Theoretical Foundation of the Stretch Energy Minimization for Area-Preserving Mappings

Mei-Heng Yueh
yue@ntnu.edu.tw
Department of Mathematics
National Taiwan Normal University

Abstract

The stretch energy is a fully nonlinear energy functional that has been applied to the numerical computation of area-preserving mappings. However, this approach lacks theoretical support and the analysis is complicated due to the full nonlinearity of the functional. In this paper, we provide a theoretical foundation of the stretch energy minimization for the computation of area-preserving mappings, including a neat formulation of the gradient of the functional, and the proof of the minimizers of the functional being area-preserving mappings. In addition, the geometric interpretation of the stretch energy is also provided to better understand this energy functional. Furthermore, numerical experiments are demonstrated to validate the effectiveness and accuracy of the stretch energy minimization for the computation of square-shaped area-preserving mappings of simplicial surfaces.

1 Introduction

An area-preserving mapping is also called an equiareal mapping or authalic mapping, which is a bijective mapping that preserves the area. In application aspects, area-preserving mappings could serve as parameterizations of surfaces in 3D space. It has been applied to processing issues in computer vision and graphics such as resampling, remeshing, and registration of surface mesh models [7, 12, 8]. Classical methods for computing area-preserving mapping includes stretch-minimizing method [11, 14], Lie advection method [22], optimal mass transportation method [6, 21, 13], and diffusion-based method [4].

In recent years, a series of efficient numerical algorithms [15, 13, 20, 17, 16, 15, 19] for the computation of parameterizations of surfaces and 3-manifolds have been well-developed based on minimizing nonlinear energy functionals of the form

\[ E(f) = \frac{1}{2} \text{trace} \left( \begin{bmatrix} f(v_1) & \cdots & f(v_n) \end{bmatrix} L(f) \begin{bmatrix} f(v_1)^T \\ \vdots \\ f(v_n)^T \end{bmatrix} \right), \]

where \( f : \mathcal{M} \to \mathbb{R}^d \) is a simplicial mapping defined on a simplicial \( d \)-complex with the image \( f(v_\ell) \) being a column vector in \( \mathbb{R}^d \), for \( \ell = 1, \ldots, n \), and \( L(f) \) is a Laplacian matrix with weights being dependent on \( f \). In particular, the area-preserving mapping is computed by minimizing the stretch energy [13, 20]. Noting that the stretch energy functional is fully nonlinear, still, the minimizer can be effectively obtained by iteratively approximating the critical point of the functional, and the resulting mapping preserves the area well. Compared to other state-of-the-art algorithms, stretch energy minimization for area-preserving mappings has advantages in both effectiveness and accuracy. However, this approach still lacks a theoretical foundation.
Contribution  In this paper, we establish the theoretical foundation of the stretch energy minimization for the computation of area-preserving mappings. The contributions of this paper are three-fold.

(i) We derive a neat gradient formula of the stretch energy functional that is easy to implement.
(ii) We provide the geometric interpretation of the stretch energy functional.
(iii) We prove that minimizers of the stretch energy functional are area-preserving mappings.

In addition, we demonstrate the associated efficient energy minimization algorithm for the computation of square-shaped area-preserving mappings of simply connected open surfaces to validate the effectiveness and accuracy of the stretch energy minimization.

Notations  In this paper, we use the following notations.

- Bold letters, for instance, $\mathbf{f}$, denote real-valued vectors or matrices.
- Capital letters, for instance, $L$, denote real-valued matrices.
- Typewriter letters, for instance, $I$, $B$, denote ordered sets of indices.
- $f_i$ denotes the $i$th entry or row of the vector or matrix $f$.
- $f_i$ denotes the subvector or submatrix of $f$ composed of $f_i$, for $i \in I$.
- $L_{i,j}$ denotes the $(i,j)$th entry of the matrix $L$.
- $L_{1,J}$ denotes the submatrix of $L$ composed of $L_{i,j}$, for $i \in I$ and $j \in J$.
- $\mathbb{R}$ denotes the set of real numbers.
- $[v_0, \ldots, v_k]$ denotes the $k$-simplex with vertices $v_0, \ldots, v_k$.
- $||v_0, \ldots, v_k||$ denotes the volume of the $k$-simplex $[v_0, \ldots, v_k]$.
- $\mathcal{M}$ denotes a simplicial 2-complex of $n$ vertices and $m$ triangular faces.
- $\mathbf{0}$ and $\mathbf{1}$ denote the zero and one vectors or matrices of appropriate sizes.

Outline  The remaining parts of the paper is organized as follows. In Section 2 we introduce the notation of simplicial surfaces and mappings as well as their matrix representations. Then, we provide the theoretical foundation of the stretch energy minimization in Section 3. The associated algorithm for the computation of area-preserving mappings is demonstrated in Section 4. Numerical results are demonstrated and discussed in Subsection 5 to validate the effectiveness and accuracy of the stretch energy minimization for the computation of square-shaped area-preserving mappings. Ultimately, concluding remarks are given in Section 6.

2 Simplicial surfaces and mappings

In numerical computation, surfaces and mappings are usually approximated by simplicial surfaces and mappings. A simplicial surface can be represented as a triangular mesh composed of vertices, edges, and triangular faces, and a simplicial mapping can be interpreted as a change of embedding of vertices of the simplicial surface.
2.1 Simplicial surfaces

A simplicial surface or triangular mesh is a simplicial 2-complex $\mathcal{M}$ composed of vertices
\[ V(\mathcal{M}) = \{ \mathbf{v}_i = (v_{i1}, v_{i2}, v_{i3})^\top \in \mathbb{R}^3 \}_{i=1}^n, \]
edges
\[ E(\mathcal{M}) = \{ [\mathbf{v}_i, \mathbf{v}_j] \subset \mathbb{R}^3 \}, \]
and triangular faces
\[ \mathcal{F}(\mathcal{M}) = \{ \tau_\ell = [\mathbf{v}_{i\ell}, \mathbf{v}_{j\ell}, \mathbf{v}_{k\ell}] \subset \mathbb{R}^3 \}_{\ell=1}^m, \]
where the bracket $[\mathbf{v}_{i\ell}, \mathbf{v}_{j\ell}, \mathbf{v}_{k\ell}]$ denotes the convex hull of vertices $\{\mathbf{v}_{i\ell}, \mathbf{v}_{j\ell}, \mathbf{v}_{k\ell}\}$, defined as
\[ \tau_\ell = [\mathbf{v}_{i\ell}, \mathbf{v}_{j\ell}, \mathbf{v}_{k\ell}] := \{ \alpha_1 \mathbf{v}_{i\ell} + \alpha_2 \mathbf{v}_{j\ell} + \alpha_3 \mathbf{v}_{k\ell} \mid \alpha_1 + \alpha_2 + \alpha_3 = 1 \} \subset \mathbb{R}^3. \]
In other words, a simplicial surface $\mathcal{M} = \bigcup_{\ell=1}^m \tau_\ell$ is a surface composed of triangular faces. In practice, the sets of vertices $V(\mathcal{M})$ and triangular faces $\mathcal{F}(\mathcal{M})$ are represented as matrices
\[
V = \begin{bmatrix}
1 & 2 & 3 \\
1 & 2 & 3 \\
\vdots & \vdots & \vdots \\
1 & 2 & 3 \\
\end{bmatrix} = \begin{bmatrix}
v_1^1 & v_1^2 & v_1^3 \\
v_2^1 & v_2^2 & v_2^3 \\
\vdots & \vdots & \vdots \\
v_n^1 & v_n^2 & v_n^3 \\
\end{bmatrix} := [v^1 \ v^2 \ v^3] \quad \text{and} \quad F = \begin{bmatrix}
i_1 & j_1 & k_1 \\
i_2 & j_2 & k_2 \\
\vdots & \vdots & \vdots \\
i_m & j_m & k_m \\
\end{bmatrix},
\]
respectively.

2.2 Simplicial mappings

A simplicial mapping on a triangular mesh $\mathcal{M} = \bigcup_{\ell=1}^m \tau_\ell$ is a particular piecewise affine mapping $f : \mathcal{M} \to \mathbb{R}^2$ solely determined by the mapping of vertices $f|_{V(\mathcal{M})} : V(\mathcal{M}) \to \mathbb{R}^2$. More explicitly, suppose $f|_{V(\mathcal{M})}$ is given by $f(\mathbf{v}_i) = \mathbf{f}_i \in \mathbb{R}^2$, for $i = 1, \ldots, n$. For the point $\mathbf{v} \in \tau_\ell \in \mathcal{F}(\mathcal{M})$,
\[
f|_{\tau_\ell}(\mathbf{v}) = \frac{1}{|\mathbf{f}_\ell|} (||\mathbf{v}, \mathbf{v}_{\ell j}, \mathbf{v}_{\ell k}|| \mathbf{f}_\ell + ||\mathbf{v}_{\ell i}, \mathbf{v}, \mathbf{v}_{\ell k}|| \mathbf{f}_{\ell j} + ||\mathbf{v}_{\ell i}, \mathbf{v}_{\ell j}, \mathbf{v}|| \mathbf{f}_{\ell k}).
\]
In practice, the function $f$ is represented as a real-valued matrix
\[
f = \begin{bmatrix}
f_1^1 & f_1^2 \\
f_2^1 & f_2^2 \\
\vdots & \vdots \\
f_n^1 & f_n^2 \\
\end{bmatrix} := [f^1 \ f^2].
\]
In other words, a simplicial map on $\mathcal{M}$ is an embedding of $\mathcal{M}$ on the parametric domain $f(\mathcal{M})$.

3 Analysis of the stretch energy functional

The stretch energy functional \cite{15, 20} for simplicial surfaces is defined as
\[
E_S(f) = \frac{1}{2} \text{trace}(f^\top L_S(f) f) = \frac{1}{2} (f_1^\top L_S(f) f_1 + f_2^\top L_S(f) f_2),
\]
where $L_S(f)$ is the weighted Laplacian matrix given by
\[
[L_S(f)]_{i,j} = \begin{cases} 
\frac{1}{2} \sum_{[v_i, v_j, v_k] \in \mathcal{F}(\mathcal{M})} \cot(\theta_{i,j}^k(f)) \frac{|f([v_i, v_j, v_k])|}{|[v_i, v_j, v_k]|} & \text{if } [v_i, v_j] \in E(\mathcal{M}), \\
-\sum_{\ell \neq i} [L_S(f)]_{i,\ell} & \text{if } j = i, \\
0 & \text{otherwise}
\end{cases}
\]
in which $\theta_{i,j}^k(f)$ is the angle opposite to the edge $[f_i, f_j]$ at the point $f_k$ on $f(\mathcal{M})$, as illustrated in Figure 1. The stretch energy functional is defined in \cite{15, 20} by imposing the area-preserving condition to the cotangent weights in the discrete Dirichlet energy.
Theorem 1. The gradient of the stretch energy functional \( (1) \) can be written in a neat formulation that can be implemented easily.

Laplacian matrix \( L \)

We now derive a simplified formulation for the gradient of the stretch energy \( (1) \). Noting that the Laplacian matrix \( L_S(f) \) in \( (2) \) is dependent on \( f \) so that the gradient

\[
\nabla f E_S(f) := \left[ \frac{\nabla f^1 E_S(f)}{\nabla f^2 E_S(f)} \right]
\]

with

\[
[\nabla f^i E_S(f)]_\ell = \frac{1}{2} \left( \sum_{i=1}^n [L_S(f)]_{i,\ell} f^1_i + \sum_{i=1}^n f^1_i [L_S(f)]_{i,\ell} + \sum_{i=1}^n \sum_{j=1}^n f^1_i f^1_j \frac{\partial}{\partial f^\ell_i} [L_S(f)]_{i,j} + \sum_{i=1}^n \sum_{j=1}^n f^2_i f^2_j \frac{\partial}{\partial f^\ell_i} [L_S(f)]_{i,j} \right)
\]

involves a 3D tensor \( \frac{\partial}{\partial f^\ell_i} [L_S(f)]_{i,j} \), which would cause difficulty for sparse computation. Fortunately, we can prove in the following theorem that the third and fourth terms in the right-hand-side of \( (3) \) can be written in a neat formulation that can be implemented easily.

Theorem 1. The gradient of the stretch energy functional \( (1) \) is formulated as

\[
\nabla f E_S(f) := \left[ \frac{\nabla f^1 E_S(f)}{\nabla f^2 E_S(f)} \right] = \begin{bmatrix} L_S(f) f^1 + h^1 \nabla f E_S(f) \end{bmatrix} \begin{bmatrix} L_S(f) f^2 + h^2 \end{bmatrix}
\]

with

\[
h^1 = \frac{1}{2} \sum_{\tau = [\mathbf{v}_i, \mathbf{v}_j, \mathbf{v}_k] \in \mathcal{N}_F(\mathbf{v}_i)} \frac{|f(\tau)|}{|\tau|} (f^2 - f^2_k),
\]

and

\[
h^2 = \frac{1}{2} \sum_{\tau = [\mathbf{v}_i, \mathbf{v}_j, \mathbf{v}_k] \in \mathcal{N}_F(\mathbf{v}_i)} \frac{|f(\tau)|}{|\tau|} (f^1_k - f^1_j),
\]

where \( \mathcal{N}_F(\mathbf{v}_i) = \{ \tau \mid \mathbf{v}_i \subset \tau \} \) denotes the set of neighboring triangular faces of \( \mathbf{v}_i \).

Proof. For each triangular face \( \tau = [\mathbf{v}_i, \mathbf{v}_j, \mathbf{v}_k] \in \mathcal{F}(\mathcal{M}) \), we let

\[
\omega_{i,j,k}(f) = \frac{\cot(\theta^k_{i,j}(f)) |f([\mathbf{v}_i, \mathbf{v}_j, \mathbf{v}_k])|}{2 |\mathbf{v}_i, \mathbf{v}_j, \mathbf{v}_k|}.
\]

Then, \( L_S(f) \) is written as

\[
[L_S(f)]_{i,j} = \begin{cases} - \sum_{[\mathbf{v}_i, \mathbf{v}_j, \mathbf{v}_k] \in \mathcal{F}(\mathcal{M})} \omega_{i,j,k}(f) & \text{if } [\mathbf{v}_i, \mathbf{v}_j] \in \mathcal{E}(\mathcal{M}), \\
- \sum_{\ell \neq i} [L_S(f)]_{i,\ell} & \text{if } j = i, \\
0 & \text{otherwise}.
\end{cases}
\]
Noting that $E_S$ can be written as
\[ E_S(f) = \frac{1}{2} \text{trace} (f^T L_S(f) f) = \frac{1}{2} \sum_{s=1}^{2} \sum_{i,j,k} f_i^s [L_S(f)]_{i,j} f_j^s, \]

the entries of gradient $\nabla f E_S(f) = ((\nabla f_1 E_S(f))^\top, (\nabla f_2 E_S(f))^\top)^\top$ can be written as
\begin{align*}
(\nabla f_1 E(f))_{i} &= \frac{1}{2} \frac{\partial}{\partial f_i^s} \left( \sum_{j,k} f_i^s [L_S(f)]_{i,j} f_j^s + \sum_{j,k} f_i^s f_j^s \frac{\partial}{\partial f_i^s} [L_S(f)]_{i,j} \right) \\
&= \frac{1}{2} \left( \sum_{j,k} [L_S(f)]_{i,j} f_j^s + \sum_{j} f_j^s \frac{\partial}{\partial f_i^s} [L_S(f)]_{i,j} \right) = \frac{1}{2} \sum_{j,k} [L_S(f)]_{i,j} f_j^s \\
(\nabla f_2 E(f))_{i} &= \frac{1}{2} \frac{\partial}{\partial f_i^s} \left( \sum_{j,k} f_i^s [L_S(f)]_{i,j} f_j^s + \sum_{j,k} f_i^s f_j^s \frac{\partial}{\partial f_i^s} [L_S(f)]_{i,j} \right) \\
&= \frac{1}{2} \left( \sum_{j,k} [L_S(f)]_{i,j} f_j^s + \sum_{j} f_j^s \frac{\partial}{\partial f_i^s} [L_S(f)]_{i,j} \right) = \frac{1}{2} \sum_{j,k} [L_S(f)]_{i,j} f_j^s \\
\end{align*}

for $s = 1, 2$. To further simplify $\frac{\partial}{\partial f_i^s} L_S(f)$ in [8], we reformulate $\omega_{i,j,k}$ in [7] as
\[ \omega_{i,j,k}(f) = \frac{\cos(\theta_{i,j,k}(f))}{2 \sin(\theta_{i,j,k}(f))} f_i f_j f_k \]
\[ = \frac{1}{4} \frac{f_i f_j f_k}{||f_i f_j f_k||^2} \]

From the definition of the stretch energy ($2$), each triangular face $\tau = [v_i, v_j, v_k]$ contributes a $3 \times 3$ partial submatrix
\[ L_\tau(f) = L_{[v_i, v_j, v_k]}(f) = \frac{1}{4|\tau|} \begin{bmatrix} (f_i - f_k) f_j - f_k & -(f_i - f_k) f_j - f_k & -(f_i - f_k) f_j - f_k \\ (f_i - f_k) f_j - f_k & (f_i - f_k) f_j - f_k & (f_i - f_k) f_j - f_k \\ (f_i - f_k) f_j - f_k & (f_i - f_k) f_j - f_k & (f_i - f_k) f_j - f_k \end{bmatrix} \]

By taking the partial derivatives of [9] with respect to $f_i^s$, $f_j^s$ and $f_k^s$, respectively, we obtain
\begin{align*}
\frac{\partial}{\partial f_i^s} L_\tau(f) &= \frac{1}{4|\tau|} \begin{bmatrix} 0 & f_k^s - f_j^s & f_j^s - f_k^s \\ f_k^s - f_j^s & 0 & f_j^s - f_k^s \\ f_k^s - f_j^s & f_j^s - f_k^s & 0 \end{bmatrix}, \\
\frac{\partial}{\partial f_j^s} L_\tau(f) &= \frac{1}{4|\tau|} \begin{bmatrix} 2 f_j^s - 2 f_k^s & f_k^s - f_j^s & f_j^s - f_k^s \\ f_k^s - f_j^s & 0 & f_j^s - f_k^s \\ f_k^s - f_j^s & f_j^s - f_k^s & 0 \end{bmatrix}, \\
\frac{\partial}{\partial f_k^s} L_\tau(f) &= \frac{1}{4|\tau|} \begin{bmatrix} 2 f_k^s - 2 f_j^s & f_k^s - f_j^s & f_j^s - f_k^s \\ f_k^s - f_j^s & 0 & f_j^s - f_k^s \\ f_k^s - f_j^s & f_j^s - f_k^s & 0 \end{bmatrix}. 
\end{align*}
Let \( f_s^e = f_{v_i, v_j, v_k}^e := (f_i^e, f_j^e, f_k^e)^T \). With the formulation of (10)–(12), it can be verify by direct computation that

\[
\begin{align*}
    f_s^e \top \frac{\partial}{\partial f_i^s} L_r(f) f_s^e &= 0, \quad (13) \\
    f_s^e \top \frac{\partial}{\partial f_j^s} L_r(f) f_s^e &= 0, \quad (14) \\
    f_s^e \top \frac{\partial}{\partial f_k^s} L_r(f) f_s^e &= 0, \quad (15) \\
    f_s^e \top \frac{\partial}{\partial f_{ij}^s} L_r(f) f_s^e &= 0, \quad \ell \notin \{i, j, k\}, \quad (16)
\end{align*}
\]

for \( s = 1, 2 \), and

\[
\begin{align*}
    \frac{\partial}{\partial f_i^s} L_r(f) f_i^s &= \frac{1}{2|\tau|} (f_j^s - f_k^s)(f_i^s f_j^s - f_i^s f_k^s + f_k^s f_j^s - f_k^s f_i^s + f_i^s f_j^s f_k^s - f_i^s f_j^s f_k^s) = \frac{|f(\tau)|}{|\tau|} (-1)^{(i, j, k)} (f_j^s - f_k^s), \quad (17) \\
    \frac{\partial}{\partial f_j^s} L_r(f) f_j^s &= \frac{1}{2|\tau|} (f_k^s - f_i^s)(f_j^s f_k^s - f_j^s f_i^s + f_i^s f_k^s - f_i^s f_j^s + f_j^s f_k^s f_i^s - f_j^s f_k^s f_i^s) = \frac{|f(\tau)|}{|\tau|} (-1)^{(i, j, k)} (f_k^s - f_i^s), \quad (18) \\
    \frac{\partial}{\partial f_k^s} L_r(f) f_k^s &= \frac{1}{2|\tau|} (f_i^s - f_j^s)(f_k^s f_i^s - f_k^s f_j^s + f_j^s f_i^s - f_j^s f_k^s + f_k^s f_j^s f_i^s - f_k^s f_j^s f_i^s) = \frac{|f(\tau)|}{|\tau|} (-1)^{(i, j, k)} (f_i^s - f_j^s), \quad (19) \\
    \frac{\partial}{\partial f_{ij}^s} L_r(f) f_{ij}^s &= 0, \quad \ell \notin \{i, j, k\}, \quad (20)
\end{align*}
\]

for \( (s, t) \in \{(1, 2), (2, 1)\} \).

Noting that the matrix \( L_S(f) \) is assembled by \( \{L_r(f)\}_{\tau \in \mathcal{F}(\mathcal{M})} \), therefore, (4)–(6) hold by (8) and (13)–(20).

The formulas of \( h^1 \) and \( h^2 \) in (5) and (6) look pretty and familiar. Indeed, they are exactly the same as \( L_S(f) f^1 \) and \( L_S(f) f^2 \), respectively, which is stated in the following corollary.

**Corollary 1.** The vectors \( h^1 \) and \( h^2 \) in (5) and (6) satisfy

\[
    h^1 = L_S(f) f^1 \quad \text{and} \quad h^2 = L_S(f) f^2,
\]

respectively. As a result, the gradient of \( E_S \) can also be formulated as

\[
    \nabla_f E_S(f) := \left[ \nabla f \top E_S(f) \right] = 2 \left[ L_S(f) f^1 \right] / \left[ L_S(f) f^2 \right]. \quad (21)
\]

**Proof.** Let \( \tau = [v_i, v_j, v_k] \in \mathcal{F}(\mathcal{M}) \). By multiplying \( L_r(f) \) in (9) with \( f_s^e := (f_i^e, f_j^e, f_k^e)^T \), \( s = 1, 2 \), we obtain

\[
    L_r(f) f_i^s = \frac{1}{4|\tau|} \left[ (f_j^s - f_k^s)(f_i^s f_j^s - f_i^s f_k^s + f_k^s f_j^s - f_k^s f_i^s + f_i^s f_j^s f_k^s - f_i^s f_j^s f_k^s) \right] = \frac{|f(\tau)|}{2|\tau|} \left[ f_j^s - f_k^s \right], \quad (22)
\]

and

\[
    L_r(f) f_j^s = \frac{1}{4|\tau|} \left[ (f_k^s - f_i^s)(f_j^s f_k^s - f_j^s f_i^s + f_i^s f_k^s - f_i^s f_j^s + f_j^s f_k^s f_i^s - f_j^s f_k^s f_i^s) \right] = \frac{|f(\tau)|}{2|\tau|} \left[ f_k^s - f_i^s \right]. \quad (23)
\]

By summing the formulas over every \( \tau \in \mathcal{F}(\mathcal{M}) \), we obtain

\[
    L_S(f) f^1 = \sum_{\tau = [v_i, v_j, v_k] \in \mathcal{F}(\mathcal{M})} \frac{|f(\tau)|}{2|\tau|} (f_j^2 - f_k^2),
\]

and

\[
    L_S(f) f^2 = \sum_{\tau = [v_i, v_j, v_k] \in \mathcal{F}(\mathcal{M})} \frac{|f(\tau)|}{2|\tau|} (f_j^1 - f_k^1).
\]

By summing the formulas over every \( \tau \in \mathcal{F}(\mathcal{M}) \), we obtain

\[
    L_S(f) f^1 = \sum_{\tau \in \mathcal{F}(\mathcal{M})} \frac{|f(\tau)|}{2|\tau|} (f_j^2 - f_k^2),
\]

and

\[
    L_S(f) f^2 = \sum_{\tau \in \mathcal{F}(\mathcal{M})} \frac{|f(\tau)|}{2|\tau|} (f_j^1 - f_k^1).
\]

By summing the formulas over every \( \tau \in \mathcal{F}(\mathcal{M}) \), we obtain

\[
    L_S(f) f^1 = \sum_{\tau \in \mathcal{F}(\mathcal{M})} \frac{|f(\tau)|}{2|\tau|} (f_j^2 - f_k^2),
\]

and

\[
    L_S(f) f^2 = \sum_{\tau \in \mathcal{F}(\mathcal{M})} \frac{|f(\tau)|}{2|\tau|} (f_j^1 - f_k^1).
\]
and
\[ L_S(f) f^2 = \sum_{\tau = \{v_i, v_j, v_k\} \in \mathcal{F}(\mathcal{M})} \left( \frac{|f(\tau)|}{2|\tau|} (f_i^k - f_j^1) \right), \]
which are exactly the same as the formulas of \( h^1 \) and \( h^2 \) in (5) and (6).

\[ \square \]

### 3.2 Geometric interpretation of the stretch energy

Next, we show in the following theorem that the stretch energy \( E_S \) can be written as the sum of squares of the image area of triangular faces divided by the triangular face area, which is a geometric interpretation of the stretch energy.

**Theorem 2.** The stretch energy defined in (1) can be reformulated as
\[ E_S(f) = \sum_{\tau \in \mathcal{F}(\mathcal{M})} \frac{|f(\tau)|^2}{|\tau|}. \]

**Proof.** By the explicit formula of \( L_\tau(f) \) in (9), a direct derivation yields that
\[ E_S(f) = \frac{1}{2} \sum_{s=1}^2 \sum_{[v_i, v_j, v_k] \in \mathcal{F}(\mathcal{M})} \left[ f_s^i f_j^s f_k^s \right] L_{\{v_i, v_j, v_k\}}(f) \begin{bmatrix} f_s^i \\ f_j^s \\ f_k^s \end{bmatrix} \left[ f_s^i \\ f_j^s \\ f_k^s \right]^T \]
\[ = \frac{1}{4|[v_i, v_j, v_k]|} \sum_{\tau \in \mathcal{F}(\mathcal{M})} \left( f_1^1 f_2^2 - f_2^1 f_1^2 + f_2^1 f_3^2 - f_3^1 f_2^2 + f_1^1 f_3^2 - f_3^1 f_1^2 + f_2^1 f_3^2 - f_3^1 f_2^2 \right)^2 \]
\[ = \sum_{\tau \in \mathcal{F}(\mathcal{M})} \frac{|f(\tau)|^2}{|\tau|}. \]

The last equality held by the identity
\[ |f([v_i, v_j, v_k])| = \frac{1}{2}(f_1^1 f_2^2 - f_2^1 f_1^2 + f_2^1 f_3^2 - f_3^1 f_2^2 + f_1^1 f_3^2 - f_3^1 f_1^2 + f_2^1 f_3^2 - f_3^1 f_2^2). \]

\[ \square \]

In particular, when \( f \) is an area-preserving mapping, the stretch energy would be the area of its image, as stated in the following corollary.

**Corollary 2.** For an area-preserving mapping \( f : \mathcal{M} \to \mathbb{R}^2 \),
\[ E_S(f) = |f(\mathcal{M})|. \]

**Proof.** Since \( f \) preserves the area, \( |f(\tau)| = |\tau| \), for every \( \tau \in \mathcal{F}(\mathcal{M}) \). By Theorem 2,
\[ E_S(f) = \sum_{\tau \in \mathcal{F}(\mathcal{M})} |f(\tau)| = |f(\mathcal{M})|. \]

\[ \square \]

### 3.3 Minimization of the stretch energy

We now provide the foundation of the stretch energy minimization for the computation of area-preserving mappings, which is stated in the following theorem.
Theorem 3. Given a simplicial surface \( \mathcal{M} \). Under the constraint that the total area of the surface remains unchanged, the minimal value of \( E_S \) occurs only at area-preserving mappings, i.e.,

\[
f = \arg\min_{g(\mathcal{M})=|\mathcal{M}|} E_S(g)
\]

if and only if \(|f(\tau)| = |\tau|\), for every \( \tau \in \mathcal{F}(\mathcal{M}) \).

Proof. Without loss of generality, we suppose that the total area of surface the image are normalized to be 1, i.e.,

\[
\sum_{k=1}^{m} |\tau_k| = |\mathcal{M}| = 1 \quad \text{and} \quad \sum_{k=1}^{m} |f(\tau_k)| = |f(\mathcal{M})| = 1.
\]

Let \( y_k := |\tau_k| \in (0,1) \) and \( x_k := |f(\tau_k)| \in (0,1), \ k = 1, \ldots, m \), be the area of \( k \)th triangular face and its image, respectively. Since \( \sum_{k=1}^{m} x_k = 1 \) and \( \sum_{k=1}^{m} y_k = 1 \), we write

\[
x_m = 1 - \sum_{k=1}^{m-1} x_k \quad \text{and} \quad y_m = 1 - \sum_{k=1}^{m-1} y_k,
\]

respectively. By (22) together with Theorem 2, it suffices to show

\[
E_S(x_1, \ldots, x_{m-1}) = \sum_{k=1}^{m-1} \frac{x_k^2}{y_k} + \frac{(1 - \sum_{k=1}^{m-1} x_k)^2}{1 - \sum_{k=1}^{m-1} y_k} \geq 1,
\]

and the equality holds if and only if \( x_k = y_k \), for \( k = 1, \ldots, m \). Noting that the critical points of \( E_S(x_1, \ldots, x_{m-1}) \) satisfies

\[
\frac{\partial E_S(x_1, \ldots, x_{m-1})}{\partial x_k} = \frac{2x_k}{y_k} - \frac{2(1 - \sum_{k=1}^{m-1} x_k)}{1 - \sum_{k=1}^{m-1} y_k} = 0,
\]

for \( k = 1, \ldots, m-1 \), i.e.,

\[
\frac{2x_k(1 - \sum_{k=1}^{m-1} y_k) - 2y_k(1 - \sum_{k=1}^{m-1} x_k)}{y_k(1 - \sum_{k=1}^{m-1} y_k)} = 0.
\]

Since \( y_k > 0 \) and \( 1 - \sum_{k=1}^{m-1} y_k > 0 \), (23) implies

\[
x_k \left( 1 - \sum_{k=1}^{m-1} y_k \right) = y_k \left( 1 - \sum_{k=1}^{m-1} x_k \right).
\]

Without loss of generality, let

\[
x_k \frac{y_k}{1 - \sum_{k=1}^{m-1} y_k} = x_m \frac{y_m}{1 - \sum_{k=1}^{m-1} y_k} = c,
\]

for \( k = 1, \ldots, m-1 \), where \( c \) is a constant, i.e., \( x_k = cy_k \), for \( k = 1, \ldots, m \). Since \( \sum_{k=1}^{m} y_k = 1 \) and \( \sum_{k=1}^{m} x_k = 1 \),

\[
c = c \sum_{k=1}^{m} y_k = \sum_{k=1}^{m} x_k = 1.
\]

Therefore, \(|f(\tau_k)| = x_k = y_k = |\tau_k|, k = 1, \ldots, m| \). \( \square \)
Theorem 3 provides a rigorous proof that the stretch energy is the right functional to minimize when computing area-preserving mappings. Noting that the stretch energy is always larger or equal to the image area, and the equality holds if and only if the mapping is area-preserving. It is natural to further define the authalic energy as

\[ E_A(f) = E_S(f) - A(f), \]  

(24)

where \( A(f) \) measures the area of the image of \( f \). Then, a quick corollary follows from Theorem 3.

**Corollary 3.** The authalic energy \( (24) \) satisfies \( E_A(f) \geq 0 \) and the equality holds if and only if \( f \) is area-preserving.

Based on Corollary 3, the authalic energy \( (24) \) can be used to measure how far a mapping is from area-preserving.

### 4 Stretch energy minimization for square-shaped authalic mappings

In this section, we demonstrate the stretch energy minimization algorithm for the computation of square-shaped area-preserving mappings of simply connected open simplicial surfaces. First, we introduce the square-shaped boundary constraint in Subsection 4.1. Then, we compute a harmonic mapping as the initial mapping, which is introduced in Subsection 4.2. Finally, we introduce the iterative procedure for the stretch energy minimization in Subsection 4.3.

#### 4.1 Square-shaped boundary constraints

We denote the index sets of boundary vertices sorted in counterclockwise order and the interior vertices as

\[ B = \{ b \mid v_b \in \partial \mathcal{M} \} \quad \text{and} \quad I = \{ i \mid v_i \notin \partial \mathcal{M} \}, \]

respectively. Suppose the corner points \( v_{C_1}, \ldots, v_{C_4} \) are selected and mapped to \((0,0), (1,0), (1,1), \) and \((0,1)\), respectively. Based on the selected corner indices \( C = \{C_1, C_2, C_3, C_4\} \), the boundary indices \( B \) is classified into 4 categories

\[ Y_0 = \{C_1, \ldots, C_2\}, \quad X_1 = \{C_2, \ldots, C_3\}, \quad Y_1 = \{C_3, \ldots, C_4\}, \quad \text{and} \quad X_0 = \{C_4, \ldots, C_1\}. \]

Each category is a subset of \( B \) sorted in counterclockwise order. The square-shaped constraint is then formulated as

\[ f^1_{x_0} = 0, \quad f^1_{x_1} = 1, \quad f^2_{y_0} = 0, \quad \text{and} \quad f^2_{y_1} = 1. \]

We denote \( I_s \) and \( B_s \) as the sets of indices of unknown entries and constrained entries of \( f^s \), respectively, for \( s = 1, 2 \). Then,

\[ I_1 = I \cup Y_0 \cup Y_1, \quad B_1 = X_0 \cup X_1, \quad I_2 = I \cup X_0 \cup X_1, \quad \text{and} \quad B_2 = Y_0 \cup Y_1. \]  

(25)

#### 4.2 Initial mapping

The initial mapping is chosen to be a harmonic mapping \( f : \mathcal{M} \to \mathbb{R}^2 \) computed by minimizing the Dirichlet energy functional \( [9, 5] \) defined as

\[ E_D(f) = \frac{1}{2} \left( f^{1\top} L_D f^1 + f^{2\top} L_D f^2 \right), \]
with $L_D$ being the Laplacian matrix

$$[L_D]_{i,j} = \begin{cases} 
-\frac{1}{2} \sum_{[v_i, v_j, v_k] \in E(M)} \cot \theta_{i,j}^k & \text{if } [v_i, v_j] \in E(M), \\
-\sum_{\ell \neq i} [L_D]_{i,\ell} & \text{if } j = i, \\
0 & \text{otherwise}
\end{cases}$$

in which $\theta_{i,j}^k$ is the angle opposite to the edge $[v_i, v_j]$ at the vertex $v_k$ on $M$, as illustrated in Figure 1. A minimizer $f$ of $E_D$ is called a harmonic mapping that satisfies

$$\nabla f E_D(f) := \begin{bmatrix} \nabla f_1 E_D(f) \\ \nabla f_2 E_D(f) \end{bmatrix} = [L_D f_1] = 0,$$

for $s = 1, 2$. Under the square-shaped boundary constraints, the equations for the minimization of $E_D$ are formulated as the linear systems

$$[L_D]_{i,\ell} f_{s,i} = -[L_D]_{i,\ell} f_{s,\ell},$$

for $s = 1, 2$, where $I_s$ and $B_s$ are given in (25).

### 4.3 Iterations for the stretch energy minimization

With the gradient formula (21) in Corollary 1, the critical points of $E_S$ would satisfy

$$\nabla f \cdot E_S(f) = 2L_S(f) f^s = 0,$$

for $s = 1, 2$. Under the square-shaped boundary constraints, the minimization can be achieved by the fixed-point iterations

$$[L_S(f^{(n)})]_{i,\ell} f_{s,i}^{(n+1)} = -[L_S(f^{(n)})]_{i,\ell} f_{s,\ell}^{(n)}$$

for $s = 1, 2$, where $I_s$ and $B_s$ are given in (25).

The computational procedure in detail of the stretch energy minimization for area-preserving mappings is summarized in Algorithm 1.

---

**Algorithm 1** Stretch energy minimization for square-shaped area-preserving mappings

**Input:** A simply connected open surface $M$, indices of corner vertices $C_1, \ldots, C_4$.

**Output:** A square-shaped area-preserving mapping $f$.

1. Let $B = \{ b \mid v_b \in \partial M \}$ be the index set of boundary vertices sorted in counterclockwise order.
2. Let $I = \{ i \mid v_i \notin \partial M \}$ be the index set of interior vertices.
3. Classify indices of boundary vertices $B$ into $Y_0 = \{ C_1, \ldots, C_2 \}, X_1 = \{ C_2, \ldots, C_3 \}, Y_1 = \{ C_3, \ldots, C_4 \},$ and $X_0 = \{ C_4, \ldots, C_1 \}$.
4. Let $f_{s,i_0} = 0$, $f_{s,i} = 1$, $f_{s,0} = 0$, and $f_{s,1} = 1$.
5. Let $I_1 = I \cup Y_0 \cup Y_1$, $B_1 = X_0 \cup X_1$, $I_2 = I \cup X_0 \cup X_1$, and $B_2 = Y_0 \cup Y_1$.
6. Let $L = L_D$ be the Laplacian matrix defined as (26).
7. Solve the linear systems $L_{1,1} f_{1,1} = -L_{1,1} f_{1,1}$ and $L_{1,2} f_{1,2} = -L_{1,2} f_{1,2}$, respectively.
8. **while** not convergent do
9.   Let $L = L_S(f)$ be the Laplacian matrix defined as (2).
10. Solve the linear systems $L_{1,1} f_{1,1} = -L_{1,1} f_{1,1}$ and $L_{1,2} f_{1,2} = -L_{1,2} f_{1,2}$, respectively.
11. **end while**
12. **return** the mapping $f = [f^1, f^2]$. 

---

10
5 Numerical experiments

Now, we demonstrate the numerical results of the square-shaped area-preserving mappings computed by the stretch energy minimization Algorithm 1. All the experiments are performed in MATLAB on MacBook Pro M1 with 16 GB RAM. The benchmark triangular mesh models, shown in Figure 2, are obtained from the AIM@SHAPE shape repository [1], the Stanford 3D scanning repository [2], and Sketchfab [3]. The areas of the mesh models are normalized to be 1 so that there would be no global scaling when the target domain is selected to be a unit square. Some of the mesh models are resampled or modified so that every triangular face contains at least one interior vertices.

The initial mappings, shown in Figure 3, of the stretch energy minimization Algorithm 1 is a square-shaped harmonic mapping introduced in Subsection 4.2. The resulting area-preserving mappings of the benchmark mesh models computed by the stretch energy minimization Algorithm 1 are demonstrated in Figure 4. We see that the area-preserving mappings look very different from the initial harmonic mappings.

Let $f : M \rightarrow \mathbb{R}^2$ be the resulting area-preserving mapping computed by Algorithm 1. Noting that the image $|f(M)|$ is a unit square with the area being 1. As stated in Corollary 3, the authalic energy $E_A$ defined in (24) satisfies

$$E_A(f) = E_S(f) - 1,$$

and $E_A(f) = 0$ if and only if $f$ is area-preserving. The decrease of the authalic energy $E_A$ is equivalent to the decrease of the stretch energy $E_S$. To verify the performance of the stretch energy minimization Algorithm 1 on decreasing the stretch energy $E_S$ defined in (1), in Figure 5, we demonstrate the relationship between the authalic energy $E_A$ and the number of iterations of Algorithm 1. We observe that the authalic energy is decreased drastically to nearly zero at the first three iteration steps, which indicates that Algorithm 1 performs effectively on decreasing the authalic energy.

To show the area-preserving property of resulting mappings computed by the stretch energy minimization Algorithm 1 in Figure 6, we demonstrate histograms of area ratios $R_A(f, \tau)$ defined in (27).
Figure 3: The square-shaped initial harmonic mappings of benchmark triangular mesh models computed by Step 7 of Algorithm 1.

Figure 4: The square-shaped area-preserving mappings of benchmark triangular mesh models computed by the stretch energy minimization Algorithm 1.
Figure 5: The relationship between the number of iterations of Algorithm 1 and the authalic energy $E_A$ defined in [24].
Figure 6: Histograms of area ratios $R_A(f, \tau)$ of area-preserving mappings of benchmark mesh models computed by the stretch energy minimization Algorithm 1.

We observe that the area ratios of most triangular faces are close to 1, especially for the mesh models with larger numbers of vertices. These results indicate that the mappings computed by Algorithm 1 preserve the area well.

To quantify the area-preserving property of the resulting mappings, in Table 1, we demonstrate the mean and the standard deviation (SD) of the area ratios

$$R_A(f, \tau) = \frac{|f(\tau)|}{|\tau|},$$

(27)

of the resulting area-preserving mapping $f$ computed by Algorithm 1, where the mean and the SD are taken over all triangular faces $\tau \in F(M)$. An area-preserving mapping $f$ has the property that

$$\text{mean } \tau \in F(M) R_A(f, \tau) = 1 \quad \text{and} \quad \text{SD } \tau \in F(M) R_A(f, \tau) = 0.$$ 

From Table 1, we observe that the mean and the SD of area ratios are reasonably close to 1 and 0, respectively, which indicates that the resulting mappings preserve the area well. In addition, the authalic energies of the resulting mappings are demonstrated in Table 1 as well. We see that the authalic energies of mappings are close to zero, from Corollary 3, these mappings are reasonably close to area-preserving. Furthermore, the computational time cost of each mapping is also demonstrated in Table 1. It would cost less than 80 seconds on a personal laptop to compute an area-preserving mapping of a mesh model of roughly 1 million vertices by Algorithm 1 which is quite efficient.
| Model Name         | # Faces | # Vertices | Area Ratio Mean | SD   | Authalic Energy | Time Cost |
|--------------------|---------|------------|-----------------|------|-----------------|-----------|
| Chinese Lion       | 34,421  | 17,334     | 0.9999          | 0.0252 | $6.0610 \times 10^{-4}$ | 0.78      |
| Femur              | 43,301  | 21,699     | 1.0000          | 0.0381 | $1.4043 \times 10^{-3}$ | 1.11      |
| Max Planck         | 82,977  | 41,588     | 1.0000          | 0.0135 | $1.7194 \times 10^{-4}$ | 2.18      |
| Left Hand          | 105,780 | 53,011     | 1.0000          | 0.0564 | $3.2248 \times 10^{-3}$ | 2.61      |
| Knit Cap Man       | 118,849 | 59,561     | 1.0000          | 0.0161 | $2.5890 \times 10^{-4}$ | 3.12      |
| Bimba Statue       | 433,040 | 216,873    | 1.0000          | 0.0167 | $2.7311 \times 10^{-4}$ | 14.48     |
| Buddha             | 945,722 | 473,362    | 1.0000          | 0.0053 | $2.7495 \times 10^{-5}$ | 41.30     |
| Nefertiti Statue   | 1,992,801 | 996,838  | 1.0000          | 0.0213 | $1.0074 \times 10^{-3}$ | 75.48     |

Table 1: The mean and SD of area ratios \(^\text{[27]}\), the authalic energy \(^\text{[24]}\) and the computational time cost of the area-preserving mapping computed by the stretch energy minimization Algorithm \[^\text{[1]}\].

6 Concluding remarks

In this paper, we demonstrate the theoretical foundation of the stretch energy minimization for the computation of area-preserving mappings, including the derivation of a neat closed-form formulation of the gradient, the geometric interpretation of the stretch energy, and the proof of the minimum value occurs only at area-preserving mappings. Numerical experiments are demonstrated to show the effectiveness and accuracy of the stretch energy minimization for the computation of area-preserving mapping of simplicial surfaces. The derivation can be generalized to the volumetric stretch energy minimization for volume-preserving mappings of simplicial 3-complexes, which is our future topic to investigate.

Acknowledgement The work of the author was partially supported by the Ministry of Science and Technology under grant numbers 109-2115-M-003-010-MY2 and 110-2115-M-003-014-, the National Center for Theoretical Sciences, the ST Yau Center in Taiwan, and the Ministry of Education.

References

[1] Digital Shape Workbench - Shape Repository. [http://visionair.ge.imati.cnr.it/ontologies/shapes/] (2016).
[2] The Stanford 3D Scanning Repository. [http://graphics.stanford.edu/data/3Dscanrep/] (2016).
[3] Sketchfab. [https://sketchfab.com/] (2019).
[4] G. P. T. Choi and C. H. Rycroft. Density-equalizing maps for simply connected open surfaces. SIAM J. Imag. Sci., 11(2):1134–1178, 2018.
[5] M. Desbrun, M. Meyer, and P. Alliez. Intrinsic parameterizations of surface meshes. Comput. Graph. Forum, 21(3):209–218, 2002.
[6] A. Dominitz and A. Tannenbaum. Texture mapping via optimal mass transport. IEEE T. Vis. Comput. Graph., 16(3):419–433, May 2010.
[7] M. S. Floater and K. Hormann. Surface parameterization: a tutorial and survey. In Advances in Multiresolution for Geometric Modelling, pages 157–186. Springer Berlin Heidelberg, 2005.
[8] K. Hormann, B. Lévy, and A. Sheffer. Mesh parameterization: Theory and practice. In ACM SIGGRAPH Course Notes, 2007.
[9] J. E. Hutchinson. Computing conformal maps and minimal surfaces. Proc. Centre Math. Appl., 26:140–161, 1991.

[10] Y.-C. Kuo, W.-W. Lin, M.-H. Yueh, and S.-T. Yau. Convergent conformal energy minimization for the computation of disk parameterizations. SIAM J. Imag. Sci., 14(4):1790–1815, 2021.

[11] P. V. Sander, J. Snyder, S. J. Gortler, and H. Hoppe. Texture mapping progressive meshes. In Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH ’01, pages 409–416, New York, NY, USA, 2001. ACM.

[12] A. Sheffer, E. Praun, and K. Rose. Mesh parameterization methods and their applications. Found. Trends. Comp. Graphics and Vision., 2(2):105–171, 2006.

[13] K. Su, L. Cui, K. Qian, N. Lei, J. Zhang, M. Zhang, and X. D. Gu. Area-preserving mesh parameterization for poly-annulus surfaces based on optimal mass transportation. Comput. Aided Geom. D., 46:76 – 91, 2016.

[14] S. Yoshizawa, A. Belyaev, and H. P. Seidel. A fast and simple stretch-minimizing mesh parameterization. In Proceedings Shape Modeling Applications, 2004., pages 200–208, June 2004.

[15] M.-H. Yueh. Surface Conformal and Equiareal Parameterizations with Applications. PhD thesis, National Chiao Tung University, 2017.

[16] M.-H. Yueh, H.-H. Huang, T. Li, W.-W. Lin, and S.-T. Yau. Optimized surface parameterizations with applications to Chinese virtual broadcasting. Electron. Trans. Numer. Anal., 53:383–405, 2020.

[17] M.-H. Yueh, T. Li, W.-W. Lin, and S.-T. Yau. A novel algorithm for volume-preserving parameterizations of 3-manifolds. SIAM J. Imag. Sci., 12(2):1071–1098, 2019.

[18] M.-H. Yueh, T. Li, W.-W. Lin, and S.-T. Yau. A new efficient algorithm for volume-preserving parameterizations of genus-one 3-manifolds. SIAM J. Imag. Sci., 13(3):1536–1564, 2020.

[19] M.-H. Yueh, W.-W. Lin, C.-T. Wu, and S.-T. Yau. An efficient energy minimization for conformal parameterizations. J. Sci. Comput., 73(1):203–227, 2017.

[20] M.-H. Yueh, W.-W. Lin, C.-T. Wu, and S.-T. Yau. A novel stretch energy minimization algorithm for equiareal parameterizations. J. Sci. Comput., 78(3):1353–1386, Mar 2019.

[21] X. Zhao, Z. Su, X. D. Gu, A. Kaufman, J. Sun, J. Gao, and F. Luo. Area-preservation mapping using optimal mass transport. IEEE T. Vis. Comput. Gr., 19(12):2838–2847, 2013.

[22] G. Zou, J. Hu, X. Gu, and J. Hua. Authalic parameterization of general surfaces using Lie advection. IEEE T. Vis. Comput. Graph., 17(12):2005–2014, 2011.