015

Hooke's Optimization for 3D triangular mesh

Hei Long Chan, Ho Yeung Hung, and Lok Ming Lui

A new framework for mesh optimization, the Filtered Hooke's Optimization, is proposed. With the notion of the elasticity theory, the Hooke's Optimization is developed by modifying the Hooke's law, in which an elastic force is simulated on the edges of a mesh so that adjacent vertices are either attracted to each other or repelled from each other, so as to regularize the mesh in terms of triangulation. A normal torque force is acted on vertices to guaranteed smoothness of the surface. In addition, a filtering scheme, called the Newtonian Filtering, is proposed as a supplementary tool for the proposed Hooke's Optimization to preserve the geometry of the mesh. Numerical simulations on meshes with different geometry indicate an impressive performance of our proposed framework to significantly improves the mesh triangulation without noteworthy distortions of the mesh geometry.

1. Introduction

Thanks to many efficient algorithms for its construction, 3D triangular mesh has become a widely used tool in different areas including computer visions, animations, video games, medical image analysis, etc. Nevertheless, due to either inappropriate vertex sampling or limit of the scanning hardware, the mesh may be of bad quality, consisting of many sharp triangles or unnatural spacing of vertices, for instance. However, not only for display purpose, mesh quality is also an important factor for precise numerical simulation and accurate analysis. Therefore, it is of interest to have an effective way to enhance the quality of the mesh, which is the goal of mesh optimization.

Theoretically, there are two directions to optimize a mesh. One is to perform down-sampling on the mesh to remove unnecessary vertices while preserving the geometry of the mesh, so that the data size can be decreased without much loss in quality. And the other one concentrates at repositioning the vertices of the mesh so that the quality of the mesh can be enhanced without much increase in data size. In this paper, we follow the latter route and develop a robust algorithm for mesh quality improvement.

Literature has provided a broad variety of models for enhancing mesh quality. However, not much of them has investigated the use of physics, the elasticity theory in particular, to solve the problem. In this work, the mesh optimization model is formulated using the Hooke's law from the elasticity theory. An elastic force based on the Hooke's law is defined on each edge of the mesh. Corresponding to an user-prescribed optimal edge length, the force attracts two adjacent vertices if the current edge length is too long, and repels the two vertices if the current edge length is too short.

On one hand, the elasticity theory gives a wise and effective optimization model. But on the other hand, direct implementation of the theory may trigger undesirable situations such as flipping of triangles, due to numerical error from discretization of the evolution process. Existing models based on the elasticity theory avoid those situations mainly by carefully adjusting the time step factor for the evolution process. However, modification to the evolution formula itself is not commonly considered.

Therefore, it is the contribution of our work to re-formulate the mesh optimization model by a modified version of the Hooke's law, which favors numerical implementation and no longer ruins the geometry of the surface. Moreover, through one another modification to simulate a torque force on each vertex, smoothness of the surface is also well-preserved. Finally, a filtering scheme called the Newtonian Filtering is developed based on the high-dimensional Newton's method to further preserve the local geometry of the optimized mesh. Combining the Hooke's Optimization and the Newtonian Filtering gives rise to our proposed model, the Filtered Hooke's Optimization, which significantly enhance the mesh quality while both smoothness and local geometry of the surface are greatly preserved. To this end, numerical experiments make evidence that our algorithm is very effective to improve the mesh quality.

Our model has at least two applications. Firstly, if the optimal edge length is prescribed as a global constant to all edges, our algorithm can sharply regularize the mesh such that all triangles converge to be equilateral. This supports further surface analysis and numerical simulation on the mesh. Secondly, the optimal edge length can vary among each edge, and hence our algorithm provides high flexibility on editing the mesh, provided that the prescribed edge length is an admissible one. This can be an useful tool for animation and medical image simulation.

The paper is organized as follows: In section 2, previous work on mesh optimization will be reviewed. Our model will be explained in detail in section 3 and the numerical implementation issue will be discussed in section 4. Experiment results will be demonstrated in section 5. Finally, a short conclusion will be presented in section 5.

2. Previous work

Different approaches have been employed to optimize a mesh. Literally, there are two main streams in mesh optimization. One stream aims at minimizing the sample size (i.e. the number of vertices) while preserving the important features including global and local geometry of the mesh, so that the storage requirement of the mesh file is minimized and data processing on the mesh can be conducted more efficiently. Moreover, through re-triangulating the mesh using remaining vertices only, triangulation quality may be improved as some vertices contributing to bad triangles may be removed. In [1], an algorithm is proposed to minimize an energy function which models the competing desires of conciseness of representation and fidelity to the data. Through their algorithm, meshes can be simplified to have only about 10% vertices remaining without affecting the global geometry of the meshes. And, in [2], iterative process of vertex repositioning is governed by a graph cut algorithm examining global combinations of local candidates. Moreover, in [3], the mesh optimization problem is formulated as a variational geometric partitioning problem. Using their approach, distortion error of the approximated mesh from the initial mesh is driven down through repeated clustering of faces into best-fitting regions. On the other hand, in [4], a geodesic-distance based model is developed to generate quality surface meshes which closely approximate the input. Number of vertices can be significantly reduced if there are many redundant vertices in the mesh.

While the first stream of mesh optimization assumes the existence of unnecessary vertices and tries to eliminate them by different means, the fundamental of our work is that the mesh is already simplified enough to preserve fine local geometry. Hence, we do not consider the need to reduce the number of vertices from the mesh.

Another stream of mesh optimization seeks for enhancing mesh quality, which may refer to different measurement according to context, with the given vertices. Those approaches are much closer to the foundation of our work. In [5], "dual mesh" is taken in use and vertex positions are optimized by minimizing a quadratic energy measuring the deviation of the mesh normals from the implicit surface normals computed at the vertices of the modified dual mesh. Curvature distortion is minimized in the optimization process. Also, in [6], a feature preserving smoothing model is developed which is guided by the uniformly weighted Laplacian and the discrete mean curvature normal. In their approach, edge vertices and corner vertices are taken as feature constraints that need to be fixed throughout the optimization process which is based on the Laplacian smoothing. And, in

[7], a non-iterative scheme based on the Laplacian smoothing is presented to smooth and improve the mesh while preserving certain prescribed features. Some existing works take the condition number shape measure, a measure derived from the Jacobian matrix of certain triangle mapping, as a quantity for mesh quality. In [8], mesh quality is enhanced in the sense of the condition number shape measure by repositioning vertices so that they are left on the original mesh faces and close to their original locations. On the other hand, mesh quality can be measured according to the shape of the triangles. In [9], mesh quality is enhanced in terms of the ratio of radius of inscribed and circumscribed circles. An optimizing scheme based on optimal delaunay triangulation is built to minimize the interpolation error in weighted L^1 norm. Moreover, in [10], the quality of vertex repositioning is measured by local volume distortion. In their algorithm, sharp features are automatically detected, and then a smoothing scheme based on the null-space smoothing is performed to optimize the mesh while minimizing the volume distortion. In [11, 12], adaptive remeshing algorithms are proposed to adjust the size of elements (i.e. faces) to resolve the relevant scales by minimizing energies raised from physics.

In this work, mesh quality is measured in terms of edge length. After prescribing an optimal edge length, our algorithm iteratively repositions the vertices so that edge lengths converge to the prescribed ones, while curvature distortions and surface geometry variations are minimized within the process.

3. Proposed model

Given an oriented mesh $M_0 = (V_0, E, F)$ and a prescribed optimal edge length L, our task is to build an iterative model to obtain a series of modified meshes $M_t = (V_t, E, F)$, so that the edge length $l_{e^{i,i^*}}$ of each edge $e^{i,i^*} \in E$ satisfies $l_{e^{i,i^*}} \to L$.

3.1. Hooke's Optimization – modification of the Hooke's law

The Hooke's law from the elasticity theory of physics hints the solution to accomplish the task. Below is the statement of the Hooke's law.

Theorem 1 (The Hooke's law)**.** *Given a spring S with stiffness coefficient $k_S > 0$. Suppose S has its one end attached to some fixed object, the restoring force F exerted by the spring on its free end for a displacement X satisfies*

$$F = -k_S X. \tag{1}$$

The direction of the force is opposite to the displacement X. That means when the spring is stretched, the force tends to extend it back and vice versa. The force is zero if and only if the displacement X is zero, that is, the spring restores its equilibrium state. The Hooke's law provides an intuitive formulation of our algorithm to take each edge as a spring, and the equilibrium state of each spring is taken as the user-prescribed optimal edge length L.

As a direct implementation of the Hooke's law, for each pair of adjacent vertices $v^i = v^i(t)$, $v^{i^*} = v^{i^*}(t)$, the force on v^i due to v^{i^*} is defined by:

$$F^{i,i^*}_{string} := \frac{(v^{i^*} - v^i)}{||v^{i^*} - v^i||} \times (||v^{i^*} - v^i|| - L), \tag{2}$$

and therefore, the combined force F^i_{string} on v^i by all its adjacent vertices is:

$$F^i_{string} := \sum_{\{1 \leq i^* \leq |V| : e^{i,i^*} \in E_t\}} F^{i,i^*}_{string}, \tag{3}$$

The term $||v^{i^*} - v^i|| - L$ in equation (2) corresponds to the displacement term X in the Hooke's law. The stiffness coefficient k_S is taken to be constant 1, and the fraction $\frac{(v^{i^*} - v^i)}{||v^{i^*} - v^i||}$ is simply the direction of the force.

As similar idea was proposed in [11, 12] to reposition the vertex by minimizing the spring energy $\sum_E (l_{e^{i,i^*}} - L)^2$, however, in practice, even if $||v^{i^*} - v^i||$ is too small, it may still happens that the total force F^i_{string} push v^i towards v^{i^*} due to the forces given by other adjacent vertices of v^i. As a result, flipping may occur which is undesirable. Traditional solutions turn to a robust control on the time step to avoid the situation. In this work, instead, we solve the problem by modifying the Hooke's law as in the following:

$$F^{i,i^*}_{string} := \begin{cases} \dfrac{(v^{i^*} - v^i)}{||v^{i^*} - v^i||} \times (||v^{i^*} - v^i|| - L) & \text{if } ||v^{i^*} - v^i|| \geq L \\ \dfrac{(v^{i^*} - v^i)L}{||v^{i^*} - v^i||^2} \times (||v^{i^*} - v^i|| - L) & \text{if } ||v^{i^*} - v^i|| < L \end{cases}, \tag{4}$$

and the forces F^{i,i^*}_{string} are summed up as in the equation (3). By modifying the stiffness coefficient from constant 1 to $\frac{L}{||v^{i^*} - v^i||}$ when $||v^{i^*} - v^i|| < L$, the force F^{i,i^*}_{string} has its magnitude inversely proportional to $||v^{i^*} - v^i|| - L$ when $||v^{i^*} - v^i||$ is small. Therefore, flipping barely happens and hence the above modification greatly improves the consistency of our algorithm.

3.2. Hooke's Optimization – smoothness regularization

The second feature of the proposed Hooke's Optimization is a smoothness regularization on the mesh. By acting only the force (3), $l_{e^{i,i^*}}$ converges to L but smoothness of the surface may be lost within the iteration. Therefore, some additional force is needed to preserve the smoothness of the surface. One solution turns to the normal torque force acting on the faces of the mesh, which is described in the following.

For each face f^j of M_t, denote by E_j the set of the three edges composing f^j. And for any two adjacent faces f^j, f^{j^*}, let $\theta_{j,j^*} \in [0,\pi)$ be the angle between the two unit inward normal vectors n^j_{face} and $n^{j^*}_{face}$ on the two faces respectively:

$$(5)\qquad < n^j_{face}, n^{j^*}_{face} >= \cos(\theta_{j,j^*}).$$

Let the common edge between these two faces be $e^{i,i^*} = E_j \cap E_{j^*}$. The force on the unit normal vector n^j due to the unit normal vector n^{j^*} is defined to be:

$$(6)\quad n^{j,j^*} = \begin{cases} \dfrac{(n^{j^*}_{face} - n^j_{face})}{\sin(\theta_{j,j^*})} \times \dfrac{\theta_{j,j^*}}{\cos(\theta_{j,j^*})+1} \times \dfrac{1}{d_{j,j^*}} & \text{if } \theta_{j,j^*} \in (0,\pi) \\ 0 & \text{if } \theta_{j,j^*} = 0 \end{cases},$$

where d_{j,j^*} is the distance between the centers of the two faces f^j, f^{j^*}, defined through the midpoint $m^{i,i^*} := \frac{v^i+v^{i^*}}{2}$ of the common edge e^{i,i^*}:

$$(7)\quad d_{j,j^*} := \left\| \frac{v^{f^j_1} + v^{f^j_2} + v^{f^j_3}}{3} - m^{i,i^*} \right\| + \left\| \frac{v^{f^{j^*}_1} + v^{f^{j^*}_2} + v^{f^{j^*}_3}}{3} - m^{i,i^*} \right\|.$$

Hence, the force on the normal vector n^j_{face} by all the normal vectors of adjacent faces is given by:

$$(8)\qquad n^j := \sum_{\{1 \le j^* \le |F| : E_j \cap E_{j^*} \neq \emptyset\}} n^{j,j^*}.$$

For a given $j \in [1, |F|]$, the force n^j on the unit normal vector n^j_{face} would produce a torque $T^j = n^j_{face} \times n^j$ that act to rotate the face f^j as a whole. For a given vertex v^i on the face f^j, the force acted on v^i due to torque T^j is given by:

$$\begin{aligned} F^{i,j}_{torque} &:= T^j \times \left(v^i - \frac{v^{f^j_1} + v^{f^j_2} + v^{f^j_3}}{3} \right) \\ &= \left(n^j_{face} \times n^j \right) \times \left(v^i - \frac{v^{f^j_1} + v^{f^j_2} + v^{f^j_3}}{3} \right) . \end{aligned} \tag{9}$$

and so the total force acted on a given vertex $v^i \in V$ due to torque action is given by:

$$F^i_{torque} := \sum_{\{1 \le j \le |F| : i \in I_j\}} F^{i,j}_{torque}, \tag{10}$$

where $I_j := \{f^j_1, f^j_2, f^j_3\}$.

3.3. The Newtonian Filtering

Through the Hooke's Optimization process, vertices are repositioned so that edge lengths are optimized while smoothness of the surface is guaranteed. However, one issue remained to be solved is that, as the vertices are repositioned, it is not guaranteed that they still lie on the surface that the original mesh M_0 is approximating. As a result, (local) geometry may be ruined. In order to keep the resultant meshes M_t closely resembling the original surface, a filtering scheme is needed such that after each iteration step, the vertices are projected back on M_0.

To do this, we make use of the thin plate spline function [13]. A function $\psi : \mathbb{R}^3 \to \mathbb{R}$ is called a thin plate spline function if part of its zero level set $\psi^{-1}(0) = \{x \in \mathbb{R}^3 : \psi(x) = 0\}$ approximately represents a surface.

Let n^i_{vertex} be the unit inward normal vector at v^i. The thin plate spline function ψ needs to satisfy

$$\psi(y^i) = d_i, \text{ for all } 1 \le i \le 2|V|, \tag{11}$$

and locally minimizes the harmonic energy

$$\int_{\mathbb{R}^3} |\Delta \psi|^2 dx = \int_{\mathbb{R}^3} |\psi_{xx} + \psi_{yy} + \psi_{zz}|^2 dx, \tag{12}$$

where y_i and d_i are defined as

$$y^i = \begin{cases} v^i & \text{if } 1 \leq i \leq |V| \\ v^i + \rho \cdot n^i_{vertex} & \text{if } |V| < i \leq 2|V| \end{cases}, \tag{13}$$
$$d_i = \begin{cases} 0 & \text{if } 1 \leq i \leq |V| \\ 1 & \text{if } |V| < i \leq 2|V| \end{cases},$$

and ρ is a user-prescribed parameter, which is chosen according to the shape of the input M_0.

To locally minimize the harmonic energy, ψ has to satisfy both of the following conditions:

$$\begin{cases} \int_{\mathbb{R}^3} |\Delta\psi|^2 dx < \infty \\ \int_{\mathbb{R}^3} |\Delta\psi|^2 dx \leq \int_{\mathbb{R}^3} |\Delta(\psi + \delta\psi)|^2 dx \end{cases}, \tag{14}$$

for any $\delta\psi \in C_0^\infty(\mathbb{R}^3)$ with $\delta\psi(y^i) = 0$ for all $1 \leq i \leq 2|V|$.

Using integration by parts, it could be shown that, a necessary (but not sufficient) condition for ψ to satisfy conditions (14) is that ψ satisfies the biharmonic equation:

$$\Delta^2\psi = \Delta(\Delta\psi) = 0. \tag{15}$$

Once the thin plate spline function ψ is constructed explicitly, the high dimensional Newton's method can be applied. Given a point $v^i = v^i(t)$ in the point set V_t, let $v^{i,0} := v^i$ be the initial point for the Newton's method. For any $v^{i,k} \in \mathbb{R}^3$ for $k \geq 0$, the high dimensional Newton's method writes:

$$v^{i,k+1} = v^{i,k} - \frac{\psi(v^{i,k})}{\|\nabla\psi(v^{i,k})\|^2}\nabla\psi(v^{i,k}). \tag{16}$$

and its converges once the norm difference $\left\|v^{i,k+1} - v^{i,k}\right\| < \epsilon$ for some user-prescribed error tolerance ϵ is reached at step k.

The whole process from constructing the thin plate spline function ψ to solving for the zero level set of ψ using the high dimensional Newton's method is called the Newtonian Filtering. Since the norm difference $\left\|v^i(t) - v^i(t+1)\right\|$ is small in each step of the Hooke's optimization, so it is reasonable to neglect the convergence criterion of the Newton's method and assume that the Newton's method always converge for every vertices of M_t for all t.

3.4. The filtered Hooke's Optimization

Applying the Newtonian Filtering, vertices are repositioned again to retrieve the local geometry of the surface. However, edge lengths are varied by it and so they may diverge from the prescribed optimal edge length L. Hence, the Hooke's Optimization is needed once again to retrieve the optimal again and after that the Newtonian Filtering is applied again to retrieve the geometry of the surface, and so on and so worth. Indeed, the spirit of the proposed Filtered Hooke's Optimization is the alternative composition of the Hooke's Optimization and the Newtonian Filtering.

Given M_0, L, and a prescribed tolerance $d_{tolerance}$ to control the convergence of the algorithm, let $D = max_E|l_{e^{i,i^*}}|$, the Filtered Hooke's Optimization alternatively applies the Hooke's Optimization to M_t to get an edge-length-optimized mesh M_{t+1} and applies the Newtonian Filtering to M_{t+1} to get a geometry-retrieved mesh M^*_{t+1}, until $D < d_{tolerance}$ at some time t. Once converged, the resultant mesh M^* is a mesh with edge length being optimized while local geometry and smoothness of the surface is preserved. To this end, the Filtered Hooke's Optimization has fulfilled all the required criterions to solve the mesh optimization problem.

4. Implementation

In this section, the evolution equation and the corresponding numerical implementation of the algorithm is to be discussed in detail.

4.1. Evolution equation of the Hooke's Optimization

An adaptive forward time scheme is adopted to numerically simulate our algorithm. The evolution of the vertex set $V_t = \{v^i(t) : 1 \leq i \leq |V|\}$ is governed by the following set of equations:

$$(17) \qquad \begin{cases} \dfrac{dv^i(t)}{dt} = u^i(t) \\ \dfrac{du^i(t)}{dt} = -\tau u^i(t) + \dfrac{\lambda F^i_{string}(t) + \mu F^i_{torque}(t)}{m} \end{cases}$$

with the initial conditions:

$$(18) \qquad \begin{cases} v^i(0) = v^i & \text{for } 1 \leq i \leq |V| \\ u^i(0) = \vec{0} & \text{for } 1 \leq i \leq |V| \end{cases}$$

where τ is a friction coefficient, λ is the string tension coefficient, μ is the torque coefficient and $m = 1/|V|$ simulates the mass for each vertex.

4.2. Numerical implementation of the evolution equation

Let v_n^i, u_n^i, $F_{string}^{i,n}$ and $F_{torque}^{i,n}$ be the numerical solutions of $v^i(t), u^i(t)$, $F_{string}^i(t)$ and $F_{torque}^i(t)$ at the n-th time step $t = t_n$ respectively. Define

$$l_n^{edge} := \min\left\{\left\|v_n^i - v_n^{i^*}\right\| : e^{i,i^*} \in E_n\right\}, \tag{19}$$

$$l_n^{velocity} := \max\left\{\left\|u_n^i\right\| : 1 \leq i \leq |V|\right\}, \tag{20}$$

$$l_n^{force} := \max\left\{\left\|\lambda F_{string}^{i,n} + \mu F_{torque}^{i,n}\right\| : 1 \leq i \leq |V|\right\}. \tag{21}$$

The evolutions of v_n^i and u_n^i are governed by the system:

$$\begin{cases} v_{n+\frac{1}{2}}^i = v_n^i + u_n^i \cdot \dfrac{1 - e^{-\tau \cdot dt_n}}{\tau} + \dfrac{\lambda F_{string}^{i,n} + \mu F_{torque}^{i,n}}{m} \cdot \dfrac{{dt_n}^2}{2} \\ u_{n+1}^i = e^{-\tau \cdot dt_n} \cdot u_n^i + \dfrac{\lambda F_{string}^{i,n} + \mu F_{torque}^{i,n}}{m} \cdot dt_n \end{cases}, \tag{22}$$

with the initial condition

$$v_0^i = v^i(0), u_0^i = 0 \text{ for } 1 \leq i \leq |V|. \tag{23}$$

Here, the force terms $F_{string}^{i,n}$ and $F_{torque}^{i,n}$ are computed using $\{v_n^i : 1 \leq i \leq |V|\}$ and $\{f^j : 1 \leq j \leq |F|\}$ as in equation (3) and (10) respectively.

4.3. Time step of the Hooke's Optimization

Starting with $t_0 = 0$, the time step t_n increases with $t_{n+1} = t_n + dt_n$. The time step dt_n is computed using a two step scheme motivated by the backtracking line search scheme [14]. Firstly, a base time step ω_n is computed as the largest $\omega_n > 0$ that satisfies the quadratic inequality

$$\frac{l_n^{force}}{2m} \cdot {\omega_n}^2 + l_n^{velocity} \cdot \omega_n \leq l_n^{edge}, \tag{24}$$

then dt_n is given by scaling ω_n through an adaptive time step factor ϵ_n:

$$dt_n = \omega_n \cdot \epsilon_n. \tag{25}$$

This ϵ_n is used to ensure that the deformation of M_t between two consecutive times $t = t_n$ and $t = t_{n+1}$ is small. An adaptive iterative scheme is used to find an appropriate ϵ_k. With the convention $\epsilon_{-1} := \frac{1}{2}$, the evolution of ϵ_n is given by

$$\epsilon_n = \epsilon_{n-1} \cdot 2^{z_n}, \tag{26}$$

where the index z_n is chosen to be the largest $z_n \in \mathbb{Z}$ with $z_n \leq 1$ such that all the following constraints are fulfilled:

- $\delta_{angle} < n_{face}^{j,n} \cdot n_{face}^{j,n+\frac{1}{2}} \leq 1$ (angle distortion)
- $\kappa_{j,n} \neq 0$, $\kappa_{j,n+\frac{1}{2}} \neq 0$ and $\left|1 - \frac{\kappa_{j,n+\frac{1}{2}}}{\kappa_{j,n}}\right| \leq \delta_{expansion}$ (area expansion)
- $\left\|v_{n+\frac{1}{2}}^i - v_n^i\right\| < \eta_n$ (vertex displacement)
- $\delta_{bound} < \kappa_{j,n+1}$ (area lower bound)

with $n_{face}^{j,n}$ and $\kappa_{j,n}$ being the unit inward normal vector and the area of the j-th face f^j in the mesh M_{t_n} respectively. The four threshold parameters, δ_{angle}, $\delta_{expansion}$, η_n, and δ_{bound} are some constants to be prescribed in the beginning of the iteration.

4.4. Numerical construction of the thin plate spline function

After one Hooke's Optimization iteration, a mesh $M_{t_{n+1}}$ is obtained such that edge lengths of $M_{t_{n+1}}$ are close to L, but the vertices $v_{t_{n+1}}^i$ may not lie on the original surface. Hence, the Newtonian Filtering is employed to project the vertices back on the original surface.

The Newtonian Filtering starts with the construction of the thin plate spline function. Let $\phi_y : \mathbb{R}^3 \to \mathbb{R}$ be the radial basis functions defined by

$$\phi_y(x) := \begin{cases} \|x-y\|^2 \log\left(\|x-y\|\right) & \text{if } x \neq y \\ 0 & \text{if } x = y \end{cases}, \tag{27}$$

then ϕ_y satisfies both of the following conditions:

$$\begin{cases} \Delta\phi_y(x) = 6\log\left(\|x-y\|\right) + 5 & \text{for } x \neq y \\ \Delta^2\phi_y(x) = \frac{6}{\|x-y\|^2} & \text{for } x \neq y \end{cases}. \tag{28}$$

Let $p_{b,c} : \mathbb{R}^3 \to \mathbb{R}$ be the affine function defined as:

$$p_{b,c}(x) := b \cdot x + c, \tag{29}$$

then ψ can be written as a linear combination of ϕ_{y^i}'s and p:

$$\psi(x) = \sum_{i=1}^{2|V|} a_i \cdot \phi_{y^i}(x) + p_{b,c}(x). \tag{30}$$

The coefficients $a_i, b = [b_1, b_2, b_3]$ and c are obtained by solving the $(N+4) \times (N+4)$ linear system:

$$\begin{bmatrix} r_{1,1} & r_{1,2} & \cdots & r_{1,N} & y_1^1 & y_2^1 & y_3^1 & 1 \\ r_{2,1} & r_{2,2} & \cdots & r_{2,N} & y_1^2 & y_2^2 & y_3^2 & 1 \\ \cdot & \cdot & \cdots & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdots & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdots & \cdot & \cdot & \cdot & \cdot & \cdot \\ r_{N,1} & r_{N,2} & \cdots & r_{N,N} & y_1^N & y_2^N & y_3^N & 1 \\ y_1^1 & y_1^2 & \cdots & y_1^N & 0 & 0 & 0 & 0 \\ y_2^1 & y_2^2 & \cdots & y_2^N & 0 & 0 & 0 & 0 \\ y_3^1 & y_3^2 & \cdots & y_3^N & 0 & 0 & 0 & 0 \\ 1 & 1 & \cdots & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \cdot \\ \cdot \\ \cdot \\ a_N \\ b_1 \\ b_2 \\ b_3 \\ c \end{bmatrix} = \begin{bmatrix} d_1 \\ d_2 \\ \cdot \\ \cdot \\ \cdot \\ d_N \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \tag{31}$$

where $N = 2|V|$ and $r_{i,j}$ is defined by

$$r_{i,j} := \begin{cases} \left\|y^i - y^j\right\|^2 \log\left(\left\|y^i - y^j\right\|\right) & \text{if } y^i \neq y^j \\ 0 & \text{if } y^i = y^j \end{cases}. \tag{32}$$

The first N equations of the linear system are used to ensure that

$$\psi(y^i) = d_i \text{ for } 1 \leq i \leq N, \tag{33}$$

and the last 4 equations are used to ensure that

$$\int_{\mathbb{R}^3} |\Delta\psi|^2 dx < \infty. \tag{34}$$

Hence, by solving the linear system (31) by LU decomposition, the desired thin plate spline function can be obtained easily.

4.5. Numerical implementation of the Newtonian Filtering

Given the thin plate spline function ψ as in (30), $\nabla\psi(v_{n+1}^{i,k})$ can be evaluated as

$$\nabla\psi(v_{n+1}^{i,k}) = \sum_{j=1}^{N} a_i \cdot \nabla\phi_{y^j}(v_{n+1}^{i,k}) + b. \tag{35}$$

Hence, the high dimensional Newton's method writes:
(36)

$$v_{n+1}^{i,k+1} = v_{n+1}^{i,k} - \frac{\sum_{j=1}^{N} a_i \cdot \phi_{y^j}(v_{n+1}^{i,k}) + b}{\left\|\sum_{j=1}^{N} a_i \cdot \nabla\phi_{y^j}(v_{n+1}^{i,k}) + b\right\|^2} \left(\sum_{j=1}^{N} a_i \cdot \nabla\phi_{y^j}(v_{n+1}^{i,k}) + b\right).$$

Here, an error tolerance parameter ε is prescribed, so the above iteration would stop at $v_{n+1}^{i,k}$ once $|\psi(v_{n+1}^{i,k})| \leq \varepsilon$ is found.

Define the Newtonian Filtering function $\Psi_M : V \to \bar{\mathbb{R}}^3$ by

$$\Psi_M(v_{n+1}^{i}) := \begin{cases} v_{n+1}^{i,K} & \text{if } \{k : |\psi(v_{n+1}^{i,k})| \leq \varepsilon\} \neq \emptyset \\ \infty & \text{if } \{k : |\psi(v_{n+1}^{i,k})| \leq \varepsilon\} = \emptyset \end{cases}, \tag{37}$$

where $K := \min\{k : |\psi(v_{n+1}^{i,k})| \leq \varepsilon\}$, then $\Psi_M(v_{n+1}^{i}) = v_{n+1}^{i,K}$ is the projection of v_{n+1}^{i} on M_0. The parameter ε should be chosen such that

$$\varepsilon \cdot \rho < l_\tau$$

where l_τ is a prescribed error tolerance parameter which ensures $\Psi_M(v_{n+1}^{i})$ to be the zero level set of ψ.

By processing through the Newtonian Filtering, a truncated vertex set V_{n+1}^{*} and hence the mesh M_{n+1}^{*} is obtained which closely resembles the original surface.

4.6. The numerical algorithm

Algorithm 1: The Filtered Hooke's Optimization Algorithm

```
Input: Initial mesh $M_0$, Stop criterion $d_{tolerance}$, Optimal length
       parameter $L$
Output: Optimized mesh $M^*$
Initialize $D = max_E|l_{e^{i_1,i_2}} - L|$, $t = 0$; repeat
    $M_{t+1}$ = Hooke's Optimization($M_t, d_{tolerance}$)
    $M_{t+1}^{*}$ = Newtonian Filtering($M_{t+1}$)
    $D \leftarrow max_{E_{t+1}}|l_{e^{i_1,i_2}} - L|$; $t \leftarrow t + 1$
until $D < d_{tolerance}$;
```

where *Hooke's Optimization* and *Newtonian Filtering* are two algorithms numerically given as in algorithm (2) and algorithm (3) respectively.

```
Algorithm 2: The Hooke's Optimization Algorithm
  Input: Mesh $M_t$, stop criterion $d_{tolerance}$
  Output: Mesh $M_{t+1} = (V_{t+1}, E, F)$
Initialize $n = 0$, $t_0 = t$, $m = 1/|V_t|$, $v_0^i = v^i(0)$, $u_0^i = 0$; repeat
    Compute $F^{i,n}_{string}$ and $F^{i,n}_{torque}$ as in (3) and (10) respectively.
    Compute
        $\begin{cases} l_n^{edge} = \min\left\{ \left\| v_n^i - v_n^{i^*} \right\| : e^{i,i^*} \in E_n \right\} \\ l_n^{velocity} = \max\left\{ \left\| u_n^i \right\| : 1 \le i \le |V| \right\} \\ l_n^{force} = \max\left\{ \left\| \lambda F^{i,n}_{string} + \mu F^{i,n}_{torque} \right\| : 1 \le i \le |V| \right\} \end{cases}.$
    where $F^{i,n}_{string}$ and $F^{i,n}_{torque}$ are computed as in (3) and (10) respectively.
    Compute
        $dt_n = \omega_n \cdot \epsilon_n,$
    Compute
        $\begin{cases} v^i_{n+\frac{1}{2}} = v_n^i + u_n^i \cdot \dfrac{1 - e^{-\tau \cdot dt_n}}{\tau} + \dfrac{\lambda F^{i,n}_{string} + \mu F^{i,n}_{torque}}{m} \cdot \dfrac{{dt_n}^2}{2} \\ u^i_{n+1} = e^{-\tau \cdot dt_n} \cdot u_n^i + \dfrac{\lambda F^{i,n}_{string} + \mu F^{i,n}_{torque}}{m} \cdot dt_n \end{cases},$
    $n \leftarrow n + 1$
until $max_{E_{t_n}} |l_{e^{i_1,i_2}} - L| < d_{tolerance}$;
```

As for practical use, the parameters are set as $\gamma = 1000$, $\lambda = 2$, $\mu = 1$, $\delta_{angle} = 0.9$, $\delta_{expansion} = 0.01$, $\xi = 0.01$ and $\rho = 10^{-3}$. The stop criterions $d_{tolerance}$ and l_γ controls the accuracy of the algorithms. One can balance the accuracy over the convergence rate by modifying these two parameters.

5. Experimental results

In this section, numerical experiments are conducted to validate the effectiveness of our algorithm. In order to quantify the effectiveness of our algorithm, in each experiment, the minimum angle of the original mesh and of the optimized mesh, and the standard deviation of the edge length of the original mesh and that of the optimized mesh are computed and compared.

Algorithm 3: The Newtonian Filtering Algorithm

```
Input: Initial mesh $M_t$, normal vector coefficient $\rho$, error torelance
       parameter $l_\gamma$
Output: Optimized mesh $M_t^* = (V_t^*, E, F)$
Choose $\epsilon$ such that
    $$\epsilon \cdot \rho < l_\gamma$$
Initialize $v_t^{i,0} = v_t^i$, $k = 0$; repeat
    Solve the following sysmte by LU decomposition to obtain coefficients
    $a_i, b, c$:
    $$\begin{bmatrix} r_{1,1} & r_{1,2} & \cdots & r_{1,N} & y_1^1 & y_2^1 & y_3^1 & 1 \\ r_{2,1} & r_{2,2} & \cdots & r_{2,N} & y_1^2 & y_2^2 & y_3^2 & 1 \\ . & . & \cdots & . & . & . & . & . \\ . & . & \cdots & . & . & . & . & . \\ . & . & \cdots & . & . & . & . & . \\ r_{N,1} & r_{N,2} & \cdots & r_{N,N} & y_1^N & y_2^N & y_3^N & 1 \\ y_1^1 & y_1^2 & \cdots & y_1^N & 0 & 0 & 0 & 0 \\ y_2^1 & y_2^2 & \cdots & y_2^N & 0 & 0 & 0 & 0 \\ y_3^1 & y_3^2 & \cdots & y_3^N & 0 & 0 & 0 & 0 \\ 1 & 1 & \cdots & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ . \\ . \\ . \\ a_N \\ b_1 \\ b_2 \\ b_3 \\ c \end{bmatrix} = \begin{bmatrix} d_1 \\ d_2 \\ . \\ . \\ . \\ d_N \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix},$$
    where $N = 2|V|$, $r_{i,j}$ is defined by
    $$r_{i,j} := \begin{cases} \left\|y^i - y^j\right\|^2 \log\left(\left\|y^i - y^j\right\|\right) & \text{if } y^i \neq y^j \\ 0 & \text{if } y^i = y^j \end{cases}.$$
    and $y^i$ and $d_i$ are defined as in (13).
    Compute
    $$\begin{cases} \psi(v_t^{i,k}) = \sum_{j=1}^{2|V_t|} a_i \cdot \phi_{y^j}(v_t^{i,k}) + p_{b,c}(v_t^{i,k}) \\ \nabla\psi(v_t^{i,k}) = \sum_{j=1}^{2|V_t|} a_i \cdot \nabla\phi_{y^j}(v_t^{i,k}) + b \end{cases}.$$
    Compute
    $$v_t^{i,k+1} = v_t^{i,k} - \frac{\sum_{j=1}^N a_i \cdot \phi_{y^j}(v_t^{i,k}) + b}{\left\|\sum_{j=1}^N a_i \cdot \nabla\phi_{y^j}(v_t^{i,k}) + b\right\|^2} \left(\sum_{j=1}^N a_i \cdot \nabla\phi_{y^j}(v_t^{i,k}) + b\right).$$
    $k \leftarrow k + 1$
until $|\psi(v_t^{i,k})| < \epsilon$;
```

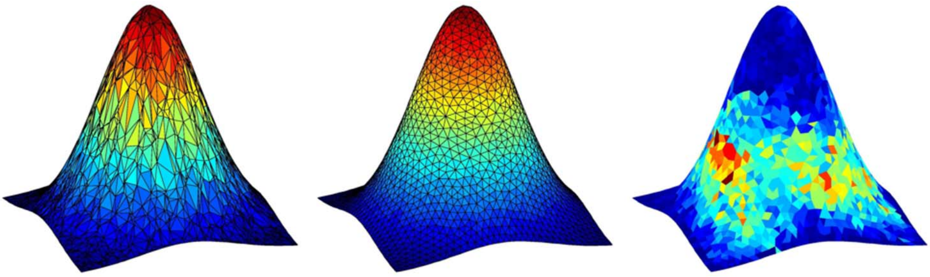


Figure 1: [Subject title: G1]. Initial mesh (left), Optimized mesh (middle), Vertex displacement map (right).

Table 1: Numerical analysis of the performance of the Filtered Hooke's Optimization algorithm on artificial data (rad. = radian, SD = standard deviation)

Mesh	G1	M1	M2	M3
No. of vertices	961	121	441	961
Greatest vertex displacement	0.1652	0.2318	0.1226	0.1143
Original least angle (rad.)	0.0018	0.0546	0.0270	0.0092
Optimized least angle (rad.)	0.5336	0.4140	0.5664	0.2159
Original edge length SD	0.0347	0.0777	0.4000	0.0280
Optimized edge length SD	0.0076	0.0226	0.0094	0.0071

As for an optimization of the mesh, the minimum angle is expected to increase and the standard deviation of the edge length is expected to decrease, so that the triangulation of the mesh is regularized.

5.1. A simple experiment

Firstly, a simple experiment using an artificial surface is examined. Let $z = g(x, y) = e^{-10((x-0.5)^2+(y-0.5)^2)}$, then z is a graph function of x, y. On the surface z, random points are selected as vertices and are triangulated afterwards. As a result, a mesh with many bad triangles are obtained. The Filtered Hooke's Optimization is applied to it and the result is shown in figure (1) and in table (1).

Clearly, the mesh edited by the Filtered Hooke's Optimization has significant quality enhancement from the initial mesh. While most of the vertices have large displacement as indicated, the shape of the hippocampus model is well preserved. The conclusion is also evident from the result in table (1). This shows that our algorithm can grealy improve the quality of a mesh without destroying its geometry.

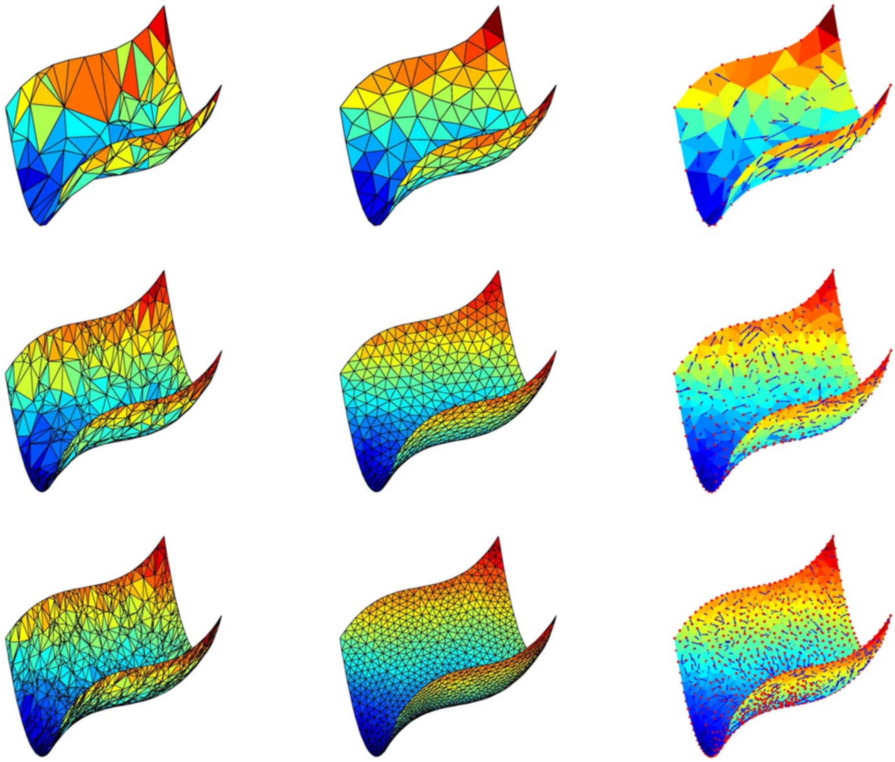


Figure 2: [Subject title: M1(top), M2(mid), M3(bot)]. Initial meshes (left), Optimized meshes (middle column), Vertex displacement maps (right).

5.2. A density experiment

In this experiment, the sensitivity of our algorithm to the density of the vertices is investigated. Consider the surface $z = g(x, y) = 2(x - 0.5)^2 + 2(y - 0.5)^3$ in the domain $[0, 1]^2$, three meshes representing the surface z is constructed in difference density. Random sampling of vertices are taken on the surface z to generate the meshes. Unnatural spacing of vertices hence appears on the meshes. The Filtered Hooke's Optimization is applied to them independently and the results are recorded in figure (2) and in table (1).

It can be seen from figure (2) that our algorithm enhances the qualities of the three meshes by around the same levels. This indicates that the effectiveness of our algorithm is density independent. No matter a coarse mesh or a dense mesh is given, our algorithm can stably enhances the quality of the mesh.

Table 2: Numerical analysis of the performance of the Filtered Hooke's Optimization algorithm on real data (rad. = radian, SD = standard deviation)

Mesh	Hippocampus	Face	Pear	Heart
No. of vertices	664	275	302	502
Greatest vertex displacement	4.4952	2.1469	3.8903	1.6358
Original least angle (rad.)	0.0031	0.1012	0.1507	0.1451
Optimized least angle (rad.)	0.6334	0.2615	0.6011	0.2519
Original edge length SD	0.6815	0.4403	1.2486	0.4247
Optimized edge length SD	0.2103	0.2656	0.2931	0.2061

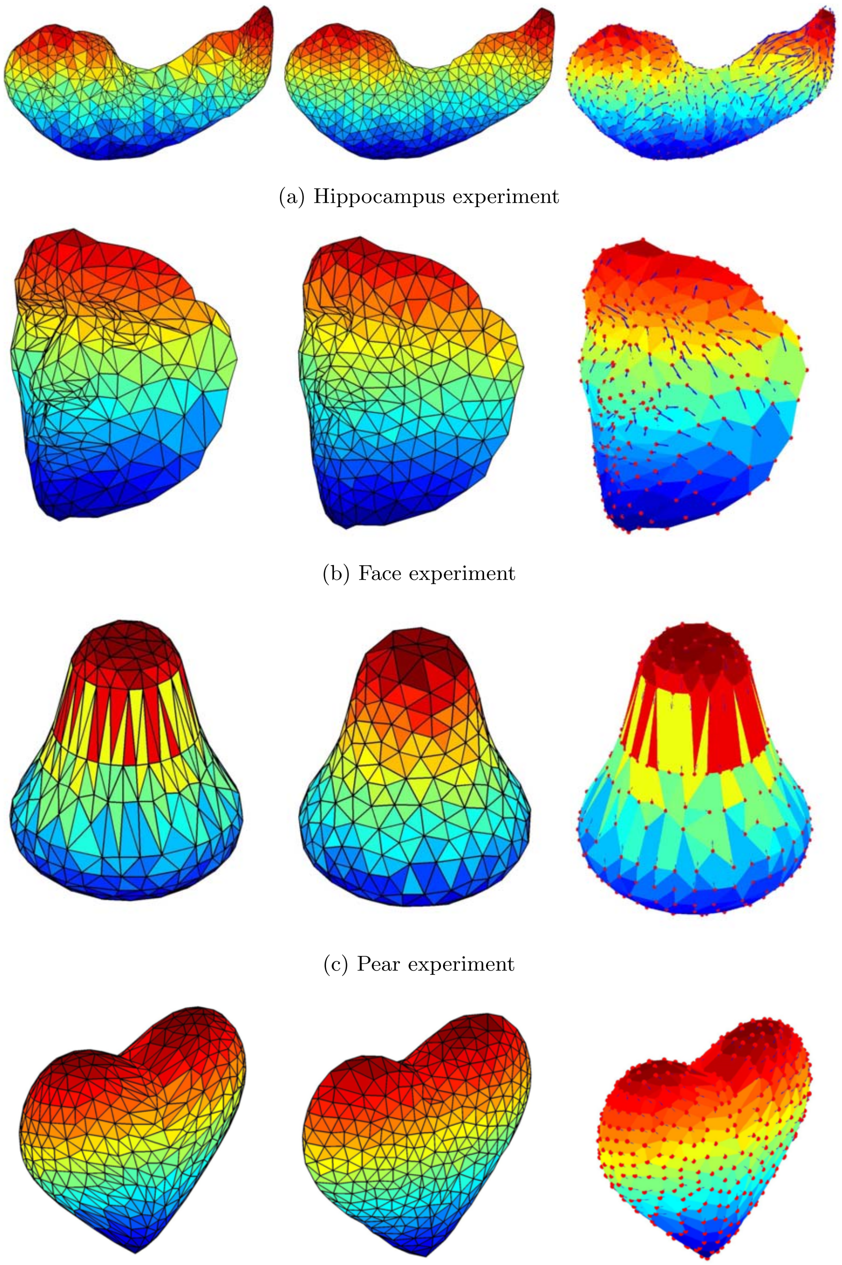
From table (1), in all of the experiments, our algorithm greatly increases the least angle on the meshes, while significantly decreases the standard deviation of the edge length of the meshes. Hence, the numerical result approves, once again, that our algorithm impressively regularize a given mesh.

5.3. Real data experiments

The effect of our model on real data is also investigated. Different objects including hippocampus of a patient, a face, a pear, and a heart-shaped mesh, are taken as input. However, due to different issues, the meshes are not well qualified in terms of vertex spacing. Sharp triangles appears on the meshes and they also contain triangles of highly fluctuating areas. Therefore, a quality enhancement is needed without affecting the geometry of the meshes, so that further numerical simulation or analysis can be performed on the optimized meshes. The Filtered Hooke's Optimization algorithm is applied to them and the result of our algorithm on the scanned meshes is recorded in figure (3) and in table (2).

From figure (3), it is obvious that the optimized meshes do not contain any unnatural spacings or sharp triangles. While great displacement of vertices are recorded as indicated in the figure, the geometry of the surfaces is highly preserved. Once again, it is evident that our model is practically effective.

As indicated in table (2), our algorithm greatly improves the triangulation of the meshes in terms of least angles and optimal edge length. As the least angles increase significantly, the optimized meshes no longer contain sharp triangles. And as the edge lengths has less fluctuation, the optimized meshes are highly regularized than the original one. To this end, the effectiveness of our algorithm in practical use is well approved.



(a) Hippocampus experiment

(b) Face experiment

(c) Pear experiment

(d) Heart experiment

Figure 3: Initial mesh (left), Optimized mesh (middle), Vertex displacement map (right).

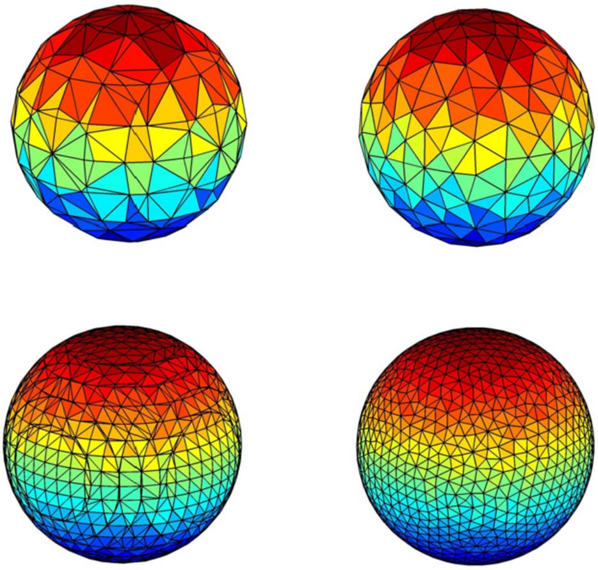


Figure 4: [Subject title: S1(top), S2(bot)] Initial meshes (left), Optimized meshes (right).

5.4. Geometry preservation of our algorithm

Once approving that our model does improves the quality of meshes, an important follow-up question is that, does our algorithm change the geometry of the given mesh? The goal of this experiment is to answer the question. This time, unit spherical meshes are taken as input. The Filtered Hooke's Optimization is applied to the meshes and curvature distortions are measured. The result is reported in figure (4) and table (3).

From table (3), while the Filtered Hooke's Optimization algorithm improves the quality of the spherical meshes, there are only negligible distortion in curvatures. This shows that our algorithm well preserves the geometry of the surface while optimizing the mesh quality in terms of triangulation.

6. Conclusion

A new framework for mesh optimization, the Filtered Hooke's Optimization, is developed by an alternatively composition of two proposed algorithms. The first one is the Hooke's Optimization, which is based on modifications to the Hooke's law from the elasticity theory of physics. By modifying the

Table 3: Numerical analysis of the performance of our algorithm on spherical meshes(rad. = radian, SD = standard deviation)

Mesh	S1	S2
No. of vertices	222	1146
Greatest vertex displacement	0.3229	0.2805
Original least angle (rad.)	0.1265	0.0147
Optimized least angle (rad.)	0.5423	0.5340
Original edge length SD	0.1129	0.0501
Optimized edge length SD	0.0509	0.0154
Original least mean curvature	0.9652	0.7745
Optimized least mean curvature	0.9471	0.9550
Original greatest mean curvature	1.0195	1.2407
Optimized greatest mean curvature	1.0415	1.0579
Original least Gaussian curvature	0.9316	0.5643
Optimized least Gaussian curvature	0.8966	0.9120
Original greatest Gaussian curvature	1.0394	1.4893
Optimized greatest Gaussian curvature	1.0847	1.1182

stiffness coefficient in the Hooke's law, an iterative scheme is developed such that an elasticity force is acted on each edge of the given mesh to either attract or repel adjacent vertices. Smoothness of the mesh is guaranteed by acting a normal torque force on faces of the mesh. Through the Hooke's Optimization, the input mesh is greatly regularized with edge lengths closely retrieve the prescribed optimal edge length. The second component of the Filtered Hooke's Optimization is the Newtonian Filtering, which is developed based on thin plate spline function and the high dimensional Newton's method, to repair the output mesh from the Hooke's Optimization to closely resembles the local geometry of the original surface. Combining the two algorithms, the Filtered Hooke's Optimization significantly regularize a given mesh with its smoothness and local geometry being well preserved. Numerical experiments validates that the algorithm greatly improves the quality of a mesh in terms of triangulation, which is measured by comparing the least angles and the statistics of vertex spacing of the input and output mesh. Lastly, numerical simulations also approves the ability of the Filtered Hooke's Optimization algorithm to practically preserve the geometry of a surface.

Acknowledegment

Lok Ming Lui is partially supported by HKRGC GRF (CUHK Project ID: 404612).

References

[1] Hugues Hoppe, et al. *Mesh optimization*, Proceedings of the 20th annual conference on Computer graphics and interactive techniques, ACM, 1993.

[2] Vincent Vidal, Christian Wolf, and Florent Dupont. *Combinatorial mesh optimization*, The Visual Computer, 28.5 (2012): 511–525.

[3] David Cohen-Steiner, Pierre Alliez, and Mathieu Desbrun. *Variational shape approximation*, ACM Transactions on Graphics (TOG), 23.3 (2004): 905–914.

[4] Oren Sifri, Alla Sheffer, and Craig Gotsman. *Geodesic-based surface remeshing*, Proc. 12th Intnl. Meshing Roundtable, 2003.

[5] Yutaka Ohtake, and Alexander G. Belyaev. *Dual/primal mesh optimization for polygonized implicit surfaces*, Proceedings of the seventh ACM symposium on Solid modeling and applications, ACM, 2002.

[6] Andrew Nealen, et al. *Laplacian mesh optimization*, Proceedings of the 4th international conference on Computer graphics and interactive techniques in Australasia and Southeast Asia, ACM, 2006.

[7] Ligang Liu, et al. *Non-iterative approach for global mesh optimization*, Computer-Aided Design, 39.9 (2007): 772–782.

[8] Rao V. Garimella, Mikhail J. Shashkov, and Patrick M. Knupp. *Optimization of surface mesh quality using local parametrization*, IMR, (2002): 41–52

[9] Long Chen, and Michael Holst. *Efficient mesh optimization schemes based on optimal Delaunay triangulations*, Computer Methods in Applied Mechanics and Engineering, 200.9 (2011): 967–984. MR2749047

[10] Xiangmin Jiao. *Volume and feature preservation in surface mesh optimization*, Proceedings of the 15th International Meshing Roundtable, Springer, Berlin, Heidelberg, 2006.

[11] Xiaoming Zheng, Anthong Anderson, John Lowengrub, and Vittorio Cristini. *Adaptive unstructured volume remeshing: Application to level-set simulations of multiphase flow*, Journal of Computational Physics, 208.2 (2005): 616–625. MR2144734

[12] Vittorio Cristini, Jerzy Bławzdziewicz, and Michael Loewenberg. *An adaptive mesh algorithm for evolving surfaces: simulations of drop*

breakup and coalescence, Journal of Computational Physics, 168.2 (2001): 445–463.

[13] Jean Duchon. *Splines minimizing rotation-invariant semi-norms in Sobolev spaces*, Constructive theory of functions of several variables, Springer, Berlin, Heidelberg, 1977. 85–100. MR0493110

[14] Larry Armijo. *Minimization of functions having Lipschitz continuous first partial derivatives*, Pacific Journal of mathematics, 16.1 (1966): 1–3. MR0191071

Hei Long Chan
Rm 222B, Lady Shaw Building
The Chinese University of Hong Kong
Hong Kong
E-mail address: hlchan@math.cuhk.edu.hk

Ho Yeung Hung
The Chinese University of Hong Kong
Hong Kong
E-mail address: yeung189@gmail.com

Lok Ming Lui
Rm 207, Lady Shaw Building
The Chinese University of Hong Kong
Hong Kong
E-mail address: lmlui@math.cuhk.edu.hk

Received 25 February 2015