Generalized Barycentric Coordinates on Irregular Polygons

Mark Meyer† Haeyoung Lee‡ Alan Barr† Mathieu Desbrun†‡

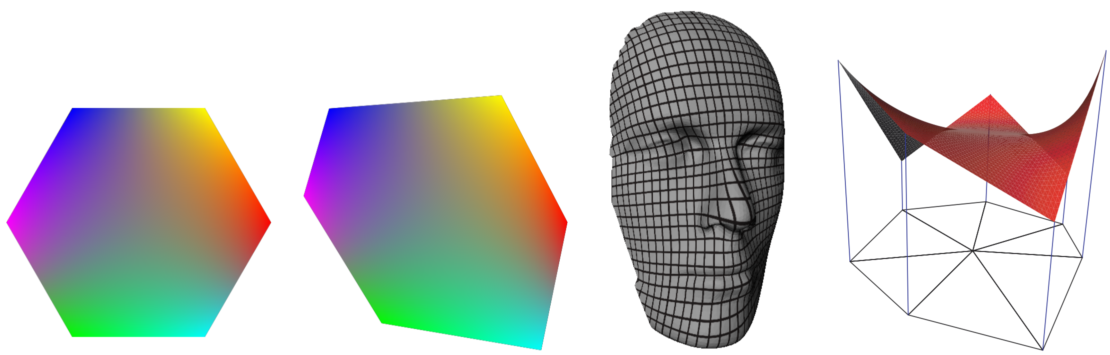
†Caltech - ‡USC

Abstract

In this paper we present an easy computation of a generalized form of barycentric coordinates for irregular, convex n-sided polygons. Triangular barycentric coordinates have had many classical applications in computer graphics, from texture mapping to ray-tracing. Our new equations preserve many of the familiar properties of the triangular barycentric coordinates with an equally simple calculation, contrary to previous formulations. We illustrate the properties and behavior of these new generalized barycentric coordinates through several example applications.

1 Introduction

The classical equations to compute triangular barycentric coordinates have been known by mathematicians for centuries. These equations have been heavily used by the earliest computer graphics researchers and have allowed many useful applications including function interpolation, surface smoothing, simulation and ray intersection tests. Due to their linear accuracy, barycentric coordinates can also be found extensively in the finite element literature [Wac75].



(a) (b) (c) (d)

Figure 1: *(a) Smooth color blending using barycentric coordinates for regular polygons [LD89], (b) Smooth color blending using our generalization to arbitrary polygons, (c) Smooth parameterization of an arbitrary mesh using our new formula which ensures non-negative coefficients. (d) Smooth position interpolation over an arbitrary convex polygon (S-patch of depth 1).*

Despite the potential benefits, however, it has not been obvious how to generalize barycentric coordinates from triangles to n-sided polygons. Several formulations have been proposed, but most had their own weaknesses. Important properties were lost from the triangular barycentric formulation, which interfered with uses of the previous generalized forms [PP93, Flo97]. In other cases, the formulation applied only to regular polygons [LD89]. However, Wachspress [Wac75] described an appropriate extension, unfortunately not very well known in Graphics[1]. We will review these techniques, and present a much simpler formulation for generalized barycentric coordinates of convex irregular n-gons.

We define the notion of generalized barycentric coordinates in the remainder of this paper as follows: let $\mathbf{q}_1, \mathbf{q}_2, ..., \mathbf{q}_n$ be n vertices of a *convex* planar polygon Q in $\mathbb{R}^2$, with $n \geq 3$. Further, let $\mathbf{p}$ be an *arbitrary* point inside Q. We call *generalized barycentric coordinates* of $\mathbf{p}$ with respect to $\{\mathbf{q}_j\}_{j=1..n}$ any set of real coefficients $(\alpha_1, \alpha_2, ..., \alpha_n)$ depending on the vertices of Q and on $\mathbf{p}$ such that all the following properties hold:

- **Property I** (Affine Combination)

$$\mathbf{p} = \sum_{j\in[1..n]} \alpha_j \, \mathbf{q}_j \ , \quad \text{with} \sum_{j\in[1..n]} \alpha_j = 1. \tag{1}$$

  This property allows us to use the polygon's vertices as a basis to locate any point inside. This partition of unity of the coordinates also makes the formulation both rotation and translation invariant.

- **Property II** (Smoothness): The $\{\alpha_j\}_{j=1..n}$ must be infinitely differentiable with respect to $\mathbf{p}$ and the vertices of Q. This ensures smoothness in the variation of the coefficients α_j when we move any vertex $\mathbf{q}_j$.

- **Property III** (Convex Combination):

$$\alpha_j \ \geq \ 0 \quad \forall j \in [1..n].$$

  Such a convex combination guarantees no under- or over-shooting in the coordinates: all the coordinates will be between zero and one.

[1] This approach has also been generalized for convex polytopes by Warren [War96]

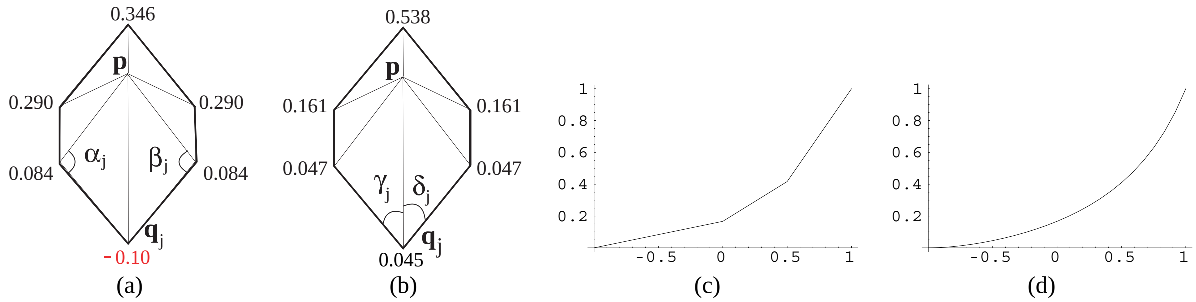


Figure 2: *While (a) Polthier's formula [PP93] can give negative coefficients for the barycentric coordinates, our generalization (b) has guaranteed non-negativity. (c) and (d) show a plot of a single barycentric coefficient, generated by linearly varying* **p***, for (c) Floater's method and (d) our new method. Note the sharp derivative discontinuities apparent when Floater's method crosses a triangle boundary, while our method produces smoothly varying weights.*

The usual triangular barycentric coordinates ($n = 3$) obviously satisfy the aforementioned properties. Note also that Equation (1) can be rewritten in the following general way:

$$\sum_{j \in [1..n]} \omega_j (\mathbf{q}_j - \mathbf{p}) = 0. \tag{2}$$

A simple normalization allows us to find partition-of-unity barycentric coordinates:

$$\alpha_j = \omega_j / (\sum_k \omega_k). \tag{3}$$

2 Previous Work

Several researchers have attempted to generalize barycentric coordinates to arbitrary n-gons. Due to the relevance of this extension in CAD, many authors have proposed or used a generalization for *regular n-sided polygons* [LD89, Kur93, Lod93]. Their expressions nicely extend the well-known formula to find barycentric coordinates in a triangle:

$$\begin{aligned} \alpha_1 &= \mathcal{A}(\mathbf{p}, \mathbf{q}_2, \mathbf{q}_3)/\mathcal{A}(\mathbf{q}_1, \mathbf{q}_2, \mathbf{q}_3) \\ \alpha_2 &= \mathcal{A}(\mathbf{p}, \mathbf{q}_3, \mathbf{q}_1)/\mathcal{A}(\mathbf{q}_1, \mathbf{q}_2, \mathbf{q}_3) \\ \alpha_3 &= \mathcal{A}(\mathbf{p}, \mathbf{q}_1, \mathbf{q}_2)/\mathcal{A}(\mathbf{q}_1, \mathbf{q}_2, \mathbf{q}_3) \end{aligned} \tag{4}$$

where $\mathcal{A}(\mathbf{p}, \mathbf{q}, \mathbf{r})$ denotes the signed area of the triangle $(\mathbf{p}, \mathbf{q}, \mathbf{r})$. Unfortunately, none of the proposed affine combinations leads to the desired properties for irregular polygons. However, Loop and DeRose [LD89] note in their conclusion that barycentric coordinates defined over arbitrary convex polygons would open many extensions to their work.

Pinkall and Polthier [PP93], and later Eck *et al* [EDD$^+$95], presented a conformal parameterization for a triangulated surface by solving a system of linear equations relating the positions of each point $\mathbf{p}$ to the positions of its first ring of neighbors $\{\mathbf{q}_j\}_{j=1..n}$ as:

$$\sum_j (cot(\alpha_j) + cot(\beta_j))(\mathbf{q}_j - \mathbf{p}) = 0 \tag{5}$$

where the angles are defined in Figure 2(a). As Desbrun *et al.* showed in [DMSB99], this formula expresses the gradient of area of the 1-ring with respect to $\mathbf{p}$, therefore Property II is immediately satisfied. The only problem is that the weights can be negative even when the boundary of the n-sided polygon is convex (as indicated in Figure 2(a) by the red colored weights), violating Property III.

Floater [Flo97, Flo98] also attempted to solve the problem of creating a parameterization for a surface by solving linear equations. He defined the barycentric coefficients algorithmically to ensure Property III [Flo97]. Additionally, most of the other properties are also enforced by construction; alas, due to the algorithmic formulation used, Property II does not hold, as proven by a cross-section in Figure 2(c). These barycentric coefficients are only C^0 as the point $\{\mathbf{p}\}$ is moved within the polygon.

However, in 1975, Wachpress proposed a construction of rational basis functions over polygons that leads to the appropriate properties. For the non-normalized weight ω_j (see Equation 2) corresponding to the point $\mathbf{q}_j$ of Figure 3(a), Wachspress proposed [Wac75] to use a construction leading to the following formulation:

$$\omega_j = \mathcal{A}(\mathbf{q}_{j-1}, \mathbf{q}_j, \mathbf{q}_{j+1}) \;.\; \Pi_{k \notin \{j,j+1\}} \mathcal{A}(\mathbf{q}_{k-1}, \mathbf{q}_k, \mathbf{p}) \tag{6}$$

Thus, each weight ω_j is the product of the area of the jth "boundary" triangle formed by the polygon's three adjacent vertices (shaded in Figure 3), and the areas of the $n-2$ interior triangles formed by the point $\mathbf{p}$ and the polygon's adjacent vertices (making sure to exclude the two interior triangles that contain the vertex $\mathbf{q}_j$). The barycentric weights, α_j, are then computed using equation 3. We refer the reader to the appendix for a very short proof of Property I. Properties II and III obviously stand since we use positive areas for convex polygons, continuous in all the points. Notice also that, in addition to the *pseudoaffine property* defined in [LD89], this formulation also enforces *edge-preservation*: when $\mathbf{p}$ is on a border edge $\mathbf{q}_i \mathbf{q}_{i+1}$ of the polygon, these barycentric coordinates reduce to the usual linear interpolation between two points.

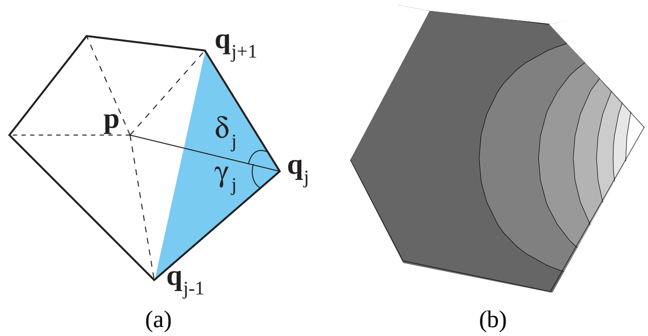


Figure 3: *(a) Our expression for each generalized barycentric coordinates can be computed locally using the edge* $\mathbf{pq}_j$ *and its two adjacent angles. (b) For an arbitrary convex polygon, the basis function of each vertex is smooth as indicated by the isocontours.*

3 Simple Computation of Generalized Barycentric Coordinates

In this section we provide the simplest form of generalized barycentric coordinates for irregular N-sided polygons that retains Property I (affine combination), Property II (smoothness) and Property III (convex combinations) of the triangular barycentric coordinates. This new barycentric formulation is *simple, local and easy to implement*. In addition, our formulation reduces to the classical equations when $n = 3$, and is equivalent to the analytic form of [Wac75], yet much simpler to compute.

If **p** is **strictly** within the polygon Q, we can rewrite the non-normalized barycentric coordinates given in equation 6 as:

$$\begin{aligned}\omega_j &= \frac{\mathcal{A}(\mathbf{q}_{j-1}, \mathbf{q}_j, \mathbf{q}_{j+1})}{\mathcal{A}(\mathbf{q}_{j-1}, \mathbf{q}_j, \mathbf{p})\mathcal{A}(\mathbf{q}_j, \mathbf{q}_{j+1}, \mathbf{p})} \\ &= [sin(\gamma_j + \delta_j)||\mathbf{q}_j - \mathbf{q}_{j-1}||\ ||\mathbf{q}_j - \mathbf{q}_{j+1}||]\ / \\ &\quad [sin(\gamma_j)||\mathbf{q}_{j-1} - \mathbf{q}_j||\ ||\mathbf{q}_j - \mathbf{p}||^2 sin(\delta_j)||\mathbf{q}_j - \mathbf{q}_{j+1}||] \\ &= \frac{sin(\gamma_j + \delta_j)}{sin(\gamma_j)sin(\delta_j)||\mathbf{p} - \mathbf{q}_j||^2}\end{aligned}$$

Therefore, using trigonometric identities for the sine function, we obtain the condensed, local formula:

$$\boxed{\omega_j = \frac{cot(\gamma_j) + cot(\delta_j)}{||\mathbf{p} - \mathbf{q}_j||^2}} \tag{7}$$

This formulation has the main advantage of being *local*: only the edge $\mathbf{pq}_j$ and its two adjacent angles γ_j and δ_j are needed. The cotangent should however not be computed through a trigonometric function call, for obvious accuracy reasons. It is far better to use a division between the dot product and the cross product of the triangle involved. Still, compared to the original Equation 6, we obtain locality, hence simplicity of computation. A simple normalization step to compute the real barycentric coordinates $\{\alpha_j\}_{j=1..n}$ using Equation 3 is the last numerical operation needed. The pseudocode in Figure 4 demonstrates the simplicity of our barycentric formulation when the point **p** is strictly within the polygon Q.

```
// Compute the barycentric weights for a point p in an n-gon Q
// Assumes p is strictly within Q and the vertices q_j of Q are ordered.
computeBarycentric(vector2d p, polygon Q, int n, real w[ ])
    weightSum = 0
    foreach vertex q_j of Q:
        prev = (j + n − 1) mod n
        next = (j + 1) mod n
        w_j = (cotangent(p,q_j,q_prev) + cotangent(p,q_j,q_next))/||p − q_j||^2
        weightSum += w_j
    // Normalize the weights
    foreach weight w_j:
        w_j /= weightSum

// Compute the cotangent of the non-degenerate triangle abc at vertex b
cotangent(vector2d a, vector2d b, vector2d c)
    vector2d ba = a - b
    vector2d bc = c - b
    return ( bc·ba / ||bc×ba|| )
```

Figure 4: *Pseudocode to compute the barycentric weights.*

Note that the above formulation is only valid when **p** is **strictly** within the polygon Q. We remedy this problem, as well as avoid numerical problems (such as divisions by extremely small numbers) through a special case. If the point **p** is within ϵ of any of the boundary segments (determined, for instance, by $\|(\mathbf{q}_{j+1} - \mathbf{q}_j) \times (\mathbf{p} - \mathbf{q}_j)\| \le \epsilon\|\mathbf{q}_{j+1} - \mathbf{q}_j\|$) the weights can be computed using a simple linear interpolation between the two neighboring boundary points (or even using the non-local equation 6).

4 Applications

As mentioned in the abstract, the use of a barycentric coordinate system is extremely useful for a wide range of applications. Since our new formulation easily extends this notion to arbitrary polygons, many domains can benefit from such a simple formula. We describe three very different example applications.

4.1 Interpolation Over N-sided Polygons

The obvious first application is to use our formula directly for interpolation of any scalar or vector field over polygons. In Figure 1, we demonstrate the smoothness for various six-sided polygons. While the regular case matches with previous formulations [LD89, Kur93], the extension to irregular polygons provides an easy way to still guarantee smoothness and non-negative coefficients.

As mentioned in the introduction, many previous formulations had various shortcomings. In Figure 2, we notice that the Polthier expression [PP93] leads to negative coefficients, while the Floater formulation [Flo97] only provides $\mathcal{C}^0$ continuity.

4.2 Parameterization

Parameterization of triangular meshes has recently been studied extensively, focusing on texturing or remeshing of irregular meshes. This consists of defining a piecewise smooth mapping between the triangulated surface and a 2D parameter plane (u, v). Floater [Flo97, Flo98] noticed that a necessary condition to define this mapping is that every vertex of the surface is mapped to a linear combination of its neighbors' mapped positions. It turns out that our formulation provides a new and appropriate way to satisfy this condition. Indeed, we can now compute directly on the surface the area-weighted barycentric coordinates of every interior point of a mesh with respect to its 1-ring neighbors, *simply by using Equation 7*. Assuming that no triangles are degenerate, we will obtain a linear combination for every vertex on the surface with respect to its 1-ring neighbors. Solving the resulting linear system of equations (using a conjugate gradient solver to exploit sparsity) will provide a quick and safe way to obtain a parameterization of an arbitrary mesh [DMA02]. Figure 1(c) demonstrates this technique: starting from an irregular face mesh, we can smoothly map a grid texture on it at low computational cost and in a robust way.

4.3 Surface Modeling and CAD

In [LD89], Loop and DeRose proposed a generalization of Bézier surfaces to regular n-sided polygons, defining what they call S-patches. Relying on a generalization of barycentric coordinates, they unify triangular and tensor product Bézier surfaces. However, they were limited to *regular polygons* with their formulation, which added hard constraints to the modeling process. Similarly, other modeling techniques (see for instance [VMT98]) use generalized barycentric coordinates, but are again constrained to regular base polygon meshes.

The new formulation we describe provides a very general way to extend the notion of Bézier surfaces. Any convex polygon provides an adequate convex domain to compute a polynomial surface. Figure 1(d) demonstrates the smoothness of a patch of order one, defined over an irregular convex domain, using our barycentric coordinate formula.

5 Conclusion

In this paper, we have introduced a straightforward way to compute smooth, convex-preserving generalized barycentric coordinates. We believe that this simple expression allows many existing works to be extended to irregular polygons with ease. We are currently investigating various avenues, ranging from generalizing this expression to 3D polyhedra, to applying it for the smoothing of meshes.

Web Information A simple C++ code implementation is available on the web at *http://www.acm.org/jgt/papers/MeyerEtAl02*.

Acknowledgements

This work has been partially supported by the Integrated Media Systems Center, a NSF Engineering Research Center, cooperative agreement number EEC-9529152, the NSF STC for Computer Graphics and Scientific Visualization (ASC-89-20219), and the CAREER award CCR-0133983.

References

[DMA02] Mathieu Desbrun, Mark Meyer, and Pierre Alliez. Intrinsic Parameterizations of Surface Meshes. In *Eurographics '02 Proceedings*, 2002.

[DMSB99] Mathieu Desbrun, Mark Meyer, Peter Schröder, and Alan Barr. Implicit Fairing of Arbitrary Meshes using Laplacian and Curvature Flow. In *ACM SIGGRAPH'99 Proceedings*, pages 317–324, 1999.

[EDD+95] Matthias Eck, Tony DeRose, Tom Duchamp, Hugues Hoppe, Michael Lounsbery, and Werner Stuetzle. Interactive Multiresolution Surface Viewing. In *ACM Siggraph'95 Conference*, pages 91–98, August 1995.

[Flo97] Michael S. Floater. Parametrization and smooth approximation of surface triangulations. *Computer Aided Geometry Desing*, 14(3):231–250, 1997.

[Flo98] Michael S. Floater. Parametric Tilings and Scattered Data Approximation. *International Journal of Shape Modeling*, 4:165–182, 1998.

[Kur93] S. Kuriyama. Surface Generation from an Irregular Network of Parametric Curves. *Modeling in Computer Graphics, IFIP Series on Computer Graphics*, pages 256–274, 1993.

[LD89] Charles Loop and Tony DeRose. A multisided generalization of B'ezier surfaces. *ACM Transactions on Graphics*, 8:204–234, 1989.

[Lod93] S. Lodha. Filling N-sided Holes. *Modeling in Computer Graphics, IFIP Series on Computer Graphics*, pages 319–345, 1993.

[PP93] Ulrich Pinkall and Konrad Polthier. Computing Discrete Minimal Surfaces and Their Conjugates. *Experimental Mathematics*, 2:15–36, 1993.

[VMT98] Pascal Volino and Nadia Magnenat-Thalmann. The SPHERIGON: A Simple Polygon Patch for Smoothing Quickly your Polygonal Meshes. In *Computer Animation '98 Proceedings*, 1998.

[Wac75] Eugene Wachpress. A Rational Finite Element Basis. *manuscript*, 1975.

[War96] Joe Warren. Barycentric Coordinates for Convex Polytopes. *Advances in Computational Mathematics*, 6:97–108, 1996.

A Derivation of Equation 6

We present here a simple proof of Equation 6. As before, let's call $\mathbf{q}_1, ...\mathbf{q}_n$ the n vertices of a convex polygon Q, and $\mathbf{p}$ a point inside Q. If we write the triangular barycentric coordinates for the point $\mathbf{p}$ with respect to a "boundary" triangle $T = (\mathbf{q}_{j-1}, \mathbf{q}_j, \mathbf{q}_{j+1})$, we get (using Equation 4):

$$\mathcal{A}(T)\ \mathbf{p} = \mathcal{A}(\mathbf{q}_j, \mathbf{q}_{j+1}, \mathbf{p})\ \mathbf{q}_{j-1} + \mathcal{A}(\mathbf{q}_{j-1}, \mathbf{q}_j, \mathbf{p})\ \mathbf{q}_{j+1}$$
$$+(\mathcal{A}(T) - \mathcal{A}(\mathbf{q}_j, \mathbf{q}_{j+1}, \mathbf{p}) - \mathcal{A}(\mathbf{q}_{j-1}, \mathbf{q}_j, \mathbf{p}))\ \mathbf{q}_j$$

Since none of these areas can be zero when $\mathbf{p}$ is inside the polygon, we rewrite the previous equation as:

$$\frac{\mathcal{A}(T)}{\mathcal{A}(\mathbf{q}_j, \mathbf{q}_{j+1}, \mathbf{p})\mathcal{A}(\mathbf{q}_{j-1}, \mathbf{q}_j, \mathbf{p})}(\mathbf{p} - \mathbf{q}_j) =$$
$$\frac{1}{\mathcal{A}(\mathbf{q}_{j-1}, \mathbf{q}_j, \mathbf{p})}[\mathbf{q}_{j-1} - \mathbf{q}_j] + \frac{1}{\mathcal{A}(\mathbf{q}_j, \mathbf{q}_{j+1}, \mathbf{p})}[\mathbf{q}_{j+1} - \mathbf{q}_j]$$

By summing the contributions of all boundary triangles, the terms on the right hand side of the previous equation will cancel two by two, and we are left with Equation 6. ■