

Discrete Lie flow: A measure controllable parameterization method



Kehua Su^a, Chencheng Li^a, Shifan Zhao^b, Na Lei^{c,e,*}, Xianfeng Gu^{d,e}

^a School of Computer Science, Wuhan University, Wuhan, China

^b School of Mathematics and Statistics, Wuhan University, Wuhan, China

^c DUT-RU ISE, Dalian University of Technology, Dalian, China

^d Computer Science Department, State University of New York at Stony Brook, NY, United States of America

^e Beijing Advanced Innovation Center for Imaging Theory and Technology, Beijing, China

ARTICLE INFO

Article history:

Available online 4 June 2019

Keywords:

Measure controllable parameterization

Area-preserving parameterization

Discrete Lie advection

Optimal transport

ABSTRACT

Computing measure controllable parameterizations for general surface is a fundamental task for medical imaging and computer graphics, which is designed to control the measures of the regions of interest in the parameterization domain for more accurate and thorough detection and examination of data. Previous works usually handle just some certain kind of topology and boundary shapes, or are computationally complex. In this paper, a modified approach based on the technique of lie advection is presented for the measure controllable parameterization of geometry objects in the general context of 2-manifold surfaces. Given a general surface with arbitrary initial parameterization without flips but usually with great area distortion, the Lie derivative is introduced to eliminate the difference between the initial parameterization and the prescribed measure. The vertices flow in the directions derived through the Lie derivative and finally converge to the ideal measure, and by its geometric meaning, this method will be called as DLF (Discrete Lie Flow) intuitively. Compared with previous methods based on Lie derivative, two key modifications were made: an adaptive step-length scheme resulting in a substantive acceleration and robustness and a measure controllable function. Area preserving mapping can be generated easily through our DLF algorithm as a special case for measure controllable parameterization. With various algorithms developed for mesh parameterization based on energy optimization approaches in recent years, our DLF is the minority that is supported by a solid differential geometric theory. We tested our method on plenty of cases, including disk models with convex and non-convex boundaries, and spherical models. Experimental results demonstrate the efficiency of the proposed algorithm.

© 2019 Elsevier B.V. All rights reserved.

1. Introduction

Parameterization is a foundational problem in computer graphics and geometry processing and is relevant to many other processes such as texture mapping, surface registration (Zeng and Gu, 2013), shape correspondence (Van Kaick et al., 2011), remeshing (Su et al., 2019) and so on. Moreover, many techniques of parameterization have recently been proposed

* Corresponding author at: DUT-RU ISE, Dalian University of Technology, Dalian, China.

E-mail address: nalei@dlut.edu.cn (N. Lei).

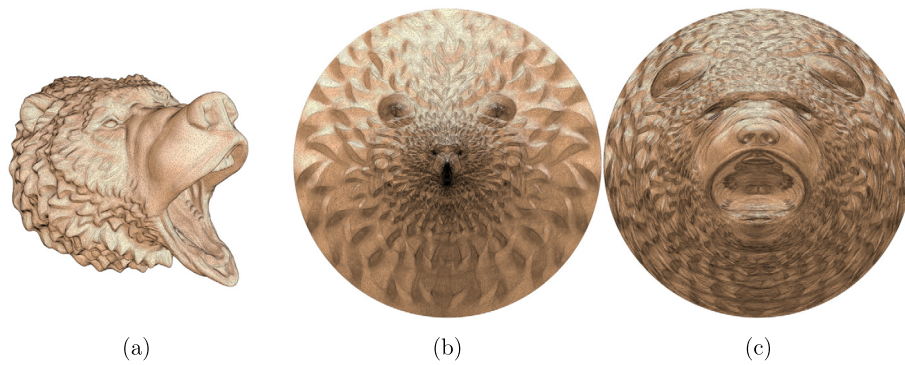


Fig. 1. Flowing process of area-preserving mapping via DLF using Tutte embedding as initial parameterization.

to medical imaging (Wang et al., 2012). For example, obtaining the flattened representations of the highly undulated and branched brain surface, which is important in the study of neural activities. Also, flattened representation will be a good complement for CT colonoscopy or virtual colonoscopy.

Among the various approaches of mesh parameterization, usually the key feature is the minimization of distortion, i.e. angle distortion and area distortion, which is like the two sides of a coin, we can only achieve either of them, or alternatively seek for a compromise. For minimizing angle distortion, the problem has been extensively studied with a plethora of algorithms. See Levy et al. (2002), Gu and Yau (2003), Jin et al. (2008), Wang et al. (2009), Bright et al. (2017), and the references therein. On the other hand, for minimizing area distortion, to the best of our knowledge, only a limited number of mature methods have been invented. See Angenent et al. (2000), Zhu et al. (2003), Zou et al. (2011), also Aigerman et al. (2015) and references therein.

Measure controllable parameterization, further on, can be regarded as a more general form of area-preserving parameterization. It enables users to select Regions Of Interests (ROIs) to enlarge (or shrink) the area elements for more advanced applications, such as more accurate detection and thorough examination in medical imaging. It is also a prerequisite in modeling and rendering techniques, like remeshing, morphing, texture mapping, etc. A very popular method for measure controllable is Optimal Mass Transport (OMT for short), we refer readers to Zhao et al. (2013), Dominitz and Tannenbaum (2010), Su et al. (2016a; 2016b; 2017) and Lei et al. (2019). However, OMT is based on Brenier's theorem for the optimal and unique mapping, and the governing Partial Differential Equations (PDE for short) Monge-Ampere (Bonnotte, 2013) is highly non-linear. The construction and maintenance of the geometric data structure is complicated, and with high spacial complexity with the increase of the dimension. Furthermore, convexity of the source surface is required, otherwise the resulting diffeomorphic map will not be able to be extended to the boundary.

In our paper, we generalize the work in Angenent et al. (2000) and Zou et al. (2011), a modified approach based on the technique of Lie advection is presented for the measure controllable parameterization of geometry objects in the general context of 2-manifold. We call this method **Discrete Lie Flow** (DLF for short). The construction of our algorithm stems from three observations: firstly, the gradient of vertex positions can be approximated from the solution of Poisson equation; secondly, the optimal steps can be derived by finding the critical flip-preventing condition; lastly, a density function can be assigned to parameterization domain for controlling ROIs measure explicitly.

Given a general surface with arbitrary initial parameterization without flips but usually with great area distortion, DLF quantifies the difference between initial parameterization and prescribed measure, then leads the vertices to flow in the directions which is designed to decrease the measure difference, and finally converge to the minimal area distortion parameterization. In Fig. 1, for the model Bear, we use a simple Tutte embedding (Tutte, 1963) for flattening the Bear onto a disk parameterization domain and produce an area-preserving parameterization via DLF. Visually, we can see the triangles huddled in the center flow evenly throughout the entire mesh. For more precision, in another Fig. 2, we made the Gargoyle model textured, in the initial parameterization through Tutte embedding we can see the checkerboard texture pulled back to the model is highly area-distorted. After the flowing process, every blocks of the texture become equal in area.

In our DLF method, the governing PDE is the linear Poisson equation, which is fundamentally superior to that of OMT solving non-linear Monge-Ampere equation. Although DLF may not be optimal in the theoretical category, our area preservation ability in practice turns out to be generally better than OMT. Also, the convex restriction for source domain can be removed, and we can derive measure controllable parameterization from initial conditions where OMT can never handle.

In summary, our method DLF has plenty of prominent features, listed as follows:

- **Measure controllable parameterization supported by a rigorous and unified theory.** We present an efficient and robust method based on discrete Lie derivative to generate measure controllable parameterizations.
- **Adaptive Step-length Scheme.** We propose the Adaptive Step-length Scheme, which greatly enhances the efficiency and robustness compared with Zou et al. (2011), a latest work also based on discrete Lie derivative.

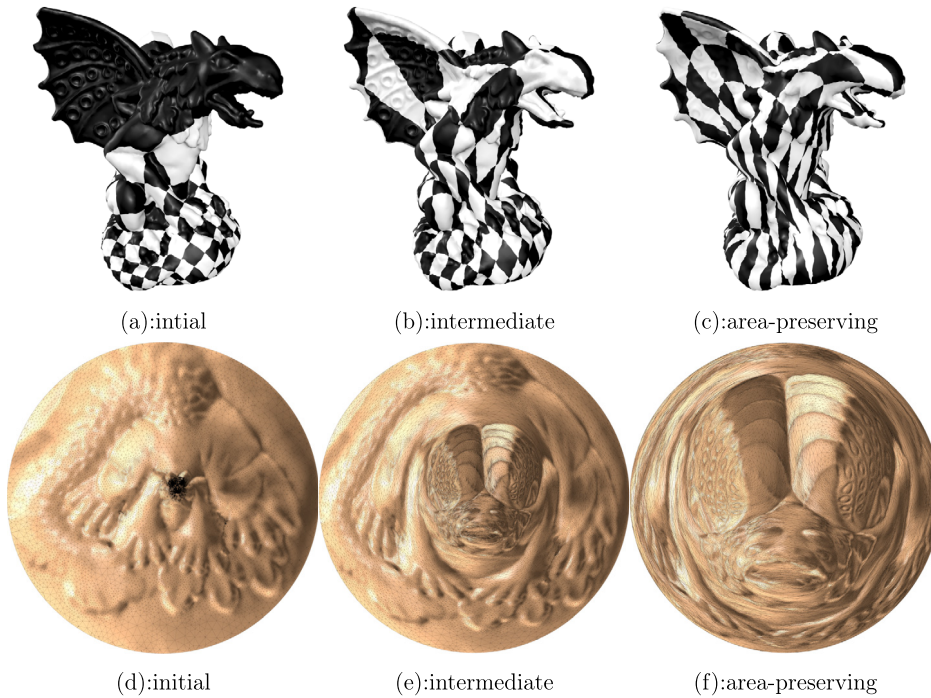


Fig. 2. Flowing process of texture mapping (top); flowing process of parameterization domain mesh (bottom).

- **A good approximate for optimal transportation map.** Our method also offers a good approximation for computing optimal transportation map and Wasserstein distance, whose governing PDE is the linear Poisson equation.

2. Previous works

Surface parameterization was initially introduced to the computer graphics community as a method for mapping textures onto surfaces, which is one of the most researched subjects in computer graphics, and specifically the problem of reducing area distortion of surface parameterization has garnered a lot of attention in past decades. In this section, we mention only the most related works on the topic of area-preserving and measure controllable parameterization. For in-depth surveys of parameterization, we refer to Hormann et al. (2008) and Sheffer et al. (2006).

For an arbitrary surface, there is no really isometric parameterization that preserves both angles and areas. Nevertheless, there is a class of methods called low stretch parameterization. They usually try to minimize a distortion energy that consists of both angle errors and area errors, like Hormann and Greiner (2000) did and Wang et al. (2002) did in their works. For low-stretch parameterization, there are two categories of approaches, including linear and non-linear methods. Linear methods compute mesh parameterization by solving a linear system, where each vertex is represented as a weighted average of a $\mathfrak{N}_1$ ring, like in Aigerman and Lipman (2015), where the authors proposed to parametrize topological disks with low-stretch property. However, the majority of methods for this problem are non-linear, with numerous non-linear deformation energies proposed in the literature for isometric distortion. Standard optimization methods are typically used to minimize these energies, such as Newton, Gauss-Newton, quasi-Newton, and second-order cone programming. For example, in Angenent et al. (2000), the authors combined the ideas of conformality derived via the minimization of the Dirichlet integral and area preservation to describe a new approach to area preserving diffeomorphism. Another example is Yoshizawa et al. (2004), where they proposed a fast and simple method based on the work in Floater (1997); given a triangle mesh, they start from the Floater shape preserving parameterization and then improve the parameterization gradually. Moreover, in Aigerman et al. (2015), authors introduced a weaker condition for bijective mapping allowing them to just optimize locally parameterization of two surfaces into the plane, and finally by a convexification trick they derive a low-stretch mapping; in Rabinovich et al. (2017), a fast and efficient algorithm was proposed based on optimization of flip-preventing energies; in Hu et al. (2018), authors proposed a very robust hierarchical algorithm for spherical parameterization; in Zhao et al. (2019) introduced Unit Normal Flow for planar and spherical parameterization based on constant mean curvature deformation.

Although a low stretch mapping can be derived, it is not actually the area-preserving mapping, not to mention being measure controllable. For purely area-preserving and measure controllable parameterization, as is mentioned above, OMT is a popular approach. In Zhu et al. (2003), authors utilize OMT to flatten closed anatomical surfaces in an area-preserving way. Dominitz and Tannenbaum (2010) point that OMT can be used to derive an optimal mapping with area-preserving and minimal angle-distortion based on Monge-Kantorovich approach, which is time consuming and numerically complex.

Later in Zhao et al. (2013), based on Monge–Brenier’s Approach, authors reduced the computational cost. Recently, in Su et al. (2016a), area-preserving mappings can be generated for poly-annulus surfaces. Su et al. (2016b) enable one obtaining measure controllable parameterization for volumetric mesh based on the technique of OMT. However, the main drawbacks of OMT may be its computational difficulty, and requirements on topology and convexity. In addition to the OMT approach, Angenent et al. (2000) first use Lie derivative to generate area-preserving mappings for spherical cases, later in Zou et al. (2011), they demonstrate that the method can also be applied to a topological disk case based on exactly the same method.

Although this paper is not the first proposal for Lie derivative method, we must stress that, until Zou et al. (2011), a major problem of Lie derivative in discrete conditions still remained unnoticed, unmentioned and therefore unsolved — selecting step-lengths. A valid step-length, in our experiments, may vary irregularly in several **orders of magnitude**, for example, from $5E - 10$ to $5E - 1$, among different models with different initial parameterizations, and among different iterations in a complete flowing procedure. This problem makes previous works based on Lie derivative far from practical applications, as the parameterization easily gets collapsed with incorporate steps. Moreover, no measure controllable approach has been mentioned before.

In this paper, we firstly prove a more general mathematical framework for computing measure controllable parameterizations which adapts to various topology and boundary conditions. Based on an any-dimensional theorem, our DLF is, further more, able to handle high dimension measure controllable parameterizations, e.g. for tetrahedral meshes. Besides, we incorporate an adaptive step-length scheme into DLF, which not only greatly accelerates the flow process, but also makes it a robust and reliable parameterization method for the first time. Our algorithm is specifically designed to apply to various cases with affordable costs, which is different from most existing methods that handle just a limited number of cases, e.g. disk cases and sphere cases etc. With various algorithms having been developed for mesh parameterization based on energy optimization approaches in recent years, our DLF is the minority that is supported by the solid differential geometric theory.

3. Overview

In this section, we outline an overview of the following sections of our paper for the convenience of readers.

- We firstly give a stretch of the mathematical framework of our method.
- Then we discretize the Lie derivative on triangle meshes.
- Next we illustrate the key process of our methods, the adaptive step-length scheme. Also we summarize the algorithm and pipeline.
- Finally we test our algorithm using plenty of models with multi-scale mesh and compare our method with several other methods.

4. Sketch of relevant mathematical theory

In this section, we outline the mathematical justification of our mapping procedure for area preserving parameterization. This is based on the idea of correcting the initial parameterization’s area defects through DLF, which will decrease the distortion gradually. Finally, the flow will converge to a minimal area distortion state. Repeating the above process with a density function will allow us to extend this method to a measure controllable case.

By area preserving mapping f for two manifolds M and N with ω_M and ω_N as volume form respectively, we actually mean $f^*(\omega_N) = \omega_M$, where $f^*(\omega_N)$ is the pullback differential form of ω_N via f . Therefore the problem of seeking an area preserving mapping is reduced to find a mapping preserving the volume form. In the proof of main theorem, we will use some notations and concepts in differential geometry, we will explain some of main concepts roughly.

Basic concepts The Lie derivative measures the change of a tensor field along the flow of another vector field. More precisely, given a differentiable tensor field T of rank (q,r) and a differentiable vector field Y , then the Lie derivative of T along Y can be defined as follows. For some open interval I around 0, $\phi : M \times I \rightarrow M$ be the one-parameter diffeomorphism group of M induced locally by Y and denote $\phi_t(p) := \phi(p, t)$. The Lie derivative of T is defined at a point p by

$$(\mathcal{L}_Y T)_p = \frac{d}{dt} \Big|_{t=0} ((\phi_{-t})^* T_{\phi_t(p)}) = \frac{d}{dt} \Big|_{t=0} ((\phi_t)^* T)_p$$

where $(\phi_t)_*$ is the push-forward along the diffeomorphism and $(\phi_t)^*$ is the pullback along diffeomorphism. Also we will utilize Cartan’s magic formula: $\mathcal{L}_Y = d \circ (i_Y) + i_Y \circ d$, where i_Y is the interior product which defined as $i_Y(\omega)(X) = \omega(Y, X)$ where X is a differentiable vector field, ω is a two-form. For more details, we refer to Levy (1964). Then we state our main theorem as follows: **Given any initial parameterization f of smooth 2-manifold M to Ω , we can always find a one parameter group $\phi_t(p) : \Omega \rightarrow \Omega$, $t \in [0, 1]$, such that $\phi_1 \circ f$ is an area-preserving parameterization.**

Proof. First, let the volume element of M and Ω be denoted by n -forms ω_M and ω_Ω . If we wanna find an area-preserving mapping $\mu : M \rightarrow \Omega$, then we must have $\mu^*(\omega_\Omega) = \omega_M$. For this, we can just need to find an automorphism $\phi : \Omega \rightarrow \Omega$, which satisfies $\phi^*(\omega_\Omega) = (f^{-1})^*(\omega_M)$, since

$$(\phi \circ f)^*(\omega_\Omega) = f^* \circ \phi^*(\omega_\Omega) = \omega_M$$

In order to find this ϕ , we let $X \in \Gamma(T\Omega)$ be a section of tangent bundle of Ω . And $\gamma(p, t)$ is the integral curve of X with the initial point $p \in \Omega$, that is,

$$\begin{aligned} \frac{d\gamma(p, t)}{dt} &= X(\gamma(p, t)) \\ \gamma(p, 0) &= p \end{aligned}$$

let ϕ_t be the one parameter group generated by the vector field X , then we have $\phi_t(p) =: \gamma(p, t)$. This is observed by Zou et al. (2011). That

$$\omega_t = (1 - t)\omega_\Omega + t(f^{-1})^*(\omega_M), t \in [0, 1]$$

flow in the direction of our desired result. So inspired by this observation, we let our pullback volume element by ϕ_t flow in the same direction. By this we mean,

$$\frac{d\phi_t^*(\omega_\Omega)}{dt} = (f^{-1})^*(\omega_M) - \omega_\Omega$$

However, this is just the Lie derivative of ω_t w.r.t. X . According to Cartan's magic formula, we have

$$\mathcal{L}_X \omega_t = d \circ \iota_X \omega_t + \iota_X \circ d \omega_t,$$

if ω_M and ω_Ω are denoted by $\omega_\Omega = g_\Omega dx^1 \wedge dx^2$ and $(f^{-1})^*(\omega_M) = g_M dx^1 \wedge dx^2$. We yield to

$$d \circ \iota_X \omega_t = (f^{-1})^*(\omega_M) - \omega_\Omega \quad (1)$$

we found that

$$X(t) = \frac{1}{(1 - t)g_\Omega + tg_M} \nabla h,$$

where h satisfies

$$\Delta h = g_M - g_\Omega$$

solves our equation. Since

$$\begin{aligned} \iota_X \omega_t &= \left(\frac{\partial}{\partial x^1}, \frac{\partial}{\partial x^2}, \dots, \frac{\partial}{\partial x^i}, \dots, \frac{\partial}{\partial x^n} \right) \\ &= (-1)^{i+1} \frac{\partial h}{\partial x^i} \end{aligned}$$

so finally, we get

$$\iota_X \omega_t = \sum_{i=1}^n (-1)^{i+1} \frac{\partial h}{\partial x^i} dx^1 \wedge dx^2$$

We yield to

$$d \circ \iota_X \omega_t = \Delta h dx^1 \wedge dx^2$$

That is,

$$\Delta h = g_M - g_\Omega$$

is the desired h .

Since we found that $\phi_1^*(\omega_\Omega) = (f^{-1})^*(\omega_M)$. That is, the ϕ_1 is desired diffeomorphism, which is derived by integrating the vector field $X(t)$ from 0 to 1. We get

$$\phi_1 = \int_0^1 X(t) dt = \int_0^1 \frac{1}{(1 - t)g_\Omega + tg_M} \nabla h dt = \frac{\ln g_M - \ln g_\Omega}{g_M - g_\Omega} \nabla h$$

Then we compose ϕ_1 and f , finally we can derive the volume-preserving mapping $\phi_1 \circ f$ we want. $\square$

5. Discrete Lie flow

In this section, our main purpose is to derive an efficient and robust algorithm, based on our theorem, to compute measure controllable parameterizations. Source domain will be denoted by (M, V, E, F) , standing for model M with vertices V , edges E and triangle faces F . Given an initial parameterization ϕ , the induced mesh on parameterization domain will be denoted by $(\Omega, \hat{V}, \hat{E}, \hat{F})$.

5.1. Initial parameterization for triangular meshes

The DLF approach for measure controllable parameterization is qualified for any types of initial parameterization. For testing the efficiency and efficacy, the DLF will be applied to several different initial parameterizations. We won't discuss the initial parameterization process in details otherwise attention will be distracted. However, reference for these methods will be listed. For conformal parameterization, we used the method in Gu and Yau (2003). The work in Rabinovich et al. (2017) is implemented for Scalable Locally Injective Mappings. Also Least Square Conformal Maps in Levy et al. (2002) and Harmonic Global Parameterization in Bright et al. (2017) are utilized in our paper.

There is no strict requirement for the initial parameterization. However, we would like those that have no flips. Although some simple methods may not meet this demand intrinsically, e.g., harmonic maps (Floater, 1997) may cause a flip when the sum of opposite angles of an edge is larger than π . Nevertheless, the minor flips of the initial parameterization can be mended simply. For example, we can recalculate the coordinate of a flipped vertex, by an affine combination of the good vertices nearby.

Mesh Optimization. To explain why we need optimization, imagine a common occasion where, in area-preserving parameterizations regardless of angle distortion, many triangles are deformed long and thin to better preserve areas. As we will see in the following section, we need to compute the cotangent Laplacian of the flowing mesh. If the mesh is not Delaunay triangulation, the angle may cause some singular entries of the Laplacian matrix, as the cotangent value of 0 and π is infinity. To deal with that, we need to do edge-flipping for the flowing mesh every time before further operations.

5.2. Solving the Laplacian equation

As is proved in the previous section, the main theorem of DLF can be demonstrated by the equation

$$\Delta h = g_\Omega - g_M$$

To discretize this problem, we need to explain the discrete version of area elements and the cotangent Laplacian on triangular mesh. By area element, we mean the $\mathfrak{A}_1(i)$ – ring area for any vertex i . More specifically,

$$\mathcal{A}(i) = \frac{1}{6} \sum_{ijk \in F} |\mathbf{e}_{ij} \times \mathbf{e}_{ik}| \quad (2)$$

Now we let g_Ω and g_M be two functions defined on parameterization Ω , where g_Ω is the original area elements on Ω and g_M is the pullback area elements on Ω through the inverse of initial parameterization f^{-1} , e.g. the corresponding area elements on M for area-preserving purpose.

Cotangent Laplacian The well-known cotangent Laplacian is actually derived by first order element method through Galerkin's approach, which is defined as follows:

$$L_{ij} = \begin{cases} \frac{1}{2} \sum_{\mathbf{e}_{ij} \in \hat{E}} (\cot \alpha_j + \cot \beta_j), & i=j, \\ -\frac{1}{2} (\cot \alpha_j + \cot \beta_j), & \mathbf{e}_{ij} \in \hat{E} \\ 0, & \text{otherwise} \end{cases}$$

Of course there are some other approaches to discretize Laplacian, while cotangent Laplacian is the most suitable one for our context

Von Neumann boundary value problem When we address with the topological disk, we are actually solving the following Von Neumann boundary value problem:

$$\begin{cases} \Delta h = g_\Omega - g_M \\ \frac{\partial h}{\partial \mathbf{n}} = 0 \end{cases} \quad (3)$$

where the $\mathbf{n}$ is the unit normal vector of the parameterization disk.

5.3. Generate one parameter group

According to our main theorem, we know that

$$X(t) = \frac{1}{(1-t)g_\Omega + tg_M} \nabla h$$

where g_Ω is the area elements of parameterization domain Ω and g_M is the pullback area elements through the inverse of initial parameterization f^{-1} . And the mapping we desire is

$$\int_0^1 X(t)dt = \frac{\ln g_M - \ln g_\Omega}{g_M - g_\Omega} \nabla h$$

So we want to use the gradient ∇h to approximate our auto-diffeomorphism. For convenience, we made some simplification. Firstly, we normalized g_Ω and g_M , then it's easy to find

$$\phi_1 = \frac{\ln g_M}{g_M - 1} \nabla h$$

When g_M is close to 1, we have

$$\lim_{g_M \rightarrow 1} \frac{\ln g_M}{g_M - 1} \nabla h = \nabla h$$

This inspires us to approximate ϕ_1 via ∇h . And we found it achieves a well-behaved performance in our experiments. In a word, every time we will apply the displacement vector field ∇h to vertices of the parameterization domain mesh.

Gradient decomposition In order to let the mesh flow on the sphere, we have to decompose our gradient ∇h along the normal and tangent direction, since we'll use the ∇h to approximate our diffeomorphism ϕ_1 and finally use ϕ_1 to generate the new mesh. When we use ∇h , we have killed the normal components of it and let mesh flow along the tangent direction. To be more specific, we compute the normal component of ∇h as follows.

$$\nabla h^\perp = \langle \nabla h, \vec{n} \rangle \vec{n}$$

$\vec{n}$ is still the unit normal vector of the sphere. Finally, we yield to the tangent directional component

$$\nabla h^\parallel = \nabla h - \nabla h^\perp$$

Then we can use $\nabla h^\parallel$ to replace ∇h and use it to flow our original mesh on the parameterization domain. More precisely, for each vertex $p \in \Omega$ on parameterization domain mesh, the displacement vector field will be applied as follows $\hat{p} = p + \lambda_i \nabla h$. Where λ_i is the i -th step for the displacement.

5.4. Adaptive step-length scheme

Importantly, a key problem is that we have to select the step-length in the directions of the vector field to make our parameterization flow. Rather than a fixed step, or any tentative ones, we creatively take a series of precise and reliable adaptive step-lengths strongly relative to the status of the current flowing mesh for each iteration.

A example for applying steps that's too large, is shown in Fig. 3. We can see that, once the flowing mesh collapsed one time, it may even collapse worse rather than rehabilitate itself, and finally no valid result is guaranteed. On the other hand, if we cautiously select steps that is too small, the deformation requires hundreds and thousands of iterations to converge.

We investigate the step principle in previous work (Zou et al., 2011). They firstly selected the amount of iterations K as a hyper-parameter, and derive a step for the k -th iteration from K and the current per-vertex surface/domain area ratio. Obviously, this operation did not turn out to be a solution satisfying enough, and the paper did not even give a convincing scheme how we can compute or estimate K in practical applications.

In our paper, we propose the adaptive step-length scheme and incorporate it into the DLF. The adaptive step-length is decided by the flip-preventing condition which was explained in the previous section. With the aim of preventing flip, consider that the critical time of collapsing is when some points of a triangle just reach its subtense. Given this numerical prerequisite, when flowing on planes, for any triangle in F we definitely have the coordinates for three vertices, say $v_1(x_1, y_1)$, $v_2(x_2, y_2)$, $v_3(x_3, y_3)$, and the gradient $\nabla h(v_i) = (h_i^1, h_i^2)^T$ for $i = 1, 2, 3$; for spherical conditions, the coordinates and gradient can be isometrically rotated into the XY plane by each triangle, to convert to a 2-dimensional situation. We let $\tilde{\lambda}$ be the critical step for any triangle, then we have three lines as follows.

$$l_{12} : (y_2 - y_1 + \tilde{\lambda}(h_2^2 - h_1^2))x - (x_2 - x_1 + \tilde{\lambda}(h_2^1 - h_1^1))y +$$

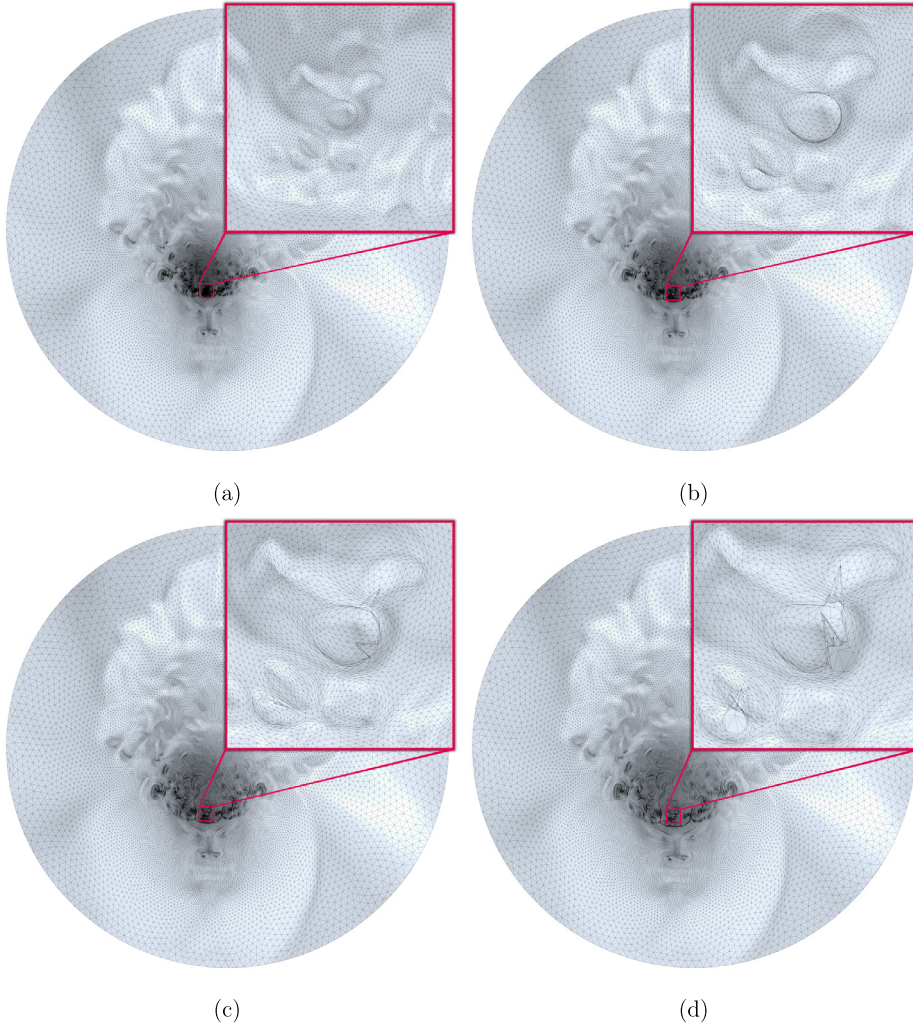


Fig. 3. DLF flowing without adaptive step-lengths. (a): original mesh; (b): mesh beginning to collapse; (c): mesh just collapsed; (d): mesh collapsed worse.

$$x_2 y_1 - x_1 y_2 + \tilde{\lambda}(y_1 h_2^1 - y_2 h_1^1) + \tilde{\lambda}^2(x_2 h_1^2 - x_1 h_2^2) = 0$$

l_{23} and l_{31} are just similar, readers can derive them by cycle the index. By assuming the vertex v_3 is just lying on the line l_{12} , we can know the step $\tilde{\lambda}$ explicitly, by solving the following equation for $\tilde{\lambda}$

$$(y_2 - y_1 + \tilde{\lambda}(h_2^2 - h_1^2))(x_3 + \tilde{\lambda}h_3^1) - (x_2 - x_1 + \tilde{\lambda}(h_2^1 - h_1^1))(y_3 + \tilde{\lambda}h_3^2) + x_2 y_1 - x_1 y_2 + \tilde{\lambda}(y_1 h_2^1 - y_2 h_1^1) + \tilde{\lambda}^2(x_2 h_1^2 - x_1 h_2^2) = 0$$

we yield to

$$\Lambda \tilde{\lambda}^2 + S \tilde{\lambda} + Q = 0$$

where

$$\begin{aligned} \Lambda &= x_2 h_1^2 - x_1 h_2^2 + h_3^1(h_2^2 - h_1^2) - h_3^2(h_2^1 - h_1^1) \\ S &= h_3^1(y_2 - y_1) + x_3(h_2^2 - h_1^2) - h_3^2(x_2 - x_1) - y_3(h_2^1 - h_1^1) + y_2(h_2^1 - h_1^1) \\ Q &= x_2 y_1 - x_1 y_2 - y_3(x_2 - x_1) \end{aligned}$$

if we have $\Lambda = 0$, we simply derive $\tilde{\lambda} = -\frac{Q}{S}$; or else we can as well get $\tilde{\lambda}$ by the quadratic formula,

$$\tilde{\lambda} = \frac{-S \pm \sqrt{S^2 - 4\Lambda Q}}{2\Lambda} \quad (4)$$

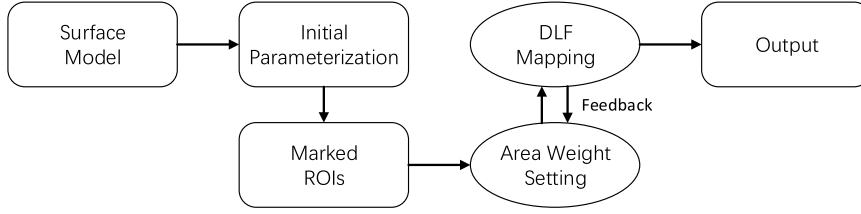


Fig. 4. The pipeline of our DLF based measure controllable framework.

if we have the quadratic discriminant $S^2 - 4\Lambda Q \geq 0$.

For each triangle $t \in F$, we pick only the minimal positive real value of $\tilde{\lambda}$ and note that value as $\tilde{\lambda}_t^+$, which stands for the maximum step for the flip-preventing condition of triangle t . If $\tilde{\lambda}_t^+$ does not exist – e.g. both the roots are negative, or the quadratic discriminant is negative – we alternatively assume it to be $+\infty$, which means t itself has a flip-preventing gradient for any positive steps. Then the adaptive step-length λ for the entire mesh, finally, can be derived through

$$\lambda = \mu \cdot \min(\tilde{\lambda}_t^+) \quad (5)$$

where $\mu < 1$ is a ratio value. In our experiments, μ can be a relatively considerable proportion from 0.2 to 0.8, which promotes fast and stable convergence. The time complexity of computing one adaptive step-length is $O(n)$, and is ignorable under that of solving a Poisson equation. In later experiments, without special remarks, we will always use this scheme to select steps.

Generally, adaptive step-length prevents triangle flipping intrinsically, as it is taken from the step-length that brings about the very first triangle flipping. Although it could be imagined that a vertex might keep tending towards an edge infinitely and cause a degeneration to line, the edge-flipping operation we proposed previously would then replace the edge that is getting in the way, so that the vertex could move on flowing.

5.5. Measure controllable parameterizations and ROIs

Another prominent features of our approach and is that it can be applied to generate not only area-preserving mapping but also measure controllable parameterizations. We first mark our regions of interests, known as ROIs, then we multiply a vector density function $\rho(x)$ with the area elements on the original model g_M . We can write it mathematically

$$\hat{g}_M = \rho(x) \otimes g_M \quad (6)$$

Actually, the thought behind our approach is very natural. In area-preserving mappings, g_Ω is iteratively moving closer to g_M ; while in measure controllable conditions, we replace g_M by $\hat{g}_M$ so as the derived parameterization can have arbitrary regions enlarged or shrunk.

5.6. Algorithm

We summarize algorithm of our methods for measure controllable parameterization as Algorithm 1. By simply assigning $\hat{g}_M = g_M$, we can also obtain an area-preserving parameterization from the pipeline, since it is a special case of measure controllable parameterization (see Fig. 4).

Algorithm 1 Discrete Lie Flow for measure controllable parameterization.

Input:

The original model M ; an initial parameterization $f : M \rightarrow \Omega$;
prescribed measure density function $\rho(x)$

Output:

A mapping $\phi : \Omega \rightarrow \Omega$, such that $u = \phi \circ f$ is the expected measure controllable mapping.

- 1: Initialize $\phi(v_i) = v_i$ and $h_M = (f_* (\omega_M))$;
 - 2: Optimize the mesh on parameter domain Ω by edge-flipping;
 - 3: Compute $h_\Omega = (\rho(x)\phi^*(\omega_\Omega))$ and scale it, such that $\int h_\Omega = \int h_M$;
 - 4: Solve Poisson Equation $\Delta g = h_M - h_\Omega$ with Neumann boundary condition $\partial h / \partial n = 0$;
 - 5: Compute vertex gradient ∇g from g ;
 - 6: Compute adaptive step-length $\lambda = \mu \cdot \min(\tilde{\lambda}_t^+)$;
 - 7: Apply displacement vector field to ϕ , such that $\phi(v_i) = \phi(v_i) + \lambda \nabla g$;
 - 8: Repeat step 2 to step 7 until $\|h_M - h_\Omega\|^2 < \text{threshold}$;
 - 9: **return** ϕ .
-

Table 1
Four kinds of distortions.

Wasserstein Distance	$\sqrt{\sum_{t=0}^n (S_t - \hat{S}_t)^2}$
D^{area}	$\sum_{t=1}^n \rho_t (\sigma_{1,t} \sigma_{2,t} + \frac{1}{\sigma_{1,t} \sigma_{2,t}})$
ARAP	$\sum_{t=1}^n \rho_t [(\sigma_{1,t} - 1)^2 + (\sigma_{2,t} - 1)^2]$
Symmetric Dirichlet	$\sum_{t=1}^n (\sigma_{1,t}^2 + \sigma_{1,t}^{-2} + \sigma_{2,t}^2 + \sigma_{2,t}^{-2})$



Fig. 5. Area-preserving mapping to a square for Buddha via DLF using harmonic mapping as initial parameterization.

6. Experimental results

To demonstrate the efficiency and robustness of our algorithm, we test our algorithm on hundreds of meshes with a variety of initial parameterizations, including topological disk meshes with circle, square and irregular non-convex boundaries, and topological sphere meshes. More importantly We have implemented our algorithms using parallel C++ on a Windows 10 platform, with 4-core 3.2 GHz i5-6500 CPU, 8 GB DDR4 memory. All the geometric data come from public resources, either manually modeled, acquired from real life by laser scanning or reconstructed from geological data. All the geometric data are represented as triangle meshes.

Measurements Usually in computer graphics, energies are introduced to measure the area and angle distortion. Our DLF are not designed to minimize a certain energy, yet we will use four area-related energies for convenience to compare with other methods. Wasserstein Distance, As-Rigid-As-Possible(ARAP), Symmetric Dirichlet and D^{area} , these four kinds of measurements are listed in Table 1. We have to stress that our DLF, unlike other methods based on energy optimizing, is not designed to minimize any of these energies, through it may appear to perform well in some. We cite these energies for a comprehensive evaluation of our method, and for convenience in comparing with other unmentioned methods.

Benchmark We test the practical robustness on a data set consisting of hundreds of models. Our method successfully finds the area-preserving parameterizations for all models with arbitrary ROIs.

6.1. Area-preserving parameterizations

We handle models with different topology and different kinds of initial parameterizations to realize ordinary area-preserving parameterizations and demonstrate its efficiency and efficacy. Fig. 5 and Fig. 6 shows DLF on a topological disk model with squared boundary, and Fig. 7 shows a detailed procedure of DLF on a topological disk model with circle boundary. The initial parameterization methods we used here are Pinkall and Polthier (1993), Tutte (1963) and Jin et al. (2008).

We also noticed that for many low-stretch parameterization approaches, they require the convexity of parameterization domain, otherwise, for example, OMT in the interior is homeomorphic, but the homeomorphism cannot be extended to the boundary. DLF overcomes this shortcoming, and doesn't need the convexity; as is shown in Fig. 10, we map the Monkey onto an arbitrary domain and implement the DLF to it to generate a non-convex area-preserving mapping. While Fig. 8 and Fig. 9 are good examples for topological sphere models. The method we used here to generate initial parameterizations is CMC (Kazhdan et al., 2012).

Accuracy To verify the accuracy of the computational results visually we make 2 histograms of logarithmic area ratios of model Maxplanck and model Buddha. More precisely, for each vertex i we compute corresponding the logarithmic area ratio p_i by

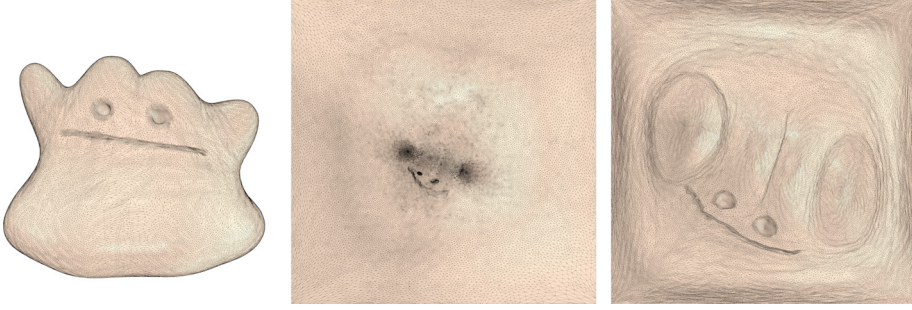


Fig. 6. Area-preserving mapping of Slime to a square via DLF using Tutte embedding as initial parameterization.

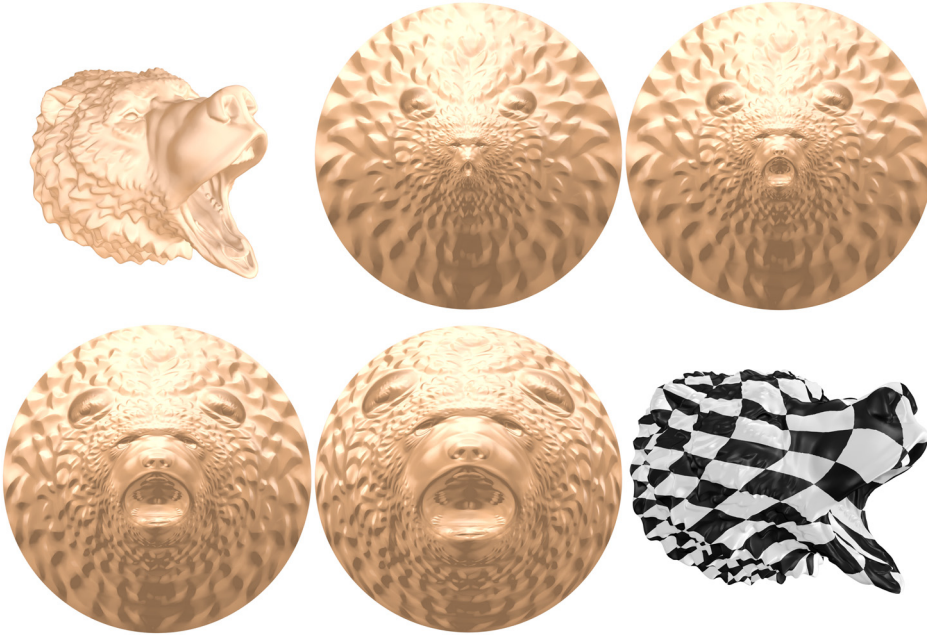


Fig. 7. Area-preserving mapping of Bear to a disk via DLF using Riemann mapping as initial parameterization.



Fig. 8. DLF deformation of Yoda to a sphere.

$$p_i = \log\left(\frac{\mathcal{A}_\Omega(i)}{\mathcal{A}_M(i)}\right)$$

where $\mathcal{A}_\Omega(i)$ is the area element of vertex i of Ω computed by Eq. (2), and $\mathcal{A}_M(i)$ is that of M . The result is shown in Fig. 11, where initial parameterizations are colored in orange and DLF area-preserving parameterizations are colored in blue. As can be seen directly, area ratios of the initial parameterizations are distributed dispersedly; and right after DLF,

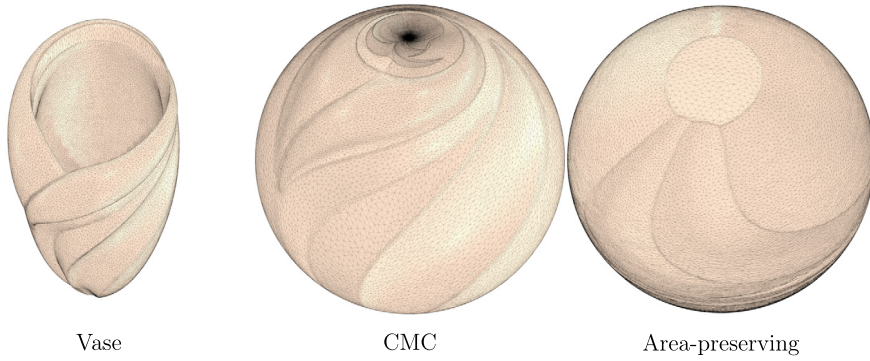


Fig. 9. DLF deformation of Vase to a sphere. The original model is like a double-cover surface.

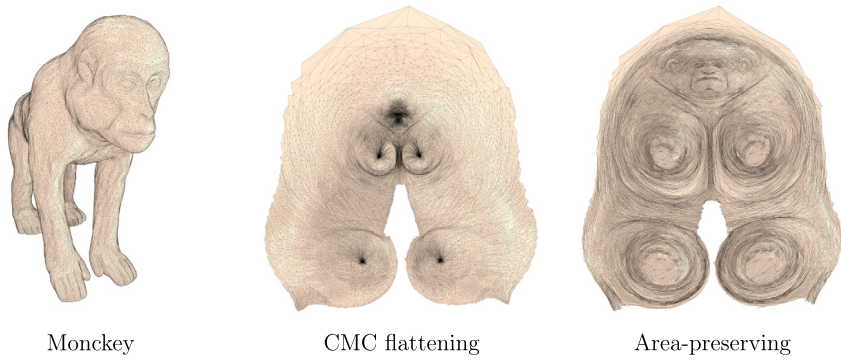


Fig. 10. DLF deformation of model with non-convex boundary proves that it can handle non-convex domains.

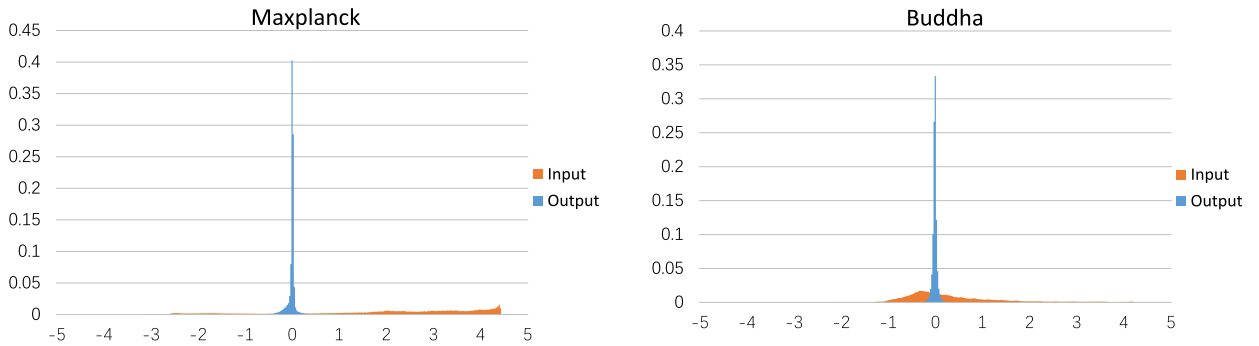


Fig. 11. Histograms of logarithmic area ratios between input initial parameterizations and output area-preserving parameterizations derived via DLF. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

they are obviously centralized to 0, which means areas of the derived parameterization are highly concordant with those of the original mesh. For rigorous numerical comparisons we will provide them in later sections.

6.2. Adaptive step-length scheme

Through these two experiments Fig. 12 and Fig. 13, we can have a glimpse of the fact that DLF is unpredictably sensitive to step-lengths. It is hard to determine even the rough magnitude by hand or by experience, for it differs enormously in various cases. Due to insufficient investigations in selecting steps, previous works are far from being applicable.

Our adaptive step-lengths determined by the formula Eq. (5) is the mathematical interpretation of the flip-preventing condition that guarantees mesh will not collapse in midway of the flowing. It determines a safe and effective step reliably for every single steps of DLF. In Fig. 12 our adaptive step-lengths reduce the number of iterations needed from 548 to 25 on model David; and in Fig. 13 especially, our adaptive step-lengths make DLF accomplish in 37 iterations, which is not excessive to call a miracle in previous discrete Lie derivative methods. For more cases, we refer readers to Table 2.

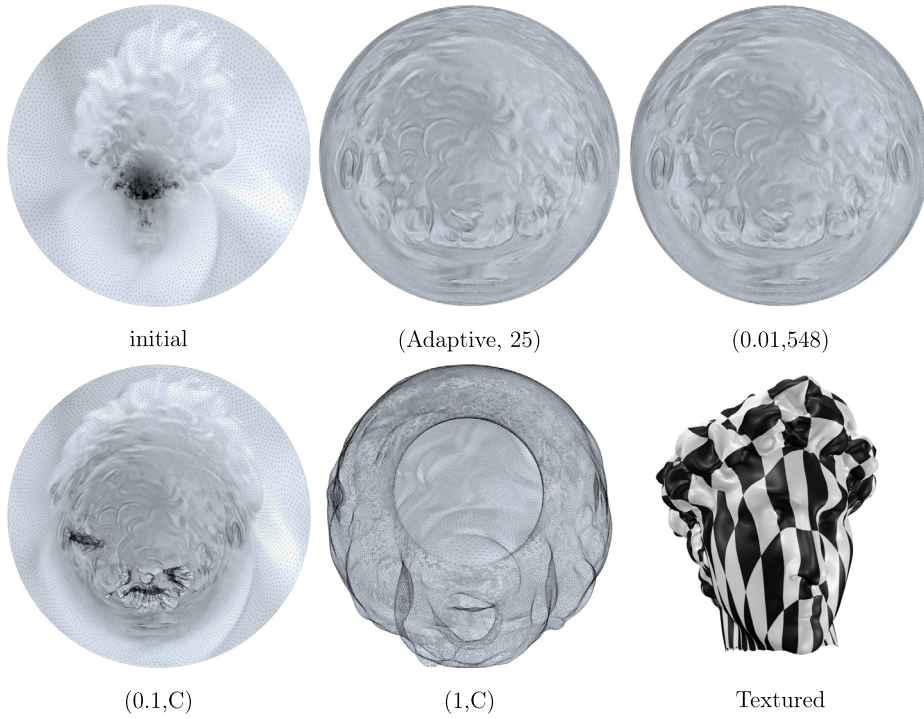


Fig. 12. Flowing results with different step choices of model David. The first entry of the binary below represents the scale of step, the second represents the number of iterations. C means the midway collapsing of mesh.

Table 2

The number of iterations for various models with step 0.1, 0.01, and 0.001, where c means the collapsing of mesh.

Model	#F	0.1	0.01	0.001	Adaptive
Man	347360	c	681	>2000	56
Chinese lion	100002	c	c	3975	48
Bimba	11253	c	369	>2000	21
Superman	190471	c	525	>2000	133
Buddha	470507	c	657	>2000	34
Dragon head	26843	c	c	c	66
Moai	16493	c	378	>2000	13
Mouse	25440	c	461	>2000	23
Horse	21423	c	c	c	19
Max Planck	84705	c	594	>2000	50
Bodhisattva	160098	c	c	4050	106

According to Fig. 14, we can see that the absolute value of slope of the line representing the DLF with our adaptive step-length is prominently larger than that with (a most successful) fixed step. With our adaptive step-length, area-preserving mappings can usually be derived in merely dozens of iterations.

6.3. Measure controllable parameterizations

By adding a density function Eq. (6) to the area elements of the original mesh M , our DLF can realize the measure controllable purpose to get any regions of interests (ROIs) enlarged or shrunk to produce any user-defined parameterizations accurately. For testing the totipotency of our approach, we do the following experiment for a topological sphere Earth model. As is shown in Fig. 15, starting from the ordinary Earth model with texture, we apply DLF to scale Europe by different factors. Another Fig. 16 demonstrates our measure controllable parameterizations for Killer Crock model possessing a disk topology.

6.4. Comparisons with other methods

Our DLF method is compared with several state-of-the-art parameterization methods to demonstrate its accuracy and efficiency. In Yoshizawa et al. (2004), they proposed a fast and simple methods for stretch-minimizing parameterization

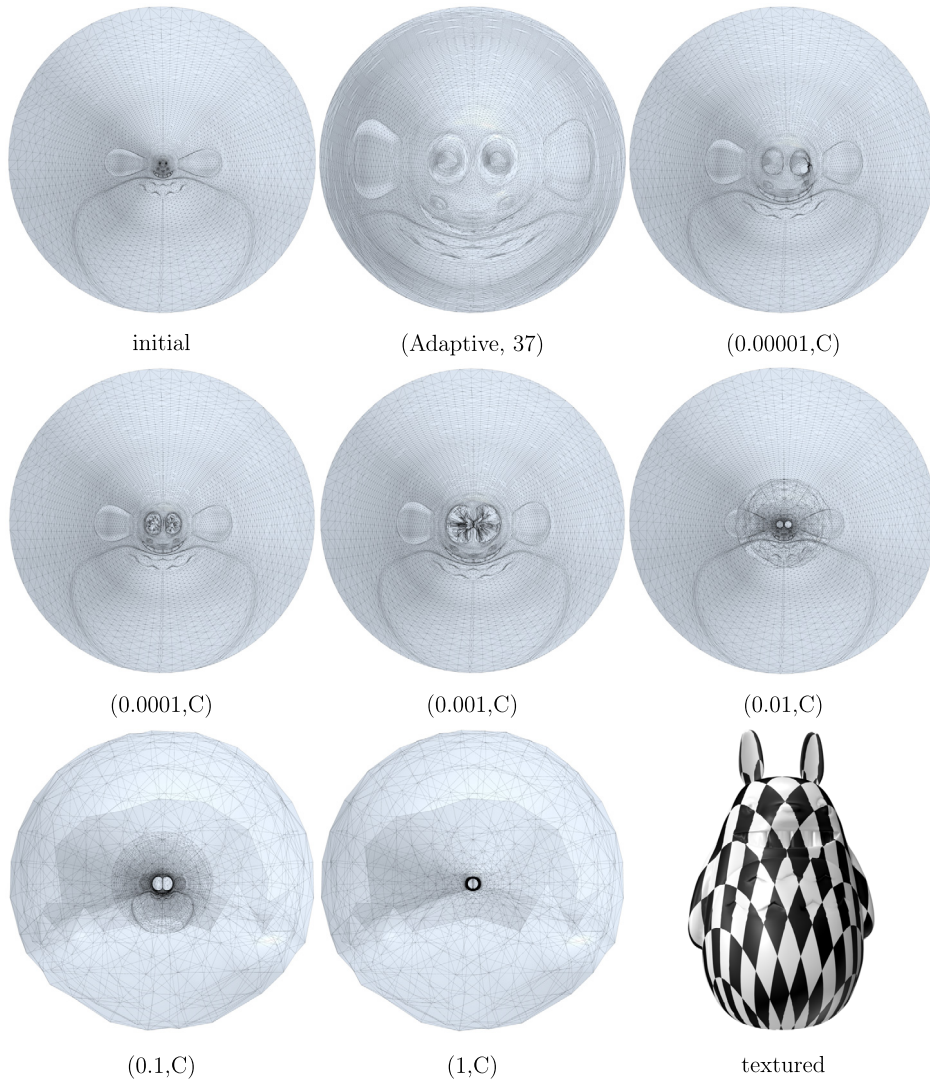


Fig. 13. Flowing results with different step choices of model Totoro. The first entry of the binary below represents the scale of step, the second represents the number of iterations. C represents the mesh collapsed.

with square boundaries, for convenience, we will call their method FSP. Later in the work Su et al. (2016a) and (2016b), the OMT is utilized as an efficient way to generate area-preserving and measure controllable parameterization, and their code implementation handles circle boundaries. When comparing with FSP and OMT, we will use DLF with their corresponding boundary shapes to eliminate unnecessary variables, and note DLFC for DLF with circle boundaries, DLFS for DLF with square boundaries. And, for topological sphere, we will compare with the method in Hu et al. (2018), where the authors proposed an efficient method for spherical parameterization, and for convenience, we will call it AHSP.

Timings Generally speaking, the typical running times of the DLF for a model with dozens of thousands of vertices are usually in ten seconds. The detailed results are listed in Table 3 and Table 5. Compared to FSP, our DLF is usually slightly slower with small models with no more than 100K faces, but in larger cases we become drastically faster. Compared to OMT, although with much reputation, our DLF is much faster in all cases, and the disparity becomes larger for huge models with more than 200K faces. Compared to AHSP, our method is also superior in speed.

Distortion The measurements are summarized in Table 4 and 3. We can see that the area-preserving ability of DLF, which is denoted by D^{area} , is outstanding among both the other methods with no doubt with absolute predominance, even beyond the theoretically optimal OMT. However, we have to admit that, for other energies that more or less take angle distortions into account, we won't deny that our DLF may not behave the best. We will also list these energies in our following tables for readers to have an objective understanding of DLF.

Table 3

Timing comparison (in sec.) for Authalic parameterization between different methods.

Model	#F	#V	OMT.t	DLFC.t	FSP.t	DLFS.t
Bear	296K	148K	439	45	91	39
Bimba	11K	56K	10	2	1	2
David	48K	24K	33	6	4	6
Buddha	470K	235K	594	34	135	36
Maxplanck	84K	42K	58	8	8	5
Nicolo	99K	50K	52	4	8	4
Slime	103K	51K	68	9	12	8
Superman	190k	96k	1989	47	40	33
Torsowoman	88k	44k	51	5	6	3
Totoro	20k	10k	28	7	1	7

Table 4

Measurements of several different energies of DLF, OMT and FSP. SD is short for Symmetric Dirchlet.

Model	#F	#V	Methods	D^{area}	SD	ARAP
Bear	296K	148K	OMT	3.19	$1.01 * 10^{10}$	1.21
			DLFC	2.07	$1.21 * 10^8$	0.79
			FSP	2.10	$2.33 * 10^6$	1.31
			DLFS	2.08	$2.27 * 10^7$	1.09
Bimba	11K	5K	OMT	2.06	$3.22 * 10^5$	2.25
			DLFC	2.09	$2.50 * 10^5$	2.22
			FSP	2.24	$1.23 * 10^5$	3.40
			DLFS	2.24	$9.41 * 10^6$	2.86
Buddha	470K	235K	OMT	2.46	$5.69 * 10^9$	0.63
			DLFC	2.02	$8.97 * 10^6$	0.39
			FSP	2.17	$2.93 * 10^6$	0.35
			DLFS	2.02	$4.10 * 10^6$	0.56
David	48K	24K	OMT	2.44	$2.48 * 10^9$	1.85
			DLFC	2.08	$8.73 * 10^5$	1.58
			FSP	2.23	$3.96 * 10^5$	1.89
			DLFS	2.11	$1.61 * 10^6$	2.51
Maxplanck	84K	42K	OMT	2.04	$1.67 * 10^6$	1.72
			DLFC	2.03	$3.28 * 10^6$	1.62
			FSP	2.22	$6.95 * 10^5$	1.83
			DLFS	2.05	$6.51 * 10^6$	2.51
Nicolo	100K	50K	OMT	2.18	$1.57 * 10^7$	0.37
			DLFC	2.02	$3.06 * 10^6$	0.22
			FSP	2.11	$5.24 * 10^5$	0.36
			DLFS	2.01	$7.40 * 10^5$	0.49
Slime	103K	51K	OMT	3.01	$1.36 * 10^8$	0.83
			DLFC	2.05	$1.17 * 10^7$	0.52
			FSP	2.05	$6.21 * 10^5$	0.70
			DLFS	2.04	$1.88 * 10^6$	0.75
Totoro	20K	10K	OMT	2.33	$4.38 * 10^8$	1.61
			DLFC	2.09	$6.18 * 10^5$	1.23
			FSP	2.19	$2.39 * 10^5$	2.55
			DLFS	2.14	$7.22 * 10^6$	1.83

Table 5

Comparison of distortion and running time between DLF and AHSP.

Model	#F	#V	Methods	Time	Areatotal	D^{area}
Armadillo	92K	46K	DLF	15	0.21	2.18
			AHSP	105	0.67	2.85
Armchair	100K	50K	DLF	10	0.06	2.01
			AHSP	54	0.25	2.12
Bear	35K	17K	DLF	5	0.08	2.05
			AHSP	57	0.55	2.51

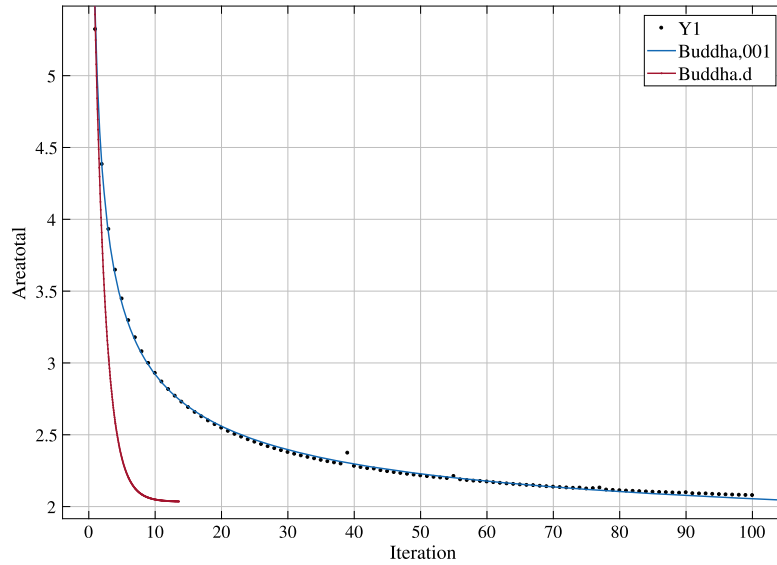


Fig. 14. D^{area} decreases along the iterations. Records of our adaptive step-length is colored in red, while that of a tentative step 0.01 is colored in blue.

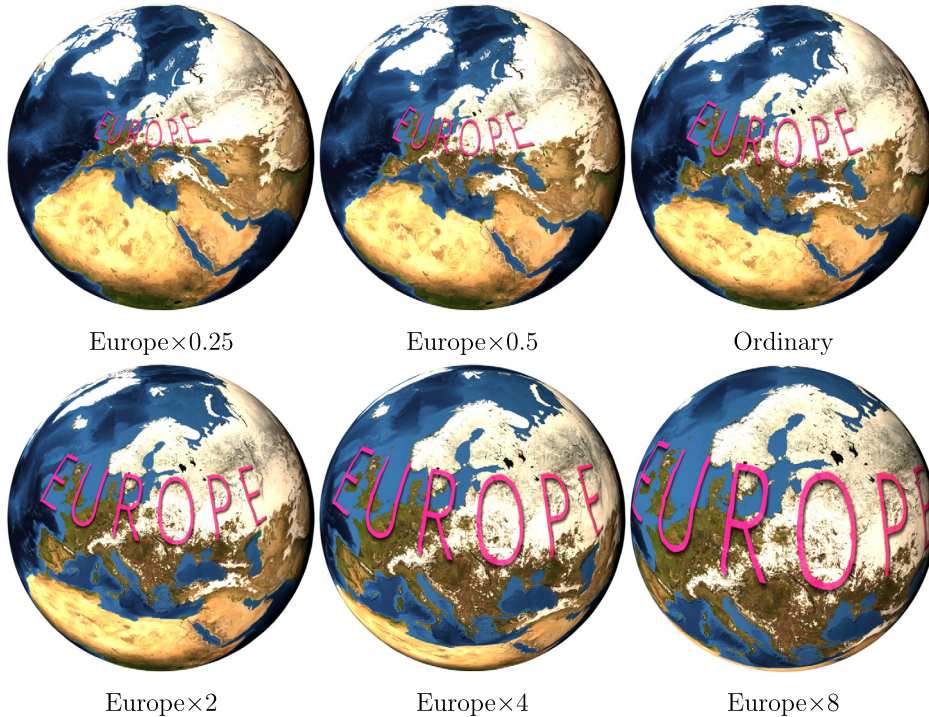


Fig. 15. Measure controllable mapping for Earth. Europe is selected as ROIs, the factor is listed below.

6.5. Measure controllable parameterization and comparisons with OMT

In theory, DLF is capable of obtaining measure preserving maps, however we don't have a proof that it is the optimal solution. By the optimal solution we mean the Wasserstein distance of it is smallest among the set of area preserving mappings. The OMT method (Optimal Mass Transport, see Dominitz and Tannenbaum (2010), Zhao et al. (2013), Su et al. (2016a) and (2016b)) is another approach for area-preserving and measure controllable parameterization with a good reputation, and from the mathematical perspective, the parameterization derived through the OMT is the optimal solution.

We design an experiment to quantify the difference between DLF and OMT in order to demonstrate the between DLF and the theoretical optimality. Let (S, g) be a topological disk with associated Riemannian metric, ϕ_0 and $\phi_1: (S, g) \rightarrow \mathbb{D}^2$, where ϕ_0 is obtained by OMT, ϕ_1 by DLF. Results of these two mappings are shown in Fig. 17 and Fig. 18. We have

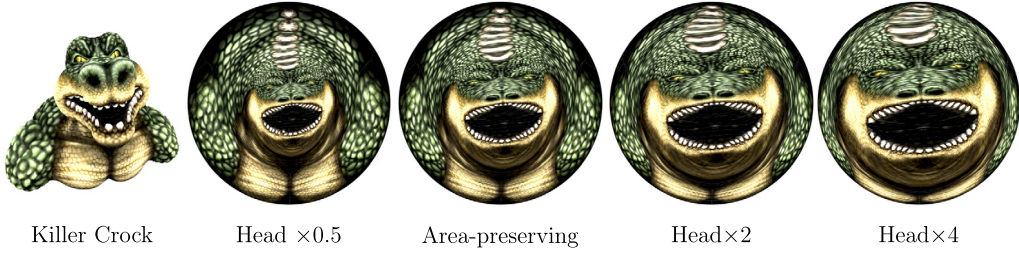


Fig. 16. Measure controllable mapping for Killer Croc. Head is selected as ROIs, the scaling factor is listed below.

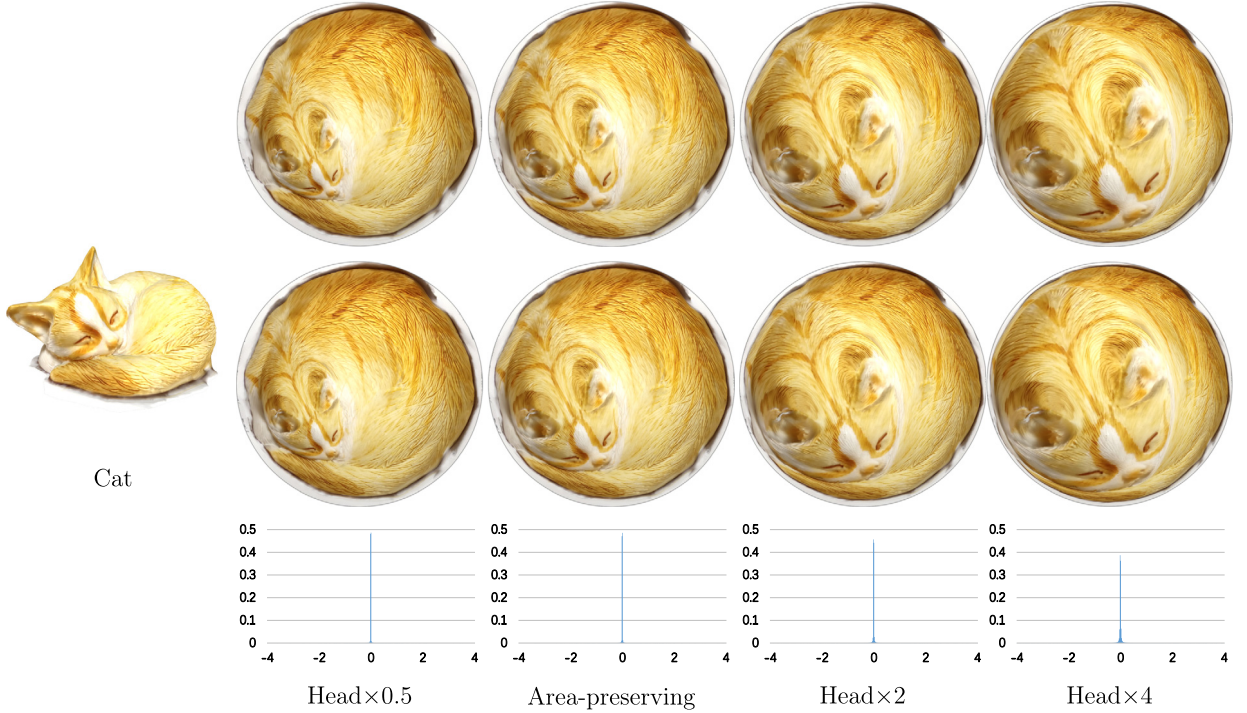


Fig. 17. Comparison between DLF and OMT for measure controllable parameterization of model Cat. We select the head of cat as ROIs and the deformation is depicted above. In turn, the measure head is shown by the factor of 0.5, 1, 2 and 4. The above row is derived via DLF, the intermediate row is computed by OMT, and the bottom row is the frequency histogram of the norm of the Beltrami coefficient μ_ψ . The result shows that there is no significant difference between DLF and OMT.

$$\psi = \phi_1 \circ \phi_0^{-1} : (\mathbb{D}^2, dx \wedge dy) \rightarrow (\mathbb{D}^2, dx \wedge dy)$$

As shown in Fig. 17 and Fig. 18, we use the Beltrami coefficient μ_ψ of ψ to measure the difference between ϕ_0 and ϕ_1 quantitatively. Recall that the Beltrami coefficient can be computed as follows:

$$\mu_\psi = \frac{\partial \psi}{\partial \bar{z}} / \frac{\partial \psi}{\partial z}$$

Let the coordinates on the result domain of the DLF and OMT be (x,y) , (u,v) respectively. Then the Beltrami coefficient can be computed through the formula:

$$\mu_\psi = \frac{\beta}{\alpha}$$

where $\alpha = \frac{yu+xv}{2xy}$ and $\beta = \frac{yu-xv}{2xy}$. The Beltrami represents the distortion, it will tend to zero as ψ approximates the identity. The Beltrami coefficients of ψ for different models are depicted in the bottom row of Fig. 17 and Fig. 18. We can see from the histogram that the Beltrami coefficients of ψ are concentrated at 0, which means the results of DLF are extremely similar to those of OMT.

Finally, we also compute the transportation cost through Wasserstein distance. Generally speaking, a difference in Wasserstein distance within 3% is a symbol for highly resembling each other. As is shown in Table 6, the average dif-

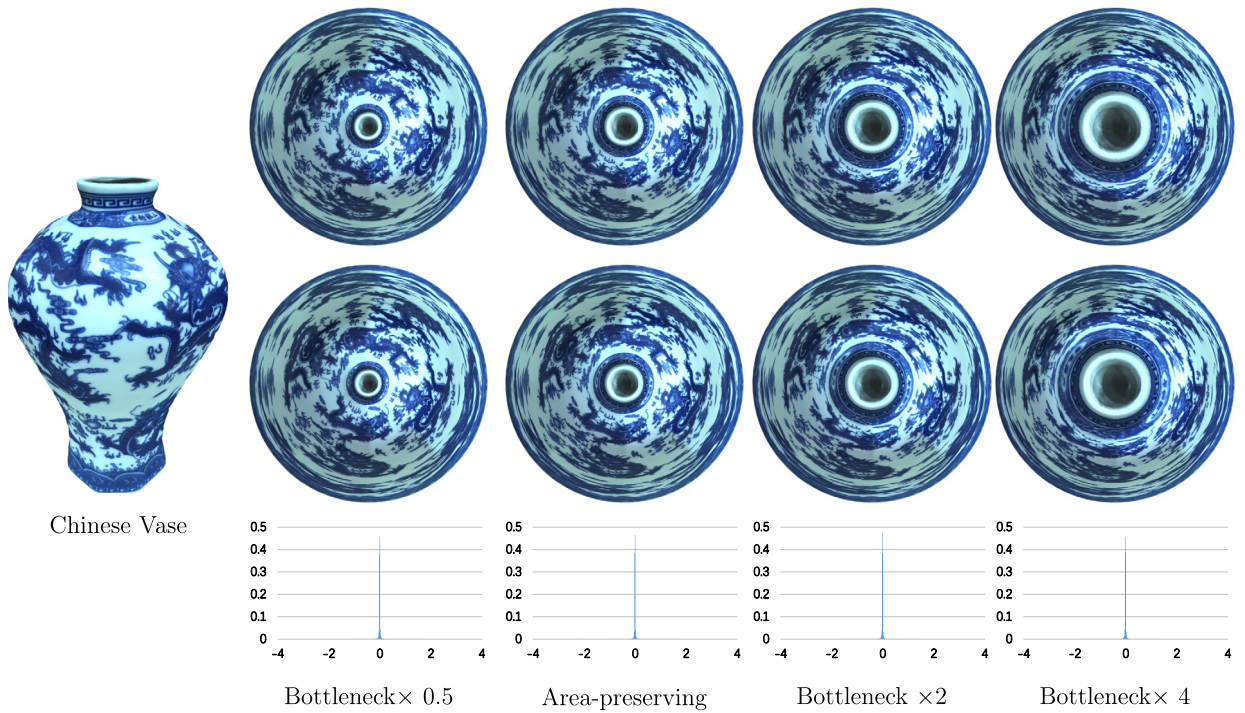


Fig. 18. Comparison between DLF and OMT for measure controllable parameterization of Chinese Vase. Measure controllable mapping for model: Chinese Vase. We select the bottleneck of Chinese Vase as ROIs and the deformation is depicted above. In turn, the bottleneck is showed by the factor of 0.5, 1, 2 and 4. The above row is derived via DLF, the intermediate row is computed by OMT, and the bottom row is the frequency histogram of the norm Beltrami coefficient μ_ψ . The result shows that there is no significant difference between DLF and OMT.

Table 6

Similarity comparison between DLF and OMT. Here we use the sum of squared Euclidean distance between DLF/OMT and the original parameterization as measure.

Model	DLF	OMT	Difference
Cat×0.5	3299.30239	3306.899436	0.2302622%
Cat×1	3457.758768	3457.04257	0.0207128%
Cat×2	6714.249024	6692.770138	0.3199001%
Cat×4	13700.85689	13461.32009	1.7483345 %
ChineseVase×0.25	12008.67629	12237.21035	1.9030745%
ChineseVase×0.5	12415.7497	12647.75603	1.8686454%
ChineseVase×1	13163.49626	13429.23322	2.0187415 %
ChineseVase×2	14573.754	14844.32246	1.8565456%
ChineseVase×4	16934.09609	17189.995955	1.5111516%
ChineseVase×8	20421.18237	20564.54963	0.7020517%
ChineseVase×16	24719.81954	24550.11919	0.6864951%

ference of 2 models among 11 different measures is 1.16%, and the maximum is 2.01%. This result also indicates the high accuracy of DLF when compared with OMT.

7. Summary

We present a efficient and robust method based on discrete Lie derivative to compute area-preserving parameterizations, and further on employ it to generate measure controllable parameterizations which is demanding in various applications such as physical modeling and medical imaging. Secondly, we invent the Adaptive Step-length Scheme and solve the evasive conundrum of selecting steps, which greatly enhances the efficiency and robustness, and make discrete Lie derivative reliable in applications for the first time. For parameterization methods, it is usually difficult to equip the algorithm with a solid mathematical framework, so researchers are more or less lacking in convincingness, but our DLF has a solid one.

7.1. Admissions

Due to constraints from a variety of aspects, we failed to obtain more abundant examples of other methods to give readers a more comprehensive view on the competitive power of our DLF method. And we also noticed that, in spherical cases, although DLF is pressure-free in the flowing procedure, a significant challenge comes from the long-existed problem in generating the initial parameterization for models with long protrudent parts. In this paper, we use conformal parameterizations as initialization mostly because it's fast to compute; however, in precious few occasions, some triangles were deformed so small that they reached the limit of double-precision floating points, and as a result DLF is unable to start flowing.

7.2. Future works

A fully reliable initial parameterization method for all kinds of models is in demand. We also need to compare more existing initializing methods and analysis the pros and cons for DLF in particular. Besides, although we have proved that DLF is capable for high dimensional situations such as generating volume-preserving parameterizations for tetrahedron meshes, yet more researches and practices need to be done to migrate DLF into higher dimensionality.

Currently, our DLF uses edge-flipping operations to keep the mesh in Delaunay triangulation and avoid singularities of cotangent values in the Laplacian. This is necessary for pushing forward the flowing process, but it also causes changes in topology. When we use the resulting vertex coordinates with the original connectivity, chances are that a minor of triangles may flip. For applications that are sensitive to the possible flips, we refer readers to the work by Liu et al. (2015) and that by Yi et al. (2018). They studied on how to construct local Delaunay without changing the connectivity.

Our method is adaptable to the Riemannian manifold in theory, although we currently realized it on 2-dimensional manifolds embedded in 3-dimensional Euclidean space in this paper. The overall framework is clear: firstly, we calculate the changes in the area elements (or volume elements, for volumetric meshes) of the vertices; secondly, we apply the changes to edge lengths in the metric space. The detailed implementation, however, still needs exploring in future works.

Declaration of Competing Interest

There is no competing interest.

Acknowledgements

The authors acknowledge the support from National Natural Science Foundation of China (Grant No. 61772379, No. 61772105, No. 61720106005, No. 61432003), the National Science Foundation of the United States (Grant No. CMMI-1762287, DMS-1737812), the grant from Ford University Research Program (URP) (Award No. 2017-9198R). The authors are also grateful to anonymous referees for their helpful comments and suggestions.

References

- Aigerman, N., Lipman, Y., 2015. Orbifold tutte embeddings. In: *International Conference on Computer Graphics and Interactive Techniques* 34 (6), 190.
- Aigerman, N., Poranne, R., Lipman, Y., 2015. Seamless surface mappings. In: *International Conference on Computer Graphics and Interactive Techniques* 34 (4), 72.
- Angenent, S., Haker, S., Tannenbaum, A., Kikinis, R., 2000. On area preserving mappings of minimal distortion. In: *System Theory*. Springer, pp. 275–286.
- Bonnotte, N., 2013. From knothe's rearrangement to brenier's optimal transport map. *SIAM J. Math. Anal.* 45 (1), 64–87.
- Bright, A., Chien, E., Weber, O., 2017. Harmonic global parametrization with rational holonomy. *ACM Trans. Graph.* 36 (4), 89. <https://doi.org/10.1145/3072959.3073646>.
- Dominitz, A., Tannenbaum, A., 2010. Texture mapping via optimal mass transport. *IEEE Trans. Vis. Comput. Graph.* 16 (3), 419–433.
- Floater, M.S., 1997. Parametrization and smooth approximation of surface triangulations. *Comput. Aided Geom. Des.* 14 (3), 231–250.
- Gu, X., Yau, S., 2003. Global conformal surface parameterization, pp. 127–137.
- Hormann, K., Greiner, G., 2000. Mips: An Efficient Global Parametrization Method. Tech. Rep., Erlangen-Nuernberg Univ. (Germany), Computer Graphics Group.
- Hormann, K., Polthier, K., Sheffer, A., 2008. Mesh parameterization: theory and practice, pp. 1–87.
- Hu, X., Fu, X.-M., Liu, L., 2018. Advanced hierarchical spherical parameterizations. *IEEE Trans. Vis. Comput. Graph.* 24 (6), 1930–1941.
- Jin, M., Kim, J., Luo, F., Gu, X., 2008. Discrete surface ricci flow. *IEEE Trans. Vis. Comput. Graph.* 14 (5), 1030–1043. <https://doi.org/10.1109/TVCG.2008.57>.
- Kazhdan, M., Solomon, J., Benchen, M., 2012. Can mean-curvature flow be modified to be non-singular? In: *Computer Graphics Forum*, pp. 1745–1754.
- Lei, N., Su, K., Cui, L., Yau, S.-T., Gu, X.D., 2019. A geometric view of optimal transportation and generative model. *Comput. Aided Geom. Des.* 68, 1–21.
- Levy, B., Petitjean, S., Ray, N., Maillot, J., 2002. Least squares conformal maps for automatic texture atlas generation, international conference on computer graphics and interactive techniques 21 (3), 362–371.
- Levy, H., 1964. Foundations of differential geometry. vol. 1. Shoshichi Kobayashi and Katsumi Nomizu. Interscience (Wiley), New York, 1963. xii + 329 pp. *Illus. Science* 143 (3603), 235.
- Liu, Y.-J., Xu, C.-X., Fan, D., He, Y., 2015. Efficient construction and simplification of delaunay meshes. *ACM Trans. Graph.* 34 (6), 174.
- Pinkall, Ulrich, Polthier, Konrad, 1993. Computing discrete minimal surfaces and their conjugates. *Exp. Math.* 2 (1), 15–36.
- Rabinovich, M., Poranne, R., Panozzo, D., Sorkine-Hornung, O., 2017. Scalable locally injective mappings. *ACM Trans. Graph.* 36 (2), 16. <https://doi.org/10.1145/2983621>.
- Sheffer, A., Praun, E., Rose, K., 2006. Mesh parameterization methods and their applications. *Found. Trends Comput. Graph. Vis.* 2 (2), 105–171.
- Su, K., Chen, W., Lei, N., Cui, L., Jiang, J., Gu, X.D., 2016b. Measure controllable volumetric mesh parameterization. *Comput. Aided Des.* 78 (1), 188–198.

- Su, K., Cui, L., Qian, K., Lei, N., Zhang, J., Zhang, M., Gu, X.D., 2016a. Area-preserving mesh parameterization for poly-annulus surfaces based on optimal mass transportation. *Comput. Aided Geom. Des.* 46, 76–91. <https://doi.org/10.1016/j.cagd.2016.05.005>. <http://www.sciencedirect.com/science/article/pii/S0167839616300528>.
- Su, K., Chen, W., Lei, N., Zhang, J., Qian, K., Gu, X., 2017. Volume preserving mesh parameterization based on optimal mass transportation. *Comput. Aided Des.* 82, 42–56.
- Su, K., Lei, N., Chen, W., Cui, L., Si, H., Chen, S., Gu, X., 2019. Curvature adaptive surface remeshing by sampling normal cycle. *Comput. Aided Des.* 111, 1–12.
- Tutte, W.T., 1963. How to draw a graph. *Proc. Lond. Math. Soc.* s3–13 (1), 743–767.
- Van Kaick, O., Zhang, H., Hamarneh, G., Cohen-Or, D., 2011. A Survey on Shape Correspondence. *Computer Graphics Forum*, vol. 30. Wiley Online Library, pp. 1681–1707.
- Wang, C.C., Smith, S.S., Yuen, M.M., 2002. Surface flattening based on energy model. *Comput. Aided Des.* 34 (11), 823–833.
- Wang, Y., Zhang, J., Chan, T.F., Toga, A.W., Thompson, P.M., 2009. Brain surface conformal parameterization with holomorphic flow method and its application to hiv/aids. *NeuroImage* 47.
- Wang, Y., Shi, J., Yin, X., Gu, X., Chan, T.F., Yau, S.-T., Toga, A.W., Thompson, P.M., 2012. Brain surface conformal parameterization with the ricci flow. *IEEE Trans. Med. Imaging* 31 (2), 251–264.
- Yi, R., Liu, Y.-J., He, Y., 2018. Delaunay mesh simplification with differential evolution. In: *SIGGRAPH Asia 2018 Technical Papers*. ACM, p. 263.
- Yoshizawa, S., Belyaev, A., Seidel, H., 2004. A fast and simple stretch-minimizing mesh parameterization. In: *Proceedings. Shape Modeling International 2004 (SMI)*, pp. 200–208.
- Zeng, W., Gu, X.D., 2013. *Ricci Flow for Shape Analysis and Surface Registration: Theories, Algorithms and Applications*. Springer Science & Business Media.
- Zhao, H., Su, K., Li, C., Zhang, B., Yang, L., Lei, N., Wang, X., Gortler, S.J., Gu, X., 2019. Mesh parametrization driven by unit normal flow. In: *Computer Graphics Forum*. Wiley Online Library, <https://doi.org/10.1111/cgf.13660>.
- Zhao, X., Su, Z., Gu, X.D., Kaufman, A.E., Sun, J., Gao, J., Luo, F., 2013. Area-preservation mapping using optimal mass transport. *IEEE Trans. Vis. Comput. Graph.* 19 (12), 2838–2847.
- Zhu, L., Haker, S., Tannenbaum, A., 2003. Area-preserving mappings for the visualization of medical structures, pp. 277–284.
- Zou, G., Hu, J., Gu, X., Hua, J., 2011. Authalic parameterization of general surfaces using lie advection. *IEEE Trans. Vis. Comput. Graph.* 17 (12), 2005–2014. <https://doi.org/10.1109/TVCG.2011.171>.