Biharmonic Distance

Y. Lipman[1*], R. M. Rustamov[2*], T. A. Funkhouser[1]

[1]Princeton University

[2]Drew University

Measuring distances between pairs of points on a 3D surface is a fundamental problem in computer graphics and geometric processing. For most applications, the important properties of a distance are that it is a metric, smooth, locally isotropic, globally "shape-aware," isometry invariant, insensitive to noise and small topology changes, parameter-free, and practical to compute on a discrete mesh. However, the basic methods currently popular in computer graphics (e.g., geodesic and diffusion distances) do not have these basic properties. In this paper, we propose a new distance measure based on the biharmonic differential operator that has all the desired properties. This new surface distance is related to the diffusion and commute-time distances, but applies different (inverse squared) weighting to the eigenvalues of the Laplace-Beltrami operator, which provides a nice trade-off between nearly geodesic distances for small distances and global shape-awareness for large distances. The paper provides theoretical and empirical analysis for a large number of meshes.

Categories and Subject Descriptors:

Additional Key Words and Phrases: Shape analysis, mesh processing, mesh distance

1. INTRODUCTION

Measuring the distances between pairs of points on a 3D surface is a classical problem in computer graphics, geometric processing, and shape analysis. It is a critical step in most shape analysis applications, including segmentation, embedding, parameterizations, deformation, and matching of 3D surface meshes.

For these applications, the important properties of a distance from a point x to another point y are that it is: 1) *metric:* non-negative, satisfies the identity of indiscernibles, symmetric, and satisfies the triangle inequality; 2) *gradual:* smooth with respect to perturbations of x and y, with no singularities except derivative discontinuity at x; 3) *locally isotropic:* approximately geodesic when y is near x;

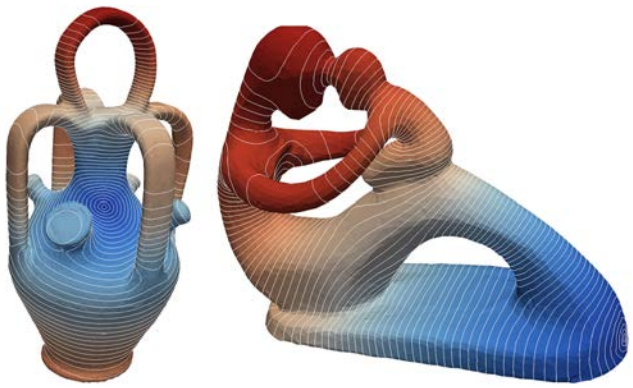


Fig. 1: *Biharmonic distance from a source point (darkest blue). Red points are furthest from the source. White lines are equally spaced in distance.*

4) *globally "shape-aware:"* reflects the overall shape of the surface when y is far from x; 5) *isometry invariant:* does not change with isometric transformations of the surface, 6) *insensitive to noise and topology:* does not change significantly with the addition of noise or changes to topology; 7) *practical to compute:* compute times between all pairs of points in common meshes take at most a few minutes; and 8) *parameter-free:* independent of any parameter that must be set differently for specific meshes or applications.

Although these properties seem fundamental, there is no current distance measure that satisfies all of them. Geodesic distance [Papadimitriou 1985; Surazhsky et al. 2005] is a metric and locally isotropic, but it is not smooth, insensitive to topology, or globally shape-aware. Alternatively, diffusion distance [Coifman and Lafon 2006] is either not locally isotropic or not globally shape-aware, depending on a parameter, and it is not necessarily a metric (when computed using only the first few eigenvalues and eigenvectors). Finally, the graph-theoretical commute-time distance [Fouss et al. 2006] cannot be defined on a continuous domain (diverges), and possesses a strong singularity at the source point.

In this paper, we introduce a novel distance operator that has all of the desired properties (Figure 1). The key idea is to balance between local and global properties of the distance by an eigenvalue normalization that is intimately related to the biharmonic differential equation. The proposed *biharmonic distance* bridges the gap between diffusion (with small t), commute-time and geodesic distances, which provide good properties locally, and diffusion distance (with large t), which provides good properties globally. Since it is practical to compute (Section 3), has provable theoretical properties (Section 4) and "shape-aware" distance measurements (Section 5), we believe that it is immediately useful for a wide variety of shape analysis applications.

* equal contributors.

DOI

http://doi.acm.org/

2. BACKGROUND AND RELATED WORK

Measuring distance, $d(x, y)$, between two points x and y on a surface is a classical problem, and thus several distance measures have previously been proposed and are commonly used in shape analysis applications.

Geodesic distance: Perhaps the most popular distance is geodesic, which measures the length of the shortest path along the surface between two points. Although this distance is intuitive and useful for many applications (e.g., path planning), it has properties that are undesirable for global shape analysis (e.g., mesh segmentation). The problems stem mainly from its *local* nature – i.e., the geodesic distance between two points on surface depends only upon an infinitely small neighborhood around the shortest path. As a result, distances are shape-oblivious (they are not affected by the rest of the surface), there are derivative discontinuities at points where two different shortest paths have the same lengths (at the collision between advancing fronts), and small perturbations to the surface (e.g., introducing a hole along the shortest path) can have significant affect on distances. Another well-known drawback of geodesic distances is its sensitivity to "topological noise"; introducing arbitrary small topological shortcuts might cause arbitrary large change in geodesic distance between pair of points. Moreover, computing exact geodesic distances between all-pairs of points is quite slow for large meshes. Exact and approximate solutions [Papadimitriou 1985; Surazhsky et al. 2005] take $O(N^{2.5} \log N)$ (on average, worst case is $O(N^3 \log N)$), for all-pairs of N vertices mesh and are difficult to implement. Thus, people generally approximate continuous geodesic distance with a discrete graph version using Dijkstra's algorithm, which still takes $O(N^2 logN)$.

Diffusion distance: For global shape analysis, it is common to model distance between points by a diffusion process within time t (a parameter). This (square of the) distance can be computed by:

$$d_D(x, y)^2 = \sum_{k=1}^{\infty} e^{-2t\lambda_k} \left(\phi_k(x) - \phi_k(y)\right)^2, \tag{1}$$

where $\phi_k(x), \lambda_k$ are the eigenfunctions and eigenvalues (resp.) of the positive definite Laplace-Beltrami operator (henceforth, the Laplacian)

$$\Delta\phi_k(x) = \lambda_k \phi_k(x),$$

where $0 = \lambda_0 < \lambda_1 \leq \lambda_2 ...$, and $t > 0$ is a time parameter. While diffusion distance is global and can be computed very quickly for all pairs of vertices (in $O(N^{1.5})$), it depends heavily on the parameter t, which intuitively specifies the amount of "diffusion time" during which paths are explored to discover connectivity between vertices. On the one hand, if t is chosen too large, then only the eigenvector(s) with the lowest eigenvalues are considered, and the result is a distance measure with nice global properties, but poor local properties (i.e., it is not locally isotropic, as shown in Figures 2 and 7). On the other hand, if t is too small, then the diffusion process runs for only a short time, and the resulting distance is useful locally, but exhibits unexpected behavior in distant areas such as near-plateaus and local extrema. Note that we adopt the notation in [Goes et al. 2008] by normalizing the time scale $t \leftarrow t/(2\lambda_1)$ to achieve scale invariance. Henceforth, by abuse of notation, diffusion times t will actually denote $t/2\lambda_1$.

Commute-time distance: Another related type of distance on graphs is the *commute-time* distance [Fouss et al. 2006; Yen et al. 2007; Qiu and Hancock 2007]. This distance can be can be intuitively described as follows: the commute-time distance between two vertices on the graph is the average time it takes a random walker to go from one vertex to the other and back. The requirement of having the random walker return back makes this distance symmetric. It can also be described using the eigenvalues and eigenvectors of the Laplacian:

$$d_C(x, y)^2 = \sum_{k=1}^{\infty} \frac{1}{\lambda_k} \left(\phi_k(x) - \phi_k(y)\right)^2, \tag{2}$$

Denoting (formally) by $g_C(x, y)$, the Green's function of the Laplacian the commute-time distance has the following equivalent definition [Fouss et al. 2006]:

$$d_C(x, y)^2 = g_C(x, x) + g_C(y, y) - 2g_C(x, y). \tag{3}$$

This distance has two main disadvantages: 1) it cannot be defined on surfaces, since its Green's function has a singularity at the diagonal, and 2) it depends only upon the conformal structure (equivalence class) of the surface which means it is the same for every two conformally equivalent surfaces.

The first drawback is related to the singularity of the harmonic Green's function at the diagonal; for example, the Green's function for simply connected planar Euclidean domain (and actually for every two-dimensional surface) has logarithmic singularity at the diagonal $g_C(x, x)$, and $g_C(x, y)$ is well-defined for all $x \neq y$. This means that the formula (3) is not defined in the continuous case. This property can also be understood by noting that the series $\sum_{k=1}^{\infty} 1/\lambda_k$ is diverging since for two dimensional surfaces $\lambda_k \sim k$ (see page 421 [Berger 2003] for compact surfaces). The second property can be understood by noting that this distance can be written in terms of the Green's function of the Laplacian $g_C(x, y)$ as Eq. (3) and that the Green's function of the Laplacian is conformally invariant.

3. APPROACH

Our new distance operator is similar to diffusion distance and commute-time distance, but it is based on a kernel that is the Green's function of the biharmonic differential equation.

The formal definition of the distance operator can be done in few equivalent ways. We will start with a continuous definition, move to the definition in the discrete case, and finish with the practical computation and approximation procedures.

3.1 Continuous construction

In the continuous case, the (square of the) distance can be simply defined using the eigenvectors and eigenvalues of the Laplace-Beltrami operator:

$$d_B(x, y)^2 = \sum_{k=1}^{\infty} \frac{\left(\phi_k(x) - \phi_k(y)\right)^2}{\lambda_k^2}. \tag{4}$$

This definition is only slightly different from the graph-theoretic commute-time distance [Yen et al. 2007] where the power of the λ in the denominator is one, and the diffusion distance where the denominator is $e^{2t\lambda_k}$. The interesting part is that this seemingly minor change produces a *fundamentally different* distance scheme with different properties than diffusion and commute-time distance. In a nut-shell, it is all related to how fast the normalized λ_k (in

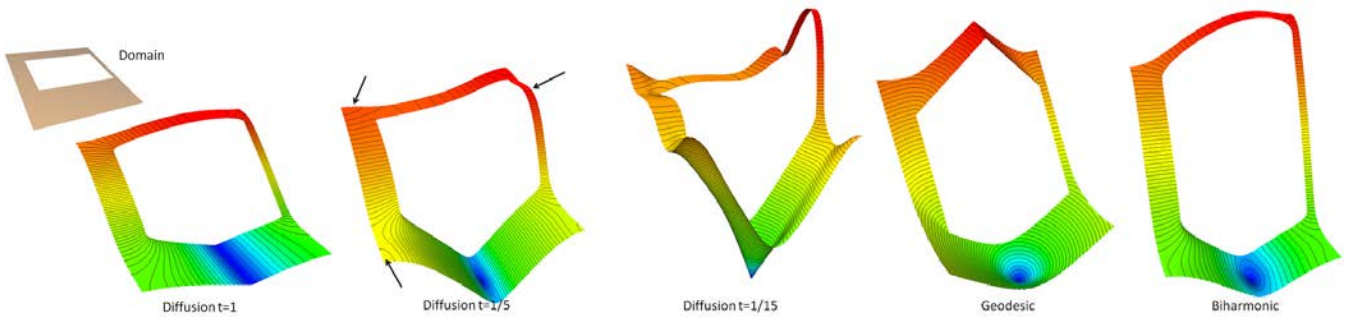


Fig. 2: *Distances measured on an Euclidean domain (top-left). We visualize the distance field from a single source point to all other points as a height function over this Euclidean domain. From left to right: diffusion distance for times $t = 1, 1/5, 1/15$ (calculated w.r.t the second largest eigenvalue, see Section 2), geodesic distance, and biharmonic distance (this paper). Note how diffusion distance either possesses many local maxima (indicated by black arrows) or locally does not reflect the source point, and the geodesic distance possesses non-smooth curve of points and the isolines are not "shape-aware" far from the source. Biharmonic distance balances the "local" and "global" properties.*

our case $1/\lambda_k^2$) decays: on one hand, if the decay is too slow (like $\sum_k 1/\lambda_k \sim \sum_k 1/k$), it will produce logarithmic singularity at the diagonal of the Green's function (harmonic Green's function, commute-time distance [Yen et al. 2007]). On the other hand if the decay is too fast, it basically ignores eigenvectors with high frequencies, and the distance is "too global". We will demonstrate that taking the quadratic normalization as done in Eq. (4) has the following justifications: it provides a good balance in the sense that it decays slow enough to get good local properties around the source point and fast enough to be shape adherent in "far" areas. This specific "balancing" it is intimately related to the biharmonic equation. In particular, Eq. (4) can be written in the following way (opening the parenthesis):

$$\begin{aligned} d_B(x,y)^2 &= \sum_{k=1}^{\infty} \frac{|\phi_k(x)|^2}{\lambda_k^2} + \sum_{k=1}^{\infty} \frac{|\phi_k(x)|^2}{\lambda_k^2} - 2\sum_{k=1}^{\infty} \frac{\phi_k(x)\phi_k(y)}{\lambda_k^2} \\ &= g_B(x,x) + g_B(y,y) - 2g_B(x,y), \end{aligned}$$

using the Green's function $g_B(x,y)$ of the biharmonic operator Δ^2:

$$g_B(x,y) = \sum_{k=1}^{\infty} \frac{\phi_k(x)\phi_k(y)}{\lambda_k^2}. \tag{5}$$

The Green's function $g_B(x,y)$ satisfies the relation

$$\Delta^2_{(x)} \int g_B(x,y) f(y) dy = f(x), \tag{6}$$

for "smooth enough" f [John 1986].

The benefits of this distance formulation in comparison to diffusion and geodesic distances can be seen in Figure 2, which shows a Euclidean domain (top-left) and the different distances visualized as height functions over this domain. Note that diffusion distance possesses good global properties for long times (left) and good local properties for small times (middle), but not both: when it is good locally it has unintuitive global behavior with many local extrema; yet, when its monotone and shape aware on far areas it produces unintuitive behavior locally. Geodesic distance is not shape aware for distant areas from the source point (observe the isolines) and possesses derivatives singularities (the kink at the top). Biharmonic strikes a balance: note that the isolines follow the shape, and the farthest point leverages the overall connectivity rather than the single shortest distance. Furthermore, it has an intuitive local behavior.

3.2 Discrete construction

The discrete definition of the biharmonic distance is based on constructing a discrete Green's function g_d of the Bi-Laplacian using formula Eq. (5).

Our discretization of the Green's function is based on the common "cotangent formula" discretization of the Laplace-Beltrami differential operator on meshes [Meyer et al. 2002; Grinspun et al. 2006]:

$$(\Delta_d u)_i = \frac{1}{A_i} \sum_{j \in Nei(i)} (cot\alpha_{ij} + cot\beta_{ij})(u_i - u_j), \tag{7}$$

where $(\Delta_d u)_i$, for a mesh function u, denotes its discrete Laplacian evaluated at vertex i (for $i = 1, 2, ..., N$, N number of vertices); A_i is the Voronoi area at i^{th} mesh vertex [Grinspun et al. 2006]; angles α_{ij}, β_{ij} are the two angles supporting the edge connecting vertices i and j (see for example [Meyer et al. 2002]). We denote the matrix of the linear transformation (7) by L_d. We also consider the area/mass matrix $A \in \mathbb{R}^{N \times N}$ which is the diagonal matrix with $A_{ii} = A_i$. Note that $L_d = A^{-1} L_c$, where L_c is the conformal discrete Laplacian [Pinkall and Polthier 1993], i.e. the matrix of the linear transformation in 7 but without the inverse area terms.

Given this discretization of the Laplacian, we define the discrete Green's function of the Bi-Laplacian, $g_d \in \mathbb{R}^{N \times N}$, by discretizing the relation in Eq. (6). Indeed, when the continuous integration is discretized we obtain

$$L_d^2 g_d A f = f, \tag{8}$$

where $f \in \mathbb{R}^{N \times 1}$ is an arbitrary vector in the image of L_d^2. As we prove in the Appendix (Theorem 3), this requirement is satisfied if we take g_d to be the *pseudo-inverse* of $L_c A^{-1} L_c$. As the pseudo-inverse of a symmetric matrix, our discrete Green's function is symmetric $g_d = g_d^t$.

Finally, having g_d at hand, the biharmonic distance on the mesh is defined via Eq. (5):

$$d_B(v_i, v_j)^2 = g_d(i,i) + g_d(j,j) - 2g_d(i,j). \tag{9}$$

Surfaces with boundaries

In order to define the biharmonic distance on surfaces with boundaries, we need to prescribe boundary conditions for the differential operator. As we will show, using the same equation, that is eq. (7), also for boundary vertices, will provide the desired solution. Therefore, surfaces with boundaries are treated in the same way.

The basic idea is to define the Laplacian with Neumann boundary conditions. That is, the eigenfunction are the eigenfunctions of the Laplacian, $\Delta\phi_k = \lambda\phi_k$, where on the boundary we force the outward normal derivative to be zero:

$$\frac{\partial\phi_k}{\partial\vec{n}} = 0,$$

where $\vec{n}$ denotes the outward normal direction on the boundary. Note that the cotangent Laplacian applied on a boundary vertex approximates the normal derivative. Figure 3 shows the biharmonic distance on a mesh with many holes and missing parts, and therefore with many boundaries. Note that the distance is largely insensitive to them.

Fig. 3: *The biharmonic distance on a raw surface mesh with many boundaries. Note that the isolines follow the shape indifferently to the missing parts and holes (e.g., the marked arrows).*

3.3 Practical computation

Given a subset of M vertices on the mesh (possibly $M = N$), we are interested in calculating all-pairs of distances in this subset. We will present two methods: 1) exact computation, and 2) approximate computation.

Exact distance computation The basic idea is that all the pairwise distances in this subset can be calculated quickly ($O(1)$) after the relevant M columns of the matrix g_d are known. Indeed, calculating the distance between vertices v_i, v_j in the subset of M vertices using Eq. (9) requires merely three values located on the i^{th} and j^{th} columns. So from this point until the end of this section, we will concentrate on how to evaluate efficiently the columns of g_d.

One way to calculate the pseudo-inverse is by performing full eigen-decomposition of $L_cA^{-1}L_c$. This is obviously a computationally expensive task. Let us present a more efficient way to calculate the j^{th} column of matrix g_d. First, we need to understand what is the matrix J in the product $L_cA^{-1}L_cg_d = J$. As proved in the Appendix (Theorem 2), $J = I - \frac{1}{N}\mathbf{1}\mathbf{1}^t$ where $\mathbf{1} \in \mathbb{R}^{N\times 1}$ is a column vector of all ones. Also, we note that each column of g_d, denoted by $(g_d)_j$, $j = 1, ..., N$, is the unique solution of the equation $L_cA^{-1}L_cy = (J)_j$ such that its entries add up to zero (Theorem 4 in the appendix). Given these observations it is rather easy to calculate each given column of g_d, $(g_d)_j$:

(1) Find a particular solution x to

$$L_cA^{-1}L_cx = (J)_j. \tag{10}$$

(2) Take

$$(g_d)_j = y = x - \frac{\mathbf{1}^tx}{\mathbf{1}^t\mathbf{1}}\mathbf{1}.$$

The particular solution x can be obtained as follows. We replace some arbitrary row and column of $L_cA^{-1}L_c$ by zeros; the diagonal entry at their intersection is set equal to 1; the corresponding row of J is replaced by zeros. We then solve the (now invertible) linear system. As proved in the Appendix (Theorem 4), given any solution x, the vector y is the sought zero-sum solution of Eq. (10).

Approximate distance computation To approximate the biharmonic distance we can follow the standard methodology for approximating the diffusion distance. That is, approximating $d_B(x, y)$ by taking first K summands in eq. (4). To this end, we compute the first K eigenvectors of the discrete Laplacian, $\Delta_d\phi_k = \lambda_k\phi_k$ which amounts to solving the generalized eigenvalue problem $L_c\phi_k = A\lambda_k\phi_k$. The distance $d_B(x, y)$ is then approximated using the truncated sum:

$$\widetilde{d}_B(x, y)^2 = \sum_{k=1}^{K} \frac{(\phi_k(x) - \phi_k(y))^2}{\lambda_k^2}. \tag{11}$$

Although the error bound is (generally) linear in $1/K$ this approximation still provides considerable speedup in tradeoff of accuracy over the exact computation (see Section 5). Furthermore, since the approximate distance uses a fixed set of eigenvectors it is smooth.

4. THEORETICAL PROPERTIES

This section is devoted to describing the theoretical properties of the biharmonic distance. In particular, we argue that the distance measure is: 1) a metric, 2) smooth everywhere except at the source point where it is continuous, and 3) efficient to calculate. The properties (1) and (2) of the *discrete* biharmonic distance should be in turn derived using approximation properties of our discretization choices.

Metric: First, the biharmonic distance is a metric. The different axioms of a metric can be checked from Eq. (4): 1) the *non-negativity* is clear, 2) the *triangle inequality* stems from the inequality $\sqrt{\sum_{k=1}^{n}(a_k - c_k)^2} \le \sqrt{\sum_{k=1}^{n}(a_k - b_k)^2} + \sqrt{\sum_{k=1}^{n}(b_k - c_k)^2}$ and taking $a_k = \phi_k(x)/\lambda_k$, $b_k = \phi_k(y)/\lambda_k$, $c_k = \phi_k(z)/\lambda_k$, where x, y, z are arbitrary points on the surface and going to the limit $n \to \infty$ to get $d_B(x, z) \le d_B(x, y) + d_B(y, z)$, 3) *symmetry* is clear by observing that the roles of x and y can be switch in Eq. (4) without changing the distance, 4) the last property is $f_B(x, y) = 0$ *only if* $x = y$. This can be understood by noting that if x and y satisfy $\phi_k(x) = \phi_k(y)$ for all k, then basically since ϕ_k is orthonormal basis then any (L_2 integrable) function f defined over the surface can be written as $f(x) = \sum_{k=1}^{\infty}\langle f, \phi_k\rangle\phi_k(x)$,

therefore

$$f(x) = \sum_{k=1}^{\infty} \langle f, \phi_k \rangle \phi_k(x) = \sum_{k=1}^{\infty} \langle f, \phi_k \rangle \phi_k(y) = f(y),$$

which means there are no functions distinguishing x and y. This necessarily means that $x = y$.

Smoothness and local isometry: The key observation about the biharmonic distance is that since on two dimensional surfaces the eigenvalues of the Laplacian λ_k, $k = 1, 2, ...$, grow like k, that is $\lambda_k \sim k$ (see [Berger 2003], and [H. P. McKean and Singer 1967]) the series $\sum_{k \geq 1} 1/\lambda_k^2$ is converging. This is the reason why the biharmonic distance is continuous everywhere and smooth everywhere except at the source point, where it has only a derivative discontinuity. Note that it has similar behavior to the regular Euclidean distance measured from a point: $d_B(x, y) = \sqrt{(x-y)^t(x-y)}$ which is continuous everywhere and smooth as long as $x \neq y$. In that sense the biharmonic distance has a natural (similar to Euclidean) local behavior.

Timing and computational complexity: Obviously, timing is a crucial aspect of a distance scheme. A good distance scheme with poor timing is hardly practical. Our exact computational scheme of the biharmonic distance, as described in Section 3, is constructed to efficiently solve the all-pairs distance problem in $O(MN)$, where M is the size of the subset on which we compute the distances. This complexity can be understood by noting that each of the M columns of g_d is a solution of the sparse and symmetric linear system of equations Eq. (10) with one row exchanged, where the matrix on the left-hand side is fixed for all j. This kind of linear system can be solved very efficiently by first performing Cholesky factorization of the matrix and then performing two back-substitutions for every given right-hand side vector. As explained in [Botsch et al. 2005] this scheme (both factorizing and back-substituting) is linear in number of vertices and therefore resulting in an $O(N)$ operations for each column evaluation of g_d.

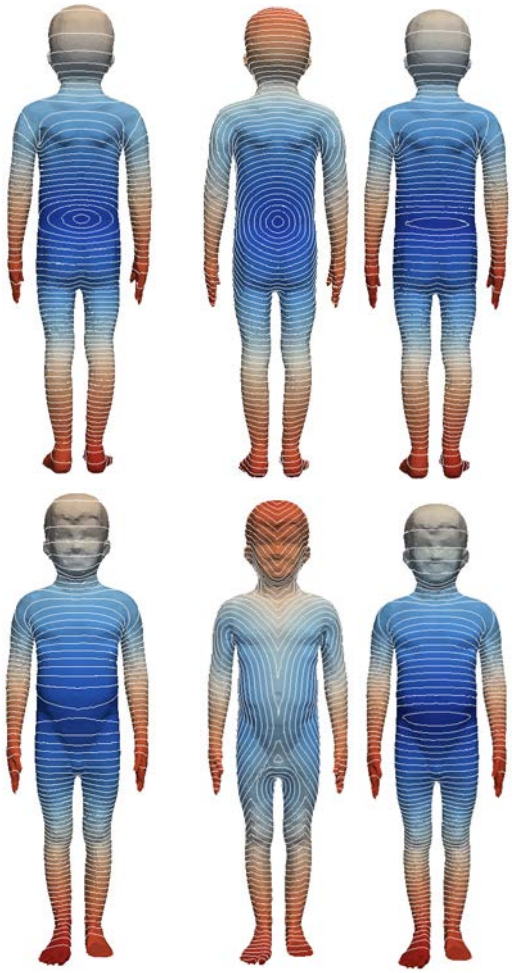
The computational complexity of approximating the biharmonic distance consists of first calculating the first K eigenvectors in $O(KN^{1.5})$, and then approximating every distance between pairs of points can be computed in $O(K)$ operations.

5. RESULTS

In order to investigate the practical properties of the biharmonic distance, we ran a set of experiment using 3D meshes from a variety of object types, including all 400 models of the Watertight Benchmark Set of SHREC 2007 [Giorgi et al. 2007]. In these experiments, our goals are to understand how the biharmonic distance behaves on complex surfaces, to test its empirical sensitivity to mesh perturbations (noise and tessellation), and to make qualitative comparisons to other commonly used distance measures.

5.1 Evaluation

Biharmonic distance examples: Our first set of results provides visualizations of the biharmonic distance computed from a single "source vertex" to all other vertices for a representative set of meshes (Figures 4(a) and 5(a)). In these visualization, distances are color-coded and Gouraud shaded, with darker blue regions indicating smaller distances, darker red regions indicating larger distances,



(a) Biharmonic (this paper) (b) Geodesic (c) Diffusion

Fig. 4: *Visualization of biharmonic (left), geodesic (middle), and diffusion distances (right) for the front and back of a boy. Note that the source vertex is on the back of the boy, but a local minimum of diffusion distance appears on the front.*

and cyan, grey, and pink regions in between. Isocontours are shown as white lines at equally spaced intervals of distance.

As can be seen from these images, our biharmonic distance (left column) is sensitive to both local and global properties of the shape: isocontours are nearly circular in the vicinity of the source vertex, and yet they follow the natural cross-sections of the shape at further distances (e.g., on the arms and legs of the boy). Also, please note that the biharmonic distance is smooth and has no local minima, except at the source point.

Comparison to geodesic distance: Figures 4 and 5 show comparisons of the biharmonic distance (left column) to the geodesic distance (middle column) computed with the exact and approximate algorithm of [Surazhsky et al. 2005]. Clearly, the *local* properties of geodesic distance are desirable: the distance is isotropic and gradually increasing in the neighborhood of the source vertex. However, the geodesic distance is not globally shape-aware, and thus: 1) distances are sensitive to the exact placement of the source, causing diagonal isocontours down the arms of the boy in Figure 4(b); and

2) distances are not smooth, causing cusps and ridges in the distance function, especially on the opposite side of the surface from the source vertex (e.g., on the stomach of the boy). These undesirable global properties of geodesic distance diminish its utility for shape analysis applications.

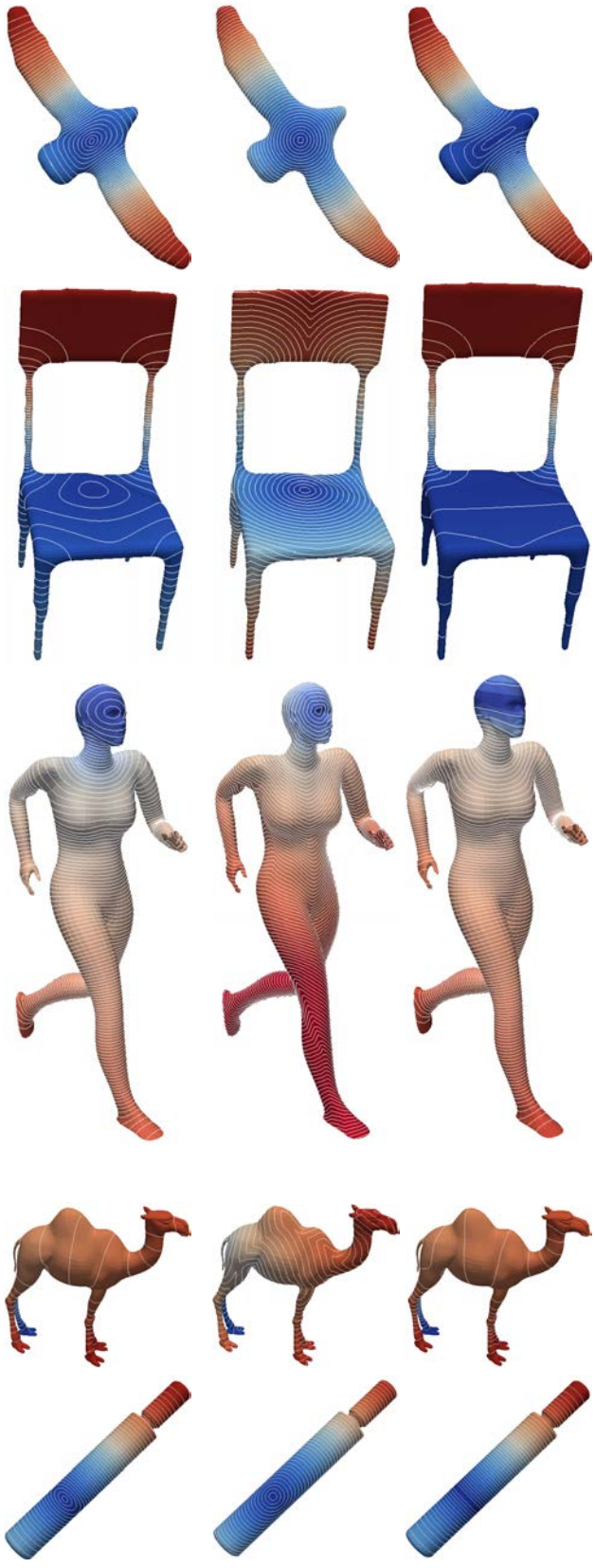
Comparison to diffusion distance: Figures 4 and 5 also show comparisons to the diffusion distance computed with $t = 1$ using the method of [Goes et al. 2008] (right column). In these images, note that the global properties of the diffusion distance qualitatively match those of the biharmonic distance. For example, isocontours are perpendicular to central axes of protrusions far from the source vertex. However, the diffusion distance does not have the nice local properties of the biharmonic distance. Note how isocontours close to the source appear to be elongated ellipses, rather than circles (e.g., on the back of the boy in Figure 4(c)), indicating the undue influence of the global shape on local distances.

Of course, the parameter t provides a trade-off between the effects of local and global shape in the diffusion distance computation (Figure 7). However, it is impossible to get the "best of both worlds" with any single setting for this parameter. If t is small (e.g., 0.125), then the diffusion process runs for only a short time, and the resulting distance is useful locally, but almost constant on for distant pairs of points (note that there are few isocontours in the palm in Figure 7a). If t is large (e.g., 0.25-1.0), then the distance captures global properties of the shape, but fails to capture local properties (note that the isocontours shown in the zoomed view of the finger Figure 7(b) are nearly parallel in the vicinity of the source vertex). In contrast, our biharmonic distance (rightmost image in Figure 7) captures both local and global properties of the shape *without any parameters.*

Sensitivity to noise, tessellation, and deformation: In order to test the robustness of the biharmonic distance, we ran experiments for meshes with different noise, tessellation, and deformations for the same shapes and visually compared the resulting distances. Representative results are shown in Figure 6. The top two images (Figure 6a-b) demonstrate insensitivity to noise – note how the coloring and isocontours of the biharmonic distance are largely unchanged by adding Gaussian noise to the mesh (σ = 400% of the average edge length). The middle row of two images (Figure 6c-d) shows insensitivity to tessellation – as the mesh is simplified from 14K vertices to 4K, the biharmonic distance remains stable (note that the visualization bilinearly interpolates the distance across triangles when rendering colors and positioning isocontours, and so the main differences between the two images are in the interpolation, not the distance function itself). The bottom row shows insensitivity to nearly isometric deformation of the Armadillo (Figure 6e-f).

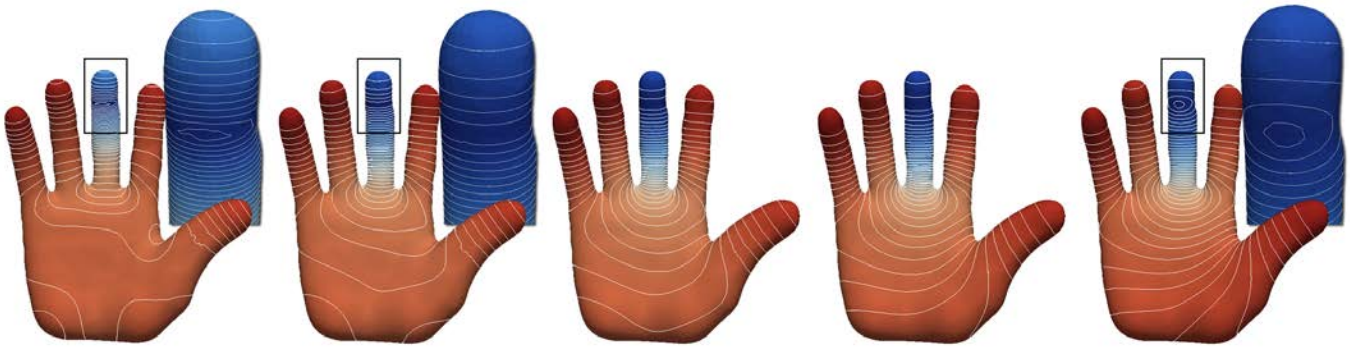
Timing: Finally, we ran experiments to test the empirical compute time (exact and approximate) of the biharmonic distance and compared it to others.

In the first experiment we took meshes with different number of vertices and timed exact computation of the biharmonic distances between all pairs of vertices ($M = N$) using a 2.2Ghz Opteron 875 processor. Looking at the results in Table I, we see that the diffusion distance is relatively fast to compute (37.2 seconds for a mesh with 16K vertices). Our biharmonic distance is not as fast (607.6 seconds), but it is much faster than geodesic distance (1098.5 seconds), which was approximated with Dijkstra's algorithm for this experiment to achieve reasonable compute times. Both theoretically and empirically, our method grows with $O(MN)$, Dijkstra's method



(a) Biharmonic (this paper) (b) Geodesic (c) Diffusion

Fig. 5: *Comparisons of biharmonic distance (left column) to geodesic (middle) and diffusion distances (right).*



(a) Diffusion ($t = 0.125$) (b) Diffusion ($t = 0.25$) (c) Diffusion ($t = 0.5$) (d) Diffusion ($t = 1.0$) (e) Biharmonic (this paper)

Fig. 7: *Comparison of diffusion distance with different parameter values for t with the biharmonic distance (rightmost image).*

grows with $O(MNlogN)$, exact geodesic computation [Surazhsky et al. 2005] grows with $O(MN^2logN)$ in the worst-case and with $O(MN^{1.5}logN)$ on average. Diffusion distance grows with $O(N^{1.5})$. Moreover, since the Biharmonic distance is a smooth function, it can in principle be approximated by sub-sampling of the mesh, while geodesic distances will exhibit artifacts and bad approximation order at points where minimizing geodesics meet.

Mesh $\|V\|$	Biharmonic	Dijkstra	Diffusion (t=0.125)
2K	6.3	14.4	1.2
4.3K	40.3	74.8	4.11
8K	131.6	233.3	8.8
12K	472.7	610.7	13.0
16K	607.6	1098.5	37.2

Table I. : *Time (in seconds) to compute distances for all vertex pairs.*

In the second experiment, we timed the exact computation of the biharmonic distance between all pairs of M vertices (where $M < N$), a scenario common in shape analysis applications (e.g., segmentation, shape matching). Table II shows timing for $M = 2, 100, 400$ and various N. Note that the case $M = 2$ corresponds to computing the distance between two points. Geodesic distance approximated with Dijkstra's algorithm is fastest for small M. Biharmonic distance has an overhead due to Cholesky factorization (see $M = 2$), which is quickly amortized as M increases, and biharmonic becomes the fastest for large M (see 112K mesh at $M = 400$). Diffusion distance computation is dominated by eigendecomposition time, and so it practically does not depend on the size of the subset M.

Lastly, we experimented with the approximation of the biharmonic distance achieved with the first K eigenvectors, as described in section 3. Figure 8 shows the log error $\log\left(\left|d_B(x,y) - \widetilde{d}_B(x,y)\right| / \max_{x,y} d_B(x,y)\right)$ as a function of the number of eigenvectors used $K = 1, 2, ..., 200$. We randomly picked 200K pairs of points of a mesh with $5K$ vertices, and computed this error; the red curve shows the worst-case (maximum) errors achieved for every K, and the blue curve shows the average errors. The computation times for the first $K = 200$ eigenvectors (resulting in the right most errors in the graph) took 12 seconds. The approximated distance between pair of points can be calculated from the first K eigenvectors in $O(K)$ (K is usually a constant, say $K = 150 - 200$). Note that computing the Green's matrix g_d took 66 seconds in this case. The pairwise distance can be computed from g_d in $O(1)$ operations.

5.2 Applications

We investigated two applications of biharmonic distance in geometric mesh processing: function interpolation on surfaces and shape matching.

Function interpolation on surfaces: An application that benefits from intrinsic distances on surfaces is interpolation. Given prescribed values at a set of surface anchor points, the aim of interpolation is to construct a function on the whole surface that attains these values at the corresponding points. The standard Shepard interpolation [Shepard 1968; Gordon and Wixom 1978] procedure solves this problem via weighted averaging: each prescribed value contributes with a weight that is inversely proportional to the distance to the corresponding point.

In our setting, let p_i be the anchor points where the scalar values f_i are prescribed. The Shepard interpolant is defined for any point p on the surface as

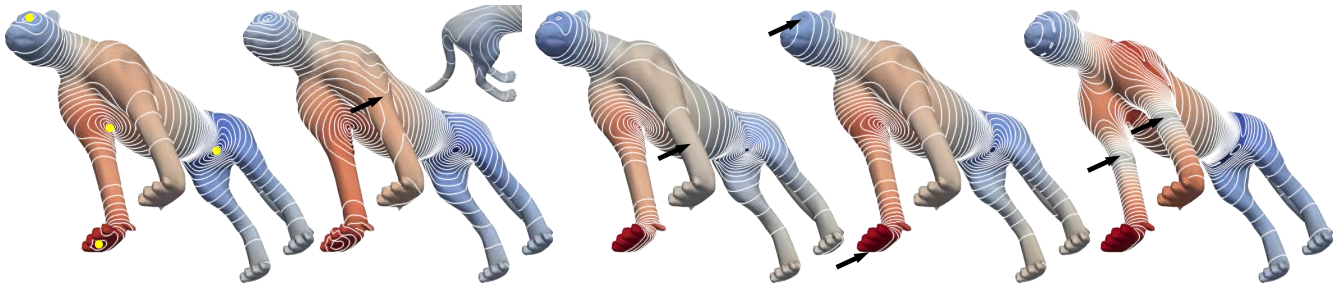
$$f(p) = \frac{\sum_i w_i(p) f_i}{\sum_i w_i(p)},$$

where $w_i(p)$ are the weights given by $w_i(p) = 1/d(p, p_i)$. Here, $d(\cdot,\cdot)$ is any distance measure on the surface. Shepard interpolant has the zeroth order precision meaning that the constant functions are reproduced, i.e. if f_i are all equal, then f is also constant.

We evaluate the effect of different choices of the distance measure $d(\cdot,\cdot)$ on the resulting interpolant. Figure 9 shows the equally spaced isocontours of the interpolant obtained using biharmonic, geodesic, and diffusion distances. Darker blue regions correspond to smaller, and darker red to larger function values. Without any parameter tuning, biharmonic distance (Figure 9(a)) provides a well-behaved interpolant both close to anchor points and far away. Geodesic distance (Figure 9(b)) interpolant displays expected behavior close to the anchors, but away from anchors diagonal isocontours and ridges appear. Experimenting with various settings for the diffusion time t parameter reveals that while acceptable results can be obtained, scrupulous tuning of the parameter is needed. For example when $t = 1/4$ (Figure 9(c)), the resulting interpolant is

Mesh size		Biharmonic			Dijkstra			Diffusion
$\|V\|$	$\|F\|$	M=2	M=100	M=400	M=2	M=100	M=400	M=2,100,400
2K	4K	0.05	0.19	1.30	0.01	0.64	2.65	1.22
4K	9K	0.15	0.62	4.31	0.03	1.62	6.40	4.09
8K	16K	0.27	1.07	8.74	0.06	2.73	10.83	1.38
16K	32K	0.64	2.66	15.53	0.12	6.23	25.67	8.80
56K	112K	2.07	9.74	46.20	0.2	10.45	41.38	74.22
111K	222K	11.18	37.13	185.28	1.05	52.8	200.39	202.20

Table II. : *Time (in seconds) to compute distances between all pairs in a subset of size* $M = 2, 100, 400$.



(a) Biharmonic (b) Geodesic (c) Diffusion ($t = 0.25$) (d) Diffusion ($t = 1.0$) (e) Diffusion ($t = 4.0$)

Fig. 9: *Function values given at anchor points (shown in yellow in (a)) are interpolated using weighted averaging. Biharmonic distance provides a balanced interpolant (a). Ridges and unintuitive isocontours appear when geodesic distance is used (b). Small scale diffusion distance results in an almost constant function in the right leg (c). Medium scale diffusion gives an acceptable result, but is insensitive to the location of the anchor point on the head (d). Large scale diffusion distance leads to spurious local extrema as seen in both of the front legs (e).*

almost constant (no isocontours) on the left front leg. When $t = 1$ (Figure 9(d)) the interpolant is insensitive to the exact location of the anchors. Finally, as we increase the value of the parameter, e.g when $t = 4$ (Figure 9(e)) the interpolant starts having unintuitive extrema, see both of the front legs.

Surface matching: In this application we test how biharmonic distance can improve predicting correspondences between two deformable surfaces $\mathcal{M}, \mathcal{N}$. In particular, given a coarse set of corresponding points, we wish to find more good candidate pairs of points that correspond. One way to do it is in the spirit of the Gromov-Hausdorff techniques [Mémoli and Sapiro 2005; Bronstein et al. 2006]: given a set of corresponding pairs $(x_i, y_i) \in \mathcal{M} \times \mathcal{N}$, $i = 1, .., k$, and a point $x \in \mathcal{M}$, we wish to find good candidates $y \in \mathcal{N}$ that corresponds with it. To this end, we represent the point x using the vector of distances to the known corresponding points on $\mathcal{M}$, that is

$$\widetilde{x} = \frac{1}{\max_i d(x, x_i)} \left(d(x, x_1), d(x, x_2), ..., d(x, x_k) \right),$$

where $d(\cdot, \cdot)$ is some distance function, and the normalization is done to be more robust to scales between $\mathcal{M}, \mathcal{N}$. Similarly for each point $y \in \mathcal{N}$,

$$\widetilde{y} = \frac{1}{\max_i d(y, y_i)} \left(d(y, y_1), d(y, y_2), ..., d(y, y_k) \right).$$

Next, we define a prediction function

$$c(y, x) = \|\widetilde{x} - \widetilde{y}\|_w = \left(\sum_{i=1}^{k} w_i(x) |d(x, x_i) - d(y, y_i)|^2 \right)^{1/2},$$

where $w_i(x) > 0$ can be set so to give higher weight to close correspondences x_i. In our experiment we took $w_i(x) = 1$. Figure 10 shows an experiment with two SCAPE human models [Anguelov et al. 2005]. Each row shows a prediction function for the red sphere (b), based on a set of three pairs of corresponding points (a). In (c) we show the prediction with the geodesic, diffusion, and biharmonic distances. Note that geodesic produces bad predictions in far areas, and diffusion produces unintuitive behavior in non-centered areas like the head on the top row.

6. CONCLUSIONS AND FUTURE WORK

This work presents a novel surface distance, called the biharmonic distance, which is based on the Green's function of the Bi-Laplacian. This distance is a metric, smooth, locally isotropic, globally shape-aware, isometry invariant, insensitive to noise and tessellation, and practical to compute. Moreover, it does not depend on any parameters.

Since measuring distances on surfaces is one of the most fundamental operations on meshes there are many interesting directions for future research. Perhaps the first is to study the applications which might benefit from the new distance. Since it is shape-aware, we expect it to be useful for shape visualization, segmentation, deformation, matching, and retrieval, for example. However, experiments (outside the scope of this paper) are required to verify this conjecture.

Another possible direction is to investigate whether there are other interesting distance measures in the biharmonic's family. That is, taking perhaps the squared biharmonic operator. More generally,

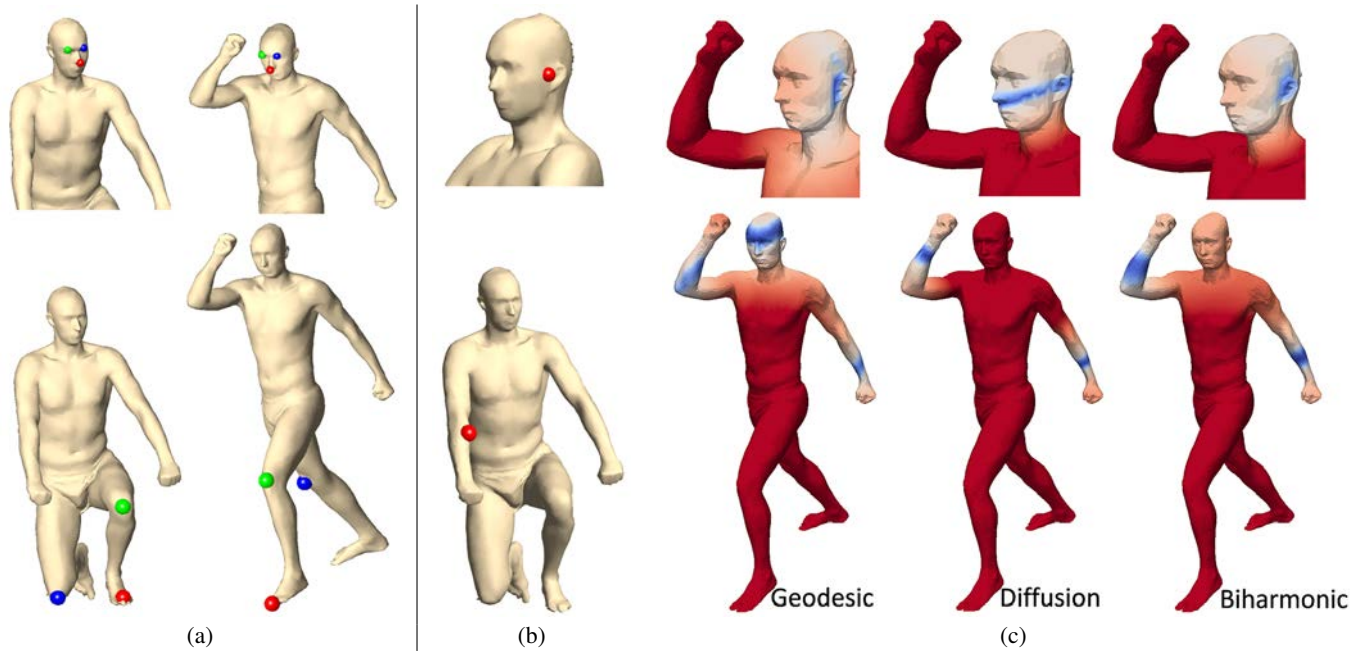


Fig. 10: *Finding candidate correspondences based on coarse prescribed set of correspondences. Given a coarse set of correspondences (a), we compute a prediction function (c) for the corresponding point to the point marked by red sphere at (b). The prediction range from high confidence (dark blue) to low (dark red), and is shown for three different distance functions: geodesic, diffusion, and biharmonic. The top row demonstrates local predictions, while the bottom row depicts extreme extrapolation. Note that geodesic distance produces bad predictions far from the known correspondences (bottom row), and diffusion distance produces unintuitive result for small not-centered areas like the head (top row). The color map is scaled to exaggerate small (blue) values.*

any normalization $1/\lambda_k^\alpha$, $\alpha > 1$ of the eigenvalues will produce a new and maybe useful distance measure. Mapping out this and related families of distance measures and analyzing their utilities with respect to different applications is a large topic for future work.

ACKNOWLEDGMENTS

We thank AIM@SHAPE for data and NSF (CNFS-0406415, IIS-0612231, CCF-0937139, CNS-0831374, and CCF-0702672), Google, and the Rothschild Foundation for funding.

Appendix

This appendix is devoted to the proofs of the statements that lead to the exact computation procedure described in the text for the discrete Green's function g_d of the bi-Laplacian. Before embarking, we remind that the kernel of the conformal Laplacian L_c consists of constant vectors (all entries are equal), and the image of L_c consists of all vectors whose entries add up to zero.

THEOREM 1. *The kernels of the matrices $L_cA^{-1}L_c$ and L_c are equal and consist of constant vectors: $Ker(L_cA^{-1}L_c) = Ker(L_c) = \{c\mathbf{1} \in \mathbb{R}^{N\times 1}\}$.*

Proof: Let x be in the kernel of $L_cA^{-1}L_c$, that is $L_cA^{-1}L_cx = 0$. Therefore $A^{-1}L_cx$ is in the kernel of L_c. Since the kernel of L_c consists of only the constant vectors we get $A^{-1}L_cx = c\mathbf{1}$, for some number c. Multiplying both sides with A we get $L_cx = cA\mathbf{1}$. The image of matrix L_c consists of vectors with entries that add up to zero; since $A\mathbf{1} = (A_1, ..., A_N)^t$ is a vector whose entries add up to a positive number (A_i are the areas) except when $c = 0$, it follows that necessarily $c = 0$, and x must be in the kernel of L_c.■

THEOREM 2. *The product $L_cA^{-1}L_cg_d = J$, where $g_d = (L_cA^{-1}L_c)^+$ is the pseudo-inverse of $L_cA^{-1}L_c$, equals to $J = I - \frac{1}{N}\mathbf{1}\mathbf{1}^t$ with $\mathbf{1} \in \mathbb{R}^{N\times 1}$ being a column vector of all ones.*

Before proceeding, we remind the definition of the pseudo-inverse as it applies to our matrix. Write the diagonalization of the (symmetric) matrix $L_cA^{-1}L_c = \sum_{k=1}^N \mu_k\psi_k\psi_k^t$, where $\psi_k \in \mathbb{R}^N, k = 1, 2, ..., N$ are the orthonormal eigenvectors and μ_i are the corresponding eigenvalues of $L_cA^{-1}L_c$. Now let $\psi_1 = (1/\sqrt{N}, ..., 1/\sqrt{N})$ be the constant eigenvector spanning the eigenspace of the zero eigenvalue (same as the kernel of the matrix). As shown in Theorem 1 this eigenspace is one dimensional, and so ψ_1 is the only eigenvector with corresponding eigenvalue $\mu_1 = 0$. As a result, the *pseudo-inverse* in this case is given by

$$g_d = \sum_{k=2}^{N} \frac{1}{\mu_k}\psi_k\psi_k^t.$$

Note that this pseudo-inverse is essentially the inverse of $L_cA^{-1}L_c$ over the subspace orthogonal to ψ_1.

Proof: Remembering that $\mu_1 = 0$ and multiplying $L_cA^{-1}L_c = \sum_{k=2}^N \mu_k\psi_k\psi_k^t$ with its pseudo-inverse $g_d = \sum_{\ell=2}^N \frac{1}{\mu_\ell}\psi_\ell\psi_\ell^t$ we

Fig. 6: *Insensitivity of the biharmonic distance to noise, tessellation, and nearly isometric deformation.*

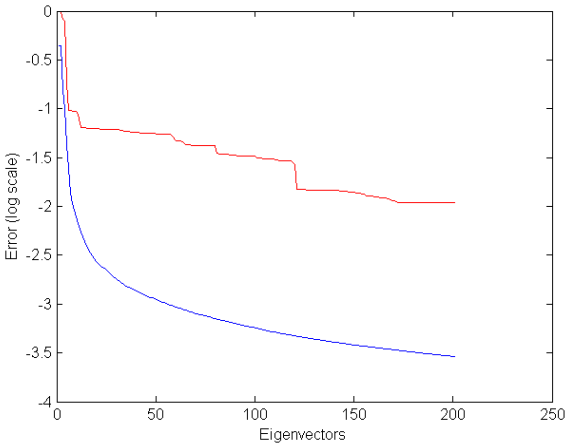


Fig. 8: *Error in approximation of the biharmonic distance using the first $K = 1, 2, .., 200$ eigenvectors is shown in log scale. The red curve demonstrated the maximum error and the blue curve the average error in approximating 200K pairwise distances randomly picked from a mesh with 5K vertices.*

get

$$L_c A^{-1} L_c g_d = \sum_{k=2,\ell=2}^{N} \frac{\mu_k}{\mu_\ell} \psi_k \psi_k^t \psi_\ell \psi_\ell^t = \sum_{k=2,\ell=2}^{N} \frac{\mu_k}{\mu_\ell} \psi_k \delta_{k,\ell} \psi_\ell^t =$$

$$\sum_{k=2}^{N} \psi_k \psi_k^t = \sum_{k=1}^{N} \psi_k \psi_k^t - \psi_1 \psi_1^t = I - \frac{1}{N} \mathbf{1}\mathbf{1}^t,$$

where $\delta_{k,\ell}$ equals one if $k = \ell$ and zero otherwise.■

THEOREM 3. *The matrix $g_d = (L_c A^{-1} L_c)^+$ satisfies the relation Eq. (8) for any f in the image of L_d^2.*

Note that it is the form of Eq. (8) that forces f to reside in the image of L_d^2.

Proof: After inserting $L_d = A^{-1} L_c$, and multiplying both sides of Eq. (8) by A, we obtain

$$L_c A^{-1} L_c g_d A f = A f,$$

and since every vector in the image of L_c has sum of entries equal to zero, so does Af. Thus, we need to show that g_d satisfies

$$L_c A^{-1} L_c g_d \tilde{f} = \tilde{f}$$

for any vector $\tilde{f}$ with entries that sum up to zero. In essence, we need to invert the singular matrix $L_c A^{-1} L_c$ on the orthogonal complement of its kernel, namely constant vectors. This is what the pseudo-inverse g_d achieves. In fact, by the previous theorem, the left hand-side of this equation is equal to $J\tilde{f}$. Now, we need to show that $J\tilde{f} = \tilde{f}$, which easily follows because $\mathbf{1}^t \tilde{f} = 0$ as the entries of $\tilde{f}$ add up to zero. ■

THEOREM 4. *The j^{th} column of matrix g_d is the unique solution of linear equation $L_c A^{-1} L_c y = J_j$ such that $\mathbf{1}^t y = 0$. If x is any solution of the linear equation $L_c A^{-1} L_c x = J_j$, then $y = x - \frac{\mathbf{1}^t x}{\mathbf{1}^t \mathbf{1}} \mathbf{1}$.*

Proof: While the second part is obvious, the first fact follows from the explicit expression for the pseudo-inverse in terms of the eigendecomposition. Indeed, by symmetry the eigenvectors constitute an orthogonal set. In particular, every eigenvector entering the formula of the pseudo-inverse is orthogonal to the constant eigenvector, and so $\mathbf{1}^t\psi_k = 0$. We get

$$\mathbf{1}^t g_d = \sum_{k=2}^{N} \frac{1}{\mu_k} \mathbf{1}^t \psi_k \psi_k^t = \mathbf{0},$$

which means that every column of g_d has sum of entries equal to zero. The fact that each column satisfies the specified linear system was proven in Theorem 2, and uniqueness of zero-sum solution follows from Theorem 1. ■

REFERENCES

ANGUELOV, D., SRINIVASAN, P., KOLLER, D., THRUN, S., RODGERS, J., AND DAVIS, J. 2005. Scape: shape completion and animation of people. *ACM Trans. Graph. 24,* 3, 408–416.

BERGER, M. 2003. *A panoramic view of Riemannian geometry*. Springer-Verlag, Berlin.

BOTSCH, M., BOMMES, D., AND KOBBELT, L. 2005. Efficient linear system solvers for mesh processing. 62–83.

BRONSTEIN, A. M., BRONSTEIN, M. M., AND KIMMEL, R. 2006. Efficient computation of isometry-invariant distances between surfaces. *SIAM J. Sci. Comput. 28,* 5, 1812–1836.

COIFMAN, R. R. AND LAFON, S. 2006. Diffusion maps. *Applied and Computational Harmonic Analysis 21,* 1 (July), 5–30.

FOUSS, F., PIROTTE, A., MICHEL RENDERS, J., AND SAERENS, M. 2006. Random-walk computation of similarities between nodes of a graph, with application to collaborative recommendation. *IEEE Transactions on Knowledge and Data Engineering 19*, 2007.

GIORGI, D., BIASOTTI, S., AND PARABOSCHI, L. 2007. SHREC:SHape REtrieval Contest: Watertight models track, http://watertight.ge.imati.cnr.it/.

GOES, F. D., GOLDENSTEIN, S., AND VELHO, L. 2008. A hierarchical segmentation of articulated bodies. *Computer Graphics Forum (Special Issue of Symposium on Geometry Processing) 27,* 5, 1349–1356.

GORDON, W. J. AND WIXOM, J. A. 1978. Shepard's method of "metric interpolation" to bivariate and multivariate interpolation. *Mathematics of Computation 32,* 141 (Jan), 253–264.

GRINSPUN, E., GINGOLD, Y., REISMAN, J., AND ZORIN, D. 2006. Computing discrete shape operators on general meshes [Eurographics 2006 Best Paper, 3rd Place]. *Eurographics (Computer Graphics Forum) 25,* 3, 547–556.

H. P. MCKEAN, J. AND SINGER, I. M. 1967. Curvature and the eigenvalues of the laplacian. *J. Differential Geom. 1*.

JOHN, F. 1986. *Partial Differential Equations*. Springer-Verlag.

MÉMOLI, F. AND SAPIRO, G. 2005. A theoretical and computational framework for isometry invariant recognition of point cloud data. *Found. Comput. Math. 5,* 3, 313–347.

MEYER, M., DESBRUN, M., SCHRDER, P., AND BARR, A. H. 2002. Discrete differential-geometry operators for triangulated 2-manifolds.

PAPADIMITRIOU, C. 1985. An algorithm for shortest-path motion in three dimensions. *IPL 20*, 259–163.

PINKALL, U. AND POLTHIER, K. 1993. Computing discrete minimal surfaces and their conjugates. *Experimental Mathematics 2*, 15–36.

QIU, H. AND HANCOCK, E. R. 2007. Clustering and embedding using commute times. *IEEE Trans. Pattern Anal. Mach. Intell. 29,* 11, 1873–1890.

SHEPARD, D. 1968. A two-dimensional interpolation function for irregularly-spaced data. In *Proceedings of the 1968 23rd ACM national conference*. ACM, New York, NY, USA, 517–524.

SURAZHSKY, V., SURAZHSKY, T., KIRSANOV, D., GORTLER, S. J., AND HOPPE, H. 2005. Fast exact and approximate geodesics on meshes. *ACM Trans. Graph. 24,* 3, 553–560.

YEN, L., FOUSS, F., DECAESTECKER, C., FRANCQ, P., AND SAERENS, M. 2007. Graph nodes clustering based on the commute-time kernel. In *In Proceedings of the 11th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD 2007). Lecture notes in Computer Science, LNCS*.