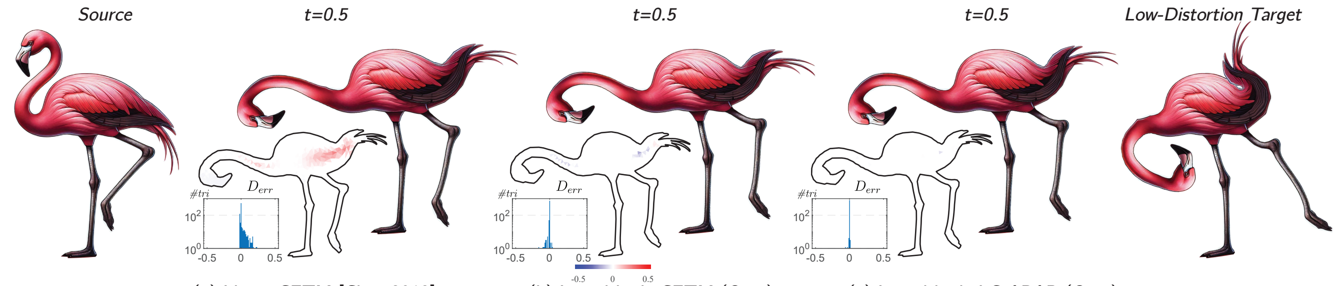
On Planar Shape Interpolation With Logarithmic Metric Blending

ALON FELDMAN, Technion Israel Institute of Technology, Israel
MIRELA BEN-CHEN, Technion Israel Institute of Technology, Israel



(a) Linear CETM [Chen 2013] (b) Logarithmic CETM (Ours) (c) Logarithmic LG-ARAP (Ours)

Fig. 1. Interpolation of an As-Killing-As-Possible deformation, with texture and the corresponding area distortion error D_{err} plots and histograms. Unlike previous methods, our logarithmic blending solutions (b)(c) minimize the area distortion, *in addition to* the conformal distortion which all methods minimize.

We present an interpolation method for planar shapes using logarithmic metric blending. Our approach generalizes prior work on pullback metrics to a framework, allowing us to employ different techniques, such as logarithmic blending of symmetric positive definite matrices, to have precise control over *both* conformal and area distortions. Key contributions include generalizing the continuous blending scheme and its adaptation to discrete mesh interpolation through different conformal and isometric parameterizations. Experimental results demonstrate that our method outperforms existing techniques in achieving bounded distortions, making it a compelling choice for applications in animation and morphing.

CCS Concepts: • **Computing methodologies** → **Shape analysis**.

Additional Key Words and Phrases: geometry processing, shape interpolation

ACM Reference Format:
Alon Feldman and Mirela Ben-Chen. 2025. On Planar Shape Interpolation With Logarithmic Metric Blending. In *Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers (SIGGRAPH Conference Papers '25), August 10–14, 2025, Vancouver, BC, Canada.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3721238.3730697

1 Introduction

Shape interpolation plays a key role in many areas of graphics and geometry processing. For example, blending two poses of the same shape is essential for creating smooth transitions in animations, while merging multiple shapes in a process called morphing, is helpful for designing and exploring new forms. Since different applications have unique requirements for shape interpolation, there is no single method that works perfectly in all cases. However, an important goal is to retain as much of the original geometric details of the input shapes as possible. Given that the shapes that are blended often differ, some level of distortion is inevitable.

We generalize the recent work on blending the pullback metric [Chen et al. 2013], which encodes local map distortion, into a broader framework of two components. The *blending component* interpolates the metric, ensuring smooth shape transitions while managing conformal and area distortions. The *realization component* reconstructs intermediate shapes from the blended metric, embedding them into a geometric domain that meets the desired interpolation properties. This enables us to provide a novel interpolation solution which is pointwise bounded in *both* conformal and area distortions, by modifying the blending method of the pullback metric from linear to *logarithmic*.

Moreover, using our framework, we show that previously proposed linear blending of the Jacobians [Alexa et al. 2000] is related to linearly blending the *root* of the pullback metric, allowing us to prove that by selecting a different realization method, this common blending machinery is sufficient to bound conformal distortion. We demonstrate our results on a variety of deformations: conformal, low-distortion and high-distortion, showing that our solution produces less area distortion, while also minimizing angle distortion.

1.1 Related Work

Among the diverse techniques developed for generating smooth transitions between 2D geometric models, intrinsic methods, and As-Rigid-As-Possible (ARAP) methods, which focus on maintaining the geometry structure and creating realistic deformations during these transitions, have emerged as pivotal. While other approaches exist, we focus on intrinsic and ARAP based solutions for their better preservation of shapes. For an extensive review of the subject, we refer to the work of Zhang et al. [2015].

Our findings also relate to works on interpolation of symmetric positive definite matrices, primarily in Diffusion-Tensor Imaging.

Intrinsic Methods. Intrinsic methods use geometric properties which are independent of the coordinate embedding, such as edge

Authors' Contact Information: Alon Feldman, Technion Israel Institute of Technology, Haifa, Israel, alon123213@gmail.com; Mirela Ben-Chen, Technion Israel Institute of Technology, Haifa, Israel, mirela@cs.technion.ac.il.

SIGGRAPH Conference Papers '25, Vancouver, BC, Canada

ACM ISBN 979-8-4007-1540-2/25/08
https://doi.org/10.1145/3721238.3730697

lengths, angles, and curvature, to guide shape interpolation. The 2D polygon interpolation method by Sederberg et al. [1993] was the first to linearly interpolate edge lengths and angles. This edge interpolation approach was adopted by later works to 3D mesh interpolation [Winkler et al. 2010]. Chen et al. [2013] introduced bounded-distortion techniques based on the surface metric, enabling smooth transitions with minimal conformal distortion. Later, Vaxman et al. [2015] introduced Möbius transformations for shape interpolation, developing a framework for interpolating conformally meshes that are conformal-equivalent [Springborn et al. 2008], in both planar and 3D settings. Chien et al. [2016a] proposed blending harmonic mappings in closed form, resulting in injective, distortion-bounded interpolations that are computationally efficient. Their approach produced C^∞ mappings with guaranteed bounds on both conformal and area distortions, albeit limited to a subclass of deformations. Aharon et al. [2019] extended metric interpolation to 3D interpolation by building on Chien et al. [2016b]'s convex parametrization framework in the discrete metric space, enabling volumetric mappings with bounded Jacobian singular values.

Unlike previous methods, our approach ensures both bounded conformal and area distortions for conformal deformations, and in practice bounds both distortions for general planar deformations.

As-Rigid-As-Possible (ARAP) Methods. ARAP methods focus on keeping parts of the shape as rigid as possible during transitions. This approach was introduced by Alexa et al. [2000], who showed how to realize a linearly blended Jacobian, minimizing deviation from the desired Jacobians. Xu et al. [2006] and Baxter et al. [2008] explored area-weighted realization and rotational consistency.

In our work, we propose a different approach for ARAP realization based on Local/Global approach [Liu et al. 2008], minimizing the deviation on the Jacobian singular values, which we find to be more suitable for metric blending then previous approaches.

Blending Symmetric Positive Definite Matrices. Logarithmic blending of two symmetric positive definite (SPD) matrices M_1, M_2, defined as $M(t) = exp\left((1-t)\cdot log(M_1) + t\cdot log(M_2)\right)$ (often called geometric, log-euclidean or exponential blending), has the unique property of interpolating the determinants of the inputs geometrically [Jung et al. 2015]. While commonly used in the field of Diffusion-Tensor Imaging to minimize scale changes and rotations during interpolation, it is less common in computer graphics. The most significant use of logarithmic matrix blending in the field is by Alexa [2002], highlighting its benefits for 3D transformations.

To our knowledge, logarithmic SPD blending has not been applied to metric blending or shape interpolation.

1.2 Contributions

Our main contributions are:

- We provide formal proofs establishing the benefits of continuous logarithmic metric blending in ensuring bounded area and conformal distortions, and its ability to flatly interpolate the metric for conformal deformations.
- We propose conformal and isometric realizations based on a novel use of the cotan-weights Laplace-Beltrami operator induced by corners, for realizing a blended discrete metric.
- We demonstrate our interpolation solution on different deformations types, showcasing its superiority compared to previous approaches.

2 The Interpolation Problem

The goal of interpolating between a source and target shape is to produce intermediate shapes with desired properties. These are governed by two principles: First, the transition should be smooth. Second, the intermediate shapes' deformations, relative to the source and target, should maintain "simplicity."

From the first principle, several fundamental properties emerge:

(1) **Lagrange Property:** The intermediate shapes must reproduce the source and target shapes at their respective frames.
(2) **Smoothness Property:** The transitions between frames should occur smoothly over time.
(3) **Symmetry Property:** The interpolation should be invariant to swapping the source and target shapes.

The second principle is more open to interpretation. A "simple" interpolation implies that if the transformation between the source and target shapes is inherently "simple" (e.g., affine, conformal, or area-preserving), the intermediate shapes should reflect this simplicity. Thus, for example, if the conformal distortion or area distortion induced by the mapping from the source to the target is bounded, the intermediate shapes should also have a bounded distortion. Note that we focus exclusively on the planar case, where both the source and target shapes lie in a plane, and the intermediate shapes are constrained to remain planar as well.

We first propose an interpolation framework in the continuous setting, understanding how each blending method behaves. With this theory in mind, we design corresponding discrete blending schemes, with the goal of obtaining similar properties.

3 The Continuous Setting

Consider two domains $S_0, S_1 \subset \mathbb{R}^2$ and a mapping $\phi : S_0 \to S_1$. The changes in lengths and angles induced by ϕ at any point is characterized by a bilinear form $\mathbb{R}^2 \times \mathbb{R}^2 \to \mathbb{R}$, called the *pullback metric*, denoted as g_ϕ. Since both domains are flat, it is established that $g_\phi = {J_\phi}^\top J_\phi$, where J_ϕ is the Jacobian of the mapping ϕ, and ${J_\phi}^\top$ represents its transpose. By definition, g_ϕ is symmetric positive definite. For a trivial mapping of a shape to itself, $\{f(x) = x \mid x \in S\}$, the metric reduces to the identity matrix I.

The conformal distortion $K(p)$ and area distortion $D(p)$ of a mapping at a point $p \in S_0$ are derived from the eigenvalues λ_1, λ_2 of the metric g_ϕ, where λ_1 and λ_2 represent the maximal and minimal eigenvalues. These distortions are commonly defined as:

$$K(p) = \sqrt{\lambda_1/\lambda_2}, \quad D(p) = \sqrt{\lambda_1 \cdot \lambda_2} \tag{1}$$

3.1 Metric Interpolation

We introduce the following interpolation scheme, referred to as the "Metric Interpolation Scheme", which consists of three key stages:

(1) **Metric Blending:** Compute interpolated metrics g_t, $0 \le t \le 1$.
(2) **Metric Flattening:** Flatten the intermediate metrics, as they are not necessarily flat.

(3) **Metric Realization:** Realize the flat metric using Poisson's equation., i.e. find an embedding that corresponds to the metric.

This scheme was first described by Chen et al. [2013], where the pullback metric g_ϕ was *linearly* blended as:

$$g_t = (1-t)\cdot I + t\cdot g_\phi. \tag{2}$$

We propose instead a *logarithmic* blending of the metric:

$$g_t = I^{1-t}\cdot {g_\phi}^t = {g_\phi}^t. \tag{3}$$

To demonstrate the full scope of our "Metric Interpolation Scheme", and show the advantages of logarithmic blending over previous works, we also propose the *square-root blending* variant:

$$g_t = \left((1-t)\cdot I^{0.5} + t\cdot g_\phi^{0.5}\right)^2 \tag{4}$$

Square-root blending enables a novel analysis of As-Rigid-As-Possible Interpolation [Alexa et al. 2000].

3.2 Lagrange, Smoothness and Symmetry Properties

The straightforward derivation of the Lagrange, smoothness, and symmetry properties is provided in Supplemental Sec. 3.

3.3 Bounded Distortion Properties

A desired property for interpolation is simplicity. In this context, simplicity is achieved when the distortions induced by the intermediate deformations are bounded between the distortions of the target deformation and the identity deformation. We define the distortion error as the log distance from the allowed distortion range.

$$K_{err} = \mathrm{Dist}\big(\ln(K_t), [0, \ln(K_\phi)]\big) \tag{5}$$

$$D_{err} = \mathrm{Dist}\big(\ln(D_t), [0, \ln(D_\phi)]\big) \tag{6}$$

$$\mathrm{Dist}(x,[a,b]) = \begin{cases} 0, & \text{if } min(a,b) \le x \le max(a,b), \\ x - min(a,b), & \text{if } x < min(a,b), \\ x - max(a,b) & \text{if } x > max(a,b). \end{cases}$$

This distortion error measures deviation from the allowable distortion range, as is implied by the source and target shapes. It is expressed logarithmically to make the error symmetric for expansion and contraction. This ensures that the error remains consistent if the source and target are swapped.

3.3.1 *Logarithmic Blending.*

Proposition 3.1. *Given $g_t = {g_\phi}^t$, the eigenvalues of the intermediate metric are geometric interpolations of the target metric eigenvalues:*

$$\lambda_{t,i} = \lambda_{\phi,i}^t \tag{7}$$

Proof. By definition of SPD matrix exponentiation. □

Corollary 3.2. *Given $g_t = {g_\phi}^t$, the conformal and area distortions induced by g_t are geometric interpolations of the corresponding distortions induced by g_ϕ. Consequently, both distortions are bounded.*

Proof.

$$K_t(p) = \sqrt{\frac{\lambda_{t,1}}{\lambda_{t,2}}} = \sqrt{\frac{{\lambda_{\phi,1}}^t}{{\lambda_{\phi,2}}^t}} = \left(\sqrt{\frac{\lambda_{\phi,1}}{\lambda_{\phi,2}}}\right)^t = K_\phi(p)^t$$

$$D_t(p) = \sqrt{\lambda_{t,1}\cdot\lambda_{t,2}} = \sqrt{{\lambda_{\phi,1}}^t\cdot{\lambda_{\phi,2}}^t} = (\sqrt{\lambda_{\phi,1}\cdot\lambda_{\phi,2}})^t = D_\phi(p)^t$$

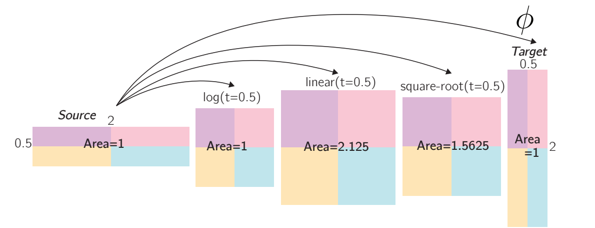


Fig. 2. Interpolation of a simple equi-areal deformation. The source has unit area, and is stretched in one axis and squeezed in the second by a factor of 4. The resulting target also has unit area. Different interpolation schemes lead to different intermediate area values, with only logarithmic blending preserving the area.

Since both distortions are exponential functions and monotonic, they are bounded by the distortion from the source to the target. □

3.3.2 *Linear Blending.*

Chen et al. [2013] proved bounds on conformal distortion for the continuous setting using the properties of linearly blended SPD matrices. However, they did not derive explicit interpolation formulas for eigenvalues and distortions. For linear blending, the eigenvalues and distortions are:

$$\lambda_{t,i} = 1 + t(\lambda_{\phi,i} - 1), \quad K_t(p) = \sqrt{\frac{1+t(\lambda_{\phi,1}-1)}{1+t(\lambda_{\phi,2}-1)}} \tag{8}$$

$$D_t(p) = \sqrt{\big(1+t(\lambda_{\phi,1}-1)\big)\big(1+t(\lambda_{\phi,2}-1)\big)}$$

3.3.3 *Square-Root Blending.*

For square-root blending, the interpolation formulas are:

$$\lambda_{t,i} = \big(1+t(\sqrt{\lambda_{\phi,i}}-1)\big)^2, \quad K_t(p) = \frac{1+t(\sqrt{\lambda_{\phi,1}}-1)}{1+t(\sqrt{\lambda_{\phi,2}}-1)} \tag{9}$$

$$D_t(p) = \big(1+t(\sqrt{\lambda_{\phi,1}}-1)\big)\big(1+t(\sqrt{\lambda_{\phi,2}}-1)\big)$$

Both blending proofs are derived by diagonalizing g_ϕ (see Supplemental Sections 1 and 2).

As with linear blending, the conformal distortion is monotonic and bounded for $0 \le t \le 1$, due to the convexity of linear-fractional functions when the denominator is positive [Boyd and Vandenberghe 2004, example (3.32)]. The area function is a quadratic function, hence the monotonicity depends on $\lambda_{\phi,1}$ and $\lambda_{\phi,2}$.

In Fig. 2, we show how only logarithmic blending maintains the area of a rectangle interpolating a simple equiareal deformation.

3.4 Flat Interpolation Property

A significant challenge of the interpolation scheme is that the intermediate metrics are not necessarily flat. A well-known trade-off of flattening curved surfaces is that flattening the metric may induce either angle distortion or area distortion, or both, depending on the chosen approach. For a general deformation and blending method, the intermediate metric will accumulate curvature. Uniquely, we

Table 1. Properties of continuous blending methods. *conformal only.

Property	Root	Linear	Logarithmic
Lagrange/Smooth/Symmetry	●	●	●
Pre-Realization Bounded Angle-Dist	●	●	●
Pre-Realization Bounded Area-Dist	○	○	●
Flat Interpolation	○	○	**Conformal***

show that when the deformation is conformal and the blending method is logarithmic, the intermediate metric is *flat*.

Conformally Equivalent Domains: Two domains S_0, S_1 are conformally equivalent if there exists a smooth function $u : S_0 \to \mathbb{R}$, called the *conformal scaling factor*, such that $g_1 = e^{2u} g_0$. The conformal scaling factor satisfies $\Delta u = e^{2u} K_1 - K_0$, known as the Yamabe equation. For flat input domains, this simplifies to $\Delta u = 0$.

Proposition 3.3. *Given two flat domains $S_0, S_1 \subset \mathbb{R}^2$ and a conformal mapping $\phi : S_0 \to S_1$, the logarithmic blending of metrics $g_t = {g_0}^{1-t} {g_1}^t$ produces intermediate pullback metrics g_t that are conformal and flat, with a linearly blended conformal factor $u_t = ut$.*

Proof. Since ϕ is conformal, there exists a conformal factor u such that $g_1 = e^{2u} g_0$. Substituting into the log blending formula:

$$g_t = {g_0}^{1-t} {g_1}^t = {g_0}^{1-t} \left(e^{2u} g_0\right)^t = \left(e^{2u}\right)^t {g_0}^{1-t} {g_0}^t$$

Simplifying, we obtain *linearly interpolated conformal factors*:

$$g_t = e^{2ut} g_0, \quad u_t = ut \tag{10}$$

Since u_t is a scalar multiple of u by t, and $\Delta u = 0$ since S_0 and S_1 are flat, the curvature induced by u_t is also zero. Thus, g_t is flat:

$$\Delta u_t = \Delta(ut) = t \Delta u = 0. \qquad \square$$

This property is unique to logarithmic blending, making it particularly advantageous for interpolating flat conformal deformations.

3.5 Continuous Setting Summary

We summarize the blending methods and properties (Table 1).

Pre-flattening, both linear and square-root blendings result in bounded conformal distortions, while logarithmic blending uniquely ensures *both* bounded area and conformal distortions. Realizing both bounds simultaneously, is impossible for non developable surfaces as flattening the metric inherently invalidates one of them.

Logarithmic blending stands out when interpolating planar *conformal* domains. In addition to preserving both conformal and area distortion bounds pre-realization, it also yields a *flat* metric, thus preserving these properties post-realization. This makes it superior to linear or square-root blending for such cases.

Lemma 3.4. *For conformal deformations, logarithmic metric blending achieves post-flattening deformations that bound conformal and area distortion.*

Proof. Immediate from Corollary 3.2 and Proposition 3.3 $\square$

4 The Discrete Setting

Consider two planar triangular meshes, $S = (V, E, F)$ and $\tilde{S} = (\tilde{V}, E, F)$, where $V, \tilde{V} \in \mathbb{R}^2$, and both meshes share the same connectivity and face orientation. We abuse the usual notation slightly, and denote by V both the vertex set and its embedding in $\mathbb{R}^2$.

The deformation function ϕ is defined on the vertices as $\phi : V \to \mathbb{R}^2$, such that $\tilde{V} = \phi(V)$. We extend the mapping linearly to the faces using barycentric coordinates. This extends ϕ to a piecewise linear function $\phi : S \to \tilde{S}$, defined throughout the mesh domain.

Using this piecewise linear extension, the discrete metric is defined as $g_f = {J_f}^\top J_f$, where J_f is the piecewise constant Jacobian of the mapping on face f. The conformal and area distortions are then computed per face using the eigenvalues of the piecewise metric:

$$K_f = \sqrt{\lambda_{f,1} / \lambda_{f,2}}, \qquad D_f = \sqrt{\lambda_{f,1} \cdot \lambda_{f,2}}, \tag{11}$$

where $\lambda_{f,1}, \lambda_{f,2}$ are the maximal and minimal eigenvalues of g_f.

Given a scalar $0 \le t \le 1$, our objective is to generate an intermediate mesh $S_t = (V_t, E, F)$ such that the properties established in the continuous setting are preserved in the discrete setting.

5 Blending

Our discrete solution is divided into two components, the blending component, and the realization component. For the blending component, we propose two approaches: a discrete metric blending approach, and an edge length blending approach.

The realization component depends on the blending approach chosen. For the discrete metric blending, we propose novel conformal, equiareal and isometric realizations. The edge blending approach is more limited, allowing only conformal realization using the scheme proposed by Chen et al. [2013]. We explore the edge length approach to show that our logarithmic blending is superior independently of the realization component, as we simply swap the blending formula in an existing solution.

5.1 Discrete Metric Blending

A straightforward way to discretize the continuous metric blending is to blend the discrete piecewise metric, according to Eq. (7), (8) and (9). We use the spectral decomposition $g_f = V_f \Lambda_f V_f^\top$, to compute the matrix exponentiation and addition operators (see Alg. 1).

Alternatively, it is possible to blend the Jacobians directly (see Alg. 2). We define the Jacobian interpolation that corresponds to the respective metric interpolations using the Singular Value Decomposition (SVD) $J_f = U_f \Sigma_f V_f^\top$, where $\Lambda = \Sigma^2$. We prefer this over the polar decomposition, commonly used in animation [Alexa et al. 2000], for simplicity: the relation between the g_f formulas and the J_f formulas is clearer with SVD, as these are related through $g(t) = J(t)^\top J(t)$. Thus, for blending Jacobians we have:

$$\textbf{Logarithmic:} \qquad J(t) = U \Sigma^t V^\top \tag{12}$$

$$\textbf{Linear:} \qquad J(t) = U \left((1-t) \cdot I^2 + t \cdot \Sigma^2\right)^{0.5} V^\top \tag{13}$$

$$\textbf{Square-Root:} \qquad J(t) = U\left((1-t) \cdot I + t \cdot \Sigma\right) V^\top \tag{14}$$

We note that Eq. (14) is the common linear blending of Jacobians, used in ARAP interpolation [Alexa et al. 2000] (albeit in the polar decomposition form), and other previous work.

Algorithm 1: Discrete Metric Blending

Input: Target face metric g_f, timestep t
Output: Intermediate metric $g_f(t)$

$[\Lambda_f, V_f] = \text{Spectral}(g_f)$;
switch *blending mode* **do**
  Logarithmic: $g_f(t) = V_f {\Lambda_f}^t V_f^\top$;
  Linear: $g_f(t) = V_f\left((1-t)\cdot I + t\cdot \Lambda_f\right) V_f^\top$;
  Square-Root: $g_f(t) = V_f\left((1-t)\cdot I^{0.5} + t\cdot \Lambda_f^{0.5}\right)^2 V_f^\top$;

Algorithm 2: Discrete Jacobian Blending

Input: Target face jacobian J_f, interpolation timestep t
Output: Intermediate jacobian $J_f(t)$

$[U_f, \Sigma_f, {V_f}^\top] = \text{SVD}(J_f)$;
switch *blending mode* **do**
  Logarithmic: $J_f(t) = U_f {\Sigma_f}^t {V_f}^\top$;
  Linear: $J_f(t) = U_f\left((1-t)\cdot I^2 + t\cdot \Sigma_f^2\right)^{0.5} {V_f}^\top$;
  Square-Root: $J_f(t) = U_f\left((1-t)\cdot I + t\cdot \Sigma_f\right) {V_f}^\top$;

Validity. Chen et al. [2013] shows that it is enough for our new metrics to be SPD, which they are, to produce positive edge lengths $\sqrt{e_{ij}^\top \cdot g(t)\cdot e_{ij}}$ that fulfill the triangle inequality.

It is important to note that intermediate metrics/Jacobians that belong to adjacent faces do not necessarily agree on the change imposed on the shared edge. Namely $J_{f_1}(t)\cdot e_{ij} \neq J_{f_2}(t)\cdot e_{ij}$, where f_1, f_2 share the edge e_{ij}. Thus, we also discuss *edge length blending*, that does not suffer from this limitation.

5.2 Edge Length-Squared Blending

Compatible edge lengths might be desirable as it adds geometrical meaning to the intermediate mesh such as curvature, and increases the number of geometrical processing tools available.

Chen et al. [2013] demonstrated that their linear metric blending produces compatible edge lengths between neighboring faces, as the interpolated edge lengths are independent of face selection:

$$\|e_{ij}(t)\|^2 = (1-t)\|e_{ij}\|^2 + t\|\tilde{e}_{ij}\|^2. \tag{15}$$

In addition to being exactly equivalent, it also has the same structure as Eq. (2). Inspired by this, we propose to use the structures of Eq. (3), (4) to formulate a blending scheme using the edge length-squared. Thus, we define logarithmic edge length interpolation:

$$\|e_{ij}(t)\|^2 = (\|e_{ij}\|^2)^{1-t}(\|\tilde{e}_{ij}\|^2)^t \;\Rightarrow\; \|e_{ij}(t)\| = \|e_{ij}\|^{1-t}\|\tilde{e}_{ij}\|^t. \tag{16}$$

Similarly, we define the square-root edge length interpolation:

$$\begin{aligned}\|e_{ij}(t)\|^2 &= \left((1-t)\sqrt{\|e_{ij}\|^2} + t\sqrt{\|\tilde{e}_{ij}\|^2}\right)^2 \\ \|e_{ij}(t)\| &= (1-t)\|e_{ij}\| + t\|\tilde{e}_{ij}\|.\end{aligned} \tag{17}$$

Additionally, logarithmic edge length interpolation can be derived directly using the concept of "Discrete Conformal Factors" and the linear conformal factor equation (10). See Supplemental Section 4.

Limitations of Log Edge Length Interpolation. Unlike linear and square-root blending of edge lengths, logarithmic blending can violate triangle inequality. This is a known pitfall when working with discrete conformal factors [Springborn et al. 2008]. In practice, this typically arises when the input deformation has significant angle distortion, hence logarithmic edge blending with CETM realization is less viable for high-distortion applications (see Fig. 3).

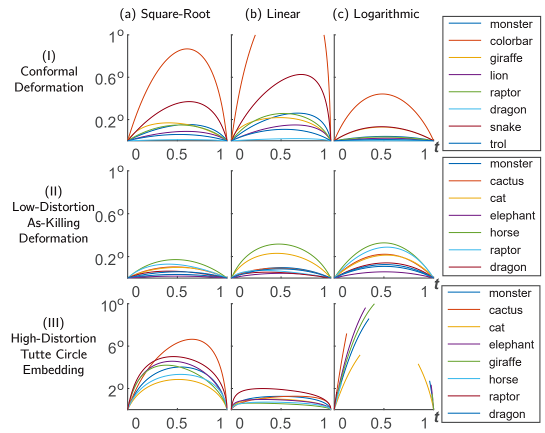


Fig. 3. Average discrete Gaussian curvature of the interpolated mesh, for different input deformations and edge blending schemes. The discrete curvature of a vertex v is $360^o - \sum_{\alpha\in \text{angles}(v)} \alpha$. For each deformation we show the average curvature at time t as a separate plot. For the high-distortion Tutte deformation, logarithmic edge blending violates triangle inequality.

Length Cross-Ratio Interpolation. Vaxman et al. [2015] proposed interpolating edge length cross-ratios, followed by realizing the intermediate cross-ratios with As-Möbius-As-possible nonlinear optimization method. In 2D, their blending is equivalent to logarithmic edge length blending, as they mention. The main difference of their approach from the direct edge length approach is that edge blending allows the use of standard realization methods such as CETM/BFF, as Chen et al. [2013] did, which offer computational advantages.

6 Realization

The realization of a mesh given the blended metrics or edge lengths can be done using existing intrinsic parameterizations techniques, which find the best fitting planar positions for the vertices, flattening and stitching the faces simultaneously.

6.1 Realization of the Interpolated Metric/Jacobians

We propose conformally realizing the discrete piecewise metric with LSCM parameterization [Desbrun et al. 2002; Lévy et al. 2002], for which we define a *corner Laplace-Beltrami.* For an equiareal or

Algorithm 3: Discrete Edge Length Blending

Input: Source and target edge lengths $\|e_{ij}\|$, $\|\tilde{e}_{ij}\|$, timestep t
Output: Intermediate edge length $\|e_{ij}(t)\|$

switch *blending mode* **do**
  Logarithmic: $\|e_{ij}(t)\| = \|e_{ij}\|^{1-t}\|\tilde{e}_{ij}\|^t$;
  Linear:: $\|e_{ij}(t)\| = \sqrt{(1-t)\|e_{ij}\|^2 + t\|\tilde{e}_{ij}\|^2}$;
  Square-Root:: $\|e_{ij}(t)\| = (1-t)\|e_{ij}\| + t\|\tilde{e}_{ij}\|$;

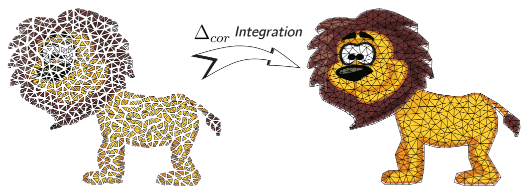


Fig. 4. Integration of a triangle soup is done using the corner Laplace-Beltrami Δ_{cor} induced by the "broken" triangles.

isometric realization we propose an adaptation of the Local/Global algorithm [Liu et al. 2008] (see Alg. 4).

6.1.1 Corner Laplace-Beltrami Operator. The intermediate metric and Jacobians are piecewise constant, and thus there doesn't necessarily exist an embedding that realizes the metric with the connectivity of the input mesh. However, if we consider each triangle separately, such an embedding does exist, and is easy to realize triangle-wise as a "triangle soup", see Fig. 4. The term "triangle soup" usually relates to the case where the *connectivity* of the mesh is broken, whereas in our case the *embedding* is broken: the coordinate functions are discontinuous, and multi-valued at the vertices, i.e., are *corner* functions (a function of the vertex *and* the face).

Given the interpolated Jacobians, we reconstruct the deformed edges (up to global translation) using $e_{ij,f}(t) = J_f(t) \cdot e_{ij}$, where $e_{ij,f}(t)$ is the half edge e_{ij} that belongs to the triangle f at time t. If only the interpolated metric is given, we reconstruct a triangle up to global rotation and translation using the new edge lengths. In both cases, the information we have about the interpolated triangles is enough to construct a *Corner Laplace-Beltrami Operator*.

This operator is built, as is standard, by summing per triangle contributions. Namely, given two faces f_1, f_2 which share an edge e_{ij} the entry $\Delta_{ij}(t)$ is computed by summing the per triangle contribution of e_{ij,f_1}, e_{ij,f_2}. Unlike the case of a standard mesh, the edge lengths of two half edges belonging to the same edge may be different, but this does not affect the computation. We denote this operator by Δ_{cor}. Interestingly, this operator is symmetric and local, similarly to the standard cotan LB operator. Further, the usual decomposition of the cotan LB operator as $\Delta = \frac{1}{4}E^T A_F^{-1} E$, where $E \in \mathbb{R}^{3|F|\times|V|}$ is a sparse matrix encoding the rotated *half* edges, and $A_F \in \mathbb{R}^{3|F|\times 3|F|}$ is a diagonal matrix of face areas, still holds. This is true because each half edge has a separate row in E. Hence, $x^T \Delta x = \|A_F^{-\frac{1}{2}} Ex\|^2 \geq 0$, and thus the corner Laplacian is also PSD.

6.1.2 Conformal Realization. LSCM is an intrinsic parameterization method, which requires only a Laplace-Beltrami operator. We therefore use our Δ_{cor} operator with LSCM to generate a conformal embedding of the interpolated Jacobians or metric. As LSCM has a free parameter of global scaling, we add a global area normalization to the interpolated mesh, by scaling it to have the same total area as the triangle soup. This operation does not affect the conformal distortion, as it is not affected by global scaling.

6.1.3 Isometric realization. To trade off the conformal distortion for area distortion when realizing the interpolated mesh, we propose a variant of the Local/Global parameterization by Liu et al. [2008]. In brief, the steps of Local/Global parameterization are:

(1) Given a 3D mesh M_{3D}, flatten the faces to a triangle soup M_{soup}.
(2) Compute an initial parameterization of M_{3D}, denoted M_{2D}, using other techniques, e.g., LSCM.
(3) Compute Jacobians of the $M_{soup} \to M_{2D}$ transformation.
(4) *Local Step*: Modify the singular values of the Jacobians such that the Jacobians will be more rigid or conformal.
(5) *Global Step*: Integrate the Jacobians using the Laplace-Beltrami and angles of M_{3D} to achieve a new parameterization M_{2D}.
(6) Compute the energy E, if not converged, repeat steps (3) to (6).

The algorithm setting for steps (4) and (6) for As-Rigid-As-Possible (ARAP) parameterization is given by [Liu et al. 2008]:

$$\sigma_{1,f}(i+1) = \sigma_{2,f}(i+1) = 1, \tag{18}$$

$$E(i) = \sum_{f=1}^{|F|} A_f \left[\left(\sigma_{1,f}(i) - 1\right)^2 + \left(\sigma_{2,f}(i) - 1\right)^2 \right], \tag{19}$$

where i is the optimization iteration, $|F|$ is the number of faces of M, A_f is the area of the face f in the triangle soup, and $\sigma_{1,f}, \sigma_{2,f}$ are the singular values of the Jacobian of f.

We suggest a variant of steps (4) and (6) for equiareal parameterization, termed "Local/Global As-Equiareal-As-Possible" (AEAP) to show the possible area-bounding benefits of logarithmic blending:

$$\sigma_{1,f}(i+1) = \frac{\sigma_{1,f}(i)}{\sqrt{\sigma_{1,f}(i)\sigma_{2,f}(i)}}, \quad \sigma_{2,f}(i+1) = \frac{\sigma_{2,f}(i)}{\sqrt{\sigma_{1,f}(i)\sigma_{2,f}(i)}} \tag{20}$$

$$E(i) = \sum_{f=1}^{|F|} A_f \left(\frac{\sigma_{1,f}(i)}{\sigma_{2,f}(i)} + \frac{\sigma_{2,f}(i)}{\sigma_{1,f}(i)} \right) (\sqrt{\sigma_{1,f}(i)\sigma_{2,f}(i)} - 1)^2 \tag{21}$$

Our selected AEAP singular values are area-preserving $\sigma_{1,f}(i+1) \cdot \sigma_{2,f}(i+1) = 1$, while minimizing error in Log-Euclidean metric.

$$\left(\ln(\sigma_{1,f}(i)) - \ln(\sigma_{1,f}(i+1))\right)^2 + \left(\ln(\sigma_{2,f}(i)) - \ln(\sigma_{2,f}(i+1))\right)^2$$

Additionally, they preserve the aspect ratio which avoids unnecessary angle distortion. For derivation see Supplemental Sec. 5.

To apply the Local/Global realization to the piecewise discrete metric, we require three minor, but key, modifications:

- Use M_{soup} instead of the input 3D mesh, skipping (1).
- Use Δ_{cor} with LSCM to compute the initial M_{2D}, skipping (2).
- Use Δ_{cor} and the angles of M_{soup} for the integration process.

6.2 Realization of the Interpolated Edge Lengths

We realize the interpolated edge lengths using CETM [Springborn et al. 2008] as proposed by Chen et al. [2013], or the newer linear BFF [Sawhney and Crane 2017] (Alg. 5). Both lead to a conformal realization.

7 Results

We implemented our algorithm in MATLAB. For mesh visualization and texture patching, we used the tool provided by Kroon [2024].

We re-implemented ARAP interpolation [Baxter et al. 2008], Local/Global [Liu et al. 2008] and BFF [Sawhney and Crane 2017].

Algorithm 4: Discrete Metric Interpolation Scheme

Input: Triangle meshes $S = (V, E, F)$, $\tilde{S} = (\tilde{V}, E, F)$, timestep t
Output: Intermediate mesh $S_t = (V_t, E, F)$

for $f \in F$ **do**
  J_f = Jacobian of transformation from $V(f)$ to $\tilde{V}(f)$;
  $J_f(t) = \textbf{JacobianBlending}(J_f, t)$;
  for $i \in \{0, 1, 2\}$ **do**
    $e_i = V(f(i+2)) - V(f(i+1))$; // i mod 3
    $E_t(f, i) = J_f(t) \cdot e_i$; // Compute new edge
    $L(f, i) = \|E_t(f, i)\|$; // Compute length
  $TriangleSoup(f) = [[0, 0], E_t(f, 2), -E_t(f, 1)]$;
Δ_t = LaplaceBeltrami(L); V_t = LSCM(Δ_t);
if *mode is ARAP or AEAP* **then**
  V_t = LocalGlobal$(\Delta_t, TriangleSoup, init = V_t, mode)$;
Apply global rigid transformation on V_t using external constraints;

LSCM [Lévy et al. 2002] and CETM [Springborn et al. 2008] were obtained from the software provided by Chen et al. [2013].

We tested different combinations of blending and realization, on deformations of various types. We generated conformal deformations using Cauchy Coordinates [Weber et al. 2009], and low-distortion deformations using As-Killing-As-Possible vector fields [Solomon et al. 2011]. Finally, we used a Tutte embedding [Tutte 1963] to the unit circle, to generate high-distortion deformations.

Visualization. We demonstrate the differences between the different blendings and realizations in two ways. First, we plot the distortion errors over the interpolated mesh for a given t, or alternatively, present them as a histogram, providing a complete view of the distortion for all the faces. Second, we plot the distortions $K(t)$, $D(t)$ in log-space as a function of t, for a subset of the triangles. The conformal distortion $K(t)$ is shown for the 25 least distorted (purple) and 25 most distorted faces (red) (50 total). The area distortion $D(t)$ is shown for the 25 least-distorted (green), most-scaled-down (yellow), most-scaled-up (blue) faces (75 total). A monotonic graph indicates bounded distortion. In addition, a *linear* graph indicates a "constant speed" growth of the log distortion during the interpolation, and thus a more natural interpolation sequence.

The full tensor of possibilities: input deformation / blending / realization is quite large, and we show only of subset of interesting behaviors, where swapping the blending or realization components makes a large difference to the resulting distortions. See also Supplemental Sec. 6 and the accompanying video for more examples.

7.1 Blending

Here we test the three blending schemes, where we tailor the realization to the input deformation. For a conformal deformation we use CETM or LSCM, for a high-distortion deformation we use ARAP, and for a low-distortion deformation we may use any realization. Fig. 1 shows that for a low-distortion deformation modifying Chen et al. [2013] (a) from linear to logarithmic blending reduces the area distortion (b). Replacing the realization with our isometric local/global ARAP realization (c) completely eliminates the error. Fig. 5 shows that for a conformal deformation, LSCM consistently produces smooth and conformally bounded results for all three blending options (b-d), as does the previous method (a). Logarithmic blending, however, also eliminates the area distortion (d). In Fig. 7, we demonstrate that when realized with conformal techniques (LSCM, CETM) (first row), or isometric Local/Global-ARAP (second and third rows), all three blendings effectively constrained conformal distortion, regardless of deformation type. However, logarithmic blending with LG-ARAP (d) effectively constrained *both* area and conformal distortions for the high-distortion deformations (two bottom rows), unlike linear and square-root blending (b-c).

Algorithm 5: Edge Length Interpolation Scheme

Input: Triangle meshes $S = (V, E, F)$, $\tilde{S} = (\tilde{V}, E, F)$, timestep t
Output: Intermediate mesh $S_t = (V_t, E, F)$

for $e_i j \in E$ **do**
  $\|e_{ij}\|, |\tilde{e}_{ij}\|$ are the edge lengths of e_{ij} in $S, \tilde{S}$ respectively;
  $L(e_{ij}) = \textbf{EdgeLengthBlending}(\|e_{ij}\|, |\tilde{e}_{ij}\|)$;
V_t = [CETM/BFF]$(L, U_{Boundary} = 0)$;
Apply global rigid transformation on V_t using external constraints;

7.2 Realization

Here we consider the different realization approaches, while keeping the blending method mostly fixed. In Figs. 9 and 7, we compare our Local/Global realization, based on minimizing singular values deviation, to Baxter et al. [2008]'s ARAP, which minimizes deviation from target Jacobians. Our approach consistently achieves smoother results, with lower area distortion. Fig. 6 compares our different realizations on a low-distortion deformation, using logarithmic blending. We note, as expected, that LSCM (a) minimizes the deviation of the conformal distortion, leading to a log-linear error graph for $K(t)$, whereas LG-AEAP (b) minimizes the deviation of the area distortion yielding a log linear graph of $D(t)$. Finally, LG-ARAP (c) balances both deviations. For a low-distortion deformation we have the option of using edge-length blending with BFF realization, which is faster than LSCM realization. Fig. 8 shows that for both the square-root (a,b) and logarithmic blending (c,d) LSCM and BFF lead to a similar conformal distortion, with BFF yielding a somewhat lower area distortion than LSCM for log blending (d). In Fig. 10 we show that LG-AEAP (a-c) minimizes area deviation, yielding cohesive area distortions at the expense of conformal deviation.

8 Discussion and Conclusion

Our novel logarithmic blending, combined with the Local/Global ARAP approach, proved highly effective in minimizing both distortions across a range of deformations. This method outperformed linear and square-root blending in overall distortion control. For low-distortion deformations, the simpler logarithmic edge length blending using BFF is a faster alternative. Detailed computation times are provided in Supplemental Sec. 6.

Our framework has several limitations. To address the curvature of the interpolated mesh, we use parameterization methods that do not guarantee pointwise area- or conformal-preserving embeddings. Future work could explore alternative blending methods to identify one capable of bounding these distortions while producing perfectly flat deformations—or prove that such blending is inherently impossible. Unlike Chen et al. [2013], our continuous framework is restricted to blending between a single source and a single target due to its reliance on the flexibility of the identity matrix I as one of the blended metrics. Consequently, it cannot be easily extended to multi-target or pose blending. This is another area for future work.

We believe our mathematical framework, blending techniques, and realization methods enhance the understanding of existing interpolation solutions and lay the groundwork for future advancements.

Acknowledgments

Mirela Ben-Chen acknowledges the support of the Israel Science Foundation (grant No. 1073/21). We thank the anonymous reviewers for the elaborated feedback and many useful suggestions.

References

Ido Aharon, Renjie Chen, Denis Zorin, and Ofir Weber. 2019. Bounded Distortion Tetrahedral Metric Interpolation. *ACM Trans. Graph.* 38, 6, Article 182 (2019), 17 pages. doi:10.1145/3355089.3356569

Marc Alexa. 2002. Linear Combination of Transformations. *ACM Trans. Graph.* 21, 3 (2002), 380–387. doi:10.1145/566654.566592

Marc Alexa, Daniel Cohen-Or, and David Levin. 2000. As-Rigid-as-Possible Shape Interpolation. In *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '00)*. 157–164. doi:10.1145/344779.344859

William Baxter, Pascal Barla, and Ken-ichi Anjyo. 2008. Rigid Shape Interpolation Using Normal Equations. In *Proceedings of the 6th International Symposium on Non-Photorealistic Animation and Rendering (NPAR '08)*. 59–64. doi:10.1145/1377980.1377993

Stephen Boyd and Lieven Vandenberghe. 2004. *Convex Optimization.* Cambridge University Press. Section 3.4.

Renjie Chen, Ofir Weber, Daniel Keren, and Mirela Ben-Chen. 2013. Planar Shape Interpolation with Bounded Distortion. *ACM Trans. Graph.* 32, 4, Article 108 (2013), 12 pages. doi:10.1145/2461912.2461983

Edward Chien, Renjie Chen, and Ofir Weber. 2016a. Bounded Distortion Harmonic Shape Interpolation. *ACM Trans. Graph.* 35, 4, Article 105 (2016), 15 pages. doi:10.1145/2897824.2925926

Edward Chien, Zohar Levi, and Ofir Weber. 2016b. Bounded Distortion Parametrization in the Space of Metrics. *ACM Trans. Graph.* 35, 6, Article 215 (2016), 16 pages. doi:10.1145/2980179.2982426

Mathieu Desbrun, Mark Meyer, and Pierre Alliez. 2002. Intrinsic Parameterizations of Surface Meshes. *Computer Graphics Forum* 21, 3 (2002), 209–218. doi:10.1111/1467-8659.00580

Sungkyu Jung, Armin Schwartzman, and David Groisser. 2015. Scaling-Rotation Distance and Interpolation of Symmetric Positive-Definite Matrices. *SIAM J. Matrix Anal. Appl.* 36, 3 (2015), 1180–1201. doi:10.1137/140967040

Dirk-Jan Kroon. 2024. Texture Patch. https://www.mathworks.com/matlabcentral/fileexchange/28106-texture-patch. MATLAB Central File Exchange. Retrieved December 25, 2024.

Bruno Lévy, Sylvain Petitjean, Nicolas Ray, and Jérome Maillot. 2002. Least Squares Conformal Maps for Automatic Texture Atlas Generation. *ACM Trans. Graph.* 21, 3 (2002), 362–371. doi:10.1145/566654.566590

Ligang Liu, Lei Zhang, Yin Xu, Craig Gotsman, and Steven J. Gortler. 2008. A Local/Global Approach to Mesh Parameterization. In *Proceedings of the Symposium on Geometry Processing (SGP '08)*. 1495–1504.

Rohan Sawhney and Keenan Crane. 2017. Boundary First Flattening. *ACM Trans. Graph.* 37, 1, Article 5 (2017), 14 pages. doi:10.1145/3132705

Thomas W. Sederberg, Peisheng Gao, Guojin Wang, and Hong Mu. 1993. 2-D Shape Blending: An Intrinsic Solution to the Vertex Path Problem. In *Proceedings of the 20th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '93)*. 15–18. doi:10.1145/166117.166118

Justin Solomon, Mirela Ben-Chen, Adrian Butscher, and Leonidas Guibas. 2011. As-Killing-As-Possible Vector Fields for Planar Deformation. *Computer Graphics Forum* 30 (2011), 1543–1552. doi:10.1111/j.1467-8659.2011.02028.x

Boris Springborn, Peter Schröder, and Ulrich Pinkall. 2008. Conformal Equivalence of Triangle Meshes. *ACM Trans. Graph.* 27, 3 (2008), 11. doi:10.1145/1360612.1360676

William T. Tutte. 1963. How to Draw a Graph. *Proceedings of The London Mathematical Society* 13 (1963), 743–767. https://api.semanticscholar.org/CorpusID:13517317

Amir Vaxman, Christian Müller, and Ofir Weber. 2015. Conformal Mesh Deformations with Möbius Transformations. *ACM Trans. Graph.* 34, 4, Article 55 (2015), 11 pages. doi:10.1145/2766915

Ofir Weber, Mirela Ben-Chen, and Craig Gotsman. 2009. Complex Barycentric Coordinates with Applications to Planar Shape Deformation. *Computer Graphics Forum* 28 (2009), 587–597. doi:10.1111/j.1467-8659.2009.01399.x

T. Winkler, J. Drieseberg, M. Alexa, and K. Hormann. 2010. Multi-Scale Geometry Interpolation. *Computer Graphics Forum* 29, 2 (2010), 309–318. doi:10.1111/j.1467-8659.2009.01600.x

Dong Xu, Hongxin Zhang, Qing Wang, and Hujun Bao. 2006. Poisson Shape Interpolation. *Graphical Models* 68, 3 (2006), 268–281. doi:10.1016/j.gmod.2006.03.001

Zhibang Zhang, Guiqing Li, Guodong Wei, Yupan Wang, Huina Lu, and Qing Yuan. 2015. Survey on shape interpolation. *Jisuanji Fuzhu Sheji Yu Tuxingxue Xuebao/Journal of Computer-Aided Design and Computer Graphics* 27 (08 2015), 1377–1388. https://www.jcad.cn/en/article/id/7ff5196f-b160-4cbc-a985-dfaafa3b2af3 In Chinese.

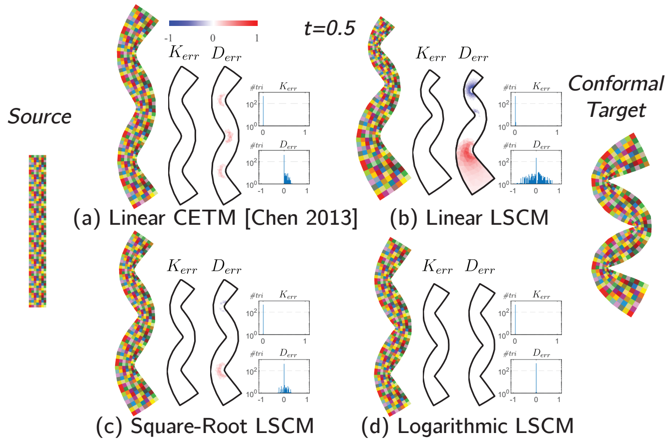


Fig. 5. Interpolation of conformal deformation. All three LSCM blendings (b-d) produced conformal results, while logarithmic (d) additionally eliminated area errors, as evidenced by the histograms and color plots.

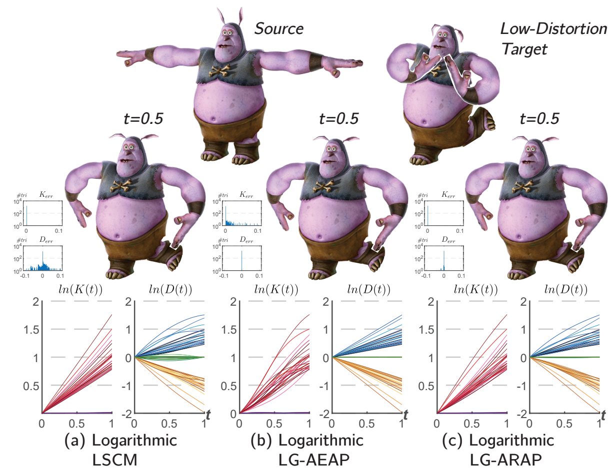


Fig. 6. Logarithmic LSCM and Local/Global realizations on low-distortion deformation. Desired distortions should be log-linear, note how each method minimizes differently distortion deviation.

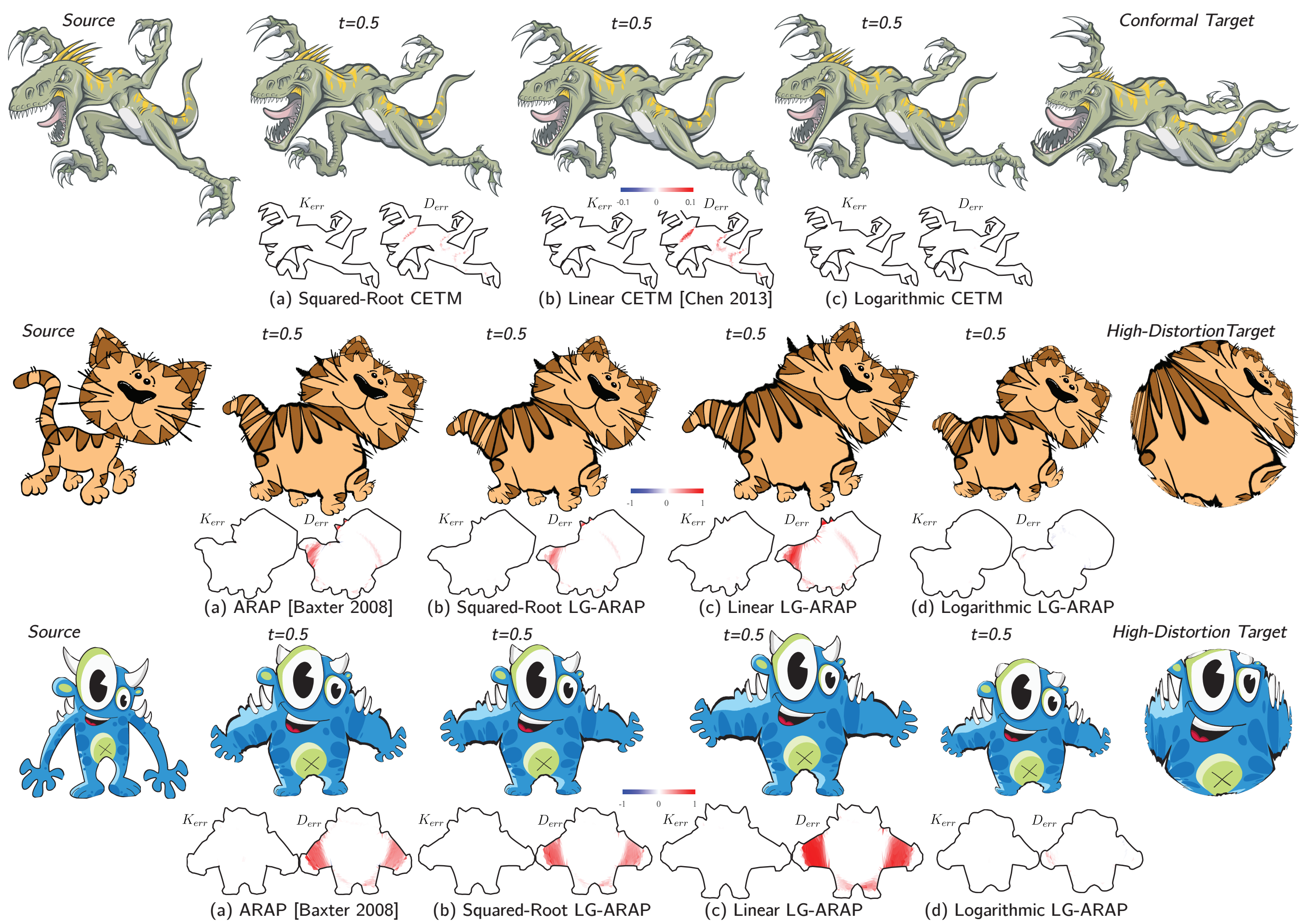


Fig. 7. Demonstration of all blending methods in various settings. Raptor's deformation is conformal. The Cat and Blue Monster deformations are Tutte embedding to circle. Note the last column in each example, where we see that logarithmic blending uniquely minimizes area distortion error D_{err}.

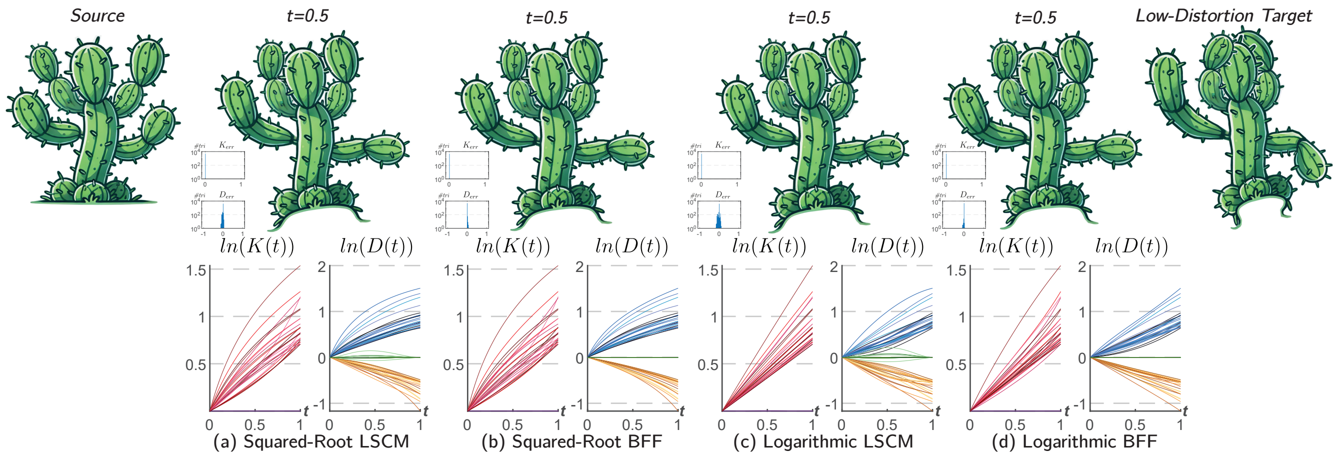


Fig. 8. Interpolation of low-distortion deformation with metric blending LSCM and edge blending BFF, both producing smooth conformal-bounded realizations. We see that BFF realization (b)(d) produces similar results to the respective counterparts (a)(c), albeit with less area distortion.

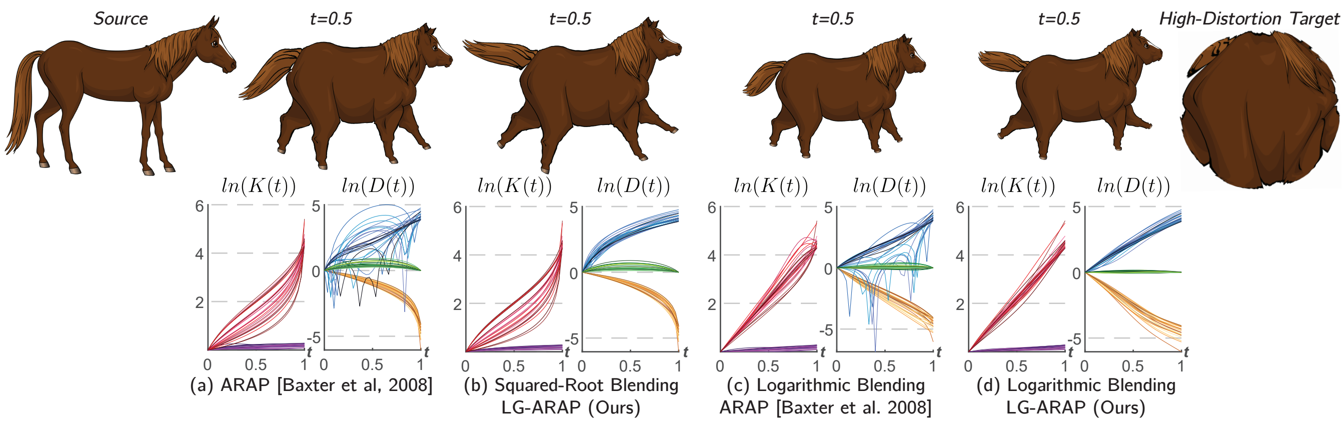


Fig. 9. Interpolation of high-distortion Tutte Circle Embedding with Baxter et al. [2008]'s ARAP realization and our Local Global ARAP realization. We can see that our realization produces smoother results, and that our logarithmic blending method produces less area distortion than square-root blending method.

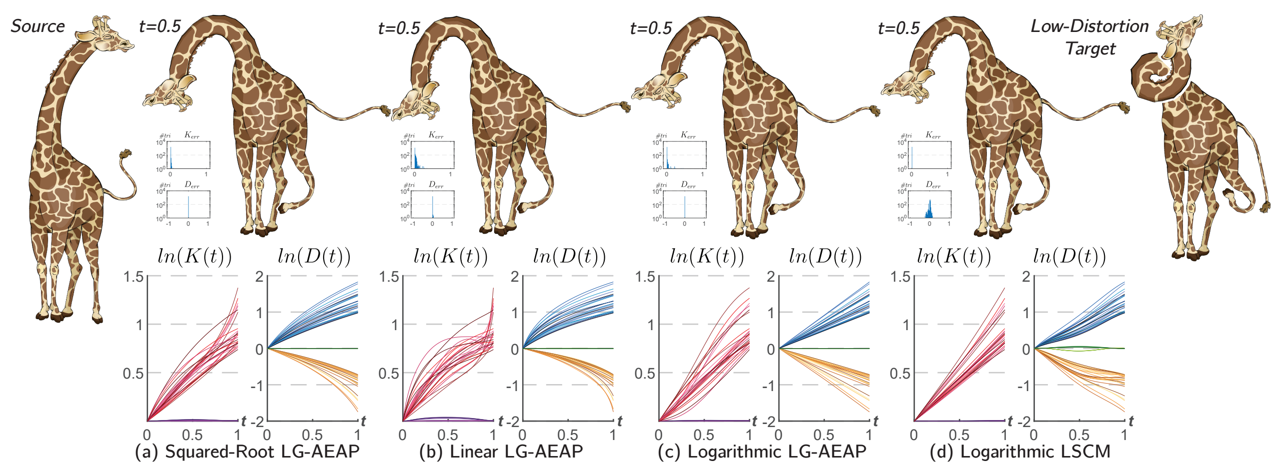


Fig. 10. Interpolation of low-distortion deformation. We can see that realizing with LG-AEAP (a-c) minimize area distortion deviation, producing smooth and cohesive area distortions for all three blendings, at the expense of added conformal distortion deviation.