

Biharmonic Coordinates

Ofir Weber¹

Roi Poranne²

Craig Gotsman²

¹Courant Institute of Mathematical Sciences, New York University, USA

²Computer Science Department, Technion, Haifa, Israel

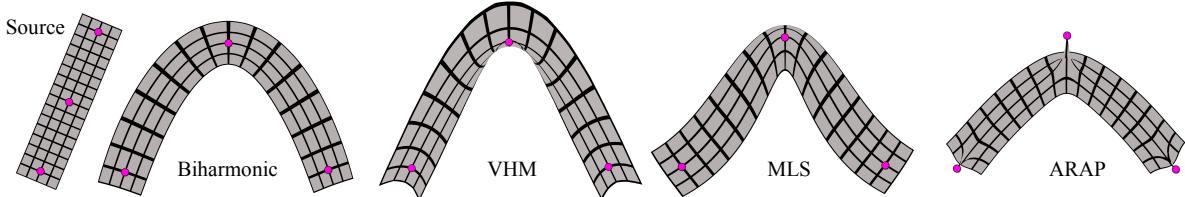


Figure 1: Source shape deformed using various algorithms by moving three control points.

Abstract

Barycentric coordinates are an established mathematical tool in computer graphics and geometry processing, providing a convenient way of interpolating scalar or vector data from the boundary of a planar domain to its interior. Many different recipes for barycentric coordinates exist, some offering the convenience of a closed-form expression, some providing other desirable properties at the expense of longer computation times. For example, harmonic coordinates, which are solutions to the Laplace equation, provide a long list of desirable properties (making them suitable for a wide range of applications), but lack a closed-form expression.

We derive a new type of barycentric coordinates based on solutions to the biharmonic equation. These coordinates can be considered a natural generalization of harmonic coordinates, with the additional ability to interpolate boundary derivative data. We provide an efficient and accurate way to numerically compute the biharmonic coordinates and demonstrate their advantages over existing schemes. We show that biharmonic coordinates are especially appealing for (but not limited to) 2D shape and image deformation and have clear advantages over existing deformation methods.

Categories and Subject Descriptors (according to ACM CCS): I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling (C.2.1, C.2.4, C.3) I.3.7 Animation

1. Introduction

Barycentric coordinates have been used extensively in the last decade for a wide variety of graphics applications. The most common applications are: mesh parameterization, rendering, shape deformation and color-and-data interpolation. Originally defined for triangles, barycentric coordinates have evolved and been generalized in many ways to perform different tasks.

In the most basic setup, we are given a planar region $\Omega \subset \mathbb{R}^2$ bounded by a polygon (sometimes called a “cage”), with m vertices $v_j = (x_j, y_j)$ and edges $e_j = (v_j, v_{j+1})$, $j = 1, \dots, m$. We use a prime superscript to distinguish between an internal point $x \in \Omega$ and a boundary point $x' \in \partial\Omega$. A set of functions $w_j(x)$ are called *barycentric coordinates*, if for any point $x \in \Omega$ they satisfy the following two properties:

identity reproduction:

$$\sum_{j=1}^m w_j(x)v_j = x \quad (1)$$

partition of unity:

$$\sum_{j=1}^m w_j(x) = 1 \quad (2)$$

Interpolating or approximating a set of values f_j given at the vertices v_j , is achieved by linearly blending the coordinate functions with the given data at the polygon vertices as coefficients:

$$f(x) = \sum_{j=1}^m w_j(x)f_j \quad (3)$$

(1) and (2) are sometimes referred to as “linear precision” and “constant precision” respectively. Combined together they imply that the interpolant $f(x)$ reproduces affine functions from their boundary values.

Beside (1) and (2), the coordinates are often required to satisfy some additional desirable properties such as smoothness, positivity, Lagrange (the ability to exactly interpolate the user data), etc. Thus, the choice of the most suitable set of coordinates is application-dependent.

Over the years, the basic concept of barycentric coordinates has been extended in many different ways. Some extensions allow the coordinate functions w_j to be complex-valued functions [WBCG09], while others allow the polygon to be an arbitrary smooth curve [Bel06] or even a 3D surface [FKR05, LLCO08]. The barycentric coordinates that we will describe here belong to a family of coordinates that supports higher order interpolation. i.e., not only do they possess the ability to exactly interpolate boundary data, but they also interpolate derivative information given along the boundary.

For this to be realized, (3) must be modified to take the following form:

$$f(x) = \sum_{j=1}^m \alpha_j(x)f_j + \beta_j(x)d_j \quad (4)$$

namely, the single set of functions $w_j(x)$ is replaced with two distinct sets of functions $\alpha_j(x)$ and $\beta_j(x)$. f_j are the prescribed values at the polygon vertices as before, and d_j are the prescribed normal derivatives along the edges of the polygon. More precisely, $d = \partial f / \partial n = \nabla f \cdot n$ where n is the unit normal to the boundary. The function f is assumed to vary linearly on the edges and the normal derivative to be constant on each edge.

Property (1) is reformulated as:

$$\sum_{j=1}^m \alpha_j(x)v_j + \beta_j(x)n_j = x \quad (5)$$

where n_j is the unit normal to the j^{th} edge, and property (2) reformulated as:

$$\sum_{j=1}^m \alpha_j(x) = 1 \quad (6)$$

The second term in (5) is the natural extension to (1) since the normal derivatives of the identity map are exactly the unit normals. Note that the β_j do not appear in (6) since the derivatives of a constant function are zero. It is important to emphasize that α_j on their own are not barycentric coordinates, as they do not satisfy (1).

Finally, for $f(x)$ to actually interpolate the boundary values and derivatives, the Lagrange property must be replaced by the analogous Hermite property:

$$\begin{aligned} \alpha_j(v_i) &= \delta_{ij} & \beta_j(x') &= 0, x' \in \partial\Omega \\ \frac{\partial \alpha_j}{\partial n}(x') &= 0, x' \in \partial\Omega & \frac{\partial \beta_j}{\partial n}(x') &= \delta_{ij}, x' \in e_i \end{aligned} \quad (7)$$

In this paper, we show how to construct a set of barycentric coordinates that satisfy (5)-(7). Additionally, they respect the shape of the object, decaying proportionally to the inner distance in a smooth manner. Such coordinates can be constructed by solving the biharmonic Dirichlet problem:

$$\begin{aligned} \Delta^2 f(x) &= 0, & x \in \Omega \\ f(x') &= g_1(x'), & x' \in \partial\Omega \\ \frac{\partial f}{\partial n}(x') &= g_2(x'), & x' \in \partial\Omega \end{aligned} \quad (8)$$

where Δ is the Laplace operator and g_1, g_2 are the prescribed boundary conditions. The biharmonic Dirichlet problem admits a unique solution under very mild conditions. This solution f minimizes the so-called Hessian energy and has a harmonic Laplacian at all points of Ω . As a consequence, these coordinates are smooth – C^∞ away from the boundary.

Since any harmonic function is also biharmonic, biharmonic functions are a natural extension to harmonic functions. Yet this extra power (compared to harmonic) can be a mixed blessing if care is not taken, as biharmonic functions do not (in general) satisfy the maximum principle and as a fourth order PDE, (8) is more prone to oscillations. Nevertheless, by incorporating a semi-automatic control mechanism over the boundary behavior we can design simple and intuitive tools to obtain results dramatically better than the relevant alternatives.

2. Previous Work

2.1 Barycentric Coordinates

There are many different constructions of barycentric coordinates for planar polygons. Wachspress [Wac75] was the first to define barycentric coordinates for convex polygons and apply them to finite elements analysis. Pinkall and Polthier [PP93] derived the so-called discrete harmonic coordinates (also called “cotangent weights”) based on a finite-element discretization of the Dirichlet energy. For non-convex polygons, these classic constructions result in discontinuous coordinates which makes them unusable.

Mean-value coordinates [Flo03] were proposed as an alternative to discrete harmonic since they are guaranteed to be positive in the kernel of a star-shaped polygon. They were further generalized [HF06] for arbitrary non-convex polygons, although the positivity property is not maintained. Figure 2 demonstrates the undesirable effect of negative coordinate values when mean-value coordinates are used to perform planar image deformation. In fact (contrary to what their name seems to imply), mean-value coordinates do not satisfy the classical mean-value property.

While all the coordinates mentioned above possess simple closed-form expressions, the need for barycentric coordinates with more tailored properties triggered further research. Lipman et al. [LKC07] modified the construction of mean-value coordinates to force them to be positive for non-convex polygons at the price of longer numerical evaluation. However, these coordinates are not smooth.

Joshi et al. [JMD*07] proposed harmonic coordinates, defined as solutions of the Laplace equation over the domain. Being harmonic functions, the coordinates obey the minimum/maximum principle and since the boundary values are forced to be in the range $[0,1]$, the coordinates are guaranteed to be positive even in highly concave regions, without sacrificing smoothness (they are C^∞ inside the domain).

On the down side, harmonic coordinates do not possess an analytic formula even for fairly simple polygons. [JMD*07] suggested using a multigrid solver to accelerate their computation. [Rus08, BCWG09] used the boundary elements method (BEM) [BD89] to construct harmonic coordinates in 3D. Recently, Weber and Gotsman [WG10] showed how to accurately and efficiently compute 2D harmonic coordinates based on a second-order discontinuous discretization of the celebrated Cauchy integral formula from complex analysis.

Hormann and Sukumar [HF08] proposed the maximum entropy coordinates (MEC) as a possible alternative to harmonic coordinates. MEC are also smooth, positive and interpolating and can be computed by solving a convex optimization problem. The important question of the existence of barycentric coordinates (for non-convex polygons), which are positive, interpolating, smooth and have a closed form, remains open.

Lipman et al. [LLCO08] proposed Green Coordinates – an extension to barycentric coordinates especially suitable for shape deformation. The deformation is represented as a linear combination of two sets of barycentric coordinates: one for the polygon’s vertices and the other for its normals. This construction is quite similar in spirit to ours, as we will see in Section 4. In fact, (4) has exactly the same form as that used in [LLCO08], although the Green coordinates do not satisfy (7).

Weber et al. [WBCG09] introduced the concept of complex barycentric coordinates by allowing the barycentric coordinates to be complex-valued functions. They proposed several recipes for such complex coordinates, the most fundamental one being the Cauchy coordinates (which is shown to be equivalent to Lipman et al.’s Green coordinates in 2D). In [WBCG09], the coordinate functions are designed to be holomorphic; thus, their linear combinations (with complex coefficients) are also holomorphic, implying that the 2D mappings they produce have the potential of being conformal maps. Even though it is meaningless to talk about positivity of complex numbers, it is quite remarkable that Cauchy coordinates (which possess a closed-form expression) are well behaved even in highly concave regions. Unfortunately, their simple and efficient computation comes at the price of being able to only approximate the boundary values.

In follow-up work, Weber and Gotsman [WG10] demonstrated that Cauchy/Green coordinates do not *always* produce conformal maps, since a complex holomorphic function will be conformal only if its first complex derivative does not vanish. They proposed Hilbert coordinates and showed how to use them in order to construct maps from the complex plane to itself which are *guaranteed* to be conformal. Their deformation algorithm allows the user to prescribe rotations along the boundary and exactly interpolate them, but no direct control over position is given.

Recently, it was shown in [WBCG*11] that complex barycentric coordinates may be used also to generate useful non-holomorphic complex maps in the plane. A general construction for complex barycentric coordinates was given, along with a closed-form recipe for (non-holomorphic) complex coordinates which interpolate the boundary values (MAGIC coordinates).

2.2 Hermite Interpolation

Biharmonic coordinates are not the first to address the problem of interpolating derivative boundary values (the so-called “Hermite setting”). Dyken and Floater [DF09] extended mean-value coordinates to the Hermite setting. However, those coordinates suffer from the same artifacts that regular mean-value coordinates do, namely, they perform poorly on non-convex shapes (see Figure 2). Furthermore, as opposed to regular mean-value, they do not possess a closed-form expression and numerical integration is required.

Manson and Schaefer [MS10] showed how to construct a family of barycentric coordinates based on moving least squares (MLS) fitting of polynomials (of arbitrary degree) and extended them to also support derivative interpolation. Unfortunately, like mean-value, they do not handle concave regions properly and suffer from the same “leakage” artifact across boundary edges. Both [DF09] and [MS10] do not provide a proof that their constructions actually satisfy the derivative constraints (although numerical experiments seem to imply that this is indeed the case).

A related concept, but one that does not entirely conform to the Hermite setting, was described by Langer and Seidel [LS08]. They gave a general recipe to convert any existing barycentric coordinates to higher-order ones. This provides the ability to interpolate boundary values as well as derivatives, but *only* at the *vertices* of the cage. Control over the rest of the boundary (i.e. edges in 2D and faces in 3D) is impossible.

2.3 Variational Methods

Barycentric coordinates lie at the heart of many deformation methods due to their smoothness, simplicity and the ability to compute them at a preprocessing step. They can be used, directly [JMD*07, WBCG09, LLCO08] or indirectly, to create a meaningful low-dimensional subspace of deformations from which an “optimal” deformation may be selected by solving a variational (i.e. optimization) problem [BCWG09, MKB*08, WG10].

Most modern deformation techniques are based on the variational approach, in which an energy functional is minimized subject to a set of user-defined constraints. Popular choices for energies are the Dirichlet (harmonic) energy [ZRK*05, JMD*07], which minimizes the magnitude of the Jacobian of the deformation map, the conformal energy (As-Similar-As-Possible - ASAP), which penalizes the deviation of the Jacobian matrix from being a similarity transformation (composition of rotation with a uniform scale) [IMH05, WBCG09], and As-Rigid-As-Possible (ARAP), which minimizes the distance of the Jacobian from the closest rotation [IMH05, BCWG09].

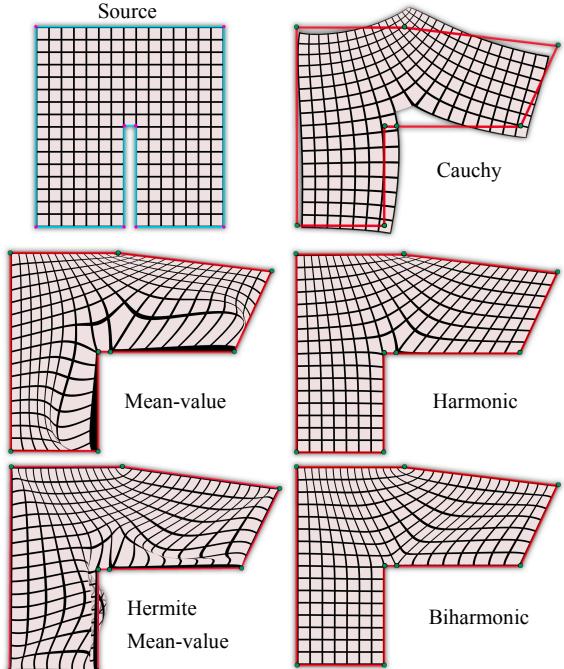


Figure 2: Deformation of the “Pants” shape (top-left). Cauchy coordinates produce angle-preserving deformation but miss the boundary. Negative values of the mean-value coordinates cause a repelling artifact between the “legs”. Harmonic coordinates overcome this problem, but result in shear on the boundary. Hermite mean-value provides the ability to specify boundary derivatives but result in severe artifacts due to the concavity of the shape. Biharmonic coordinates generate a good compromise with full control over boundary behavior.

The VHM method [BCWG09] is a variant of the ARAP method, and closely related to our deformation algorithm. VHM operates in the subspace of harmonic maps, parameterized by two functions defined on the boundary of the domain. In addition, the Jacobian of the deformation at a small set of points (so-called anchors) located along the skeleton of the shape is required to be close to rotations. Finally, a regularization term is added to compensate for the fact that rigidity is forced only at small portion of the

shape (the anchors). As we will see, our deformation algorithm has two clear advantages over VHM. First, it does not require knowledge of the skeleton of the shape. Second, it does not rely on a user knob to control the balance between smoothness and rigidity.

Botsch and Kobbelt [BK04] presented an interactive shape deformation framework which in contrast to our *space* deformation method operates on manifolds. The user assigns manipulators to different parts of the shape and applies affine transformations to them. The transformations are propagated to the rest of the shape by recursively solving the Laplace equation. Higher order Laplacians allow smoother transition between constrained and unconstrained regions and in particular, a second order Laplacian results in a C^1 continuous biharmonic function. Andrews et al. [AJC11] also solved high order harmonic systems in their surface modeling technique. This allowed them to explicitly control Dirichlet and Neumann boundary conditions simultaneously.

Recently, Jacobson et al. [JBPS11] presented a method for computing weight functions (not quite barycentric coordinates), specifically designed for shape deformation. Their main observation is that in order to achieve appealing results, the weights should be positive and at least C^1 at the control handles. The weights are computed by solving a (convex) constrained optimization problem using the finite element method (essentially decomposing the region of interest in space into tetrahedral elements). The energy is the Hessian energy with natural boundary conditions (i.e. normal derivative is zero) whose minimizer without positivity constraints is a biharmonic function. Figure 3 compares between these Bounded Bihamonic Weights (BBW) and our $\alpha_j(x)$ coordinate. We observed that in the majority of cases, $\alpha_j(x)$ tend to be positive even without explicitly constraining it. It is important to note that in general, BBW satisfy Eq. (2) but not (1). In order to achieve identity reproduction, the weights are linearly blended with transformations rather than with positions. However, without the complementary $\beta_j(x)$ coordinate, it is impossible to achieve explicit control over boundary derivatives.

3. Contributions

We formulate three different deformation methods, all based on our new biharmonic coordinates. Each method has a different user interface and provides different advantages. The most sophisticated one (Section 5.4) is a competitive alternative to ARAP, which allows the Jacobian near the boundary to be an orthogonal matrix (as opposed to orthonormal), eliminating shearing artifacts while preserving the “thickness” of the object (see Figure 1 for a sneak peek).

The main contributions of this paper are:

- 1) Proposal of biharmonic coordinates as a general mathematical tool for data interpolation.
- 2) Description of an efficient and accurate algorithm for the computation of biharmonic coordinates (closed-form expressions for the BEM basis functions used in the computation are also provided).
- 3) Introduction of a deformation algorithm based on the novel concept of shape “thickness”.

4. Biharmonic Functions

In this section, we derive the biharmonic coordinates as solutions to the biharmonic Dirichlet problem and describe a numerical procedure to compute these coordinates based

on the direct Boundary Elements Method (BEM) approach [BD89].

4.1 The Boundary Integrals

Fuglede [Fug81] provide a boundary integral equation for the biharmonic Dirichlet problem. For completeness, we provide here a slightly different, more detailed derivation for the same equation, as we have found it to be more revealing. Our starting point is Green’s second identity:

$$\int_{\Omega} \xi \Delta \eta - \eta \Delta \xi dA = \int_{\partial\Omega} \xi \frac{\partial \eta}{\partial n} - \eta \frac{\partial \xi}{\partial n} ds \quad (9)$$

where η and ξ are any twice-differentiable functions in $\Omega \subset \mathbb{R}^2$, Δ is the Laplace operator, the integral on the left is over the area of Ω and the integral on the right is over the boundary $\partial\Omega$. Now denote by G^H and G^{BH} the fundamental solutions of the 2D harmonic and biharmonic equation respectively:

$$G^H(x, x') = -\frac{1}{2\pi} \ln \|x - x'\|$$

$$G^{BH}(x, x') = -\frac{\|x - x'\|^2}{8\pi} (\ln \|x - x'\| - 1)$$

and by f a biharmonic function. Plugging $\xi = \Delta f$ and $\eta = G^{BH}$ into (9) leads to:

$$\begin{aligned} & \int_{\Omega} \Delta f \Delta G^{BH} - G^{BH} \Delta^2 f dA \\ &= \int_{\partial\Omega} \Delta f \frac{\partial G^{BH}}{\partial n} - G^{BH} \frac{\partial \Delta f}{\partial n} ds \end{aligned} \quad (10)$$

In contrast to [Fug81], we use a slightly different function for G^{BH} that has the property $\Delta G^{BH} = G^H$. Since $\Delta^2 f = 0$, we have:

$$\int_{\Omega} \Delta f G^H dA = \int_{\partial\Omega} \Delta f \frac{\partial G^{BH}}{\partial n} - G^{BH} \frac{\partial \Delta f}{\partial n} ds \quad (11)$$

Returning to (9), but this time plugging in $\xi = f$ and $\eta = G^H$ and rearranging the terms results in:

$$f = \int_{\Omega} \Delta f G^H dA + \int_{\partial\Omega} f \frac{\partial G^H}{\partial n} - G^H \frac{\partial \Delta f}{\partial n} ds \quad (12)$$

which is Green’s third identity. Finally, combining (11) and (12) leads to the *boundary integral* identity for the biharmonic function f :

$$\begin{aligned} f(x) &= \int_{\partial\Omega} f(x') \frac{\partial G^H}{\partial n} - G^H \frac{\partial f(x')}{\partial n} ds \\ &+ \int_{\partial\Omega} \Delta f(x') \frac{\partial G^{BH}}{\partial n} - G^{BH} \frac{\partial \Delta f(x')}{\partial n} ds \end{aligned} \quad (13)$$

Eq. (13) implies that we can evaluate a biharmonic function f at any point $x \in \Omega$ if we know the values of f , $\partial f / \partial n$, Δf , $\partial \Delta f / \partial n$ on the boundary. Moreover, given any such values on the entire boundary, the integral in (13) will produce a biharmonic function in Ω . However, this function will not have the specified boundary values, unless they were taken from a biharmonic function in the first place. In particular, since Δf is a harmonic function, it is uniquely defined by $\partial \Delta f / \partial n$ along the boundary. Substituting Δf instead of f in (12) leads to the following extra condition that Δf must satisfy:

$$\Delta f(x) = \int_{\partial\Omega} \Delta f(x') \frac{\partial G^H}{\partial n} - G^H \frac{\partial \Delta f(x')}{\partial n} ds \quad (14)$$

4.2 Boundary Discretization

The main idea of BEM is to represent the function we are solving for as a boundary integral, then discretize this boundary (as opposed to Finite Element Methods which discretize the entire domain) and derive the solution by solving a linear system of equations. Although the obtained solution only approximates the true solution (with high fidelity), it is a C^∞ biharmonic function expressed in a closed form.

Since our domain is polygonal, we can convert the boundary integrals (13) and (14) to a sum over the m polygon edges $e_j, j = 1, \dots, m$:

$$\begin{aligned} f(x) &= \sum_{j=1}^m \int_{e_j} f \frac{\partial G^H}{\partial n} ds - \sum_{j=1}^m \int_{e_j} \frac{\partial f}{\partial n} G^H ds \\ &\quad + \sum_{j=1}^m \int_{e_j} \Delta f \frac{\partial G^{BH}}{\partial n} ds - \sum_{j=1}^m \int_{e_j} \frac{\partial \Delta f}{\partial n} G^{BH} ds \\ \Delta f(x) &= \sum_{j=1}^m \int_{e_j} \Delta f \frac{\partial G^H}{\partial n} ds - \sum_{j=1}^m \int_{e_j} \frac{\partial \Delta f}{\partial n} G^H ds \end{aligned}$$

We are interested in a particular biharmonic function that has continuous piecewise-linear values on the polygon edges and piecewise-constant normal derivatives. Unfortunately, the Laplacian Δf and its normal derivative $\partial \Delta f / \partial n$ can be quite complicated functions. Nevertheless, in order to solve the boundary integral equation, we will assume that Δf can be faithfully approximated by a piecewise-linear function and $\partial \Delta f / \partial n$ by a piecewise-constant function on $\partial\Omega$. Note that in the special case that f is an affine function, these assumptions are accurate and f will be reproduced exactly, and since affine precision implies linear and constant precision, the advantage of using this particular construction is now clear. Namely, that equations (5) and (6) are satisfied *exactly*.

Since $\partial \Delta f / \partial n$ and $\partial f / \partial n$ are constant on each edge, they may be factored out of the integral sign. Expressing f and Δf on each edge e_j as a linear function of the values f_j, f_{j+1} and $\Delta f_j, \Delta f_{j+1}$ at the endpoints, and evaluating the integrals, leads to:

$$\begin{aligned} f(x) &= \sum_{j=1}^m \phi_j^H(x) f_j + \psi_j^H(x) d_j \\ &\quad + \phi_j^{BH}(x) l_j + \psi_j^{BH}(x) k_j \end{aligned} \tag{15}$$

$$\Delta f(x) = \sum_{j=1}^m \phi_j^H(x) l_j + \psi_j^H(x) k_j \tag{16}$$

where $f_j = f(v_j)$, $l_j = \Delta f(v_j)$ and $d_j = \partial f(x') / \partial n$, $k_j = \partial \Delta f(x') / \partial n$ for $x' \in e_j$. $\phi_j^H, \psi_j^H, \phi_j^{BH}, \psi_j^{BH}$ are the basis functions obtained from computing the integrals. Fortunately, these integrals may be solved analytically, and the exact expressions are given in Appendix A. To the best of our knowledge, the solution of the above integrals has not appeared in the literature before.

4.3 Biharmonic Coordinates

We now have all the necessary building blocks for setting up the linear system. Eqs. (15) and (16) are valid for any point $x \in \Omega$, and although the basis functions ϕ_j^H, ψ_j^H ,

ϕ_j^{BH}, ψ_j^{BH} are singular on the boundary, their limits when x approaches $x' \in \partial\Omega$ exist.

Evaluating (16) at the vertices v_i results in m linear equations:

$$l_i = \sum_{j=1}^m \phi_j^H(v_i) l_j + \psi_j^H(v_i) k_j, \quad i = 1 \dots m \tag{17}$$

which has the matrix form: $l = \Phi^H l + \Psi^H k$, where l and k are $m \times 1$ column vectors and Φ^H and Ψ^H are $m \times m$ matrices. Rearranging the equation and inverting Ψ^H leads to the following linear relation $k = Al$, where

$$A = (\Psi^H)^{-1}(I - \Phi^H)$$

Following the same path, we evaluate (15) at the vertices, obtaining the following system of equations:

$$f = \Phi^H f + \Psi^H d + \Phi^{BH} l + \Psi^{BH} k \tag{18}$$

Plugging in the expression for k , combining similar terms and inverting another matrix, we can express l as

$$l = Bf + Cd, \text{ where}$$

$$B = D(I - \Phi^H)$$

$$C = -D\Psi^H$$

$$D = (\Phi^{BH} + \Psi^{BH} A)^{-1}$$

Denoting by $\phi_j^H(x), \psi_j^H(x), \phi_j^{BH}(x), \psi_j^{BH}(x)$ the $1 \times m$ row vectors obtained by evaluating the appropriate basis functions at a given fixed x , we obtain the matrix form of Eq. (13):

$$f(x) = \phi^H(x)f + \psi^H(x)d + \phi^{BH}(x)l + \psi^{BH}(x)k$$

Plugging in the expressions derived above for l and k :

$$\begin{aligned} f(x) &= \phi^H(x)f + \psi^H(x)d \\ &\quad + (\phi^{BH}(x) + \psi^{BH}(x)A)(Bf + Cd) \end{aligned}$$

Finally, we are able to express the biharmonic function $f(x)$ using the following linear combination:

$$f(x) = \alpha(x)f + \beta(x)d$$

where

$$\begin{aligned} \alpha(x) &= \phi^H(x) + \zeta(x)B \\ \beta(x) &= \psi^H(x) + \zeta(x)C \\ \zeta(x) &= \phi^{BH}(x) + \psi^{BH}(x)A \end{aligned} \tag{19}$$

and conclude that the vectors $\alpha(x)$ and $\beta(x)$ are the biharmonic coordinates of x . A color visualization of the coordinate functions for a sample vertex and edge of a simple polygon is given in Figure 3.

To summarize: the algorithm for computing the biharmonic coordinates boils down to evaluating the four $m \times m$ matrices $\Phi^H, \Psi^H, \Phi^{BH}, \Psi^{BH}$ based on the expressions given in Appendix A and using them to compute the three matrices A, B, C . The latter involves two matrix inversions and several matrix multiplications and additions. Once this is done (typically in a preprocessing stage), Eq. (19) is used to compute the biharmonic coordinates at as many points $x \in \Omega$ as needed.

Finally, we mention that it is possible to improve the accuracy of the results by subsampling the polygon along the edges in two ways. First, (15) and (16) can be evaluated at the subsamples, resulting in additional equations for the systems (17) and (18). Since this system will be over-constrained, the matrix inversion used for computing A and B is replaced by a pseudo-inverse. This typically leads to smoother results near the boundary. The second way to improve accuracy is by adding “virtual” vertices at the subsamples, which will result in a larger, but still square

system. The prescribed boundary values of the new vertices are simply linearly interpolated from those of the original vertices, making the refinement process transparent to the user.

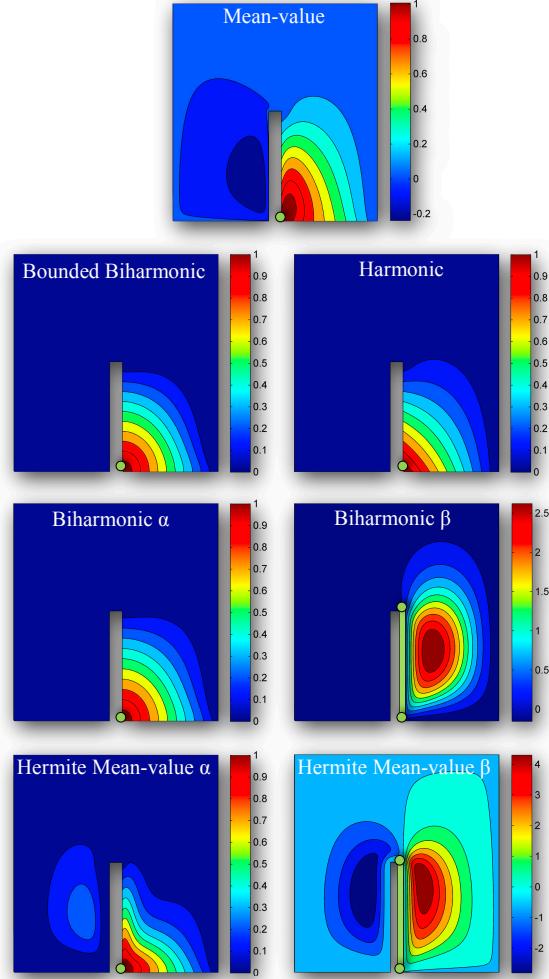


Figure 3: Visualization of various coordinate schemes for the vertex / edge marked in green. Top-down – mean-value contains undesirable negative values. Bounded biharmonic weight [JBPS11] are forced to be positive. Harmonic coordinates are positive as well due to the min / max principle. Our biharmonic α is positive for that particular vertex, which explains its similarity to [JBPS11]. Note the unintuitive local maximum on the left hand side of Hermite mean-value α . Hermite mean-value β becomes negative.

5. Shape Deformation

Although biharmonic coordinates are general enough to be applied to any kind of scalar or vector data interpolation, from now on we focus exclusively on the application of planar shape deformation.

A simple setup of a deformation algorithm based on barycentric coordinates is obtained by treating the boundary values as vectors in \mathbb{R}^2 , allowing the user to interact with the system by dragging the vertices v_j of the cage Ω to some new positions f_j . In response, Eq. (3) or (4) is invoked, mapping each point $x \in \Omega$ to a new position $f(x)$. The mathematical properties of the map $f: \mathbb{R}^2 \rightarrow \mathbb{R}^2$ and the quality of the deformation are then derived from the properties of the coordinates. In particular, the Jacobian of

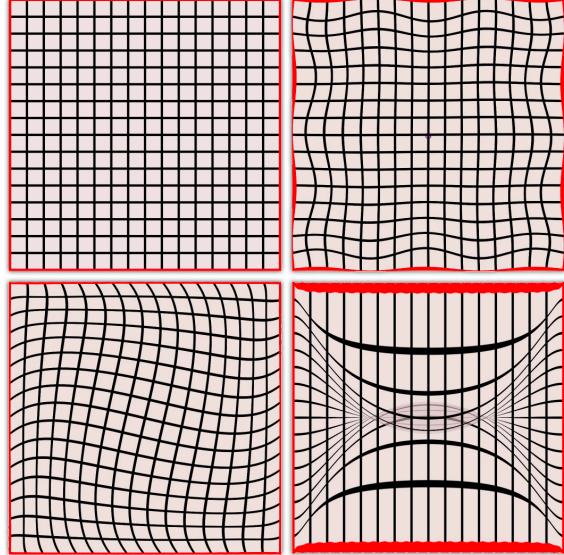


Figure 4: A square (top-left) is mapped to itself in various ways using biharmonic coordinates, varying only the normal derivatives. Forcing the derivative to be too large (bottom-right) leads to self-overlap.

the map, which depends linearly on the derivatives of the coordinates, characterizes the local behavior of the deformation.

In contrast to traditional barycentric coordinates, which allow only the prescription of boundary values, biharmonic coordinates provide the freedom to prescribe both boundary values and normal derivatives. Specifically, biharmonic coordinates allow to prescribe the complete 2×2 (piecewise constant) Jacobian matrix on the boundary. To see this, denote the j^{th} 2×1 edge vector as $e_j = (v_{j+1} - v_j)$ and the edge unit normal as $n_j = (e_j)^{\perp} / \|e_j\|$. The action of the Jacobian matrix J of f on an edge vector is given by $Je_j = f_{j+1} - f_j$. Similarly, the action of J on the normal is $Jn_j = d_j$. Hence, since e_j and n_j are fixed, specifying f_j and d_j uniquely determine J .

Figure 4 shows a variety of maps from the square to itself, obtained by keeping the boundary fixed while manually changing the normal derivative vector direction and magnitude. While this example demonstrates the flexibility and expressiveness of biharmonic coordinates, in some scenarios a user may want to prescribe only boundary positions, and avoid dealing with boundary derivatives all together. Moreover, as evident in Figure 4 bottom right, a careless specification of boundary derivatives may lead to unnatural results. To alleviate this, we now provide several methods for automatically deducing boundary derivatives from boundary positions alone. Which method to choose depends on the effect that the user wants to achieve, thus application-dependent.

5.1 Orthogonal Boundary Jacobians

A natural choice in many scenarios is to set the normal derivative vector to be perpendicular to the target edge. The corresponding Jacobians on the boundary are orthogonal but not necessarily orthonormal. Denote the unit normal to the j^{th} edge of the target cage as $\tilde{n}_j = (f_{j+1} - f_j)^{\perp} / \|f_{j+1} - f_j\|$ and consider the case where $d_j = s_j \tilde{n}_j$ for some real scalar s_j . The question that remains is then, what value should we assign to s_j ? As a

first obvious choice we set $s_j = 1$ so that d_j has unit length. This means that the deformation will not contain stretch in the normal direction. Figure 5 compares between bending a bar using harmonic coordinates and using biharmonic coordinates with orthogonal boundary Jacobians with fixed $s_j = 1$.

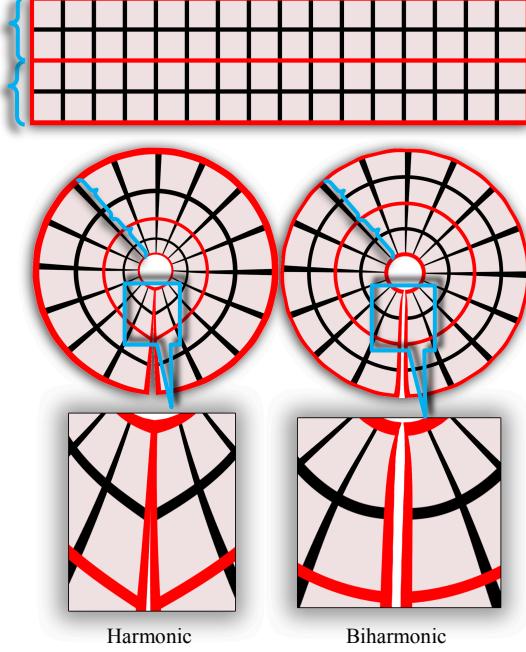


Figure 5: Bending of a bar by specifying a circular boundary. On the left, harmonic coordinates results in shear. In addition, the center line (red) is unnaturally shifted away from the center of rotation. On the right, biharmonic coordinates with orthogonal boundary Jacobians (Section 5.1) maintains the center line exactly in the middle, keeping the anisotropic scale aligned to the boundary.

5.2 Complex Barycentric Coordinates

An alternative choice is to restrict the Jacobians on the boundary to be similarity matrices. This can be achieved by setting $s_j = \|f_{j+1} - f_j\|/\|e_j\|$. The deformation will tend to preserve angles between any two intersecting curves in the vicinity of the boundary with the added value of interpolating the boundary (with possibly more shear in the interior).

Recall an elementary fact from complex analysis; a 90 degree rotation can be expressed by multiplication with the complex number i . Let us interpret the vectors d_j and f_j as complex numbers and impose $d_j = i(f_{j+1} - f_j)/\|e_j\|$. Rearranging the terms in Eq. (4), we can express the mapping formula in complex form:

$$f(z) = \sum_{j=1}^m C_j(z) f_j$$

$$C_j(z) = \alpha_j(z) + i \left(\frac{\beta_{j-1}(z)}{\|e_{j-1}\|} \right) - \frac{\beta_j(z)}{\|e_j\|}$$

where $C_j(z)$ are complex barycentric coordinates [WBCG09]. Since the real part $\alpha_j(z)$ is not harmonic, obviously these complex coordinates are not holomorphic. While Weber et al. [WBCG*11] showed how to construct non-holomorphic complex coordinates that have the La-

grange property, our coordinates are the first complex barycentric coordinates that have the Hermite property. Figure 6 demonstrates this effect and compares our result to Cauchy coordinates which preserve angles everywhere but only approximate the cage target position.

5.3 As-Harmonic-As-Possible Deformation

In more complicated scenarios, when the source and target cages are significantly different, a more sophisticated way to choose s_j is required. Since we consider harmonic coordinates to be the preferred choice for deformation among all real coordinate recipes, the closest to this is what we call the As-Harmonic-As-Possible (AHAP) map that still satisfies the user constraints.

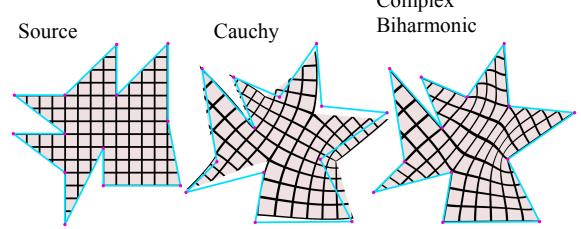


Figure 6: Deformation of the star: Cauchy coordinates produces a conformal map, but the boundary is not interpolated. Using complex biharmonic coordinates (Section 5.2) we are able to interpolate the boundary while maintaining conformality at the vicinity of the boundary.

We define the harmonicity of a biharmonic function $f(x)$ as:

$$E_{AHAP}(f) = \int_{\partial\Omega} \|\Delta f(x)\|^2 ds \quad (20)$$

where the integration is done solely on the boundary. Recall that $\Delta f(x)$ is harmonic, hence achieves minimum/maximum values on the boundary. Therefore minimizing $\|\Delta f(x)\|^2$ on the boundary bounds its values in the interior. We also experimented with integrating over the entire domain and obtained similar results. In our discrete setting, (20) becomes $\|l\|^2$, where $l = Bf + Cd$ (Section 4.3) and $d_j = s_j \tilde{n}_j$. Thus, given the vector of boundary values f , the only unknowns are the scalars s_j .

Since negative values of s_j lead to a non-bijective map, we restrict s_j to be positive. Thus, minimizing E_{AHAP} boils down to a non-negative least squares problem with a non-empty feasible region, which implies the existence of a global minimum. Figure 8 shows a square deformed into a diamond shape using harmonic coordinates, biharmonic coordinates with orthogonal unit length normal derivatives (Section 5.1) and the AHAP biharmonic map. The latter manages to create a map that satisfies the orthogonality constraints on the boundary while avoiding self-intersections. Away from the boundary, the map resembles the harmonic deformation.

5.4 Thickness-Preserving Shape Deformation

Cage-based deformation follows naturally from the use of barycentric coordinates, nonetheless, manipulating the entire cage to produce a deformation is tedious and can lead to suboptimal results. To simplify the user’s task, we follow [IMH05, SMW06, WBCG09, BCWG09] and use the point-to-point deformation metaphor. The user selects a small set of control points inside the cage (usually at joints and extremities of the shape), and controls the deformation by repositioning only these control points. The idea is to

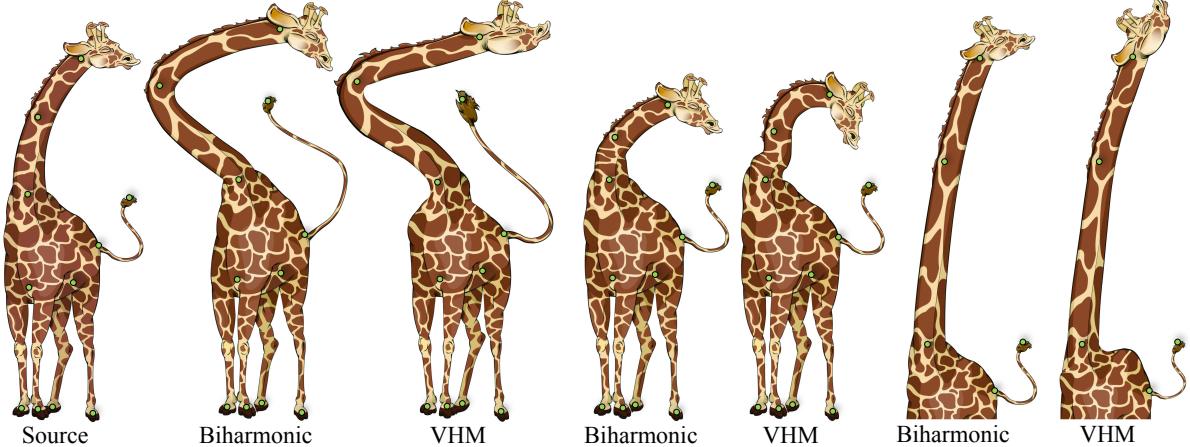


Figure 7: Comparison of the deformation of a giraffe using VHM and our thickness-preserving method for three different positions of the control points. Our method easily handles shortening or elongation of the giraffe’s neck and tail, while VHM results in artifacts at the head and tail in an effort to preserve length. Note that our method correctly deals with bending when desired.

make the cage invisible to the user, although behind the scenes we solve an inverse problem. i.e., we search for a target cage that will transform the control points to their prescribed target locations. Since the number of control points is typically much lower than the number of cage vertices, the problem is under-constrained and requires a regularization term, similarly to other variational deformation methods.

Let us define the *local thickness* $\sigma(x')$ of a map f at a boundary point x' as:

$$\sigma(x') = \tilde{n}^t(f(x'))J(x')n(x') \quad (21)$$

$J(x')n(x')$ is the normal derivative vector, denoted by $d(x')$, and \tilde{n} is the *target* unit normal (at $f(x')$). The *local thickness* $\sigma(x')$ measures the amount of stretch in the normal direction. In continuum mechanics terminology, this is simply the normal strain in the normal direction. In our discrete setting, $d(x')$ is piecewise constant and assumed to be aligned with \tilde{n} , hence the *local thickness* on the j^{th} edge is:

$$\sigma_j = \tilde{n}_j^t J_j n_j = \tilde{n}_j^t d_j = \tilde{n}_j^t \tilde{n}_j s_j = s_j.$$

We call a map that has unit local thickness at any boundary point *and* orthogonal Jacobians, a *thickness-preserving* map. Note that the maps described in Section 5.1 are *thickness-preserving* by construction. However, in this Section, instead of fixing the boundary as is done in Section 5.1, we fix only a small number of internal points. We then search the subspace of biharmonic maps and pick a map which is *thickness-preserving* while satisfying the *point-to-point* position constraints.

Consider a small offset of the boundary curve in the direction of the inward normal and the narrow band that surrounds the swept area. Intuitively, a thickness-preserving map deforms this narrow band such that the image of the band maintains its width (but not necessarily its length).

Such a map does not limit the change in the tangential component along the boundary and allows articulated objects to stretch their “bones”. Nevertheless, even though we disallow shear on the boundary (by setting d to be equal to \tilde{n}), shear and stretch may still appear as we move away from the boundary toward the interior of the shape. To tackle this, we design an energy that strives to mini-

mize changes in d . Note that since d is constant on each edge, its derivative in the tangent direction vanishes. Hence we aim at minimizing changes only in the normal direction. Formally, we would like the vector $\partial d / \partial n$ to vanish. Since in general, it is impossible for this term to vanish in the presence of positional constraints (at the control points), we aim to minimize this term by defining the following energy functional:

$$E^{\text{thickness}}(f) = \int_{\partial\Omega} \left\| \frac{\partial^2}{\partial n^2} f(x') \right\|^2 ds \quad (22)$$

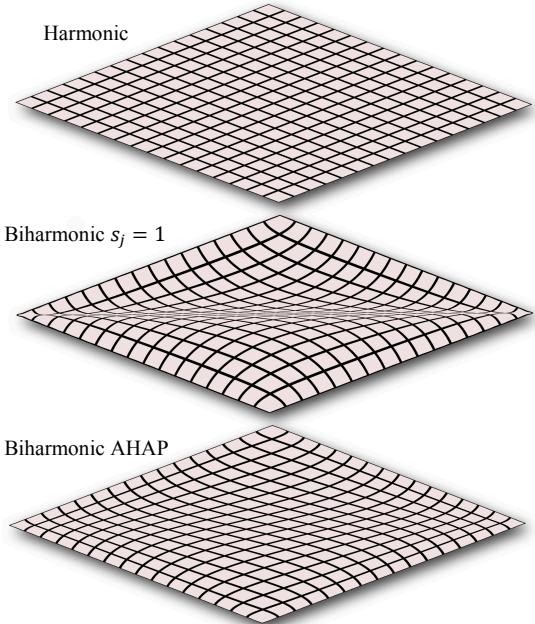


Figure 8: The unit square is mapped to a diamond shape: Harmonic coordinates produce shears near the boundary. A biharmonic map with $s_j = 1$ (Section 5.1) overlaps itself in the middle. A biharmonic As-Harmonic-As-Possible map (Section 5.3) is smooth, bijective, and perfectly aligned with the boundary.

In our discrete setting, the integral is replaced with a sum over k uniformly spaced boundary samples and the optimization problem becomes:

$$f_{optimal} = \operatorname{argmin}_{\{f_j\}_{j=1}^m} (E^{thickness}(f))$$

subject to:

$$f(p_i) = q_i, \quad i = 1, \dots, b$$

where b is the number of control points. Note that in the general case, the map f depends both on f_j and d_j , implying that the energy should be quadratic. However, we restricted the map to be thickness-preserving, meaning that $d_j = \tilde{n}_j = (f_{j+1} - f_j)^\perp / \|f_{j+1} - f_j\|$, hence, the map f and therefore the energy and the set of b constraints, depend (non-linearly) only on f_j .

Experimental Evaluation

We compare our thickness-preserving deformation method with the MLS [SMW06], ASAP and ARAP [IMH05], VHM [WBCG09] and BBW [JBPS11] methods. All share the same user interface. As a first sanity check, we place three control points on a horizontal bar; two at the far ends and one in the middle (illustrated in Figure 1). We then drag the middle control point upwards. Despite the simplicity of this setting, it is still quite challenging for existing deformation methods since the target positions of the control points imply that the bar should change its length. As a result, methods that are based on some notion of isometry (distance preserving) will try to resist this change and respond with artifacts.

As evident in Figure 1, the ARAP method results in discontinuity artifacts at the control points, while VHM produces a foldover in the middle of the bar and creates unnatural bending near the end points. Our deformation is C^∞ at the control points and is the only one that preserves the overall shape of the bar.

In Figure 7, we compare our method with VHM when deforming a giraffe shape. Note the distortion created by VHM at the head and tail of the giraffe when control points are pulled apart or brought close together. Figure 9 compares our method with the ARAP and ASAP methods. Again, the control points are positioned in a configuration that suggests that the giraffe's neck and legs should shorten. ASAP reacts by uniformly scaling down the head and legs dramatically, while ARAP, in an effort to preserve distances, bends the neck and legs. Our method simply shortens them to satisfy the positional constraint. However, this behavior can be altered easily. By adding a third control point in the middle of the neck and dragging it to the side, the shortening effect is replaced with bending (Figure 7 left).

Figure 11 compares our method with the Bounded Bi-harmonic Weights (BBW) method of Jacobson et al. [JBPS11]. BBW requires the additional user input of affine transformation per control point. In an attempt to automate this, Jacobson et al. proposed a heuristic to estimate rotation matrices from the position of control points. While this automatic method does not produce satisfying results for this particular deformation, manually tweaking the rotations and scale values results in a pleasing deformation comparable to our automatic thickness-preserving algorithm.

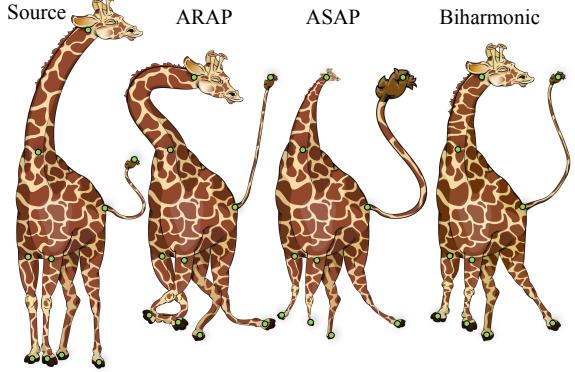


Figure 9: Deformation of a giraffe by different algorithms using identical positional constraints. ARAP bends the neck and legs in order to accommodate the user constraints. ASAP reduces dramatically the overall size of head, tail and legs, creating unnatural effect. Thickness-preserving biharmonic deformation simply shortens the neck and legs and extends the length of the tail as expected.

6. Implementation details

We have implemented our algorithms on an Intel i7 CPU with 8 GB of RAM. A typical number of cage edges for a fairly complicated model such as the Giraffe is 100. Each edge was subsampled (creating an extra 2-3 virtual vertices per edge) to achieve better accuracy.

The computation of the biharmonic coordinates (Section 4.3) was done in C++. Evaluating the basis functions and solving the relevant linear systems took 1 to 20 seconds of preprocessing time, depending on the complexity of the cage and the image resolution. All the cage-based deformation algorithms (Section 5 excluding 5.4) run in real time and achieve frame rates of at least 100fps for all the examples shown in this paper. To solve the optimization problem of Section 5.3, we used the CVXGEN quadratic programming solver [MB10] which turned out to be very efficient for problems with up to 50 variables.

The solution of the optimization problem for the thickness-preserving deformation (Section 5.4) was done in MATLAB using the non-linear constrained optimization solver (*fmincon*) from the Optimization Toolbox. We fed the solver with the analytic expression of the non-linear energy along with its gradient. Running times were somewhat slower, however we did manage to achieve interactive rates for inputs with up to 150 degrees of freedom. We believe that incorporating a dedicated solver into our C++ framework will improve performance by an order of magnitude.

To evaluate the non-linear energy (Eq. 22), we need to compute the second normal derivatives of α and β which, in turn, contain the normal derivatives of the harmonic and biharmonic basis functions. These derivatives (along with all the other basis functions) are computed during the pre-processing step. Although we were able to obtain closed-form expressions for the derivatives of the basis functions (using a symbolic solver), the expressions for the biharmonic functions turned out to be lengthy and we found that evaluating them numerically using a simple finite difference scheme leads to good results.

In addition, we found that numerically evaluating the expression for the limits of the biharmonic basis functions on the boundary (which is needed for Eqs. (17) and (18)) is

simple to implement and leads to accurate results. We simply offset the entire cage inwards by a small amount ($1e^{-6}$ in all our examples) and evaluate the expressions on the offset.

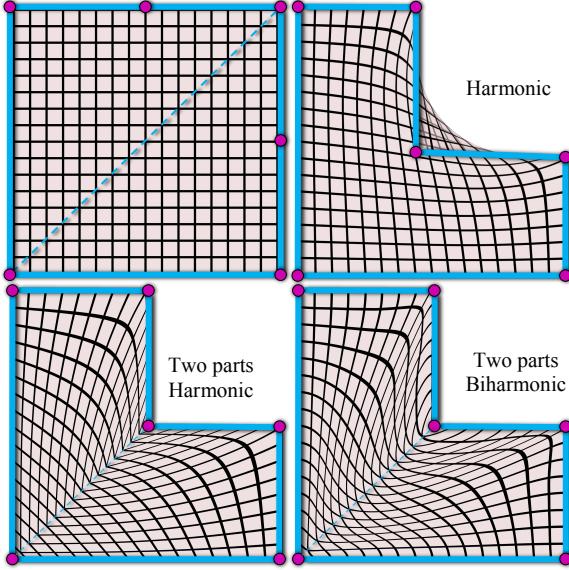


Figure 10: A square (top-left) is deformed using harmonic coordinates (top-right). Note how the interior of the square spills outside the target cage. We then split the square into two triangles, mapping each part separately and gluing the results back together. Both harmonic (bottom-left) and biharmonic (bottom-right) coordinates result in a bijective map. However, harmonic is only C^0 along the seam while biharmonic is C^1 .

7. Summary and Future Work

We have developed biharmonic barycentric coordinates as a useful tool for interpolating data values as well as normal derivatives from the boundary of a polygonal shape to its interior. Using the Boundary Elements Method, the computation of the coordinates is efficient and accurate and does not require discretization of the entire domain.

In the context of planar deformations, biharmonic coordinates enable full control over the boundary shape (as do harmonic coordinates) as well as boundary derivatives, thus can be regarded as a direct generalization to harmonic coordinates. Some of the maps within this richer subspace are superior to their harmonic counterparts (see Figures 5, 8, 10). Yet using biharmonic coordinates, in general, requires more user input to which it might be sensitive. For example, providing incorrect boundary derivative data might lead to undesirable effects. This is demonstrated in Figure 4 (bottom-right) and Figure 8 (middle). Also, our biharmonic coordinates do not possess the min / max principle that harmonic coordinates have; hence the coordinates are not guaranteed to be positive.

To alleviate these limitations and in an effort to simplify the user interaction, we provide several ways for automatically adjusting the boundary derivatives based only on the given boundary positions.

In addition, we have proposed a novel variational approach to deformation that encourages stretching in the tangent direction while eliminating shear in the vicinity of the boundary. We call it “thickness-preserving” deformation and have demonstrated its advantages over other state-of-the-art methods. This method has the advantage

that it is based *solely* on boundary formulation and does not require discretization of the interior e.g. by meshing (ARAP) or finding the skeleton of the shape (VHM).

Our biharmonic coordinates were defined in 2D. Nevertheless, they can be derived in any dimension (although closed-form expressions might be more difficult to obtain). In the future, we would like to explore their behavior on volumetric models. We believe that a 3D extension of the thickness-preserving method may be promising.

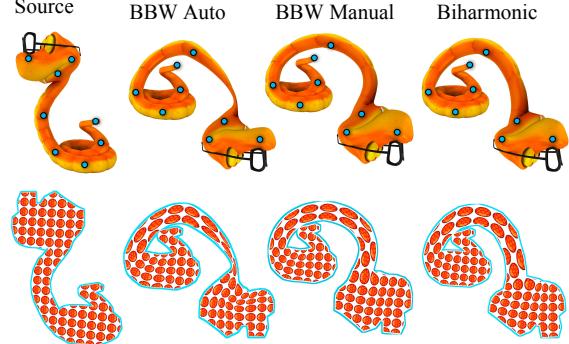


Figure 11: Deformation of the “snake with glasses”. Top row using the original image texture. Bottom row: same result textured with a pattern of circles. BBW with automatic rotation estimation results in a severe area loss. Manually tweaking the affine transformations creates a plausible result comparable to our automatic thickness-preserving algorithm (right).

The basic construction we used in this paper can also be extended to higher orders using polyharmonic functions, which are a generalization of harmonic and biharmonic functions. Although the motivation for this in the context of shape deformation is not obvious, we provide the first few steps of the derivation in Appendix B.

Figure 10 illustrates a deformation of a square to an L-shaped figure. In general, doing this with *any* barycentric coordinates will result in a “spill” outside the boundary. To overcome this, we split the square into two triangles and deform each part separately and then glue the results back together. We repeat the experiment using harmonic and biharmonic coordinates. Since harmonic coordinates do not allow any control over the normal derivative, the map is not smooth along the joint seam. In the biharmonic case, we can set the normal derivatives on both sides of the seam to be identical, resulting in a map which is smooth on the seam. This technique may be used to create C^1 basis functions on non-convex finite elements, which may be more powerful for physical simulations than the C^0 harmonic basis functions used by Martin et al. [MKB*08].

Acknowledgements

The authors are grateful to Alec Jacobson for inspiring discussions. Ofir Weber was partially supported by NSF IIS grant 0905502. We would like to thank the anonymous reviewers for their valuable comments.

References

- [AJC11] ANDREWS J., JOSHI P., CARR N.: A linear variational system for modeling from curves. Computer Graphics Forum 30, 6, (2011)

- [ACL83] ARONZAIN N., CREESE T.M., LIPKIN L.J.: Polyharmonic functions, Clarendon Press (1983).
- [BK04] BOTSCHE M., KOBELT L.: An intuitive framework for real-time freeform modeling. ACM Trans. Graphics 23, 3 (2004).
- [BCWG09] BEN-CHEN M., WEBER O., GOTSMAN C.: Variational harmonic maps for space deformation. In ACM Trans. on Graphics Vol. 28, No. 3 (Proc. SIGGRAPH 2009).
- [BD89] BREBBIA C. A., DOMINGUEZ J.: Boundary Elements: An Introductory Course, 2nd ed, Computational Mechanics Publications, McGraw-Hill, New York, 1989.
- [Bel06] BELAYEV A.: On transfinite barycentric coordinates. Proc. of SGP (2006).
- [DF09] DYKEN C., FLOATER M. S.: Transfinite mean value interpolation, Comp. Aided Geom. Design. 26 (2009).
- [FHK06] FLOATER M. S., HORMANN K., KÓS G.: A general construction of barycentric coordinates over convex polygons. Advances in Computational Mathematics 24, 1–4 (2006).
- [FKR05] FLOATER M. S., KÓS G., REIMERS M.: Mean-value coordinates in 3D. Comp. Aided Geom. Design 22 (2005).
- [Flo03] FLOATER M. S.: Mean value coordinates. Computer Aided Geometric Design 20, 1 (2003).
- [Fug81] FUGLEDE B.: On a direct method of integral equations for solving the biharmonic Dirichlet problem. ZAMM 61, 9 (1981).
- [HF06] HORMANN K., FLOATER M. S.: Mean value coordinates for arbitrary planar polygons. ACM Transactions on Graphics 25, 4 (2006).
- [HS08] HORMANN K., SUKUMAR N.: Maximum entropy coordinates for arbitrary polytopes. Computer Graphics Forum 27, 5 (2008). Proc. SGP.
- [IMH05] IGARASHI T., MOSCOVICH T., HUGHES J.-F.: As-Rigid-As-Possible shape manipulation. ACM Trans. Graph. (Proc. SIGGRAPH), 24, 3, (2005).
- [JMD*07] JOSHI P., MEYER M., DEROSSE T., GREEN B., SANOCKI T.: Harmonic coordinates for character articulation. ACM Transactions on Graphics 26. Proc. of SIGGRAPH (2007).
- [JBPS11] JACOBSON A., BARAN I., POPOVIC J., SORKINE O.: Bounded biharmonic weights for real-time deformation. ACM Trans. Graph. 30, 4, Proc. of SIGGRAPH (2011).
- [LLCO08] LIPMAN Y., LEVIN D., COHEN-OR D.: Green coordinates. ACM Transactions on Graphics 27, 3 (2008), Proc. of SIGGRAPH.
- [LS08] LANGER T., SEIDEL H.-P.: Higher order barycentric coordinates. Computer Graphics Forum (Proc. Eu-rographics) 27 (2), (2008).
- [LKC07] LIPMAN Y., KOPF J., COHEN-OR D., LEVIN D.: GPU-assisted positive mean value coordinates for mesh deformations. SGP 2007.
- [MB10] MATTINGLEY J., BOYD S.: CVXGEN: A code generator for embedded convex optimization, 2010.
- [MS10] MANSON J., SCHAEFER S.: Moving least squares coordinates. Computer Graphics Forum 29, 5. Proc. SGP (2010).
- [MKB*08] MARTIN S., KAUFMANN P., BOTSCHE M., WICKE M., GROSS M.: Polyhedral finite elements using harmonic basis functions. Computer Graphics Forum 27, 5, (2008).
- [PP93] PINKALL U., POLTHIER K.: Computing discrete minimal surfaces and their conjugates. Experiment. Math. 2, 1 (1993).
- [Rus08] RUSTAMOV R. M.: Boundary element formulation of harmonic coordinates. Technical report, Department of Mathematics, Purdue University, 2008.
- [SMW06] SCHAEFER S., MCPHAIL T., WARREN J.: Image deformation using moving least squares. ACM Transactions on Graphics 25, 3 (2006). Proc. of SIGGRAPH
- [Wac75] WACHSPRESS E. L.: A Rational Finite Element Basis, vol. 114 of Mathematics in Science and Engineering. Academic Press, New York, 1975
- [WBCG09] WEBER O., BEN-CHEN M., GOTSMAN C.: Complex barycentric coordinates with applications to planar shape deformation. Computer Graphics Forum 28, 2. Eu-rographics 2009.
- [WBCG*11] WEBER O., BEN-CHEN M., GOTSMAN C., HORMANN K.: A complex view of barycentric mappings. Computer Graphics Forum 30, 5 (SGP 2011).
- [WG10] WEBER O., GOTSMAN C.: Controllable conformal maps for shape deformation and interpolation. ACM Trans. on Graphics 29, 3 (Siggraph 2010).
- [ZRK*05] ZAYER R., RÖSSL C., KARNI Z., SEIDEL H.-P.: Harmonic guidance for surface deformation. In Computer Graphics Forum (2005).

Appendix A

Closed form expressions of the biharmonic basis functions

All of the basis functions are defined by integrals which can be solved analytically. To simplify calculations, we translate and rotate each edge:

$$e_i = v_{i+1} - v_i$$

such that it coincides with the interval $[0, c_i]$ on the x -axis, where $c_i = \|e_i\|$.

Denote the vector u_i as:

$$(u_i^x, u_i^y) = (x_0, y_0) - v_i$$

where (x_0, y_0) is the point at which we want to evaluate the basis functions.

we first apply the affine transformation $T_i(x_0, y_0)$:

$$T_i(x_0, y_0) = \frac{1}{c_i}(u_i^x e_i^x + u_i^y e_i^y, u_i^y e_i^x - u_i^x e_i^y)$$

then, instead of evaluating the solution at (x_0, y_0) , we evaluate it at $(x_i, y_i) = T_i(x_0, y_0)$.

To evaluate the basis functions, we first evaluate the following intermediate terms:

$$\begin{aligned} L_{1i} &= \log(x_i^2 + y_i^2) \\ L_{2i} &= \log(x_i^2 + y_i^2 + c_i^2 - 2c_i x_i) \\ A_i &= \tan^{-1}\left(\frac{x_i}{y_i}\right) + \tan^{-1}\left(\frac{c_i - x_i}{y_i}\right) \\ \hat{\phi}_i^H &= \frac{2x_i A_i - y_i(L_{1i} - L_{2i})}{4\pi c_i} \\ \hat{\phi}_i^{BH} &= \frac{1}{16\pi c_i} \left(4A_i x_i y_i^2 - L_{2i} y_i (x_i^2 - y_i^2 - c_i^2) \right. \\ &\quad \left. + L_{1i} y_i (x_i^2 - y_i^2) - 2y_i c_i (c_i + x_i) \right) \end{aligned}$$

Finally, the basis functions are given by:

$$\begin{aligned} \psi_i^H &= -\frac{1}{4\pi} (2A_i y_i + L_{1i} x_i - 2c_i + (c_i - x_i)L_{2i}) \\ \phi_i^H &= \hat{\phi}_{i-1}^H - \hat{\phi}_i^H + \frac{A_i}{2\pi} \\ \psi_i^{BH} &= -\frac{1}{144\pi} \left(3L_{1i} x_i (3y_i^2 + x_i^2) \right. \\ &\quad \left. + 3L_{2i} (c_i - x_i) (c_i^2 - 2c_i x_i + 3y_i^2 + x_i^2) \right. \\ &\quad \left. + 12A_i y_i^3 + 24 (c_i^2 x_i - c_i x_i^2) - 8c_i^3 - 30c_i y_i^2 \right) \\ \phi_i^{BH} &= \hat{\phi}_{i-1}^{BH} - \hat{\phi}_i^{BH} \\ &\quad + \frac{y_i}{8\pi} (2A_i y_i + L_{1i} x_i - 3c_i + L_{2i} (c_i - x_i)) \end{aligned}$$

Appendix B

Polyharmonic Coordinates

Our construction of biharmonic coordinates can be generalized to *polyharmonic* coordinates. A polyharmonic function of degree r is a function $u(x)$ that satisfies:

$$\Delta^r u(x) = 0$$

The corresponding Dirichlet conditions consist of up to the r -th derivative in normal direction of u on the bounda-

ry. The analogous Green's third identity for the poly-Laplacian [ACL83] consists of r equations, the s^{th} of which is:

$$\Delta^{s-1} u = \sum_{i=1}^s \int_{\partial\Omega} \Delta^{i-1} G_i \frac{\partial \Delta^{s-i} u}{\partial n} - \Delta^{s-i} \frac{\partial G_i}{\partial n} \Delta^{i-1} u$$

where $\Delta^0 u = u$ and G_i is the fundamental solution of the i^{th} harmonic equation:

$$G_i(x, x') = \frac{1}{2\pi} \|x - x'\|^{i-1} (a_i \ln \|x - x'\| - b_i)$$

and a_i and b_i are given by the recurrence:

$$\begin{aligned} a_1 &= 1, & b_1 &= 0 \\ a_i &= \frac{a_{i-1}}{4(i-1)^2}, & b_i &= \frac{a_{i-1}}{4(i-1)^2} \left(\frac{a_{i-1}}{i-1} + b_{i-1} \right) \end{aligned}$$

We can follow the same path as in Section 4, discretize the boundary and assume that for each r , $\Delta^r u$ is linear and $\frac{\partial \Delta^r u}{\partial n}$ is constant on each edge. Integrating results in a set of r equations similar to (15). We can then solve for f and the derivatives and form the polyharmonic coordinates.