

Moving Least Squares Coordinates

Josiah Manson, Scott Schaefer

Texas A&M University

Abstract

We propose a new family of barycentric coordinates that have closed-forms for arbitrary 2D polygons. These coordinates are easy to compute and have linear precision even for open polygons. Not only do these coordinates have linear precision, but we can create coordinates that reproduce polynomials of a set degree m as long as degree m polynomials are specified along the boundary of the polygon. We also show how to extend these coordinates to interpolate derivatives specified on the boundary.

Categories and Subject Descriptors (according to ACM CCS): I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—Boundary representations

1. Introduction

Barycentric coordinates are a standard interpolation technique in Computer Graphics. These coordinates solve a boundary value interpolation problem and can be used to interpolate discrete scalar fields, vector fields or even multidimensional fields over irregular tessellations. While barycentric coordinates were first generalized for Finite Element Analysis [Wac75], the Graphics community has made heavy use of these coordinates for applications such as texture mapping [DMA02], polygon shading [HF06], spline surfaces [LD89] and surface deformation [JSW05].

Suppose $p_i \in \mathbb{R}^2$ for $i = 1 \dots n$ are the vertices of a polygon. These points define a piecewise polynomial boundary curve $P_i(t)$ such that $P_i(t) = (1-t)p_i + tp_{i+1}$. Furthermore, assume that each edge of this polygon has an associated function $F_i(t) \in \mathbb{R}$. We define a barycentric interpolant of this function as

$$\hat{F}(x) = \sum_{i=1}^n \int_0^1 B_i(x,t) F_i(t) dt$$

where $B_i(x,t)$ is the barycentric coordinate function associated with the i^{th} edge at parameter t and $x \in \mathbb{R}^2$ is a point in the domain. If $F_i(t) = (1-t)f_i + tf_{i+1}$, then the interpolant takes on the more familiar form,

$$\hat{F}(x) = \sum_{i=1}^n b_i(x) f_i$$

where $b_i(x) = \int_0^1 (1-t)B_i(x,t) + tB_{i-1}(x,t) dt$ are the barycentric coordinates of x with respect to the vertices p_i .

For $b_i(x)$ to be barycentric, several properties must hold. First, barycentric coordinates should produce an interpolant that takes on the value of $F_i(t)$ along the boundary. Therefore,

$$\hat{F}(P_i(t)) = F_i(t).$$

Second, the coordinates should have linear precision. This means that, for all linear $L(x)$,

$$L(x) = \sum_{i=1}^n \int_0^1 B_i(x,t) L(P_i(t)) dt.$$

Note that if the coordinates have linear precision, they also form a partition of unity ($\sum_{i=1}^n \int_0^1 B_i(x,t) dt = 1$). This property trivially follows from linear precision if $L(x) = 1$. While these properties are necessary for $B_i(x,t)$ to be barycentric, we typically require that $B_i(x,t)$ is also smooth in practice.

1.1. Related Work

Barycentric coordinates were originally described by Möbius for simplices such as triangles in 2D [Möb27]. While barycentric coordinates are unique for triangles, there are many possible solutions for polygons with more sides. Wachspress extended the idea of barycentric basis function to convex polygons [Wac75] for use in finite element analysis, but Wachspress coordinates become undefined over the

interior of the polygon when the polygon is concave. A general construction for different families of barycentric coordinates defined over convex polygons [FHK06] was recently found by Floater et al. In the same paper, the authors showed that Wachspress coordinates are a member of this family.

Until Mean Value Coordinates (MVC) [Flo03, HF06] were discovered, no known coordinates were well defined over concave polygons. In MVC, the weights at a point x are calculated by integrating the values on a boundary line over the arc spanned by the line in the polar coordinate system around x . However, since the direction of integration is reversed for back-facing lines, it is possible to obtain negative values for coordinates in concave polygons. Lipman et al. [LKKOL07] modified MVC to ensure that their Positive MVC are always positive even for concave polygons. To prevent integrating over back facing lines, they integrate only over the lines that are visible from a point x . However, this means that Positive MVC are not smooth.

Joshi et al. [JMD*07] determined that Harmonic coordinates are both positive and smooth. Harmonic coordinates are the solution to Laplace's equation subject to the boundary constraints demanded by the interpolatory property of barycentric coordinates. Although it is unclear that it makes sense to talk about an optimal basis, the fact that Harmonic coordinates minimize curvature of the basis function and prevent disconnected areas from influencing each other are desirable traits. Unfortunately, calculating Harmonic coordinates requires discretizing the function domain into finite elements and solving a large linear system. Even then, the basis functions are approximate and cannot be evaluated exactly.

Hormann and Sukumar found another form of positive barycentric coordinates by adapting principles from statistics. In their Maximum Entropy Coordinates (MEC) [HS08], each vertex is given a probability distribution function that approaches infinity as a sample point x approaches the edges adjacent to that vertex. The coordinates of x are then given by the probability of a vertex being chosen with no bias at x . MEC get their name because finding probabilities with minimal bias maximizes entropy, which is the mechanism through which the probabilities are found. Since probabilities must be between zero and one, MEC are guaranteed to be positive and are probably smooth. Although MEC can be calculated far more directly and efficiently than Harmonic Coordinates, they have no closed form and must be solved for through an iterative process.

Some methods have also extended barycentric coordinates to curved (transfinite) boundaries. Several methods [WHD07, SJW07] generalize barycentric coordinates to arbitrary convex sets that can be bounded by a parameterized curve. In his theoretical analysis of barycentric coordinates [Bel06], Belyaev found a generalization of transfinite coordinates, of which he found that MVC and Wachspress coordinates are instances.

Dyken et al. [DF09] and Floater et al. [FS08] developed a method for extending MVC to interpolate derivatives (Hermite data) on concave, curved boundaries. Although these extended MVC are able to reproduce cubic functions, they require boundary derivatives as input to do so. Our method, however, can reproduce functions of degree m if the boundary values $F_i(t)$ are sampled from that degree m function without using any derivatives. Another advantage of our method is that our coordinates have a closed-form expression, whereas Hermite MVC coordinates require numeric integration. Langer et al. [LS08] also provide a method for modifying barycentric coordinate constructions to add derivative information at the vertices of the polygon.

Another approach to calculating basis functions by solving a moving least squares problem. One application this has been used for is in approximating functions from a set of point samples [Wen01]. In Image Deformation Using Moving Least Squares (IDMLS) [SMW06], Schaefer et al. apply moving least squares interpolants to image deformation while optionally restricting shear and scaling in their similarity and rigid deformations. Schaefer et al. also provide a closed form solution for finding basis functions of line segments. We show that the affine construction of IDMLS can be generalized to calculate barycentric coordinates for closed (or even non-closed) polygons. We extend this construction to higher degree curves/functions and show how to incorporate derivatives along the boundary as well.

1.2. Contributions

The progression in barycentric coordinates has been to generalize from simplices to convex polygons and finally to concave polygons. Our coordinates are well defined for those polygons, but go even further. Our coordinates are well defined for non-closed polygons and self-intersecting polygons (though the latter creates discontinuities with incompatible data at the self-intersection). Our barycentric coordinates

- create a family of barycentric coordinates that are well-defined for arbitrary 2D polygons, closed or not,
- can handle arbitrary transfinite boundary curves for which we provide closed-form expressions when the boundary is piecewise linear,
- can reproduce polynomials of arbitrary degree with the appropriate polynomial functions $F_i(t)$ provided on the boundary,
- allow for interpolation of cross-boundary derivatives.

2. Barycentric Coordinates

The basic approach that we use to create barycentric coordinates is to find a polynomial $\hat{F}(x)$ that minimizes the squared distance between $\hat{F}(P_i(t))$ and $F_i(t)$ along the boundary. If we represent $\hat{F}(x)$ in the power basis, then $\hat{F}(x) = V_m(x)C$, where

$$V_m(x) = (1 \quad x_1 \quad x_2 \quad \dots \quad x_1^m \quad x_1^{m-1}x_2 \quad \dots \quad x_2^m)$$

represents a polynomial of total degree m . The coefficients for the functions in $V_m(x)$ are given by C and $x = (x_1 \ x_2)$. The best approximating polynomial is completely described by the coefficients C , which are given by

$$\operatorname{argmin}_C \sum_{i=1}^n \int_0^1 \|P'_i(t)\| (V_m(P_i(t))C - F_i(t))^2 dt.$$

In the integral above, $\|P'_i(t)\|$ gives an arc-length parameterization of the curve so that any unit length of the boundary has equal weight. Unfortunately, this simple minimization interpolates boundary values only when the fitting error is zero.

By weighting parts of the boundary that are closer to the point of evaluation x such that the weights approach infinity as x approaches the boundary, we can interpolate arbitrary function values at the boundary. Several weight functions have this property, but a natural choice is to make the weight function inversely proportional to distance so that

$$W_i(x,t) = \frac{\|P'_i(t)\|}{\|P_i(t) - x\|^{2\alpha}}$$

where α controls the speed at which $W_i(x,t)$ decays, and $\|P'_i(t)\|$ gives an arc-length parameterization. The moving least squares minimization can then be formulated as

$$\operatorname{argmin}_C \sum_{i=1}^n \int_0^1 W_i(x,t) (V_m(P_i(t))C - F_i(t))^2 dt. \quad (1)$$

Since Equation 1 is quadratic in C , the global minimum is found where the derivative is zero. Let

$$A = \sum_{i=1}^n \int_0^1 W_i(x,t) V_m^T(P_i(t)) V_m(P_i(t)) dt.$$

Then C is given by

$$C = \sum_{i=1}^n A^{-1} \int_0^1 W_i(x,t) V_m^T(P_i(t)) F_i(t) dt.$$

If we suppose that $F_i(t)$ can be represented in a polynomial basis of degree k such as the Bernstein basis $\beta_{j,k}(t) = \binom{k}{j} (1-t)^{k-j} t^j$, then $F_i(t) = \sum_{j=0}^k \beta_{j,k}(t) f_{i,j}$ where $f_{i,j}$ represents the j^{th} coefficient of the function on the i^{th} edge. From our definition of $\hat{F}(x)$, we find that

$$\begin{aligned} \hat{F}(x) &= V_m(x)C \\ &= V_m(x) \sum_{i=1}^n \sum_{j=1}^k \left[A^{-1} \int_0^1 W_i(x,t) V_m^T(P_i(t)) \beta_{j,k}(t) dt \right] f_{i,j} \\ &= \sum_{i=1}^n \sum_{j=1}^k B_{i,j}(x) f_{i,j} \end{aligned}$$

Notice that, if $F_i(t)$ forms a continuous function, then there are duplicated entries in $f_{i,j}$ because $f_{i,k} = f_{i+1,0}$. Therefore this weighted sum above can be reindexed to remove duplicates, which provides a simple, closed-form expression for

the barycentric coordinate functions associated with the coefficients $f_{i,j}$. If $k = 1$, then this summation can be rewritten in terms of $b_i(x)$ such that

$$\hat{F}(x) = \sum_{i=1}^n b_i(x) f_i$$

where $b_i(x) = B_{i,0}(x) + B_{i-1,1}(x)$ and f_i are the function values specified at the vertices p_i of the polygon. Note that this construction produces an entire family of barycentric coordinates corresponding to different values of α .

For linear $P_i(t)$ and $F_i(t)$, this method reproduces the line segment construction for affine transformations in IDMLS [SMW06]. The difference is that we never explicitly construct the affine transformation matrix and provide a simpler form of the solution in terms of barycentric coordinates. Furthermore, our construction is far more general and Section 4 shows how to compute the integrals for $m, k > 1$ and for various values of α .

3. Derivatives

While the construction up to this point has focused on building an interpolating function $\hat{F}(x)$ for values along the boundary, it can also be useful to specify derivatives along the boundary that $\hat{F}(x)$ should interpolate. Since $\hat{F}(x)$ interpolates the boundary, all derivatives of $\hat{F}(x)$ on the boundary in the direction of the boundary curve are fully constrained. Derivatives perpendicular to the boundary curve, however, are unconstrained. We define the direction perpendicular to the domain curve at a given parameter t as

$$P_i^\perp(t) = \frac{(-P'_{i,2}(t) \ P'_{i,1}(t))}{\|P'_i(t)\|}.$$

With this notation, the derivative of $V_m(x)$ along the boundary in the direction of $P_i^\perp(t)$ is then

$$G_m(t) = P_i^\perp(t) \begin{pmatrix} \frac{\partial V_m}{\partial x_1}(P_i(t)) \\ \frac{\partial V_m}{\partial x_2}(P_i(t)) \end{pmatrix}.$$

If the user provides derivatives $F_i^\perp(t)$ along the boundary in the direction of $P_i^\perp(t)$, then we can find a set of coefficients C that minimize both the error in function values and derivatives by

$$\operatorname{argmin}_C \sum_{i=1}^n \left[\int_0^1 W_i(x,t) (V_m(P_i(t))C - F_i(t))^2 dt + \int_0^1 W_i(x,t) (G_m(t)C - F_i^\perp(t))^2 dt \right]. \quad (2)$$

We have chosen to use the same weight function for convenience of notation, but values and derivatives can have different weights. This minimization is still quadratic in C and has a global minimum given by

$$\begin{aligned} A &= \sum_{i=1}^n \int_0^1 W_i(x,t) \left(V_m^T(P_i(t)) V_m(P_i(t)) + G_m^T(t) G_m(t) \right) dt \\ C &= \sum_{i=1}^n A^{-1} \int_0^1 W_i(x,t) \left(V_m^T(t) F_i(t) + G_m^T(t) F_i^\perp(t) \right) dt. \end{aligned}$$

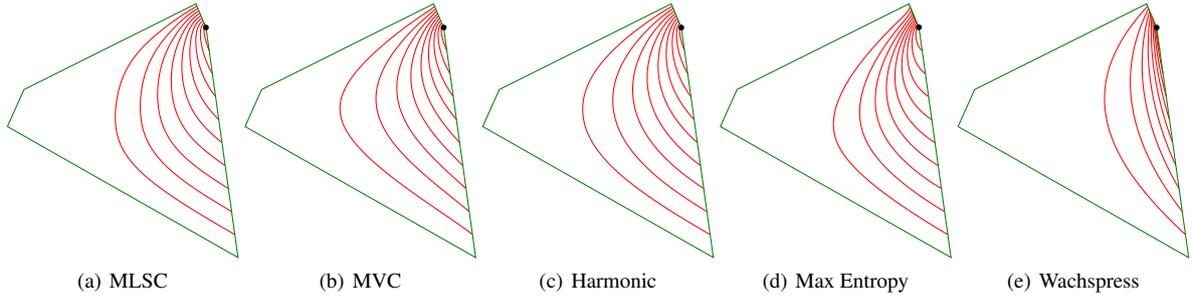


Figure 1: Comparison of basis functions over a convex polygon. Notice that (a), (b), and (c) look similar, while (d) pulls away from the top boundary and (e) has a very steep slope.

If we also represent $F_i^\perp(t)$ in the Bernstein basis such that $F_i^\perp(t) = \sum_{j=0}^{\ell} \beta_{j,\ell}(t) f_{i,j}^\perp$, then value of the interpolant reduces to

$$\hat{F}(x) = V_m(x)C = \sum_{i=1}^n \left[\sum_{j=1}^k B_{i,j}(x) f_{i,j} + \sum_{j=1}^{\ell} D_{i,j}(x) f_{i,j}^\perp \right]$$

where $f_{i,j}^\perp$ represents the j^{th} Bernstein control point for the cross-boundary derivative along the i^{th} edge.

The functions $B_{i,j}(x), D_{i,j}(x)$ are generalized barycentric basis functions and satisfy a modified set of properties from Section 1 for barycentric functions. For example, $B_{i,j}(x)$ satisfies the partition of unity property

$$\sum_{i=1}^n \sum_{j=1}^k B_{i,j}(x) = 1$$

but $D_{i,j}(x)$ does not. Together, $B_{i,j}(x)$ and $D_{i,j}(x)$ satisfy linear precision. That is, for any linear function $L(x)$, there exists coefficients $f_{i,j}$ and $f_{i,j}^\perp$ such that

$$L(x) = \sum_{i=1}^n \left[\sum_{j=1}^k B_{i,j}(x) f_{i,j} + \sum_{j=1}^{\ell} D_{i,j}(x) f_{i,j}^\perp \right].$$

In contrast to traditional barycentric coordinates, the derivative basis functions, $D_{i,j}(x)$, are required in addition to $B_{i,j}(x)$ for linear precision.

The final property, which is interpolation, is more subtle. Certainly $\hat{F}(P_i(t)) = F_i(t)$ because the least squares problem did not add any new constraints into the optimization for values along the boundary. The derivative, however, may not always be interpolated.

First, it is clear that the degree m of the function fit must be greater than or equal to the order of the derivative being fit along the boundary. If that were not the case, then the derivative function $G_m(t)$ would be identically zero and derivatives would be removed from the optimization.

Second, whether or not the derivative is interpolated de-

pends on the weight functions $W_i(x,t)$ and how quickly they approach infinity along the boundary. For $\alpha = 1$, $\hat{F}(x)$ interpolates $F_i(t)$, but not $F_i^\perp(t)$. However, when $\alpha \geq 2$, the derivatives of $\hat{F}(x)$ match that of $F_i^\perp(t)$ along the boundary. While we have no algebraic proof of this statement, we have verified this behavior numerically on many examples.

To reproduce a smooth function with the construction, $F_i(t)$ must be smooth. The data on the boundary must also be specified consistently. For $F_i(t)$, this means that intersections of boundary lines, such as at vertices, must share the same value. For $F_i^\perp(t)$, consistency means that no point on the boundary can have more than one tangent plane. At vertices p_i , the two boundary curves $F_{i-1}(t)$ and $F_i(t)$ define a unique tangent plane at p_i as long as the lines $P_{i-1}(t)$ and $P_i(t)$ are not collinear. Therefore, $F_i^\perp(t)$ cannot be specified independently of the derivatives of $F_{i-1}(t)$ and $F_{i+1}(t)$ at its end-points.

4. Properties

We have already discussed some properties of these barycentric coordinates such as interpolation of values and derivatives along the boundary of the region. These barycentric coordinates have a number of additional, interesting properties that we elaborate on below.

Arbitrary Precision

Our barycentric coordinates allow us to control the precision of the polynomials we can reproduce. For all polynomial functions $H(x)$ of total degree m , there exists coefficients $f_{i,j}$ for $j = 0 \dots m$ (and likewise $f_{i,j}^\perp$ if derivatives are specified) such that $\hat{F}(x) = H(x)$. Furthermore, $f_{i,j}$ (and $f_{i,j}^\perp$) are given by blossoming such that $F_i(t) = H(P_i(t))$. The reason this statement is true is that we fit a function of total degree m with the basis $V_m(x)$. Regardless of the weights $W_i(x,t)$, the fitting error will be zero if the boundary data is compatible with sampling from $H(x)$. Since the error in Equations 1 and 2 must be positive, $V_m(x)C = H(x)$ will be a global minimum with an error of zero in the minimization. Furthermore,

this precision is independent of whether or not $P_i(t)$ forms a closed curve or consists of disconnected curves in \mathbb{R}^2 . We know of no other barycentric coordinate construction that has this property.

We can compare our coordinates with the Hermite MVC construction from Floater et al. [FS08] as well. The authors show that their hermite interpolant has cubic precision but requires a numerical integral to evaluate. Likewise, our method (with or without derivative information specified) will have cubic precision as long as $m = 3$.

Interpolation

Our method solves a least squares problem for every point x within the boundary. For $\alpha > 0$ the weight function $W_i(x,t)$ is finite everywhere except at the boundary point $x = P(t)$ where the value is infinite. This means that the weight of the boundary point dominates the contribution from all other points and the least squares problem must produce a function such that $\hat{F}(x)$ interpolates the boundary.

Unfortunately, this argument does not hold for interpolation of derivatives. The cross-boundary derivative of $\hat{F}(x)$ for a point on the boundary depends on at least one other point that is an infinitesimal distance away from the boundary. The derivative therefore depends on a point that has finite weight contribution from all points on the boundary. Whether the derivative is interpolated therefore depends on how quickly the weight function approaches infinity at the boundary.

Smoothness

For points x that are not on the boundary, the weight function $W_i(x,t)$ changes smoothly in both x and t , and is, in fact, C^∞ . This means that the integral of $W_i(x,t)$ with respect t maintains continuity in x and that the basis functions are C^∞ over the interior of the domain.

Closed-Form

As long as the boundary curves $P_i(t)$ are linear, our coordinates have a closed-form solution regardless of the order polynomial basis $V_m(x)$ we solve for, the order of the boundary values $F_i(t)$, and the order of the derivatives $F_i^\perp(t)$. First note that all integrals are over rational functions. Also, although the term $\|P_i'(t)\|$ implicitly includes a square root, this value is constant for linear $P_i(t)$. Notice also that, once constants are factored out, the denominator of the rational functions is created solely by the weight function $W_i(x,t)$ and has a special form. Since the denominator is the squared magnitude of a linear function, the denominator is some quadratic $Q = a + bt + ct^2$ raised to the power α .

Because of the additive property of integrals, each of the summands in the numerator can be integrated separately, so

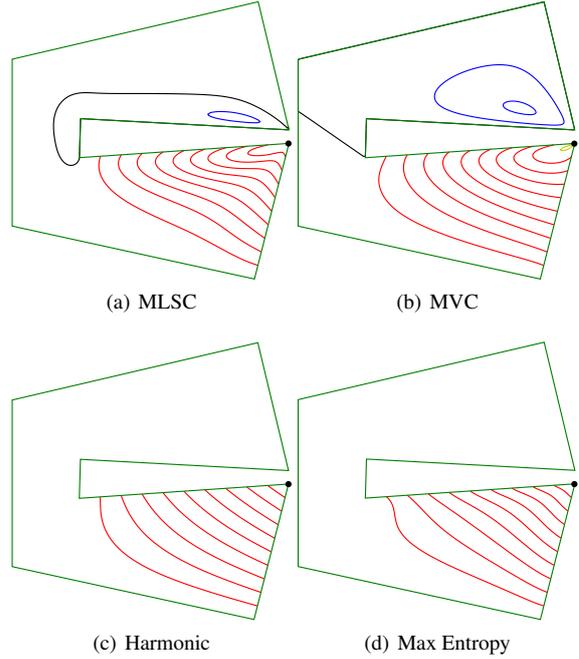


Figure 2: Comparison of basis functions over a concave, U-shaped polygon. MLSC uses $\alpha = 2$. Notice that (b) has a large negative region and values larger than 1, while (c) and (d) are always positive.

that it is sufficient to consider integrals only of the form $\int \frac{t^i dt}{Q^{j+1}}$. Although i and j can be arbitrary positive numbers, we can use recurrence relationships for integrals of this form to reduce the degree of the numerator and the denominator [Pei29]. To reduce the power of the numerator, we apply the relationship

$$\int \frac{t^i dt}{Q^{j+1}} = -\frac{t^{i-1}}{(2j-i+1)cQ^j} - \frac{b(j-i+1)}{c(2j-i+1)} \int \frac{t^{i-1} dt}{Q^{j+1}} + \frac{a(i-1)}{c(2j-i+1)} \int \frac{t^{i-2} dt}{Q^{j+1}}.$$

Once the numerators are all either constant or linear, the order of the denominator can be reduced through the recurrences

$$\int \frac{t dt}{Q^{j+1}} = -\frac{2a+bt}{j(4ac-b^2)Q^j} - \frac{b(2j-1)}{j(4ac-b^2)} \int \frac{dt}{Q^j}$$

$$\int \frac{dt}{Q^{j+1}} = \frac{2ct+b}{j(4ac-b^2)Q^j} + \frac{2c(2j-1)}{j(4ac-b^2)} \int \frac{dt}{Q^j}.$$

It is then easy to verify that the base cases of the integrals have closed-forms. Matrix inversion also has a closed-form solution, so the entire equation for calculating coordinates has a closed-form.

5. Results

Several types of barycentric coordinates have been described in the last several years, and we will compare our results with the most prominent types. For these comparisons, we fit linear polynomials ($m = 1$) and set $\alpha = 2$. First, we compare basis functions in a convex polygon with obtuse angles and short sides in Figure 1. The vertex associated with the basis function is shown as a black dot. In the image, we show contour lines for the basis functions at increments of $\frac{1}{10}$. We draw the zero contour in black, contours in between 0 and 1 in red and negative valued contours in blue.

Among the types of coordinates shown, Harmonic coordinates produce the most visually pleasing result, and both MLSC and MVC have similar looking basis functions to Harmonic coordinates. Maximum Entropy coordinates exhibit high curvature contours along the short edge. Wachspress coordinates clearly produce undesirable contours in this example, because the function value changes very quickly near the vertex of the basis function. In fact, as the angle approaches 180 degrees at a vertex, the derivatives of Wachspress coordinates are unbounded.

We also compare methods that are defined over concave polygons in Figure 2. Polygons that have points in the domain with small Euclidean distance but large geodesic distance often prove problematic for barycentric coordinates. Again, Harmonic coordinates are typically superior in this situation because they depend only on geodesic distance and minimize curvature, though at a high computational cost. MEC now compare favorably to MLSC and MVC, because they are guaranteed to be positive. This figure also illustrates a tradeoff between MLSC and MVC. MVC produce basis functions with lower curvature and more regular shapes, but can have larger negative regions and larger maximum and minimum values. For example, MVC have values greater than 1 in this figure (the contour of value 1 is drawn in yellow). Negative values are undesirable, because interpolated values may extend beyond the range of the boundary values in regions where the basis functions are negative.

In Figure 3 we show an example of the basis functions that result from derivative constraints. In this example, function values are linearly interpolated along line segments, but cross boundary derivatives use a quadratic Bezier basis. This quadratic basis leaves one degree of freedom for manipulating derivatives, because the values at the ends of the line segments are constrained to match the tangent plane defined by the function values at the corners. In this figure, the top images show the bases corresponding to a single point without derivative constraints (left) and with derivative constraints (right) at increments of $\frac{1}{10}$.

When we add derivative constraints, the basis functions corresponding to boundary values must create cross-boundary derivatives with zero magnitude. As a consequence, all contour lines are perpendicular to the boundary. We have also observed that $B_{i,j}(x)$ tend to be positive when

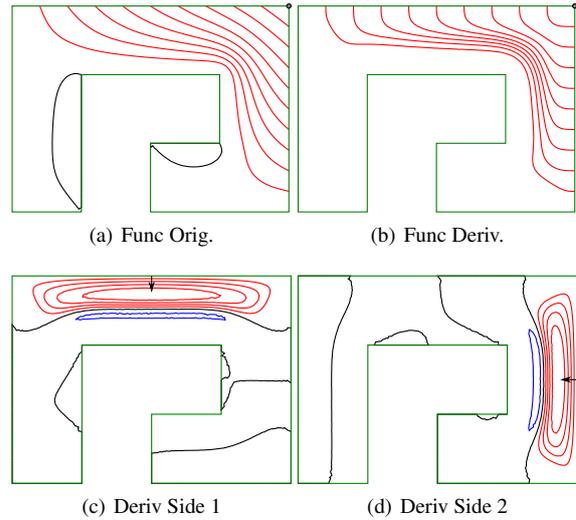


Figure 3: Basis functions of our method with derivatives over a concave polygon with $\alpha = 2$. Above: the basis function of a vertex with and without constrained derivatives. Below: the basis functions of the middle control point of the derivatives.

derivatives are specified. In fact, we have found no examples in which negative values occur no matter how convoluted the shape of $P_i(t)$ is. This is in contrast to MVC or even our own coordinates without derivative constraints, both of which routinely have negative regions in the basis functions when applied to non-convex shapes. Unfortunately, we have not yet found a proof showing that $B_{i,j}(x)$ is always greater than zero in this case.

The bottom images show the basis functions associated with the derivative control point at the center of the edge in increments of $\frac{1}{200}$. Note that, in contrast to the basis functions associated with the vertices, these derivative basis functions can and will be negative, though only very slightly so. In this example, $D_{i,1}(x) > \frac{-1}{100}$.

Our construction also defines many members of a family of coordinates that are all well-defined and have closed-form solutions. Figure 4 shows six members of this family corresponding to different values of α and using different degree polynomials m in the moving least squares optimization. In Figure 5 (top) we also show interpolation of boundary values with $m = 1, 2$ where the boundary data is given by quadratic curves. Using linear polynomials ($m = 1$) typically produces basis functions with the smallest oscillations. While all of these coordinates provide linear precision, higher values of m provide the ability to reproduce functions up to degree m assuming that the $F_i(t)$ are also of degree m . When boundary derivatives are specified, the utility of fitting higher order polynomials becomes clear. If, for example, all of the

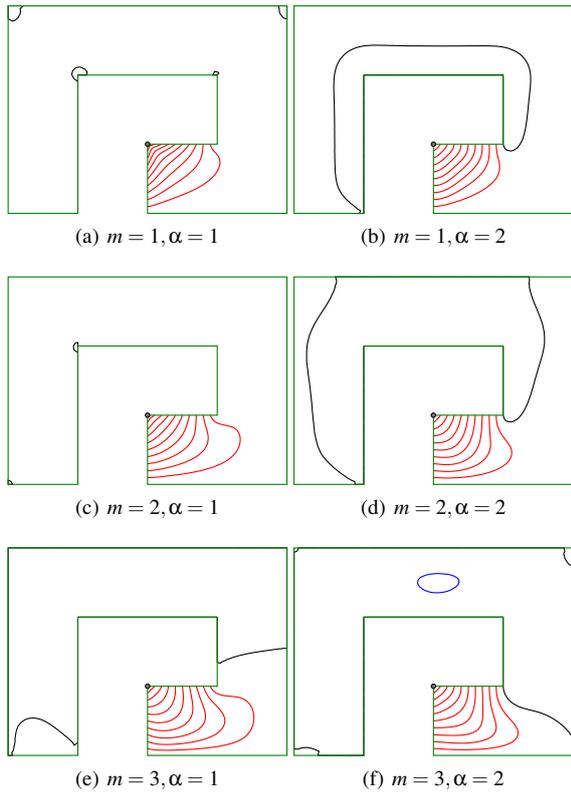


Figure 4: Several example basis functions are shown with linear precision (top), quadratic precision (middle), and cubic precision (bottom). For each precision, we also show the effect of varying α by using $\alpha = 1$ on the left, and $\alpha = 2$ on the right.

boundary derivatives are specified to be negative as is shown in Figure 5 (bottom), one expects a bowl shape like the one admitted by quadratic functions. Figure 6 shows the same example with $m = 2$ except that we modify the derivatives at the center of each edge to be different values.

Like most other types of barycentric coordinates, our coordinates extend to 3D polytopes. While in 2D we fit a polynomial basis $V_m(x)$ to the values on the boundary of a curve, in 3D we fit a polynomial basis to the values of the boundary of a surface. For simplicity, we assume that the surface is closed and has triangular faces. The coefficients C of the basis can then be found by minimizing the following expression.

$$\operatorname{argmin}_C \sum_{i=1}^n \int_0^1 \int_0^t W_i(x, s, t) (V_m(P_i(s, t))C - F_i(s, t))^2 ds dt$$

Although we do not know of any closed-form for this surface integral, it is possible to evaluate the integral numerically. Note that we do have a closed-form for the interior

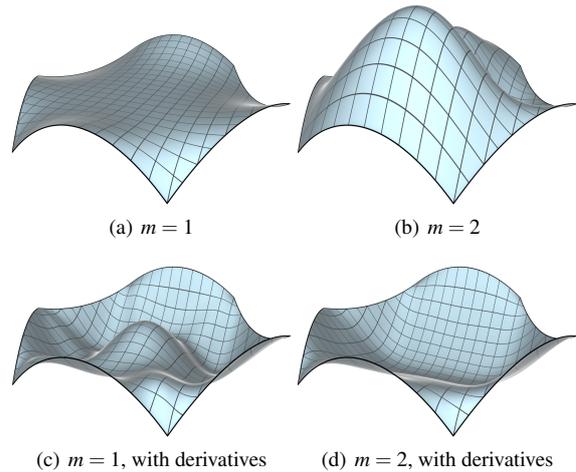


Figure 5: A 3D graph of interpolation ($\alpha = 2$) over a hexagonal boundary with $m = 1$ (left) and $m = 2$ (right) using quadratic boundary curves. The top row performs interpolation without derivatives while the bottom row uses quadratic derivative curves with negative derivatives along the center of each edge.

integral over s , which can be used to accelerate/improve the numerical integral over the surface.

One application of these 3D coordinates is to calculate deformations of an object. If the object is enclosed in a bounding polytope with vertices p_i , then every position x within the polytope can be represented by its barycentric coordinates as $x = \sum_i b_i(x)p_i$. Deformed positions \hat{x} can then be calculated for a modified polytope with vertices \hat{p}_i from the previously calculated coordinates as $\hat{x} = \sum_i b_i(x)\hat{p}_i$. Figure 7 shows an example of deformations of a 3D model performed with this method.

6. Conclusions

In this paper we have presented a barycentric coordinate basis that we believe has several useful properties. Our construction provides an entire family of barycentric coordinates with closed-form solutions that are well-defined for arbitrary polygons (even disconnected curves), reproduce functions to a specified degree m and can interpolate derivative information for some values of α .

The current limitations of these coordinates are that they are not guaranteed to be positive and we lack proofs for some observed properties such as interpolation of derivatives and for positivity of the vertex basis functions when derivatives are interpolated. Extending our construction to interpolate higher order derivatives also follows in a straight-forward manner from Section 3. However, it is unclear what restric-

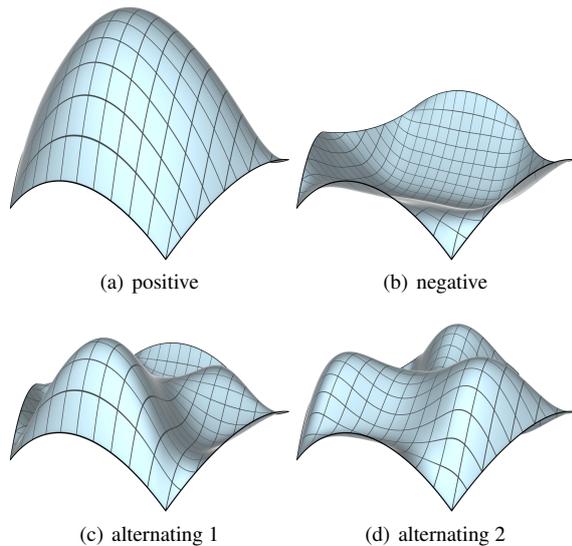


Figure 6: A hexagon with quadratic function values and derivatives specified along the edges. The derivatives at the end-points are constrained by the function values, but we modify the derivative in the center of each edge. We constrain the derivative in the center of the edge to be either all positive, negative, or alternating in sign.

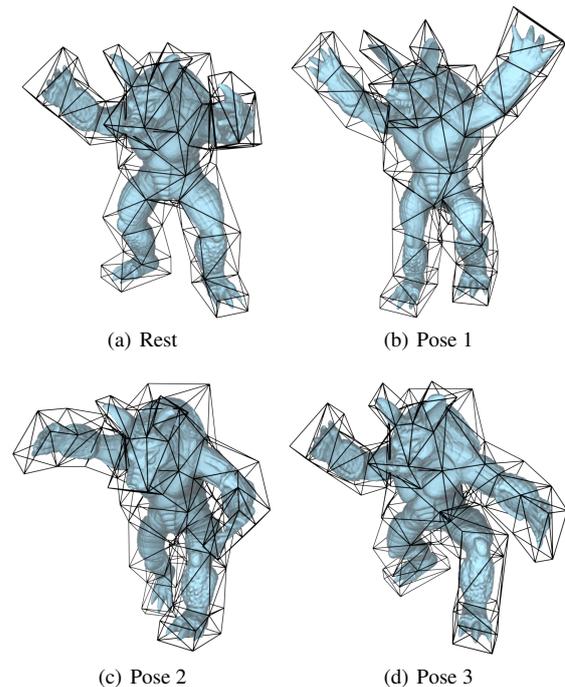


Figure 7: Example of 3D deformation using our coordinates. The rest position is shown on the top left, along with some example deformations.

tions on α must hold in order to guarantee interpolation and we would like to explore this idea in the future.

References

- [Bel06] BELYAEV A.: On transfinite barycentric coordinates. In *SGP* (2006), pp. 89–99. [2](#)
- [DF09] DYKEN C., FLOATER M. S.: Transfinite mean value interpolation. *CAGD* 26, 1 (2009), 117–134. [2](#)
- [DMA02] DESBRUN M., MEYER M., ALLIEZ P.: Intrinsic parameterizations of surface meshes. *CGF* 21, 3 (2002), 209–218. [1](#)
- [FHK06] FLOATER M., HORMANN K., KÓS G.: A general construction of barycentric coordinates over convex polygons. *Advances in Computational Mathematics* 24, 1 (2006), 311–331. [2](#)
- [Flo03] FLOATER M. S.: Mean value coordinates. *CAGD* 20 (2003), 2003. [2](#)
- [FS08] FLOATER M. S., SCHULZ C.: Pointwise radial minimization: Hermite interpolation on arbitrary domains. *CGF* 27, 5 (2008), 1505–1512. [2, 5](#)
- [HF06] HORMANN K., FLOATER M. S.: Mean value coordinates for arbitrary planar polygons. *ACM TOG* 25, 4 (2006), 1424–1441. [1, 2](#)
- [HS08] HORMANN K., SUKUMAR N.: Maximum entropy coordinates for arbitrary polytopes. *CGF (Proceedings of SGP)* 27, 5 (2008), 1513–1520. [2](#)
- [JMD*07] JOSHI P., MEYER M., DEROSE T., GREEN B., SANOCKI T.: Harmonic coordinates for character articulation. *ACM TOG* 26, 3 (2007), 71. [2](#)
- [JSW05] JU T., SCHAEFER S., WARREN J.: Mean value coordinates for closed triangular meshes. In *ACM SIGGRAPH* (2005), pp. 561–566. [1](#)
- [LD89] LOOP C. T., DEROSE T. D.: A multisided generalization of bézier surfaces. *ACM TOG* 8, 3 (1989), 204–234. [1](#)
- [LKC07] LIPMAN Y., KOPF J., COHEN-OR D., LEVIN D.: Gpu-assisted positive mean value coordinates for mesh deformations. In *SGP* (2007), pp. 117–124. [2](#)
- [LS08] LANGER T., SEIDEL H.-P.: Higher order barycentric coordinates. *CGF* 27, 2 (2008), 459–466. [2](#)
- [Möb27] MÖBIUS A. F.: *Der Barycentriche Calcul.* J. Barth, Leipzig, 1827. [1](#)
- [Pei29] PEIRCE B.: *A Short Table of Integrals.* Ginn and Company, MA, USA, 1929. [5](#)
- [SJW07] SCHAEFER S., JU T., WARREN J.: A unified, integral construction for coordinates over closed curves. *CAGD* 24, 8-9 (2007), 481–493. [2](#)
- [SMW06] SCHAEFER S., MCPHAIL T., WARREN J.: Image deformation using moving least squares. *ACM TOG* 25, 3 (2006), 533–540. [2, 3](#)
- [Wac75] WACHSPRESS E. L.: *A rational finite element basis.* Mathematics in Science and Engineering. Elsevier, 1975. [1](#)
- [Wen01] WENDLAND H.: Local polynomial reproduction and moving least squares approximation. *IMA Journal of Numerical Analysis* 21, 1 (2001), 285–300. [2](#)
- [WSD07] WARREN J. D., SCHAEFER S., HIRANI A. N., DESBRUN M.: Barycentric coordinates for convex sets. *Advances in Computational Mathematics* 27, 3 (2007), 319–338. [2](#)