

Interior Distance Using Barycentric Coordinates

R. M. Rustamov¹ Y. Lipman² T. Funkhouser²

¹ Drew University
² Princeton University

Abstract

This paper introduces a framework for defining a shape-aware distance measure between any two points in the interior of a surface mesh. Our framework is based on embedding the surface mesh into a high-dimensional space in a way that best preserves boundary distances between vertices of the mesh, performing a mapping of the mesh volume into this high-dimensional space using barycentric coordinates, and defining the interior distance between any two points simply as their Euclidean distance in the embedding space. We investigate the theoretical properties of the interior distance in relation to properties of the chosen boundary distances and barycentric coordinates, and we investigate empirical properties of the interior distance using diffusion distance as the prescribed boundary distance and mean value coordinates. We prove theoretically that the interior distance is a metric, smooth, interpolating the boundary distances, and reproducing Euclidean distances, and we show empirically that it is insensitive to boundary noise and deformation and quick to compute. In case the barycentric coordinates are non-negative we also show a maximum principle exists. Finally, we use it to define a new geometric property, barycentroid of shape, and show that it captures the notion of semantic center of the shape.

Keywords: distance measurement, shape analysis

1. Introduction

Measuring intrinsic distances on and inside surface meshes is a fundamental problem in geometric processing, with applications in path planning, shape matching, surface editing, mesh segmentation, and other geometric analysis problems. Desirable properties of such a distance are that it is a metric, fast to compute, smooth, and insensitive to deformations.

While several methods with these properties have been proposed for distances *on* surface meshes (e.g., geodesic distance [SSK*05], diffusion distance [CLL*05], etc.), methods for measuring distances between points *inside* the volume bound by a surface mesh are limited. Direct extension of the surface-based methods are computationally expensive (e.g., computing the shortest path within a 3D polyhedron is NP-hard [CR87], and computing diffusion distances on a 3D lattice is expensive), and thus people have generally resorted to Euclidean distances between points in object interiors, which of course are not intrinsic for deformable shapes.

The goal of our work is to provide an efficient method for computing distances between points in the interior of a surface mesh (Figure 1). Our method takes as input a set of pair-

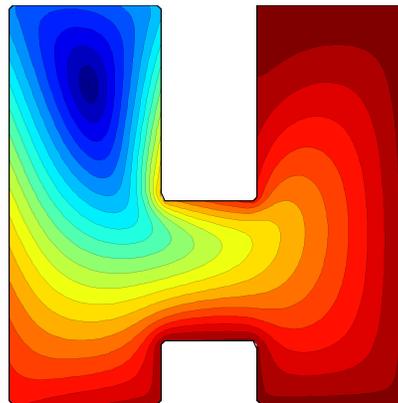


Figure 1: The interior distance measured from a single source point (black point) inside a 2D shape.

wise boundary distances between vertices of the mesh and propagates them into the interior of the mesh. The key steps are: 1) to embed the boundary mesh into a high-dimensional

space in a way that best preserves the prescribed boundary distances, 2) to define a volumetric mapping of the interior of the mesh into this high-dimensional space using barycentric coordinates, and 3) to define the interior distance between any two points simply as the Euclidean distance between their embeddings.

This procedure can be encapsulated in a simple formula: for any distance measure along the boundary, $d(v_i, v_j)$ (where v_i and v_j are mesh vertices), and any method for defining barycentric coordinates of a point in the mesh interior, $w_i(\cdot)$ (where i is a vertex index), the interior distance, $\hat{d}(p, q)$, between any two points, p and q , can be calculated via the following quadratic form:

$$\hat{d}^2(p, q) = (\vec{w}(p) - \vec{w}(q))^T A (\vec{w}(p) - \vec{w}(q)),$$

where $\vec{w}(p) = (w_1(p), \dots, w_n(p))^T$ is the vector of barycentric coordinates for point p , and A is the Gram matrix of the dot-products of the distance-preserving embeddings v_i^* of the mesh vertices v_i . That is $A_{ij} = \langle v_i^*, v_j^* \rangle$.

This method was chosen for three reasons. First, it provides a framework within which it is possible to prove desirable properties of an interior distance measure. For example, we prove that if the matrix A is conditionally positive definite, then the interior distance is a metric and interpolates the boundary distances. Second, it is easy to implement and fast to compute. For certain choices of boundary distances, this formula can be evaluated in time linear in the number of vertices of the mesh. Therefore, it is fast enough to be computed at interactive rates for large meshes, and significantly faster than any other interior distance of which we are aware. Finally, it provides a framework to define different interesting distances inside volumes which are induced naturally from different choices of boundary distances and barycentric coordinates.

The main contribution of this paper is the proposed framework for computing distance between pairs of points in the interior of a mesh (Section 3). Within this general framework, we consider the effect of different ways to embed the boundary mesh in a high-dimensional space (Section 4) and prove desirable properties of the interior distance in relation to properties of the distance along the boundary and the barycentric coordinates (Section 5). In Section 6, we present results of experiments with a particular version of the interior distance that interpolates the diffusion distance [CLL*05, GGV08] along the boundary using mean value coordinates [Flo03, JSW05]. We find that the resulting distance measure is fast to compute, insensitive to noise and deformation, and consistent with our human expectations for a distance within the objects tested. In Section 7, we use it to define a new shape property, the barycentroid of shape, which we find captures the notion of semantic center of the shape.

2. Previous work

There has been a fair amount of prior work on computing distances between points in the interior of a 3D object in computer graphics, computational geometry, robot motion planning, computer vision, medicine, and graph analysis.

The simplest and most common distance measure is the Euclidean distance. Of course, Euclidean distance is quick to compute and smooth. However, it is completely indifferent to the shape of the enclosing boundary mesh, and thus only valuable for studying extrinsic properties of a shape. In our work, we aim to find an intrinsic, shape-aware distance measure – e.g., one that does not change with isomorphic deformations of the boundary and better respects the global volumetric shape.

The second most common approach is the *inner distance* – i.e., the length of the shortest path between two points through the interior of the object. While this distance has intuitive meaning and has been found useful for shape analysis and shape matching [LJ07, LFR09], computing it is NP-hard for 3D polyhedral boundaries [CR87]. It is an instance of the Euclidean Shortest Path Problem With Polyhedral Obstacles, which has been studied extensively in computational geometry and motion planning [Pap85]. Although there are a variety of polynomial-time approximation algorithms (e.g., using visibility graphs [CSY94, BCKO08]), they are still expensive. Moreover, the inner distance has derivative discontinuities (where different shortest paths meet) and is only locally influenced by the boundary of the shape (at the points of contact between the shortest path and the boundary). Thus, smooth deformations of the boundary do not lead to smooth changes to the inner distance.

More robust, global distances can be computed using models of physical processes. For example, the *diffusion distance* between two points p and q represents the integral of a heat diffusion process [CLL*05]. Similarly, the *random walk distance* represents the expected length of random walks from p to q , and the *commute-time distance* represents the expected length of random walks from p to q and back [YFD*07]. Though these distance measures have many of the properties we desire (e.g., smoothness, non-local influence of the boundary, etc.), and boundary versions of them have been used successfully for shape analysis in computer graphics (e.g., [GGV08]), they are expensive to compute with high accuracy for volumetric interiors. Though many of these distances are defined in a continuous setting, they are typically computed approximately on a discrete graph representing the object interior (e.g., voxels, tetrahedra, or some other meshing of interior space). As a result, they suffer from approximation errors due to the discretization and/or require large storage and compute times if the discretization is very fine.

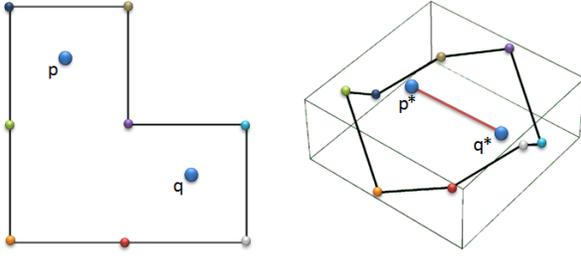


Figure 2: The method of defining the interior distance is through embedding the boundary (left) in high-dimensional space (right) s.t. the prescribed pairwise distances are preserved as much as possible. The vertices are colored to depict this embedding. The distance $\hat{d}(p, q)$ between two interior points p, q is defined as the Euclidean distance between their embeddings p^*, q^* .

3. Overview

Our goal is to define an interior distance by propagating distances provided on the boundary of a mesh into its interior. Specifically, given a set of pairwise distances $d_{ij} = d(v_i, v_j)$ on a mesh, where $v_i \in \mathbb{R}^3$, $i = 1, \dots, n$ are the vertices of the mesh, we aim to construct a bivariate interior distance function $\hat{d}(\cdot, \cdot)$ such that $\hat{d}(v_i, v_j)$ exactly/approximately equals $d(v_i, v_j)$ for every two vertices on the mesh. Furthermore, we want \hat{d} to possess properties characteristic of distance functions, e.g., being a metric in the mathematical sense.

Our approach is to propagate distances with a three-step procedure. First, we embed the vertices $v_i \in \mathbb{R}^3$ in \mathbb{R}^m (for some m) by constructing a map $v_i \mapsto v_i^* \in \mathbb{R}^m$ such that $\|v_i^* - v_j^*\|_{\mathbb{R}^m}$ exactly equals or approximates d_{ij} . Then, we extend this embedding map to the interior of the mesh via barycentric interpolation by setting $p \mapsto p^* = \sum_i w_i(p) v_i^*$, where $w_i(\cdot)$ are barycentric coordinates. Finally, we define the distance between points p and q as the *Euclidean distance* between their images:

$$\hat{d}(p, q) := \|p^* - q^*\|_{\mathbb{R}^m} \quad (1)$$

This procedure is illustrated in Figure 2: the closed polygonal line (left) is embedded into a high dimensional space (right) such that the Euclidean distances between pairs of vertices match the boundary distances on the mesh. Then, a distance between two arbitrary interior points is defined as the Euclidean distance between their images, where the image p^* of a point p is defined using the barycentric coordinates of p w.r.t. the original boundary mesh. This procedure can also be viewed as deforming the interior of the mesh into the high-dimensional space via the barycentric coordinates and measuring distances in that space.

One advantage of this approach is that it provides a formula for the interior distance expressed directly in terms of the barycentric coordinates of the points p and q and

the embedded boundary mesh vertices v_i^* , which is simple and efficient to compute. If we plug $p^* = \sum_i w_i(p) v_i^*$, and $q^* = \sum_j w_j(q) v_j^*$ in Eq. (1), we get:

$$\begin{aligned} \hat{d}^2(p, q) &:= \langle p^* - q^*, p^* - q^* \rangle_{\mathbb{R}^m} = \\ &\left\langle \sum_i (w_i(p) - w_i(q)) v_i^*, \sum_j (w_j(p) - w_j(q)) v_j^* \right\rangle_{\mathbb{R}^m} = \\ &\sum_{i,j} (w_i(p) - w_i(q)) (w_j(p) - w_j(q)) \langle v_i^*, v_j^* \rangle_{\mathbb{R}^m}. \end{aligned}$$

The last expression is a quadratic form in $w_i(p) - w_i(q)$ and can be written in matrix notation as

$$\hat{d}^2(p, q) = (\vec{w}(p) - \vec{w}(q))^T A (\vec{w}(p) - \vec{w}(q)), \quad (2)$$

where $\vec{w}(p) = (w_1(p), w_2(p), \dots, w_n(p))^T$ is the column-vector made by stacking the weights, and A is the *Gram matrix* that contains the dot-products between the embedded vertices v_i^* , that is $A_{ij} = \langle v_i^*, v_j^* \rangle_{\mathbb{R}^m}$. In other words, if Y is the $m \times n$ matrix whose i^{th} column contains the coordinates of the point v_i^* in \mathbb{R}^m , then $A = Y^T Y$.

A second advantage of this approach is that it provides a formulation in which the properties of the interior distance can be proved in terms of the matrix A and the barycentric coordinates $\vec{w}(\cdot)$. For example, it is easy to prove that $\hat{d}(p, q)$ is a metric if A is conditionally positive definite. This and other properties are proved formally in Section 5.

4. Embedding

In order to use our construction for measuring interior distances, one first has to find an embedding of boundary $v_i \mapsto v_i^* \in \mathbb{R}^m$ such that $\|v_i^* - v_j^*\|_{\mathbb{R}^m}$ exactly equals or approximates d_{ij} , the provided pairwise distances on the boundary. We look at three classes of boundary distances: 1) an exact distance-preserving embedding is available a-priori – in fact, the distance itself is defined in terms of such an embedding; 2) an explicit embedding is not available, but the distance is exactly embeddable; 3) an exact distance-preserving embedding does not exist.

Classes 1) and 2) are the most efficient for our construction, and provide an interior distance that interpolates the prescribed boundary distances. For class 3), our construction is still valid, except that the interior distance does not interpolate, but only approximates, the input boundary distances.

Class 1: The most natural boundary distances for our framework are the distances for which a distance preserving embedding is known a-priori. Diffusion distance [CLL*05] and GPS distance [Rus07] are such examples. Both of these distances are defined by embedding the mesh into a high-dimensional space. For example, to obtain the diffusion distance, one considers the diffusion map [CLL*05] that can

be defined for meshes as follows [GGV08]. Consider a discrete Laplace-Beltrami operator of the mesh, and compute its eigendecomposition $\{\lambda_k, \phi_k\}_{k=1}^n$. The diffusion map is the embedding

$$v_i \mapsto \left(e^{-\lambda_1 t} \phi_1(v_i), e^{-\lambda_2 t} \phi_2(v_i), \dots, e^{-\lambda_n t} \phi_n(v_i) \right) \in \mathbb{R}^n,$$

here $\phi_k(v_i)$ is the i^{th} entry of the eigenvector ϕ_k . Now diffusion distance is given as Euclidean distance between the image under the diffusion map,

$$d^2(v_i, v_j) = \sum_{k=1}^n e^{-2\lambda_k t} (\phi_k(v_i) - \phi_k(v_j))^2.$$

We clearly see that when using diffusion distance in our framework, the diffusion map is the distance preserving embedding that we need.

Class 2: For this class of distances the explicit embedding is not available or expensive to compute, yet our framework can be efficiently applied because we only need the Gram matrix A as seen from Eq.(2), but not the explicit embedding.

When pairwise boundary distances are given, and these distances are known to be exactly embeddable, we can use the tools from Multi-Dimensional Scaling (MDS) [BG05, CC01], to obtain the Gram matrix A . Indeed, the classical metric MDS provides a way of writing A in terms of D , the matrix of squared distances, $D_{ij} = d^2(v_i, v_j)$. This is done using the so-called “double centering matrix” J : let $\vec{1}$ be a column vector of 1’s and set

$$J = I - \frac{1}{n} \vec{1} \vec{1}^T.$$

One can show that the Gram matrix A can be written as

$$A = -\frac{1}{2} J D J. \quad (3)$$

In order for A to define an exact distance preserving embedding, A must be positive semi-definite, which allows writing $A = Y^T Y$, and the $m \times n$ matrix Y will contain the coordinates of $v_i^*, i = 1, \dots, n$ in \mathbb{R}^m . We see that this approach provides both a check of exact embeddability, and an efficient way of applying our framework via Eq.(2); also note that, alternatively, Eq.(4) can be used.

An interesting example of this class is the commute-time distance [YFD*07], for which the Gram matrix A can be applied quickly to a vector. Indeed, it can be seen that the Gram matrix is $A = L^+$, the pseudo-inverse of the Laplacian. In this case due to the sparsity of Laplacian, we can prove that the computation of our interior distance can be done via Eq.(2) in linear time in the number of mesh vertices.

Class 3: It is possible that a distance preserving boundary embedding does not exist, which is the case for e.g. geodesic distance. However, it is still possible to apply our framework by finding an approximate embedding. In this case our construction produces an interior distance that approximates the prescribed boundary distances.

The fact that an exact distance preserving embedding of the boundary into Euclidean space does not exist can be detected after computing Gram matrix A using Eq.(3). In such cases, A will not be positive semi-definite. We can obtain an approximate embedding by constructing a positive semi-definite matrix A' that is as close to A as possible in some norm. For example one such approximating matrix A' can be found by restricting A to the subspace spanned by the eigenvectors of positive eigenvalues. This allows the decomposition $A' = Y^T Y$ and we obtain a valid approximate embedding. Now one can use either Eq.(2) with A' instead of A , or the explicit embedding to compute the interior distance.

5. Theoretical properties

We will now prove the properties of the interior distance framework assuming that a distance preserving boundary embedding $v_i \mapsto v_i^*$ exists, and the weights are barycentric coordinates.

Before starting, we quickly review the properties of barycentric coordinates: *Lagrange property* requires that $w_i(v_j) = \delta_{ij}$ (the Kronecker delta); *partition of unity* requires the weights to sum up to one, $\sum_i w_i(p) = 1$ for any p in the interior; *linear precision* is that the equality $\sum_i w_i(p) v_i = p$ holds for any point p in the interior.

Independence of the embedding: The interior distance \hat{d} is uniquely determined by the input boundary distance d and the weights; it does not depend on the specific embedding used.

Proof To obtain the proof we will present an equivalent alternative formula for the interior distance that uses the input boundary distances and weights directly without any reference to the embedding. To this end, given any distance preserving boundary embedding $v_i \mapsto v_i^*$, consider the corresponding Gram matrix A . Note that from

$$d^2(v_i, v_j) = \|v_i^* - v_j^*\|^2 = \langle v_i^*, v_i^* \rangle + \langle v_j^*, v_j^* \rangle - 2\langle v_i^*, v_j^* \rangle$$

it follows that $d^2(v_i, v_j) = A_{ii} + A_{jj} - 2A_{ij}$. We will write this equality in the matrix form as follows. Introduce matrix D with $D_{ij} = d^2(v_i, v_j)$, and let $\vec{1}$ be the $n \times 1$ column vector with all entries equal to 1, and \vec{a} be the $n \times 1$ column vector that contains the diagonal of matrix A . Now we can rewrite the previous equation as $D = \vec{a} \vec{1}^T + \vec{1} \vec{a}^T - 2A$. Writing $A = \frac{1}{2}(\vec{a} \vec{1}^T + \vec{1} \vec{a}^T - D)$, and substituting this expression into Eq.(2) gives

$$\hat{d}^2(p, q) = -\frac{1}{2}(\vec{w}(p) - \vec{w}(q))^T D (\vec{w}(p) - \vec{w}(q)). \quad (4)$$

The last equation does not contain terms with \vec{a} because $\vec{1}^T(\vec{w}(p) - \vec{w}(q)) = 0$, an equality that means that the coordinates of the vector $\vec{w}(p) - \vec{w}(q)$ sum up to zero, which follows from the partition of unity. Since this formulation of the interior distance only uses the boundary distances and the

weights, we see that our interpolation scheme is independent of the particular embedding used. \square

As a by-product we get another interpretation of the defined distance. Indeed, expanding the Eq.(4) as summation we get

$$\begin{aligned} \hat{d}^2(p, q) &= \sum_{i,j} D_{ij} w_i(p) w_j(q) - \\ &- \frac{1}{2} \sum_{i,j} D_{ij} (w_i(p) w_j(p) + w_i(q) w_j(q)). \end{aligned} \quad (5)$$

Now we see that our distance is defined by the weighted average of the known pairwise distances $\sum_{i,j} D_{ij} w_i(p) w_j(q)$ subtracted by a symmetric term which forces the distance to be zero when $p = q$.

Interpolation: \hat{d} interpolates d on the boundary: $\hat{d}(v_i, v_j) = d(v_i, v_j)$.

Proof Since an exact distance preserving embedding of the boundary exists, the Lagrangian property, $w_i(v_k) = \delta_{ik}$, implies that under our embedding the vertex v_k is mapped to $\sum_i w_i(v_k) v_i^* = \sum_i \delta_{ik} v_i^* = v_k^*$, and thus, $\hat{d}(v_i, v_j) = \|v_i^* - v_j^*\| = d(v_i, v_j)$ \square

Metric: If the Gram matrix A is conditionally positive definite, then \hat{d} is a metric.

Conditionally positive definiteness of A means that for any non-zero vector \vec{x} with $\sum_i x_i = 0$ we have $\vec{x}^T D \vec{x} > 0$.

Proof We check the four properties that a metric must satisfy. Remember that $\hat{d}(p, q) = \|p^* - q^*\|$ is Euclidean distance between p^* and q^* in R^m . Now, non-negativity, $\hat{d}(p, q) \geq 0$, follows from non-negativity of Euclidean distance. Symmetry, $\hat{d}(p, q) = \|p^* - q^*\| = \|q^* - p^*\| = \hat{d}(q, p)$, follows from the symmetry of Euclidean distance. Triangle inequality, $\hat{d}(p, q) \leq \hat{d}(p, r) + \hat{d}(r, q)$, is the result of triangle inequality for Euclidean distance. Finally, we need to check that $\hat{d}(p, q) = 0$ if and only if $p = q$. Clearly, $\hat{d}(p, p) = \|p^* - p^*\| = 0$. The only remaining property is that $\hat{d}(p, q) = 0$ implies $p = q$. From the formula (2), using conditional positive definiteness of A and the fact that the sum of entries of the vector $\vec{w}(p) - \vec{w}(q)$ is zero, one can see that $\hat{d}(p, q) = 0$ implies $\vec{w}(p) = \vec{w}(q)$. Now by linear precision, we have $p = \sum_i w_i(p) v_i = \sum_i w_i(q) v_i = q$. \square

Note that we required A to be conditionally positive definite in order to ensure that $\hat{d}(p, q) = 0$ implies $p = q$. Two remarks are in order. First, for diffusion and commute-time distances, A is conditionally positive definite. In fact, A is positive definite for diffusion distance, because then A is the heat kernel, and its eigenvalues are of the form $e^{-t\lambda}$. For commute-time distance A is the pseudo-inverse of the Laplacian, and, thus, has positive eigenvalues, except for the constant eigenvector. Second, what this condition achieves is the injectivity of the map $p \mapsto p^*$. Overall, this injectivity is related to the question of whether during a cage based deformation the interior can fold over itself. Since in practice deformations using barycentric coordinates are injective (except in extreme cases, see [FK]); such cases are even

less probable in our setting because we embed into higher-dimensional spaces), one usually gets a metric even without assuming the conditional positive definiteness of A .

Same topology as Euclidean: If the Gram matrix A is conditionally positive definite, and barycentric coordinates are continuous, then \hat{d} induces the same topology in the interior of the mesh as Euclidean distance.

This property is important because it ensures that we are not augmenting the notion of closeness in the interior of the mesh – for example, a connected set in Euclidean topology does not become disconnected in the topology induced by the new distance.

Proof We need to prove for a sequence of points p_n , that $\hat{d}(p_n, p) \rightarrow 0$ if and only if $\|p_n - p\| \rightarrow 0$. By continuity of weights, $\|p_n - p\| \rightarrow 0$ implies $\vec{w}(p_n) \rightarrow \vec{w}(p)$, and then by Eq. (2) $\hat{d}(p_n, p)$ converges to zero. The opposite implication follows because if $\hat{d}(p_n, p) \rightarrow 0$, then from Eq. (2) and conditional positive definiteness of A , we get $\vec{w}(p_n) \rightarrow \vec{w}(p)$. Linear precision can be used as in the previous proof to get $\|p_n - p\| \rightarrow 0$. \square

Maximum principle: If the weights are non-negative, then the diameter of the interior under \hat{d} is not greater than the diameter of the boundary under d :

$$\max_{p, q \in \Omega} \hat{d}(p, q) \leq \max_{p, q \in \partial\Omega} d(p, q).$$

Stated differently, the maximum distance between two interior points is no more than the maximum distance on the boundary.

Proof Let $D_{max} = \max D_{ij}$, then due to non-negativity we get from equation (5) that

$$\begin{aligned} \hat{d}^2(p, q) &\leq \sum_{i,j} D_{ij} w_i(p) w_j(q) \leq D_{max} \sum_{i,j} w_i(p) w_i(q) = \\ &= D_{max} \sum_i w_i(p) \sum_j w_j(q) = D_{max}, \end{aligned}$$

where we use the partition of unity property in the last step. \square

Euclidean reproduction: Interpolating Euclidean distance from the boundary reproduces Euclidean distance in the interior: if $d(v_i, v_j) = \|v_i - v_j\|_{\mathbb{R}^3}$, then $\hat{d}(p, q) = \|p - q\|_{\mathbb{R}^3}$.

Proof Note that the identity map $v_i \mapsto v_i \in \mathbb{R}^3$ provides an embedding of vertices that preserves Euclidean distances. The extension of this map to the interior is also the identity map because it is given by $p \mapsto p^* = \sum_i w_i(p) v_i^* = \sum_i w_i(p) v_i = p$, where the last equality holds because of the linear precision. Now we have, $\hat{d}(p, q) = \|p^* - q^*\|_{\mathbb{R}^3} = \|p - q\|_{\mathbb{R}^3}$. The Euclidean reproduction property now follows from the fact that our construction is independent of the embedding used. \square

This last property of our framework should be compared with the linear precision required in the framework of

barycentric coordinates. The result of this requirement is that when barycentric interpolation is applied to a linear function, it is reproduced. Clearly, there is no intention of using barycentric coordinates to interpolate linear functions, but existence of this property leads generally to quadratic approximation order (in the diameter of the shape) and provides a validity check. In the same way the fact that Euclidean distances are reproduced by our framework provides a validity check and, most likely, a better approximation order (see Figure 11).

6. Results

In order to investigate properties of our interior distance framework empirically, we have performed several experiments. The goal of these experiments is to study how interior distance changes with perturbations to the source point, boundary geometry, boundary distances, and barycentric coordinates.

Unless otherwise specified, we use the diffusion distance [CLL*05,GGV08] as the boundary distance and mean value coordinates [Flo03,JSW05] as the barycentric coordinates. The diffusion distance was chosen because it is shape-aware, it provides all the conditions for a metric, it is defined in terms of a higher-dimensional embedding, and practically is very efficient to compute. The mean value coordinates were chosen because they are computationally efficient, since they possess closed-form formulas both in 2D [Flo03] and 3D [JSW05], and they have all the desirable properties except non-negativity (which is only relevant for the maximum principle).

For our experiments, we compute distances between pairs of points and visually evaluate the results. For ease of visualization, we generally select a single “source point” and then compute the distance to all other points in the interior of the object from that point. In general, the “source point” is depicted as a black dot, dark blue represents small distance values, dark red represents large values, and isolines are shown with equal spacing in the interior distance range. For 3D examples, we visualize slicing planes within the volume of the object.

Effect of source point: Figure 3 shows how the distance changes as the source point varies within a 2D shape. From this figure, one can see that the distance does not experience any discontinuities as the source changes. Also, the isolines confirm that the resulting distance is smooth within the shape interior.

Effect of boundary shape: Figure 4 shows how the distance from a fixed source changes as the boundary shape varies. It can be seen that the interior distance is “shape-aware” – it is affected by global shape features. For example, the farthest isoline includes the upper right corner in every case, which is intuitive. This figure is also a test of the effect of increasing concavity. It is seen that even though the mean value co-

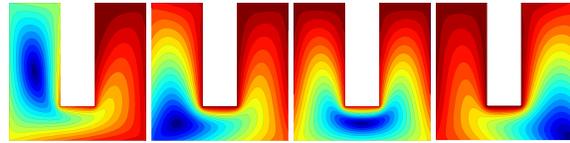


Figure 3: Variation of the interior distance as the source point varies.

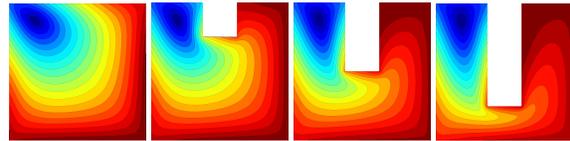


Figure 4: The effect varying shape on the interior distance.

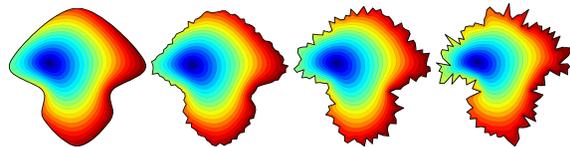


Figure 5: The effect of adding increasing amounts of boundary noise on the interior distance.

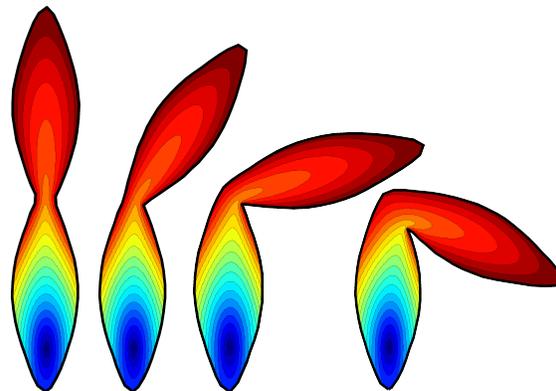


Figure 6: The effect of articulation on the interior distance.

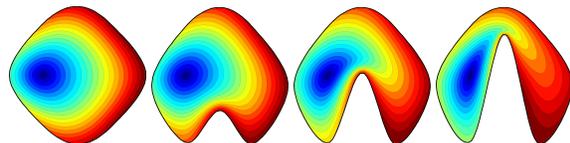


Figure 7: The effect of deformation on the interior distance.

ordinates do become negative with concavity, the resulting distance does not suffer any visual artifacts.

Effect of boundary noise: Figure 5 shows the interior distance for a series of shapes obtained by adding noise to the boundary of a smooth shape. In this example, the vertices are displaced by a Gaussian noise of standard deviation of 1%, 2%, and 4% of the shape’s diameter, from left to right.



Figure 8: 3D Distance visualization within the human, dino-pet, hand, cheetah, octopus, and fertility models.

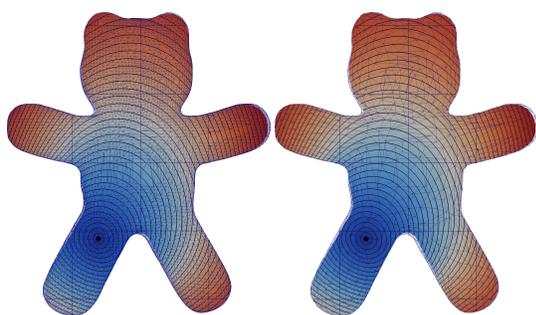


Figure 9: The effect of tessellation on the interior distance, 9K vertices (left) versus 1.5K vertices (right).

Note that the interior distance remains largely the same as the boundary noise increases.

Effect of articulation: Figure 6 shows that the interior distance is insensitive to articulation. Notice even in the case of the right-most articulation example, where the thickness in the middle changes and an increased concavity appears, the distance is still stable.

Effect of deformation: Figure 7 shows how the interior distance changes as a deformation is applied to the boundary. In this case, the interior distance deforms with the boundary – i.e., as interior points move to new locations, so too do the isolines of distance.

Effect of 3D object type: Figure 8 shows the interior distance for a planar slice through the interior of different 3D meshes with different source positions (black dots near the centers of blue regions). Isolines again confirm that the distance is smooth and shape-aware.

Effect of tessellation: Figure 9 shows the interior distance computed for two different tessellations of a 3D model: with 9K vertices on left, and 1.5K on right. The isolines confirm the insensitivity of the interior distance to tessellation. This suggests that the interior distance can be computed reliably using a simplified mesh for very large models.

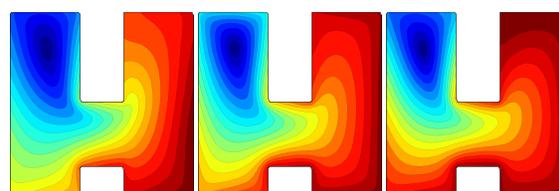


Figure 10: Interior distances derived from different boundary distances (from left to right): geodesic, commute-time, and diffusion distances.

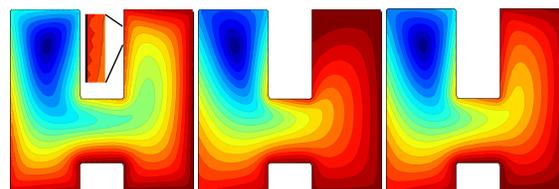


Figure 11: Interior distances derived from different vertex weights (from left to right): Shepard's weights, mean value coordinates (MVC), and maximum entropy coordinates (MEC). The latter two (MVC and MEC) are barycentric coordinates, while the Shepard weights are not (they satisfy all conditions except linear precision). The zoom-in shows the ripples appearing when using Shepard weights.

Effect of boundary distance: Figure 10 compares the interior distance defined by three different boundary distances: geodesic distance (approximating the Gram matrix A by a positive semi-definite one and embedding via MDS), commute-time distance, and diffusion distance. These three interior distances exhibit different levels of shape-awareness, and thus each of the three may be most appropriate for different applications.

Effect of barycentric coordinates: Figure 11 compares the interior distance defined by three different interpolation schemes: Shepard's weights, mean value coordinates (MVC), and maximum entropy coordinates (MEC). Shep-

ard's weights are not exactly barycentric coordinates as they do not satisfy the linear precision requirement, but they do satisfy all other requirements. The absence of linear precision seems to be the reason for the ripples seen close to the boundary on the left image. MVC satisfy all conditions (smoothness included) except non-negativity. MEC satisfy all properties including non-negativity; however, MEC are only conjectured to be smooth based on experiments [HS08].

Compute Time: Table 1 reports the time in seconds to compute the interior distance for a variety of three-dimensional models on a 2.2Ghz AMD Opteron 875 processor. Note that our implementation is in MATLAB, except the calculation of the mean value coordinates, for which we use C++. In spite of using unoptimized code, distance calculations still take milliseconds after precomputation.

V	F	#eigens	preprocess	per-distance
2.5K	4.9K	66	3.52	0.008
4.3K	8.6K	104	7.53	0.018
7.9K	15.8K	112	11.96	0.034
16K	31.9K	82	45.47	0.059
26.8K	53.6K	93	151.63	0.120

Table 1: Timing for distance evaluation in seconds is provided for 3D models of different sizes. During the preprocessing the discrete Laplacian is set up, and enough eigenvectors are evaluated to achieve the set tolerance (see Appendix). Given the preprocess data, the last column shows the time needed to compute the distance between a pair of interior points; note that it is dominated by the computation of the mean value interpolation.

7. Application - Computing the Barycentroid

Beyond computing distances, we propose a new shape analysis concept, which we call the “barycentroid” of a mesh.

The barycentroid provides a notion of the “center point” of a shape, which we model after the following definition of the usual center of mass. The center of mass (barycenter) of a set can be defined as the point for which the average squared Euclidean distance to the set elements is minimal. We replace in this definition Euclidean distance by the interior distance, and call the minimizing point the *barycentroid*. Since we are dealing with meshes, we need to actually minimize the area weighted averages; here the vertex voronoi areas provide the appropriate weighting. When diffusion distance is used on the boundary, we can prove that minimizing this area weighted average is equivalent to minimizing the following simple potential $U^2(p) = \vec{w}(p)^T A \vec{w}(p)$.

Figures 12 and 13 show the isolines of $U(\cdot)$ in several two and three dimensional cases. For calculating the barycentroid in 3D, we follow the direction of negative gradient of this function U until reaching a minimum. Figure 12 also demonstrates that the barycentroid is insensitive to noise

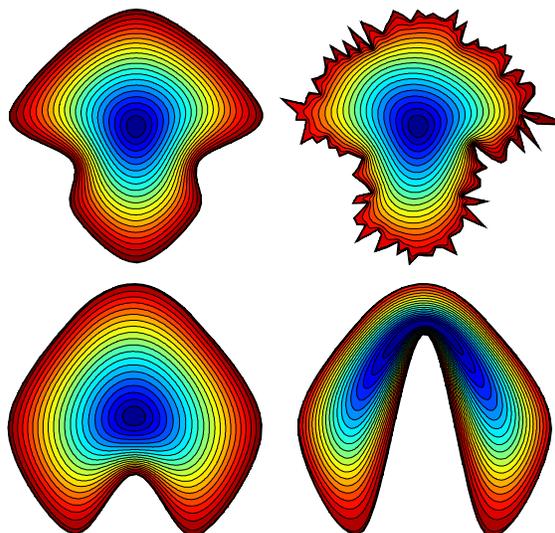


Figure 12: Visualization of isolines of $U(\cdot)$, the defining function of the barycentroid, demonstrating the robustness of the barycentroid to noise and deformation.

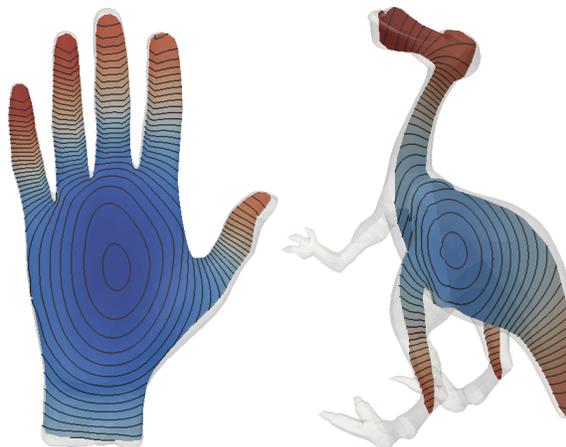


Figure 13: Isolines of $U(\cdot)$ for hand and dino-pet models.

and deformation. Figure 14 shows the insensitivity of the barycentroid to different poses and different objects from the same class. Note how the barycentroid stays approximately constant as the object undergoes pose changes, demonstrating its invariance to articulation. In addition, the location of the barycentroid remains consistent across different models that can be seen as belonging to the same class. These examples give practical evidence that the minimizer point of U is appropriate to serve as a “center point” both in 2D and 3D.

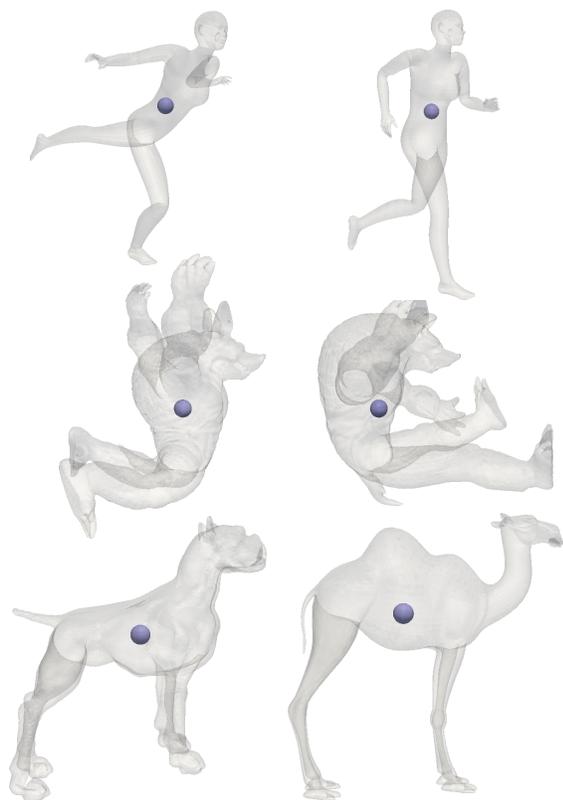


Figure 14: The barycentroid is depicted as blue spheres inside various shapes. The first two rows demonstrate the invariance of the barycentroid to pose. The last row shows its insensitivity to different objects of the same class.

8. Conclusion and Future Work

This paper presents a framework for taking a distance measure on a boundary mesh and propagating it into the interior of the mesh while preserving its properties. In particular, if a distance-preserving embedding exists, the constructed interior distance: 1) interpolates the given distance on the boundary, 2) satisfies the axioms of a metric, 4) reproduces the Euclidean distance, 5) is topologically equivalent to Euclidean distance, 6) possesses a “maximum principle” (when the barycentric coordinates are positive), and 7) is fast to compute. In addition to proving these properties theoretically, we provide empirical results for an instantiation of the framework using the diffusion distance and mean value coordinates.

This work provides a small, first step and therefore has limitations that suggest topics for future work.

A first topic suitable for further investigation is to characterize the empirical properties of our interior distance for different combinations of boundary distances and barycen-

tric coordinates. In our initial implementation, we chose to use diffusion distance and mean-value coordinates because they have desirable properties for our framework. However, perhaps others would provide even better properties. Or, perhaps we could design distances and coordinates that do.

Another interesting topic for future work would be to investigate applications of interior distances in computer graphics and geometric modeling. We describe a method for computing the barycentroid of a 3D mesh, which may have applications in shape registration, matching, skeletonization, etc. However, we imagine that several other applications are possible and/or become practical with an interior distance measure that is efficient to compute.

Acknowledgements

This project would not have been possible without the help of people and organizations who provided the data and funding. For the 3D models used in this paper, we thank Daniela Giorgi and AIM@SHAPE. For funding in support of this project, we thank the NSF (CNFS-0406415, IIS-0612231, and CCF- 0702672), Google, and The Rothschild Foundation.

References

- [BCKO08] BERG M. D., CHEONG O., KREVELD M. V., OVERMARS M.: *Computational Geometry: Algorithms and Applications*. Springer-Verlag TELOS, Santa Clara, CA, USA, 2008.
- [BG05] BORG I., GROENEN P. J. F.: *Modern multidimensional scaling*, second ed. Springer Series in Statistics. Springer, New York, 2005. Theory and applications.
- [CC01] COX T. F., COX M. A. A.: *Multidimensional scaling*, second ed., vol. 59 of *Monographs on Statistics and Applied Probability*. Chapman & Hall/CRC, London, 2001.
- [CLL*05] COIFMAN R. R., LAFON S., LEE A. B., MAGGIONI M., NADLER B., WARNER F., ZUCKER S. W.: Geometric diffusions as a tool for harmonic analysis and structure definition of data: Diffusion maps. *Proceedings of the National Academy of Sciences* 102, 21 (May 2005), 7426–7431.
- [CR87] CANNY J., REIF J.: New lower bound techniques for robot motion planning problems. In *SFCS '87: Proceedings of the 28th Annual Symposium on Foundations of Computer Science* (Washington, DC, USA, 1987), IEEE Computer Society, pp. 49–60.
- [CSY94] CHOI J., SELLEN J., YAP C.-K.: Approximate euclidean shortest path in 3-space. In *SCG '94: Proceedings of the tenth annual symposium on Computational geometry* (New York, NY, USA, 1994), ACM, pp. 41–48.
- [FK] FLOATER M., KOSINKA J.: On the injectivity of wachspress and mean value mappings between convex polygons. *Advances in Computational Mathematics (to appear)*.
- [Flo03] FLOATER M. S.: Mean value coordinates. *Computer Aided Geometric Design* 20, 1 (2003), 19–27.
- [GGV08] GOES F. D., GOLDENSTEIN S., VELHO L.: A hierarchical segmentation of articulated bodies. *Computer Graphics Forum (Special Issue of Symposium on Geometry Processing)* 27, 5 (2008), 1349–1356.

- [HF06] HORMANN K., FLOATER M. S.: Mean value coordinates for arbitrary planar polygons. *ACM Transactions on Graphics* 25, 4 (2006), 1424–1441.
- [HS08] HORMANN K., SUKUMAR N.: Maximum entropy coordinates for arbitrary polytopes. *Computer Graphics Forum* 27, 5 (July 2008), 1513–1520. Proceedings of SGP 2008.
- [JSW05] JU T., SCHAEFER S., WARREN J.: Mean value coordinates for closed triangular meshes. vol. 24, 3 (Proc. SIGGRAPH), pp. 561–566.
- [LFR09] LIU Y.-S., FANG Y., RAMANI K.: Idss: deformation invariant signatures for molecular shape comparison. *BMC Bioinformatics* 10, 1 (2009), 157–171.
- [LJ07] LING H., JACOBS D. W.: Shape classification using the inner-distance. *IEEE Trans. Pattern Anal. Mach. Intell.* 29, 2 (2007), 286–299.
- [MDSB02] MEYER M., DESBRUN M., SCHRÖDER P., BARR A.: Discrete differential geometry operators for triangulated 2-manifolds. In *Proceedings of Visual Mathematics* (2002).
- [Pap85] PAPANIMITRIOU C.: An algorithm for shortest-path motion in three dimensions. *IPL* 20 (1985), 259–163.
- [PP93] PINKALL U., POLTHIER K.: Computing discrete minimal surfaces and their conjugates. *Experimental Mathematics* 2 (1993), 15–36.
- [Rus07] RUSTAMOV R. M.: Laplace-Beltrami eigenfunctions for deformation invariant shape representation. In *Symposium on Geometry Processing* (2007), pp. 225–233.
- [SSK*05] SURAZHSKY V., SURAZHSKY T., KIRSANOV D., GÖRTLER S. J., HOPPE H.: Fast exact and approximate geodesics on meshes. *ACM Trans. Graph.* 24, 3 (2005), 553–560.
- [Xu04] XU G.: Discrete Laplace-Beltrami operators and their convergence. *Comput. Aided Geom. Des.* 21, 8 (2004), 767–784.
- [YFD*07] YEN L., FOUSS F., DECAESTECKER C., FRANCO P., SAERENS M.: Graph nodes clustering based on the commute-time kernel. In *Proceedings of the 11th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD 2007). Lecture notes in Computer Science, LNCS* (2007).

Appendix

In this appendix, we provide more detail about our implementation in the case when the boundary distance is diffusion distance and weights used are mean value coordinates.

Our computation of mean value coordinates in 2D is based on the pseudocode provided in [HF06]. For 3D we use the pseudo-code of [JSW05].

The diffusion distance is incorporated in our framework using its explicit embedding – the diffusion map. The diffusion map is based on the eigendecomposition of the Laplace-Beltrami operator. We use the discrete Laplace-Beltrami operator of the form $M^{-1}L$, where the matrices L and M defined as follows. In three dimensions, where the boundary is a triangle mesh, let L be the cotangent Laplacian [PP93], and M be the diagonal matrix that contains the voronoi areas of mesh vertices [MDSB02, Xu04]. In two-dimensions, the boundary is a polygon; to obtain L we use the stencil $[-1/d^-, 1/d^- + 1/d^+, -1/d^+]$ at each vertex, and we define the “voronoi area” of each vertex as $(d^+ + d^-)/2$. Here,

d^+ and d^- are the distances to the two neighbors of the current vertex.

The eigendecomposition is computed by solving the symmetric generalized eigenvalue problem $L\phi = \lambda M\phi$. The eigenvalues λ_k are assumed to be ordered in a non-decreasing order, and the eigenvectors ϕ_k are normalized to have the unit M -norm, $\phi_k^T M \phi_k = 1$.

Following [GGV08] we set the time scale parameter t of diffusion inversely proportional to the first non-zero eigenvalue λ_2 (remember $\lambda_1 = 0$), namely, we use $t = (8\lambda_2)^{-1}$. Note that due to the fast decay of the term $e^{-\lambda_k t}$ appearing in the diffusion map, we only need a limited number of eigenvectors to reliably approximate the exact embedding (diffusion map). We retain all the eigenvectors that satisfy $e^{-\lambda_k t} > \epsilon$, where we used $\epsilon = 10^{-6}$.

After computing these m eigenvalues/eigenvectors in pre-process, we can compute the distance between two interior points in time $O(mn)$, where n is the number of mesh vertices. Indeed, we need to extend each dimension of the boundary embedding to the interior via the mean value interpolation. There are m dimensions, and the computation of mean value interpolant takes time linear in the number of mesh vertices.