2020

Heat Kernel Voting with Geometric Invariants

Alexander Harr
Minnesota State University, Mankato

Heat Kernel Voting with Geometric Invariants

By Alexander Harr

Supervised by Dr. Ke Zhu

A thesis submitted in partial fulfillment of the requirements for the degree of Master of Arts at Mankato State University, Mankato

Mankato State University, Mankato

Mankato, Minnesota

May 2020

May 8, 2020

Heat Kernel Voting with Geometric Invariants

Alexander Harr

This thesis has been examined and approved by the following members of the student's committee:

Dr. Ke Zhu

(Supervisor)

Dr. Wook Kim

(Committee Member)

Dr. Brandon Rowekamp

(Committee Member)

Acknowledgements

This has been difficult, and anything worthwhile here is due entirely to the help of others.

I would first like to thank Dr. Ke Zhu for sharing his knowledge, and insights into the world of geometry. I am absolutely grateful for the time, effort and patience that Dr. Zhu has had with me during the writing of this paper.

I would also like to thank Dr. Kim and Dr. Rowekamp for their participation on my committee, and their taking the time to review and evaluate my work.

Special thanks goes out to all of my instructors and professors who have shared their knowledge and given their students some direction in the pursuit of the abstract truth, whatever that is.

I would like to thank all those close to me just for being them. In particular, I would like to thank my mother, Chad, Natasha, and the rest. Thanks to my friends Adam, Edwin, Brian, Kyle, Fidel and many others for the good company and conversation. I ask the forgiveness of anyone who I have forgotten to mention by name, though if you know me, you know that I am writing this at the extreme last minute. You also know that I am thankful.

Contents

Heat Kernel Voting with Geometric Invariants

Alexander Harr

A thesis submitted in partial fulfillment of the requirements for the degree of Master of Arts in Mathematics

Minnesota State University, Mankato
Mankato, Minnesota

May, 2020

Abstract

Here we provide a method for comparing geometric objects. Two objects of interest are embedded into an infinite dimensional Hilbert space using their Laplacian eigenvalues and eigenfunctions, truncated to a finite dimensional Euclidean space, where correspondences between the objects are searched for and voted on. To simplify correspondence finding, we propose using several geometric invariants to reduce the necessary computations. This method improves on voting methods by identifying isometric regions including shapes of genus greater than 0 and dimension greater than 3, as well as almost retaining isometry.

1 Introduction and Previous Work

The use of geometric properties in the study of shapes has been of interest for a long time. More recently, the field of Riemannian geometry has been used in a computational aspect in the study of shapes, networks and data. Several algorithms, such as ISOMAP, Local Linear Embedding, Eigenmap and many others have applied geometric tools to machine learning and nonlinear dimensionality reduction (NLDR). The field of graphics processing has also seen a great deal of interest, and has found the study of 2D surfaces very helpful.

Of particular interest to us here is the voting approach to identify correspondences on surfaces, which is to say maps between regions that preserve the metric as much as possible. The Möbius Voting algorithm [LF 09] provides a way to compare regions of 2D surfaces of genus 0 in three dimensions that are intrinsically similar, thought possibly extrinsically very different. We are interested here in determining which components of this algorithm keep the problem a three dimensional one. After this, we will ask the question of how this method can be generalized to shapes of higher genus and higher dimension. To that end, our goal here is to develop a voting algorithm for higher than 3 dimensions based on an almost isometric heat kernel embedding [BBG 94].

1.1 Some Background and Use of Shape Comparison

Many of the uses of shape comparison in 3 dimensions are obvious, as most physical objects that we deal with are 3D. Less clear are the applications of shape comparison in higher dimensions. We'll go over a few places where this may be of interest.

1.1.1 The Differential Geometry Perspective

This perspective looks at such matters as Gromov-Hausdorff and spectral distance estimates, which measure isometry, as well as Yamabe flow, which preserves conformal structures, and anisotropic indicators, which measure conformality.

[Chan 14] provides a framework for measuring deviations from conformality in higher dimensions. For 2 dimensional surfaces, the Beltrami coefficient serves this role. For higher dimensions, an anisotropic indicator is defined, giving a method to measure conformal deformations between two objects.

The Yamabe flow is a geometric flow on a compact manifold, that if it converges, does so to to a metric of constant scalar curvature. The metric evolving from the Yamabe flow is conformally equivalent to the initial metric.

1.1.2 3D Graphics

The techniques described here, while formulated trying to solve the problems associated with high dimensions, can certainly be used on 3D problems. Certainly, the Möbius Voting approach is a very good algorithm, but heat kernel voting has its own advantages stemming from its emphasis on isometry rather than conformality.

Additionally, the heat kernel method is capable of handling surfaces of genus greater than 0.

In the field of 3D computer graphics, our process is variously referred to as shape comparison, tracking, alignment, and registration, all having similar meanings. The 3D objects of interest in this field are dealt with from the perspective of Riemannian geometry as 2 dimensional surfaces embedded in 3 dimensions. Here, the Riemann Mapping Theorem ensures that any genus 0 closed 2D surface can be mapped to the unit sphere S^2 or the unit disk D^2 conformally. This, in addition to the closed form Möbius transformation for point alignment has created a vibrant body of work on conformal geometry for 3D surfaces. While the loss of some computational tools in higher genus and higher dimensions is ultimately what pushes us toward the isometric approach, we maintain that the isometry approach can still yield relevant information.

1.2 Möbius Voting-Shape Registration in Three Dimensions

Möbius Voting [LF 09] handles the problem of comparing two genus 0 metric surfaces in a computationally simple way. Their solution is to conformally embed the two meshes into S^2 or $\mathbb{C} \cup \{\infty\}$ (uniformization). From here, random sets of 3 points on each mesh representation of the surfaces are matched via Möbius transformations, then the Möbius transform defined by this matching is used to map the remaining points. The correspondence error for the remaining points is measured and stored for comparison with different samples in the process of voting.

The fact that Möbius voting is the primary motivator for this one, and that we are

following a very similar process, we'll go into some details. Specifically, the process is carried out as follows.

1.2.1 Sampling and Embedding the Meshes

The process begins with two discrete meshes in 3D. To reduce computation, sample vertices are taken first from local maxima of Gauss curvature, then from the most isolated point from all previous points until the desired number of samples have been taken.

Now the meshes are mapped conformally to the unit sphere using the discrete harmonic map and its conjugate given in [PP]. The specific manner of the embedding is to create a midedge mesh from the original. The midpoints of the original edges become the vertices of the mid-edge mesh, so that a vertex v_i is associated with each edge. Within each face f_{ijk} in the original, a midedge face is constructed inside with the same orientation. So for a face f_{ijk} in the original mesh, mid-edge construction is as follows

$$v_i \to V_q \qquad v_j \to V_r \qquad v_k \to V_s$$

$$f_{ijk} \to F_{qrs}$$

$$e_{ij} \to E_{qr} \qquad e_{jk} \to E_{rs} \qquad e_{ki} \to E_{sq}$$

Constructing the embedding function requires these notions to be defined. How-

ever, the embedding will only take the vertices. So it is helpful to have the explicit formula for the midedge vertex

$$V_r = \frac{v_i + v_j}{2}$$

Now this is used in the computation of the harmonic function. The function

$$u(v_i) = \sum_j u_j \phi(v_i)$$

is discrete harmonic if it satisfies

$$\sum_{j \in \mathcal{N}(i)} (u(v_i) - u(v_j))(\cot \alpha_{ij} + \cot \beta_{ij}) = 0 \quad \forall v_i \in V$$

Here, $u_i = u(v_i)$, $\phi_i(v_j)$ is the indicator function; $\phi_i(v_j) = 1$ if $i = j$ and 0 otherwise. α_{ij} and β_{ij} are the supporting angles for the edge e_{ij}. The function u is initialized for two vertices $u(v_i) = -1$, $u(v_j) = 1$, and the remaining values are solved for.

The conjugate function, u^* takes mid-edge vertices V_i, and is defined in terms of the harmonic function u

$$u^*_q - u^*_r = \frac{1}{2}((u(v_i) - u(v_j)) \cot(\theta_k) + (u(v_k) - u(v_j)) \cot \theta_i)$$

θ_k is the angle of vertex v_k. The embedding is then

$$\Phi(V_r) = \frac{u(v_i + v_j)}{2} + iu^*(V_r)$$

1.2.2 Aligning

After performing the process described previously on each input mesh, there are two meshes embedded conformally into $\hat{\mathbb{C}} = S^2$. The process of voting requires defining maps from one mesh to another. From each mesh, 3 points should be sampled. The goal is not simply to align these points, rather, we want an explicit Möbius map between the spheres that aligns these points. A Möbius transformation is defined in [Schwertfeger 1962]

Definition 1 (Möbius Transformation)**.** *Given $z \in \hat{\mathbb{C}}$, a Möbius transformation is given*

$$M(z) = \frac{az + b}{cz + d}$$

where a, b, c, d are from $\mathbb{C}$ and $ad - bc \neq 0$.

In matrix language, M acts on each $z \in \hat{\mathbb{C}}$ by

$$M(z) = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} z \\ 1 \end{bmatrix}$$

here considering homogeneous coordinates on $CP^1 \cong \hat{\mathbb{C}}$.

A useful tool here is the cross ratio

Definition 2 (Cross Ratio)**.** *Given four points $z_1, z_2, z_3, z_4 \in \mathbb{C}$, the cross ratio is defined*

$$(z_1, z_2; z_3, z_4) = \frac{(z_3 - z_1)(z_4 - z_2)}{(z_3 - z_2)(z_4 - z_1)}$$

Then the following will help specify point alignment:

Theorem 1 (Cross-Ratio Invariance [Möbius 1855]). *Given four points $z_1, z_2, z_3, z_4 \in \hat{\mathbb{C}}$, and a Möbius transformation M, and $w_j = M(z_j)$. Then the cross ratio is invariant under the Möbius transform M, i.e.*

$$(w_1, w_2; w_3, w_4) = (z_1, z_2; z_3, z_4)$$

Theorem 2 (3-Transitivity [Schwertfeger 1962]). *Given three unique points z_1, z_2, z_3 in $\hat{\mathbb{C}}$ and w_1, w_2, w_3 be three different unique points. The Möbius transformation M providing*

$$w_1 = M(z_1), \quad w_2 = M(z_2), \quad w_3 = M(z_3)$$

is uniquely determined. M is found explicitly be solving the cross ratio problem

$$(w, w_1; w_2, w_3) = (z, z_1; z_2, z_3)$$

for w.

In general, this gives the explicit formulation:

$$M = \begin{bmatrix} a & b \\ c & d \end{bmatrix} = \begin{bmatrix} w_2 - w_3 & w_1 w_3 - w_1 w_2 \\ w_2 - w_1 & w_1 w_3 - w_3 w_2 \end{bmatrix}^{-1} \begin{bmatrix} z_2 - z_3 & z_1 z_3 - z_1 z_2 \\ z_2 - z_1 & z_1 z_3 - z_3 z_2 \end{bmatrix}$$

In this way, w_1, w_2, w_3 and z_1, z_2, z_3 are mapped to $0, 1$, and ∞ for comparison (keeping in mind that we have embedded our meshes in the extended complex plane).

Clearly, these points match. Next, the rest of the sampled points σ_1 and σ_2 are mapped by M to this space.

1.2.3 Voting

Having aligned three points, it is time to 'vote' on the remaining samples. The process is given by [LF 09]. Our first task is to look for correspondences between points. To do so, we do the following:

1. Some small radius is chosen.

2. The nearest point correspondences are found. If the radius about a point is empty, we consider no correspondence found.

3. A threshold K should be chosen for the desired number of correspondences, if the correspondences found by a map M does not meet this threshold, no vote is taken. Otherwise, a vote is taken and the matrix needs to be updated.

Caution is necessary when measuring the deformation. The embedding is conformal, and does not preserve isometry in general. Point correspondences may be close, but carry isometric distortion. A first proposal for measuring the error is given, taking into account the conformal factor.

$$E(c) = \sum_k d_S(z_k, c(z_k)) \int_{\Phi(\Omega_k)} \kappa_1 dxdy$$

Here, $d_S(z_k, M(z_k)) = \frac{|z_k - M(z_k)|}{|1+\bar{z}_k M(z_k)|}$ is the distance on S^2, and κ_1 is the conformal

factor for Φ_1. Ω_k represents a cell in $\mathcal{M}_1$, and Σ_k is a cell in a partition of the surface $\mathcal{M}_1 = \cup_k \Sigma_k$ where each Σ_k contains a single point. $c(z_k)$ is correspondence with z_k, two mutually closest points from different meshes. Then

$$area(\Omega_k) = \int_{\Phi_1(\Omega_k)} \lambda dx dy$$

However, because points were sampled uniformly, the area of each region Ω_K is *roughly* the same. So the error formula has the simpler form

$$E(c) = \sum_l d_S(z_k, c(z_k))$$

This quantity is used for voting in the following way. The process begins with a correspondence matrix C with all entries initialized to 0. Some small e is given. Each time a map M is found and the number of correspondences meets the threshold K, C is updated according to the rule

$$[c_{xy}] \leftarrow [c_{xy}] + \frac{1}{e + E(M)/n}$$

for each found correspondence between x and y. After a number of iterations on M, the matrix is normalized so that each entry c_{xy} represents the confidence that two points x and y have correspondence. Lower amounts of global distortion provide stronger votes, while high amounts of global distortion provide weak votes.

1.3 Limitations of Möbius Voting

Attempting to apply this algorithm to higher dimensions fails for two reasons.

1. Pinkall and Poltier's conjugate harmonic functions, used for uniformization, are defined for 2 dimensional meshes. Generally speaking, the uniformization step is based on the Riemann mapping theorem, which does not generalize to surfaces of genus greater than 0, or manifolds higher than 2 dimensions.

2. Möbius transformations are not point-transitive for more than three points. Resolving this then, is to find a computationally tractable manner of shape comparison that does not rely on these particular tools.

This algorithm also works only for surfaces of genus 0. In our approach, we will develop an method that can compare surfaces of any genus. It is also worth considering the distortion of the metric in conformal transformations. While this nonrigid property has its advantages for shapes with large deviation, the metric still contains useful information. Our approach will still use conformality, but we also turn an eye more toward almost isometry as a tool.

1.4 Heat Kernel Voting

In the interest of bypassing the limitations described previously, we propose a method of different flavor than Möbius voting. We have two different classes of shape that we could work with.

1. n dimensional Riemannian manifolds embedded in $\mathbb{R}^m$, $(\mathcal{M}, g) \subset \mathbb{R}^m$. We will often refer to this as the smooth case.

2. Graphs embedded in $\mathbb{R}^m$, $G = \{V, E\}$. Here V is the vertex set and E is the edge set. In this situation, G is typically weighted. In practice, the edge weights we encounter are usually the Euclidean distance. From here, we use the heat kernel to transform the weights, a process which will be described in Section 1.4.2. We will often refer to this as the discrete case.

2.5 One other consideration must be made, given the growing interest in using geometric properties in data analysis. Provided the input data is a point cloud, ie a set of points $p = \{p_1, \dots p_n\} \subset \mathbb{R}^n$, no edges will be defined. To consider either method, a graph or manifold structure is required, so provisions must be made. If the data has no edges [BN 02] has proposed either connecting points within some ϵ ball, or using a K-nearest neighbor algorithm to connect points. At the point when the points have some edge structure, this is treated as a discrete graph.

The methods for smooth manifolds and discrete graphs are parallel; occasionally the methods will look fairly different, but the broad outline of the methods are the same.

1.4.1 Embedding a Smooth Manifold

The heat kernel embedding proposed by [BBG 94] approaches an isometry into l^2 as the heat flow time t goes to 0. We'll make use of this almost isometry and expand our view into isometric transformations. We'll begin by defining a few things.

The space L^2 is the space of square-integrable functions

$$\int_X |f(x)|^2 dx < \infty$$

with inner product $f, g \in L^2$ defined,

$$\langle f, g \rangle = \int_X f(x)\overline{g(x)}d\mu(x)$$

It will be helpful to think about the space of square-summable real sequences in l^2

$$\{a_i\}_{i\geq 1} \qquad \text{so that } \sum_{i=1}^{\infty} |a_i|^2 < \infty$$

considering the Fourier expansion of functions in L^2. The heat kernel embedding embeds a manifold into l^2, hence our interest.

This extends the concept of Euclidean distance for an infinite dimensional space in the following way. The distance in $\mathbb{R}^N$ is given by $\sqrt{x_1^2 + x_2^2 + \cdots + x_N^2}$. It's natural to think about this process extending to the infinite dimensional vector space by considering $N \to \infty$. In this case, we should only consider sequences that converge.

The Laplace operator (Laplacian), Δ is of central importance for our goals here.

The familiar Laplacian of a real valued function f is given

$$\Delta f = \nabla \cdot \nabla f = \left(\frac{\partial^2 f}{\partial x_1^2} + \frac{\partial^2 f}{\partial x_2^2} + \cdots + \frac{\partial^2 f}{\partial x_N^2} \right)$$

Here ∇ gives the gradient and $\cdot$ is the dot product. For this reason, the Laplacian is often described as the divergence of the gradient.

$$\Delta f = \text{div } \nabla f$$

This concept is the same for the Laplacian on a manifold. However, the coordinate representation looks quite a bit different.

Definition 3 (Smooth Laplacian on a Manifold)**.** *Given an n dimensional Riemannian manifold (M^n, g), the Laplacian is defined*

$$\Delta_g = -div_g \cdot \nabla_g$$

Where ∇_g is the gradient

$$\nabla_g \varphi = \sum_{i,j=1}^{n} g^{ij} \frac{\partial \varphi}{\partial x_i} \frac{\partial}{\partial x_j}$$

and div_g is the divergence at the point $X = \sum_{i=1}^{n} b_i \frac{\partial}{\partial x_i}$

$$div_g X = \frac{a}{\sqrt{|\det g|}} \sum_{i=1}^{n} \frac{\partial}{\partial x_i} (b_i \sqrt{|\det g|})$$

So the Laplacian is given with local coordinates

$$\Delta_g = -\frac{1}{\sqrt{|\det g|}} \sum_{ij=1}^{n} \frac{\partial}{\partial x_i} \left(g^{ij} \sqrt{|\det g|} \frac{\partial}{\partial x_j} \right)$$

Henceforth, when we say "Laplacian", we will be specifically referring to the Laplace operator on Riemannian manifolds.

The eigenvalues of the Laplacian are the numbers λ that satisfy the eigenvalue equation

$$\Delta\phi = \lambda\phi$$

If λ_j is an eigenvalue then ϕ_j is it's corresponding eigenfunction. With this, we cite the following

Theorem 3 (Sturm-Liouville Decomposition [Chavel 84])**.** *For a compact manifold $\mathcal{M}$, there exists a complete orthonormal basis $\{\phi_1, \phi_2, \phi_3, \dots\}$ of $L^2(M)$ consisting of eigenfunctions of Δ with ϕ_j having eigenvalue λ_j satisfying*

$$0 < \lambda_1 \leq \lambda_2 \leq \ldots \uparrow +\infty$$

In particular, each eigenvalue has finite multiplicity.

We begin with two n dimensional Riemannian manifolds $(\mathcal{M}_1, g_1)$ and $(\mathcal{M}_2, g_2)$. We want to embed our objects in a more manageable space. Recall that for a 2 dimensional, genus 0 manifold, the Riemann mapping theorem posits the existence of a conformal map to the sphere or unit disk. For higher dimensional Riemannian man-

ifolds, we can use heat kernel methods to approximate an isometric embedding into a canonical target for each mesh. Our approach is to use the heat kernel embedding to embed each manifold into l^2.

A selection of eigenfunctions $\{\phi_j^a\}_{j\geq 1}$ form an orthonormal basis for the Hilbert space l^2. We consider the spectrum, where $0 = \lambda_0 \leq \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_i \leq \ldots$. Then [BBG 94] gives the embedding

Definition 4 (Heat Kernel Embedding [BBG 94])**.** *Given an n-dimensional closed Riemannian manifold $(\mathcal{M}, g)$ and an orthonormal basis of real eigenfunctions of the Laplacian Δ_g, define the family of maps*

$$\Phi_t^a : M \to l^2 \tag{1}$$

$$x \to \sqrt{2}(4\pi)^{n/4} t^{(n+2)/4} \{e^{-\lambda_j t/2} \phi_j^a(x)\}_{j\geq 1} \tag{2}$$

The heat kernel embedding embeds a manifold into an infinite dimensional Hilbert space l^2, at which point we truncate the embedding to $\mathbb{R}^N$ or S^{N-1}, for a suitable N. The driving factor in our use of this embedding is due to the following theorem.

Theorem 4 ([BBG 94]: Theorem 5)**.** *Fix an n-dimensional closed Riemannian manifold $(\mathcal{M}, g)$ and an orthonormal basis of real eigenfunctions of its Laplacian. Let g_{can} be the Euclidean scalar product on l^2.*

1. *For all positive t, the map Φ_t^a is an embedding of M into l^2.*

2. The pulled-back metric $(\Phi_t^a)^* g$

$$(\Phi_t^a)^* g_{can} = g + \frac{t}{3}\left(\frac{1}{2} Scal_g \cdot g - Ric_g\right)\mathcal{O}(t^2)$$

When $t \to 0^+$.

Here Scal is the scalar curvature and Ric is the Ricci curvature tensor.

The key factor is that the heat kernel embedding approaches isometric as $t \to 0$. For this reason, we will tailor our approach to preserve isometry as much as possible.

In general, computing an l^2 basis of eigenfunctions is unfeasible, and it will be necessary to work with a finite basis of N eigenfunctions. It is possible to truncate the embedding to get within some ϵ to an isometry. [Portegies 13] holds the promise of working with in an almost isometric embedding into a finite dimensional space in the Truncation Lemma (Lemma 1, Section 1.4.3).

1.4.2 Eigenmap Method for Embedding a Graph

The exact method of heat kernel embedding will differ from the continuous case slightly in the discrete case . The embedding uses the eigenfunctions of the Laplacian, similar to the smooth case. It is the character of the Laplacian that varies. The continuous uses the Laplace-Beltrami operator; meshes will use the graph Laplacian.

In the case of a graph as a manifold, there are several formulations for the graph or mesh Laplacian. The eigenmap method for embedding is for weighted graphs, so we'll consider our graph having edge weights, although we have the implicit assumption

that the location of points in space is important. For now, we'll give the Laplacian for a graph with general edge weight w. Later, we will ammend this to refer specifically to a Gaussian edge weight.

Let $G = (V, w)$ be a graph, where V is the vertex set, or nodes, and has size $|V| = K$ and w is the set of edge weights. Specifically, if x and y are vertices, the weight of the edge between them is denoted w_{xy}. We also need to consider the weight of a vertex x, defined $w(x) = \sum_y w_{xy}$, where y is a neighbor of x.

In general we have the following definition for the weighted graph Laplacian used in Eigenmap as follows

Definition 5 (Weighted Graph Laplacian [BN 02])**.** *Let (V, w) be a locally finite weighted graph without isolated points. For any function $f : V \to \mathbb{R}$, define the function $\Delta_w f$ by*

$$\Delta_w f(x) = \sum_{x \sim y} (f(x) - f(y)) w_{xy}$$

Here $x \sim y$ indicates that x and y are adjacent.

It is necessary to construct this in an operator form. To do so, begin by defining the degree matrix

$$D_{xx} = w(x)$$

and the weighted adjacency matrix

$$W_{xy} = \begin{cases} w_{xy} & \text{if} \quad x \sim y \\ 0 & \text{otherwise} \end{cases}$$

Then the Laplacian matrix is given

$$L = D - W \tag{3}$$

The normalized Laplacian is formed by factoring out D^{-1}:

$$\Delta_G = D^{-1}L = I - D^{-1}W \tag{4}$$

Since D is diagonal and G has no isolated points, D^{-1} always exists. It is worth pointing out that there are several normalization procedures that seem to suit the purposes required of them. The differences are small. As we are using Belkin and Niyogi's Eigenmap, we will use their formulation.

Having defined the graph Laplacian, we can now narrow our focus to the particular embedding that we are using. Consider a graph $G = \{V, E\}$ embedded in $\mathbb{R}^n$. Notice here that we are not looking at a weighted graph. The specific procedure for eigenmap embedding requires us to specify the nature of the edge weight as follows.

The heat kernel of the graph is given [BN 02]

$$k(x, y, t) = \exp\left(-\frac{||x - y||^2}{t}\right)$$

In the interest of using eigenmap, [BN 02] gives the edge weight function by the kernel $w_{xy}(t) = k(x, y, t)$. As before, the vertex weight of x is defined $w_t(x) = \sum_y k(x, y)$. For the remainder of this document, the weight w of and edge will refer specifically to this weight scheme.

A few observations should be made about $k(x, y, t)$. The first is that it is inverse to the distance between two vertices: if $||x - y||$ is large, $k(x, y, t)$ will be small, if $||x - y||$ is small, $k(x, y, t)$ will be large. Another fact is that as $t \to 0$, $k(x, y, t) \to 0$. For this reason, the vertex degree also goes to 0 as $t \to 0$.

As with the normalized Laplacian, there are several variations of the heat kernel in spectral graph theory that differ slightly. Again, we'll be using the one deemed favorable to eigenmap, but the use of different formulas in other areas is no cause for alarm.

1.4.3 Truncating to Finite Dimensional Space

While a remarkable result, the embedding (1) leaves our manifolds in an infinite dimensional space. To continue with our goals, it will be necessary to work in a finite dimensional vector space. To do this, a result from [Portegies 13] will give us some hope.

Definition 6 (Dilatation)**.** *The local dilatation at p of a function f from a Riemannian manifold $\mathcal{M}, g$ to a finite dimensional vector space V is given*

$$|(df)_p| = \lim_{r\to 0} \sup_{x,y\in B_r(p)} \frac{d(f(x), f(y)}{d(x,y)}$$

An embedding is close to isometric if the local dilatation is close to one.

Lemma 1 (Truncation Lemma [Portegies 13])**.** *Given an n-dimensional Riemannian manifold $(\mathcal{M}, g)$, with Volume bounded above by V, Ricci curvature bounded below by k, and injective radius bounded below by i, we have the following.*

Let $\epsilon > 0$. Then there exists a $t_0 = t_0(n, k, i, \epsilon)$ so that for all $0 < t < t_0$, there is an $N_E = N_E(n, k, i, V, \epsilon, t)$ so that if $N \geq N_E$, the map

$$\Phi_t^N(p) := (2t)^{(n+2)/4}\sqrt{2}(4\pi)^n/4\{e^{\lambda_j t}\phi_j(p)\}_{j\geq 1}^N$$

is an embedding of $(\mathcal{M}, g)$ into $\mathbb{R}^N$ so that

$$1 - \epsilon < |(d\Phi_t^N)_p| < 1 + \epsilon$$

In reality, explicit values for the necessary geometric quantities may not be available. However, given that the truncation is done with heuristics in either case, we consider the lemma above an improvement, as now it is clear that the truncation dimension depends only on the geometry of the object under consideration. In the case of a metric network, N is uniform while the number of nodes in the network can

be much larger.

The fact that we our embedding is a near isometry will influence some of our later methods. The heat kernel embedding also (nearly) balances the points, which will be helpful when it comes to aligning the points.

1.4.4 Sampling

Samples must then be generated. We need some heavier tools than simple random sampling to approach this problem due to the very high number of points under consideration (this will become more apparent later). We should begin with samples that we deem to be more important. To do this, we consider three methods.

1. Sampling based on scalar curvature. Regions of high unsigned (positive or negative) scalar curvature are determined to be more 'important'. Obviously, this term is ambiguous, but we later give some motivation that should clear things up. We begin by sampling the critical points of scalar curvature. If there are still too few points, we sample from the scalar curvature flows between these regions.

2. Sampling the critical points and zeros of the eigenfunctions.

3. Using a Vitali covering approach to reduce the number of samples in a uniform way. Details are given in the main body.

1.4.5 Point Matching

After acquiring samples, the next task is to align these points. The issue of aligning more than 3 points with $O(n,1)$ transformations is limited. Möbius transformations *do* exist in n dimensions in the form of $O(n,1)$, often called the Lorentz group. This group is of dimension $\frac{n(n+1)}{2}$ (with 3 dimensions for $O(3,1)$). By way of example, the case of Möbius transformations on $\mathbb{R}^3$ is the action of $O(3,1)$. In Möbius voting, points are in $S^2 \subset \mathbb{R}^3$. $O(3,1)$, by the above formula, has $\frac{3(4)}{2} = 6$, corresponding with 3 point alignment, with each point having 2 degrees of freedom. The shortcoming here is generalized Möbius transformations align only 3 points.

The motivation for using Möbius transformations in Möbius voting is that in 3 dimensions, it is possible to align 3 points with any other 3 points with no other information, and the formula for doing so is very simple. Regardless, this is a conformal tranformation. Given that our embedding is almost isometric, it will be prudent to maintain this quality as much as possible. Keeping this goal in mind, we can develop alternative strategies to solve these problems.

Without Möbius transformations, we can still align more than 3 points. It is always possible to align N generic points with another N points with an $N \times N$ matrix. The case in aligning N points becomes a matrix inverse problem, since the alignment operation in this case is unique.

Let $[p_1, \ldots, p_M]$ and $[q_1, \ldots, q_M]$ be two sets of points from each manifold, translated so that $p_1 = q_1$. Keeping in mind that each p_i is a vector with N coordinates,

we find the L that aligns these points

$$L = [q_1 \dots q_N]([p_1 \dots p_N])^{-1}$$

This is worthwhile to consider, and will be the typical approach we consider. However, this is rigid. Aligning the points is not an issue, rather, it is possible that the transformation is far from isometric. If we want to loosen the type of transformation used, we can align a subset of k points and see where particular regions fall. For this reason, we can consider isometric or almost isometric transformations that align k points, with $k < N$. Then we need to find $L_{k\times k}$ that minimizes $||LL^T - I||_F$ subject to

$$L[p_1, \dots, p_k]^T = [q_1, \dots, q_k]^T \tag{5}$$

For high dimensions, this process can be computationally costly. Options are solving (5) subject to the given constraints, or further truncating the embedding at the expense of losing isometry. To reduce computation in all cases, we will in Section 2.5 provide some schemes based on geometric properties to eliminate the possibility of correspondence before we attempt to align points. By reducing the possible correspondences before calculating any alignment, we can reduce this problem further.

1.4.6 Summing up the Process

1. Construct a mesh (if necessary). If the data is a point cloud, we will want to determine edges in a manner that turns the data into a connected graph.

2. Embed each manifold into l^2. Use heat kernel embedding to embed each manifold into an infinite dimensional space.

3. Truncate to $\mathbb{R}^N$. Currently lacking a way to determine all the parameters that determine the dimension in which to truncate our embedding, according to the geometric qualities provided by [Portegies 13], determining this dimension will be determined by heuristics until such time that the relevant quantities can be reliably found in general.

4. Sample points. Uniform samples are good, but for complicated data, the sheer number of samples necessary renders this unfeasible. Instead, we'll rely on regions of high positive or negative scalar curvature, nearest orthogonal transformation, and spectral comparison.

5. Align points or get close. Depending on what your purposes, we can rely with orthogonal transforms, with voting thresholds to identify wholly isometric objects, or objects with isometric regions. The number of points aligned will depend on the truncation dimension.

6. Check similarity. The deviations from isometry (or conformality) tells us how "good" our transformation is. The heaviest tool for this is the Hausdorff dis-

tance.

7. Vote and update the fuzzy correspondence matrix.

8. Repeat steps 5 through 7 for a predetermined number of times, or until all combinations of points are exhausted.

1.5 Our Contributions

The field of shape comparison is vibrant and active. We hope to contribute material in the following ways.

1. We intend to extend voting algorithms to higher genus surfaces and higher dimensional manifolds.

2. We propose using geometric properties to improve computational speed. High dimensional information can be very complicated, and processing time threatens to be an issue if we are not careful. We intend to reduce the amount of information to process by considering several invariants that will allow us to discard unreasonable points or correspondences.

   We find that the scalar curvature can tell us what points are 'important' so we can narrow the scope of computing alignments. On the other hand, we can use the cross ratio, eigenfunctions, and center of mass to tell us whether a correspondence is worth considering.

3. We hope to develop some manner of using generalized Lorentz transformations,

ie transformations in $O(n,1)$, to attempt conformal transformations in higher than 3 dimensions.

1.6 Computational Considerations

It is one thing to show the existence of something, another to compute it. Here we seek to assure the reader that the necessary computations are possible in the scheme that we have proposed.

1.6.1 Laplacian Computation

The embedding of smooth manifolds and graphs is similar in that each method requires computing the eigenfunctions of the Laplacian. In general, the method of Rayleigh quotients can find the eigenvalues of the Laplacian on a smooth manifold. Let $H_0^{1,2}$ denote the Sobolev space consisting of functions $u, v \in H_0^{1,2}$ and with inner product

$$\langle u, v \rangle_{1,2} = \int_\Omega (\nabla u \cdot \nabla v + uv) dx$$

Definition 7 (Rayleigh Quotient)**.** *Let w be a function on the region $\Omega \subset \mathbb{R}^n$. Then the Rayleigh quotient of w is defined*

$$\frac{||\nabla w||^2_{L^2(\Omega)}}{||w||^2_{L^2(\Omega)}} = \frac{\int_\Omega |\nabla w|^2 dx}{\int_\Omega w^2 dx}$$

Theorem 5 (Maximum-Minimum Property for j^{th} Eigenvalue[CH 89])**.** *Let $u_1, u_2, \ldots, u_{j-1}$*

be the first $j-1$ eigenfunctions of Δ, chosen to be orthonormal. Let

$$Y = \{w : w \in C^2(\Omega), w \neq 0, w(x) = 0 \text{ for } x \in \partial\Omega, \langle w, v_i \rangle = 0 \text{ for } i = 1, 2, \ldots, j-1\}$$

Let u_j be a function so that

$$\lambda_j = \frac{||\nabla u_j||^2}{||u_j||^2} = \min_{w_j \in Y} \left\{ \frac{||\nabla w||^2}{||w||^2} \right\}$$

Then λ_j is the j^{th} eigenvalue and u_j is a corresponding eigenfunction.

The first eigenpair is found by the min-max principle for all functions on Ω.

This method will give the eigenvalues for a smooth manifold. For graphs, the problem is to find the eigenvalues of the Laplacian matrix, which is a real symmetric matrix. There are many algorithms for finding this, including Rayleigh Quotient Iteration (RQI). Since this has cubic convergence (fast for eigenvalue problems), RQI is a reasonable choice of method, and is preferred for aesthetic reasons. The packages SLEPc and IETL each provide implementaions of RQI.

1.6.2 Truncation

While infinite dimensional spaces give plenty of room to stretch out in, for practical purposes we usually want to work in finite dimensions.

Lemma 1 tells us that the dimension of truncation is based on the Ricci curvature, injective radius, volume, dimension of n and time. These are fixed quantities, so some dimension N to which we wish to truncate the embedding does exist, based on hour

desired ϵ within isometry. In practice, this number may be hard, or impossible to determine. We will find ourselves relying on heuristics to choose the value of N. For examples of this, see [Sharma 12], where the author finds success using the first three to five eigenfunctions.

2 Shape Comparison

Now it is time to discuss specifically our method for comparing shapes. As we mentioned earlier, the motivation for this comes from Möbius Voting. Our plan is to extend a voting method to shapes of genus higher than 0, and dimension more than 3. There are of course, several technical challenges, and quite a few computational challenges; overcoming these issues constitutes most of the details of the outline given previously.

Before we talk about comparing shapes, we need a metric by which to compare them. This, as it turns out is not so simple from a computational standpoint, but a strong theoretical method exists.

2.1 Measuring Similarity: The Gromov-Hausdorff Distance

Gromov-Hausdorff distance (d_{GH}) is a commonly used method for matching shapes, and will be the distance that we'll use here. The Gromov-Hausdorff distance gives a sense of how far two shapes are from being isometric.

To begin the discussion, we first consider the Hausdorff distance of two subsets of

a metric space.

Definition 8 (Hausdorff Distance [Gromov 81]). *For two subsets X and Y of a metric space Z with metric d, the Hausdorff distance is defined as*

$$d_H^Z(X,Y) = \inf\{\epsilon \geq 0; X \subset N_\epsilon(Y), Y \subset N_\epsilon(X)\}$$

Where N_ϵ is an ϵ neighborhood of a set.

This metric gives a method of measuring the distance between subsets of a metric space. Our task however, will ultimately be to compare manifolds, which we generally won't expect to be embedded in the same space or have the same metric. We want to know how similar two shapes are. For this we turn to the Gromov-Hausdorff distance. This can be defined in terms of the Hausdorff distance as follows

Definition 9 (Gromov-Hausdorff Distance [Gromov 81]). *Given two metric spaces X and Y, the Gromov-Hausdorff distance is given*

$$d_{GH}(X,Y) = \inf_{f,g,Z} d_H^Z(f(X), g(Y))$$

where $f : X \to Z$ and $g : Y \to Z$ are isometric embeddings into Z and d_H^Z is the Hausdorff distance in Z

In particular

$$d_{GH}(X,Y) = 0 \quad \text{if and only if } X \text{ and } Y \text{ are isometric}$$

Now we can compare correspondences. Computing the Gromov-Hausdorff distance is now a matter of computing the distortion between all possible isometric correspondences between manifolds. Our method here involves embedding two manifolds in the same space via heat kernel embedding (which is almost isometric), aligning points, then computing the Hausdorff distance of the sets. The reader will notice that the theoretical motivation for what we are doing is in fact estimating the Gromov-Hausdorff distance described here.

2.2 Spectral-Hausdorff Distance

The eigenfunctions and eigenvalues of our manifolds play a central role in the process of shape comparison. The spectral Hausdorff distance describes the distance between manifolds in terms of the Laplacian eigenfunctions. From these distances, we can find some helpful continuity results.

2.2.1 Smooth Spectral Hausdorff Distance

[BBG 94] gives a measure for isometry between manifolds. Given a Riemannian manifold $(\mathcal{M}, g)$ with an orthonormal basis of eigenfunctions $a = \{\phi_j^a\}_{j\geq 1}$, define the maps

$$I_t^a(x) = \sqrt{\mathrm{Vol}(\mathcal{M})}\{e^{-\lambda_j t/2}\phi_j^a(x)\}_{j\geq 1} \tag{6}$$

describing the action of the volume form of the manifold on a basis of eigenfunctions, ϕ^a. Using this, we have a definition of spectral distance:

Definition 10 (Spectral Distance [BBG 94])**.** *Given two Riemannian manifolds* $(\mathcal{M}_1, g_1)$ *and* $(\mathcal{M}_2, g_2)$*, the spectral distance is defined*

$$d_t(\mathcal{M}_1, \mathcal{M}_2) = \max\left\{ \sup_{a_1}\left(\inf_{a_2} d_H(I_t^{a_1}(\mathcal{M}_1), I_t^{a_2}(\mathcal{M}_2))\right), \sup_{a_2}\left(\inf_{a_1} d_H(I_t^{a_2}(\mathcal{M}_2), I_t^{a_1}(\mathcal{M}_1))\right)\right\} \tag{7}$$

Where I_t^a *is given above and* d_H *is the Hausdorff distance in* l^2*.* $a_i, i = 1, 2$ *is an orthonormal basis of eigenfunctions* $\{\phi_j^a\}_{j\geq 1}$ *in* $L^2(\mathcal{M}, g)$*.*

2.2.2 Graph Spectral Distance

Consider now the case of a graph. Consider a finite, connected graph $G = (V, \mu)$. In this model, the similarity between points (vertices) is given by the weight of the edge joining them. Spectral graph theory [Gu] looks at what can be inferred by studying the spectrum of the Laplacian of a graph.

Let $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$ be the set of eigenvalues of Δ_{G_1}, the graph Laplacian (4) of G_1, and $\mu_1 \leq \mu_2 \leq \cdots \leq \mu_n$ be the set of eigenvalues of Δ_{G_2}, defined the same way.

If $|G_1| = |G_2|$, the spectral measure is defined for graphs G_1 and G_2

$$d(G_1, G_2) = \frac{1}{n}\left(\sum_{i=1}^{n}(\lambda_i - \mu_i)^2\right)^{1/2}$$

where n is the number of vertices in each graph.

If the number of vertices is not the same, define a spectral measure,

$$m_{\sigma(G_1)} = \frac{1}{N} \sum_i \delta_{\lambda_i}$$

with associated cumulative distribution function F_{G_1}. The inverse CDF is defined

$$F_{G_1}^{-1}(x) = \inf\{t \in \mathbb{R} : F_{G_1} > x\}$$

Then the spectral distance is the Wasserstein distance between the spectral measures of G_1 and G_2

$$d^W(m_{\sigma(G_1)}, m_{\sigma(G_2)}) = \left(\int_0^1 ||F_{G_1}^{-1}(x) - F_{G_2}^{-1}(x)||^2 dx \right)^{\frac{1}{2}}$$

2.2.3 Conclusions

The problem of shape comparison ultimately comes down to solving one of these problems. The most direct way is to use the Gromov-Hausdorff distance. Shapes with a small Gromov-Hausdorff distance are similar; those with a large distance are dissimilar. However, the simplicity of this ends here. Actually computing the Gromov-Hausdorff distance requires us to find the best possible correspondence with respect to the Hausdorff distance, a number that is itself hard to compute. The reader can imagine that there are far too many correspondences to consider this tractable.

Our method is entirely concerned with finding the best correspondence between meshes (the continuous case is viewed as a mesh of sampled points). So long as we can achieve this goal reasonably simply, estimating the Gromov-Hausdorff distance becomes significantly more achievable.

The next possibility that we are presented with is spectral Hausdorff distance. We can compare shapes in this way; small spectral Hausdorff distance indicates similar shapes; the converse is true for large distances. The discrete case initially looks promising, we just need to sum the difference of all the eigenvalues. However, it's important to remember that there may be lots of eigenvalues, there are methods and software packages (SLEPc, IETL) to calculate this, but each eigenvalue calculation adds non-trivial software overhead.

The smooth case has challenges of its own. The method exists for finding eigenfunctions; though it may be difficult. In some instances, the eigenfunctions may already be known, such as S^N and $\mathbb{C}P^N$. The maps I_t^a (see (6))are possible to compute; an instance of I_t^a is the same as a heat kernel map, differing only by a constant. The next problem may not be so simple to solve. It is necessary to find the supremum and infinum over all possible choices of eigenfunction basis for each manifold, and calculate the Hausdorff distance each time. Attempting this directly is not a suitable approach.

However, this gives us tools to work with. Spectral Hausdorff distance tells us something about the continuous dependence of the distance on the eigenfunctions. This gives us some tools for excluding certain correspondences before wasting com-

putational resources on them.

2.3 Embedding

Let $(\mathcal{M}_1, g_1)$ and $(\mathcal{M}_2, g_2)$ be two manifolds in m dimensions. Ultimately, we want to compare these shapes as they exist in some ambient space. In order to do so, we need to embed them into some canonical target, such as $\mathbb{R}^N$ or $\mathbb{S}^{N-1}$. The heat kernel embedding is the first step in doing so. Given below is the method for smooth and discrete manifolds.

2.3.1 Smooth Manifolds

Ideally, we consider manifolds as they exist in a Euclidean space. For our embedding however, we need to make a detour to l^2 before truncating. Berard, Besson and Gallot [BBG 94] give a method of embedding a manifold into the Hilbert space of square summable sequences, l^2. Given a Riemannian manifold $(\mathcal{M}, g)$, the Laplacian eigenvalue equation is given

$$\Delta_g \phi = \lambda \phi$$

Where Δ_g is the Laplacian of $\mathcal{M}$ with respect to the metric g. If λ_j is a solution to this equation, the corresponding function ϕ_j is the eigenfunction associated with λ_j. Using these eigenpairs, the embedding described in (1) approaches isometric as $t \to 0$

It is worth noting that [BBG 94] also provides a renormalized embedding from M

to l^2 that embeds the manifold into S^∞, the infinite dimensional sphere.

$$K_t^a : x \to \frac{1}{\sqrt{\sum_{j\geq 1} e^{-\lambda_j t}\phi_j^2(x)}} \left\{e^{-\lambda_j t/2}\phi_j(x)\right\}_{j\geq 1}$$

This embedding approaches isometry by

$$(K_t^a)^*_{can} = \frac{1}{2t}\left(g - \frac{t}{3}\text{Ricci} + \mathcal{O}(t^2)\right)$$

The map into S^∞ will be of interest with respect to Möbius geometry on spheres.

At this point finding the embedding is a matter of solving the eigenvalue problem above. For some structures, these are known, in other cases, they must be computed explicitly, using Rayleigh-Quotient methods. The fact that this embedding is nearly an isometry will inform all of our further actions. We can't exactly work with infinite dimensional data, but there are results for truncation, which we will look at later.

2.3.2 Heat Kernel Embedding of a Graph

The general formulation of isometric comparison for discrete manifolds is similar to smooth manifolds. The main thrust in using the eigenfunctions of the Laplacian to embed a manifold into l^2 is the the same for discrete and smooth manifolds. For the specific procedure we'll be following the Eigenmap method, given by [BN 02].

The general case is for a graph $G = \{V, E\}$ embedded in $\mathbb{R}^n$, which is to say that every vertex has a coordinate in $\mathbb{R}^n$.

Those working with point clouds will find that the edge set is not defined for the

points. In order to create a graph structure, [BN 02] provides two solutions.

1. One approach is to form edges from one vertex to all vertices within some ϵ disk of the vertex. The difficulty is in picking a value for ϵ: the ideal value may not be obvious. It is also possible that this method will disconnect the graph.

2. It is possible to connect edges using the k-nearest neighbors algorithm (KNN). Given a point x, KNN will classify k closest points in the class connected to x. Here, k may be hard to determine, however we are guaranteed a connected graph.

Once edges are determined, the next step is to calculate the weights. First, we use the Gaussian function will give the edge weights as we did earlier

$$\mu_{xy}(t) = k(x, y, t) = \exp\left(-\frac{||x - y||^2}{t}\right)$$

This takes a higher value for close points, in a sense giving a notion of the strength for a given connection.

Now consider a map

$$\Phi : G \to \mathbb{R}^N$$

$$\Phi(v_i) = \phi_i$$

Where N is the number of vertices in G. Φ has the matrix representation: $\Phi = [\phi_1, \phi_2, \ldots, \phi_N]$. Note that at this point we aren't necessarily considering ϕ_j to be an

eigenvector of the Laplacian, although we will soon show that this is the case. The goal of our embedding is to minimize

$$\sum_{i,j} ||\phi(v_i) - \phi(v_j)||^2 W_{ij}$$

Here, W_{ij} can be thought of as a penalty for separating points with strong connections. We can reformulate the problem

$$\begin{aligned}
\sum_{i,j} ||\phi(v_i) - \phi(v_j)||^2 &= \sum_{i,j} \left((\phi_1(v_i) - \phi_1(v_j))^2 + \cdots + (\phi_N(v_i) - \phi_N(v_j))^2\right) \\
&= \sum_{i,j} (\phi_1(v_i)^2 + \phi_1(v_j)^2 - 2\phi_1(v_i)\phi_1 v(j) + \dots \\
&\cdots + \phi_N(v_i)^2 + \phi_N(v_j)^2 - 2\phi_N(v_i)\phi_N(v_j))W_{ij} \\
&= \sum_i \sum_k \phi_k(v_i)^2 D_{ii} + \sum_j \sum_k \phi_k(v_j)^2 D_{jj} - 2\sum_{i,j} \sum_k \phi_k(v_i)\phi_k(v_j) W_{ij} \\
&= \mathrm{Tr}(\Phi^T D \Phi) + \mathrm{Tr}(\Phi^T D \Phi) - 2\mathrm{Tr}(\Phi^T W \Phi) \\
&= 2\mathrm{Tr}(\Phi^T L \Phi)
\end{aligned}$$

So the best embedding is the one that minimizes

$$\mathrm{Tr}(\Phi^t L \Phi)$$

subject to

$$\Phi^T D \Phi = I$$

[BN 02] gives the solution to this by the eigenvalue problem:

$$L\phi = \lambda D\phi$$

To find the embedding then, we need to find the eigenvalues of the normalized Laplace operator, given in Section 1.4.2, (4). The eigenvalue problem for Δ is given by

$$\Delta\phi = \lambda\phi$$

$$(D^{-1}L)\phi = \lambda\phi$$

$$L\phi = \lambda D\phi$$

Since the graph consists of a finite number of elements, the spectrum is finite. So the eigenfunctions are the columns of the matrix:

$$[\phi_1, \phi_2, \ldots, \phi_N]$$

corresponding to eigenvalues

$$0 \leq \lambda_1 \leq \lambda 2 \leq \cdots \leq \lambda_N$$

The embedding (Eigenmap) is then given [BN 02]

$$\Phi : G \to \mathbb{R}^N$$

defined by

$$x \to (\phi_1(x), \phi_2(x), \ldots, \phi_N(x))$$

Remark: It is important to note here that the solution in the discrete embedding is that the above eigenvalue problem *minimizes*, but does not necessarily fully eliminate metric distortion in mapping a graph to Euclidean space. We don't find an isometry, just the closest to isometry that we can get in $\mathbb{R}^N$.

2.3.3 Theoretical Basis

We will later like to align our Riemannian manifolds $(\mathcal{M}_1, g_1)$ and $(\mathcal{M}_2, g_2)$ with orthogonal transformations. To do this, we need the following result

Theorem 6. *If Riemannian manifolds $(\mathcal{M}_1, g_1)$ and $(\mathcal{M}_2, g_2)$ are isometric and $\Phi_t^{g_1}$ and $\Phi_t^{g_2}$ are isometric embeddings, then there exists an orthogonal transform A of l^2 so that $A(\Phi_t^{g_1}(\mathcal{M}_1)) = \Phi_t^{g_2}(\mathcal{M}_2)$.*

Proof. Assume that $\mathbf{M}_1 = (\mathcal{M}_1, g_1)$ and $\mathbf{M}_2 = (\mathcal{M}_2, g_2)$ are isometric, so that

$$f : \mathbf{M}_1 \to \mathbf{M}_2$$

is an isometry.

By the Sturm-Liouville theorem, the eigenvalues of the Laplacian have finite multiplicity, each basis of eigenfunctions $\alpha = \{\phi_j^{g_1}\}_{j\geq 1}$ and $\beta = \{\phi_j^{g_2}\}_{j\geq 1}$ consists only of eigenfunctions of finite multiplicity. Since $\mathbf{M}_1$ and $\mathbf{M}_2$ are isometric, the eigenvalues are identical and the eigenfunctions are the same up to isometry in each eigenspace. So for eigenfunctions $\phi_i^{g_1}$ and $\phi_i^{g_2}$ with multiplicity m, define $A_i \in O(m)$ satisfying

$$\phi_i^{g_1} = A_i \phi_i^{g_2}$$

Define the block matrix A, where the i^{th} block is the matrix A_i

$$A = \begin{bmatrix} A_1 & & & 0 \\ & A_2 & & \\ & & \ddots & \\ 0 & & & A_N \end{bmatrix}$$

So A is an orthogonal transform, and

$$A[\phi_1^{g_2}, \phi_2^{g_2}, \dots \phi_N^{g_2}] = [\phi_1^{g_1}, \phi_2^{g_1}, \dots, \phi_N^{g_1}]$$

For each coordinate $x \in \mathcal{M}_1$, $f(x) \in \mathcal{M}_2$. So the embedded manifolds can be mapped to one another

$$\Phi_t^{g_1}(\mathcal{M}_1) = A\Phi_t^{g_2}(f(\mathcal{M}_2))$$

□

This ensures that the images of almost isometric manifolds under the heat kernel embedding can be related by almost orthogonal linear transformations in l^2 or $\mathbb{R}^N$.

2.4 Sampling

For finding our alignments, we need to sample a number of points from each manifold. Let Σ_1 be the set of sampled points from $\Phi_t^{g_1}(\mathcal{M}_1)$, and let $|\Sigma_1| = s_1$. Shape comparison in high dimensions has size issues. For an object in N dimensions, the maximum number of k samples that can be aligned is N. A vital part of voting is comparing unmatched points, so the k aligned points are only a part of Σ_1. In addition, the Truncation lemma gives the notion that the total number of points in the mesh, s, will be much higher than N, in the sense that the truncation dimension is based on the geometric complexity of the object.

Clearly, for smooth manifolds, it is necessary to sample points to find correspondences. Even for graphs, in which the points are discrete sets, the number of vertices will often be too high to feasibly find possible correspondences for all points. It is clear that a sampling method is necessary to overcome these issues.

2.4.1 Vitali Samplings

This method is less precise than the others, but is well suited for entirely discrete data, such as point clouds or rasters. We'll begin with a useful result, e.g. [Tao 11]

Lemma 2 (Vitali Covering Lemma)**.** *Let $B_1, B_2, \ldots, B_n$ be a finite collection of open balls in $\mathbb{R}^N$ (not necessarily disjoint). Then there exists a subcollection $B'_1, \ldots B'_m$ of*

disjoint balls in this collection such that

$$\cup_{i=1}^{n} B_i \subset \cup_{j=1}^{n} 3B'_j$$

The Vitali covering lemma tells us that it is possible to create a family of balls $\mathcal{C}$, of size ϵ that is both disjoint, and has the property that tripling the radius of the balls in $\mathcal{C}$ so that they cover our space, with the idea that the centers of the balls provide a semi-uniform sampling of the region. The proof of the covering lemma is constructive; an algorithm for covering can be extracted from the proof. For this reason, we cite it here

Proof. Let $B_1, B_2, \ldots B_M$ be open balls in $\mathbb{R}^N$. Define $B = \{B_1, B_2, \ldots M_M\}$. Pick the largest ball and give it the name B'_1. Now assume inductively that a subcollection $B'_1, B'_2, \ldots B'k$ have been chosen. If there is no B_i disjoint from $B'_1, B'_2, \ldots B'k$, we are done. Otherwise, pick the largest B_i disjoint from $B'_1, B'_2, \ldots B'k$ and add it to the collection so that $B_i = B'_{k+1}$. This is continued until no balls are remaining. Call the final set $B' = \{B'_1, B'_2, \ldots, B'_m\}$.

This completes the algorithm for choosing a covering. Next is to show that this works and satisfies the condition $\cup B_i \subset \cup 3B'_j$

Pick a ball B_i in the original collection B. B_i must intersect with at least one ball in B', let B'_j be the first such ball. Then B_i is disjoint from $B'_1, \ldots B'_{j-1}$. Therefore, the radius of B_i is no larger than that of B'_j. Let B_i have radius r_i and center p_i and let B'_j have radius r_j and center p_j.

We have established that $r_i \leq r_j$. It is clear that $||p_i - p_j|| \leq r_i + r_j \leq 2r_j$. Since B_i has radius r_i, the furthest extent of B_i is $2r_j + r_i \leq 3r_j$ from p_j. Hence, $B_i \subset 3B'_j$.

□

According the the Vitali covering lemma, each center point is between $2 - \epsilon$ and $6 - \epsilon$ away from any other point. At this point, edges can be specified as connecting points within $6 - \epsilon$ of one another.

Two significant issues must be addressed with this method: complexity and loss of precision. Ultimately, the latter problems depend on the choice of ϵ, so we'll address that.

By clearing wide radii around a point, it may be that the sampling is missing some important structure to the data. For instance, consider the situation

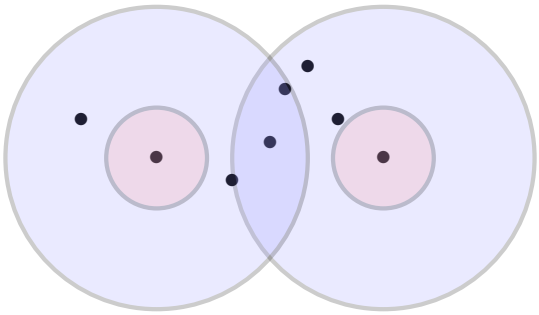

Figure 1:

Here we can see that in the first image, the choice of ϵ is too small to capture the overall arrangement of the points. The second image shows an arguably more accurate image. However, smaller choice of epsilon requires more computation, and is more sensitive to noise.

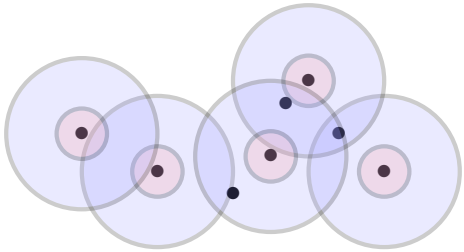

Figure 2:

The tradeoff to using a smaller epsilon is that the net construction increases in complexity. Every additional point requires checking that it is at least $2 - \epsilon$ away from all previously added points.

This could also be used to sample points on a smooth manifold by doing this process in the embedding space. This may produce uniform samples because the second fundamental form is close to constant in the embedded space [WZ 15]. Consider why this is tractable. Provided we use the renormalized embedding so that the target space is S^{∞}, truncated to S^{N-1}. A circle packing scheme on the sphere is a very difficult problem to solve. Generally, we can't hope to achieve this. However, using the Vitali covering method, we can provide a loose circle packing on the sphere.

If the truncated space is too high of dimension, and if the number of samples is too high, the balancing act between sampling enough so that the geometry is captured, while keeping the number of computations as low is possible, is delicate. For this reason, our sampling method should probably have some focus on more 'important' regions of the mesh, particularly for high dimensional objects. In the event that this sampling is still too dense for our computers to work with, we can use the following

methods to reduce this point set.

2.4.2 Scalar Curvature

We claim that regions of high scalar curvature have greater importance than regions of low scalar curvature. There are several ways to come to this conclusion, we will use the following.

The reason for this can be explained from the perspective of uniform sampling in an ambient space. This itself is a difficult problem, and has issues. To see how scalar curvature relates to this, consider the following expression for scalar curvature:

$$\frac{\text{Vol}(B_\epsilon(p) \subset M)}{\text{Vol}(B_\epsilon(0) \subset \mathbb{R}^n)} = 1 - \frac{S(p)}{6(n+2)}\epsilon^2 + \mathcal{O}(\epsilon^4)$$

For a fixed dimension n and radius ϵ.

If the scalar curvature is 0 (Euclidean space) the ball in M is the same volume as in $\mathbb{R}^n$. If the scalar curvature is high and positive, then $\text{Vol}(B_\epsilon(p) \subset M) < \text{Vol}(B_\epsilon(0) \subset \mathbb{R}^n)$. On the other hand, if the scalar curvature is high and negative, then $\text{Vol}(B_\epsilon(p) \subset M) > \text{Vol}(B_\epsilon(0) \subset \mathbb{R}^n)$. In either case, the geometry of the manifold differs from that of Euclidean space. In other words, there is more geometric complexity in these highly curved regions.

This perspective of scalar curvature leads us to see another benefit. Based on the idea that regions of high scalar curvature correspond to more 'important' regions, forcing rigid correspondences here makes sure that important parts of shapes are

compared, while less important regions have the freedom to be quite different. This can be adapted to more pose invariant comparisons.

2.5 Filtering Correspondences

Given point samples $\Sigma_1 \subset \Phi_t^{g_1}(\mathcal{M}_1)$ and $\Sigma_2 \subset \Phi_t^{g_2}(\mathcal{M}_2)$, the next task is to align them; the details of this are given in the next section. Before this, it is worth considering that some correspondences are immediately not valid. This is especially important in our case, as filtering correspondences can cut down the number of matrix computations in finding a best alignment. The near isometry of the embedding, and the goal of near isometric correspondence allow us to use some tools to disregard correspondences immediately.

2.5.1 Cross Ratio

We'll begin with the well known concept of the cross ratio as it's used in 2 dimensions.

Definition 11 (Cross Ratio in $\mathbb{R}^n$)**.** *Given* 4 *points* A, B, C, D *in* $\mathbb{R}^2$ *or* $\hat{\mathbb{C}}$*, the cross ratio is given*

$$[A, B; C, D] = \frac{(C - A)(D - B)}{(C - B)(D - A)}$$

Typically, this is defined for points on the number line or (extended) complex plane. We seek a formulation of the cross ratio in $\mathbb{R}^N$. We first need to define a particular metric.

Definition 12 (Chordal Metric)**.** *The Chordal metric*

$$d_\pi : \hat{\mathbb{R}}^n \to \mathbb{R}$$

Is defined as the Euclidean distance under stereographic projection π:

$$d_\pi(x, y) = |\pi(x) - \pi(y)|$$

More explicitly, for $x, y \in \mathbb{R}^n$, the chordal distance is expressed:

$$d_\pi(x, y) = \frac{2|x - y|}{\sqrt{1 + |x|^2}\sqrt{1 + |y|^2}}$$

With the special case

$$d_\pi(x, \infty) = \frac{2}{\sqrt{1 + |x|^2}}$$

Then the cross ratio can be defined by multiplying distances rather than scalars.

Definition 13 (Cross Ratio on $\mathbb{R}^n$)**.** *[Ratcliffe 06] Let u, v, x, y be points in $\hat{\mathbb{R}}^n$ with $u \neq v$ and $x \neq y$. The cross ration of these points is the number*

$$[u, v, x, y] = \frac{d_\pi(u, x)d_\pi(v, y)}{d_\pi(u, v)d_\pi(x, y)}$$

It is easy to check that two sets that differ by an orthogonal (isometric) transform have the same cross ratio. In this sense, the cross ratio can be used to develop a filter for isometric transformations. A stronger result also holds:

Theorem 7 ([Lie 1871])**.** *A transformation is conformal if and only if it preserves cross ratios.*

This property provides a filter for conformal registration as well. Better still, the cross ratio is continuous. So nearly isometric sets of points will have similar cross ratios. From this, if the cross ratios between nearby points are excessive, the correspondence should be discarded. Given the points specified earlier, we propose the filter based on the cross ratio. If given 4 points from each embedded manifold $z_1, z_2, z_3, z_4 \in \Phi_t^N \subset \mathbb{R}^n$ and $w_1, w_2, w_3, w_4 \in \Phi_t^N \subset \mathbb{R}^n$, the cross ratio satisfies for some $\epsilon \geq 0$.

$$|[z_1, z_2, z_3, z_4] - [w_1, w_2, w_3, w_4]| < \epsilon$$

then there *may* be a good correspondence between the points. Otherwise not, and computational resources should not be spent calculating a map aligning them.

2.5.2 Scalar Curvature Matching

We have determined that the scalar curvature is an important quantity for sampling. We now propose that the critical points of scalar curvature should be close for similar manifolds. Describing the scalar curvature in local coordinates;

Given Christoffel symbols

$$\Gamma_{kl}^i = \frac{1}{2} g^{im} \left(\frac{\partial g_{mk}}{\partial x^l} + \frac{\partial g_{ml}}{\partial x^k} + \frac{\partial g_{kl}}{\partial x^m} \right)$$

Then the scalar curvature is written

$$S = g^{ab}\left(\frac{\partial \Gamma^c_{ab}}{\partial x^c} - \frac{\partial \Gamma^c_{ac}}{\partial x^b} + \Gamma^d_{ab}\Gamma^c_{cd} - \Gamma^d_{ac}\Gamma^c_{bd}\right)$$

What we see from this is that the scalar curvature is continuous with respect to the metric. So isometric manifolds will have the same extrema. Also, nearly isometric manifolds should have extrema that are close to each other.

With this in mind, we propose the following scheme. These critical points can be ordered from least to greatest. Now because we don't have an explicit error term for the scalar curvature with respect to the metric, it would be unwise to try to measure the 'amount' of isometry between $(\mathcal{M}_1, g_1)$ and $(\mathcal{M}_2, g_2)$. However, if these terms are wildly off, especially in the early terms, we can quite confidently disregard the correspondence.

After this, point correspondences can be calculated between these points of high scalar curvature, based on the justification given in Section 2.4.2. This is to say that the local critical points of scalar curvature can be used as a subset to sample for alignment.

2.5.3 Discrete Scalar Curvature

We'll now give a measure of scalar curvature for graphs. As is often the case in the discrete setting, there are multiple, non-equivalent formulations for scalar curvature.

We will be looking at two notions of Ricci Curvature:

1. Ollivier-Ricci (OR) Curvature [Ollivier 09].

2. Forman-Ricci (FR) Curvature [Forman 03].

Often, notions carried from the smooth setting to the discrete realm, in this case Ricci curvature, capture different qualities about the curvature [CW 17]. However, the relation between Olivier-Ricci and Forman-Ricci curvature is not clear. In the medical imaging setting, both formulations performed similarly. For this reason, we'll recommend use of the FR formulation, since it is easier to compute.

Take $G = (V, \mu)$ to be a graph with edge weights μ, and let $x \in V$. Then the Forman-Ricci curvature of G is given explicitly as

$$\mathrm{Ric}_{FR}(x, y) = \mu_{xy} \left(\frac{\mu(x)}{\mu_{xy}} - \sum_{z \neq y} \frac{\mu(x)}{\sqrt{\mu_{xy}\mu_{xz}}} + \frac{\mu(y)}{\mu_{xy}} - \sum_{s \neq y} \frac{\mu(y)}{\sqrt{\mu_{xy}\mu_{sy}}} \right)$$

Where z is a neighbor of x and s is a neighbor of y [WSJ 16]. From here, we have the Forman-Ricci formulation of scalar curvature

$$S_{FR}(x) = \sum_{y} \mathrm{Ric}_{FR}(x, y)$$

where y is a neighbor of x. Again in this explicit formula, we find that scalar curvature is a continuous function of the metric μ on G.

2.5.4 Critical Points of Eigenfunctions

Recall the spectral distance given earlier by [BBG 94], (7). From this, we may be able to get useful information from the eigenfunctions. This motivates us to compare the zeros and critical points of the jth eigenfunction for each manifold. We hope to infer some result about the relative locations of the zeros and critical points of the eigenfunctions, provided we can find continuity results.

The continuity of the eigenfunctions is only guaranteed if the multiplicity is 1. There is a specific case when this is true.

Theorem 8 ([Uhlenbeck 72])**.** *The eigenspaces of the Laplacian are one dimensional for a generic metric.*

Generic here means that the metric is in the set of residuals of the metrics on a manifold, the complement of which is meager. If g is generic, we are done. Otherwise, since the non-generic metrics are in a meager set of all metrics, g should be able to be made generic by a small perturbation.

Theorem 9 ([Abdalla 12])**.** *Let $(\mathcal{M}, g(t))$ be a family of Riemannian manifolds such that $g(t)$ is analytic in $t \in (0,T)$, and let $\{\phi_j(t)\}_{j\geq 0}$ be a complete orthonormal basis of eigenfunctions of $\Delta_{g(t)}$, with corresponding eigenvalues λ_j, all analytic in $t \in (0,T)$. Then the map*

$$\Phi_t : (\mathcal{M}, g(t)) \to l^2, \quad t \in (0,T)$$
$$x \to \sqrt{2}(4\pi)^{n/4} t^{(n+2)/4} \left\{ e^{-\frac{\lambda_j(t)t}{2}} \phi_j(t,x) \right\}_{j\geq 1}$$

is an embedding for all $t \in (0, T)$ *and*

$$(\Phi_t)^*_{can} = g(t)(V, V) + \frac{t}{3}\left(\frac{1}{2}Scal_{g(t)}g(t) - Ric_{g(t)}\right) + \mathcal{O}(t^2)$$

when $t \to 0_+$

The eigenfunctions $\phi(t, x)$ of $\Delta_{g(t)}$ are continuous with respect to t (since they are analytic with respect to t). With this, we see that it is possible to perturb the metric in a continuous manner, and since g is in a meager set, this perturbation based on t can be made small enough that $g(t)$ is generic so that all eigenvalues of $\Delta_{g(t)}$ have multiplicity one.

We should then expect that the critical points of eigenfunctions of nearly isometric manifolds should be close to each other, or those of perturbed eigenfunctions $g(t)$. Interpreted as a part of our filter, any large deviation between critical points of $\phi_j^{g_1}$ and $\phi_j^{g_2}$ is grounds for disregarding a correspondence. Specifically, if c_i is a critical point of $\phi_j^{g_i}$, $i = 1, 2$, we should see, for some ϵ, the critical values

$$|\phi_j^{g_1}(c_1) - \phi_j^{g_2}(c_2)| < \epsilon$$

If this fails, the correspondence should be reconsidered. Another purpose that this can serve is to evaluate the choice of basis for the embedding. Almost isometric is not the same as isometric. Analyzing the critical points of the eigenfunctions can also be used to evaluate the *choice* of eigenfunctions, specifically whether the choice

for each embedding is reasonable for matching.

Remark: Note that it is necessary here to consider the eigenfunctions, and not simply the eigenvalues. It is possible, and in fact counterexamples have been found, of non-isometric manifolds with identical Laplacian spectrums. Comparing eigenvalues alone will not serve our purpose.

2.5.5 Orthogonal Procrustes Error

We wish to see how close two shapes are to isometric. Our perspective has us looking to align a number of points and then measure the deviation from isometry. We may be able to disregard point correspondences by working from the other direction. Given a set of points X from one shape and Y from the other, we can try to find an orthogonal transformation that most closely matches the points X to the points Y.

Considering matrices X and Y in which the rows of X are the vectors $\boldsymbol{x}$ and the rows of Y are the vectors $\boldsymbol{y}$ which we would like to send them to. Then we phrase the problem: how can we find A that minimizes

$$||X - AY||_F$$

with A orthogonal? This is known as the orthogonal Procrustes problem [Schönemann 66]. The solution to this problem, given by Peter Schönemann in 1964, is described as follows.

Theorem 10 (Orthogonal Procrustes Problem [Schönemann 66])**.** *Let X and Y be*

matrices. Define the matrix $M = YX^t$. *The singular value decomposition is given*

$$M = USV^t$$

Define

$$A = UV^t$$

Then A *minimizes* $||X - AY||_F$ *subject to* $AA^T = I$.

In this way, we can find the orthogonal matrix A that most closely matches the chosen samples Σ_X to Σ_Y. If the deviation between some $\boldsymbol{x} \in \Sigma_X$ and $\boldsymbol{y} \in \Sigma_Y$

$$|\boldsymbol{x} - A\boldsymbol{y}| > \epsilon,$$

then orthogonal transformations are not capable of aligning $\boldsymbol{x}$ with $\boldsymbol{y}$ to a reasonable amount, and the sampled points should be reconsidered. This method is particularly well suited when the number of points being aligned is much smaller than the truncated dimension, as this is an underdetermined problem and there is more flexibility in mapping points.

2.6 k-Point Alignment

Having embedded each mesh into a canonical domain and truncated to a suitable amount, we now come to the point that in Möbius voting corresponds to Möbius matching.

2.6.1 The Shortcoming of Möbius Alignment

It is important to note that Möbius transformations exist in higher than three dimensions. These are typically defined

Definition 14 (General Möbius Transformation [Ratcliffe 06]). *A Möbius Transformation of $\hat{\mathbb{R}}^n$ is a finite composition of reflections of $\hat{\mathbb{R}}^n$ in spheres.*

More interesting for us is the following result:

Theorem 11 (Liouville's Theorem on Conformal Maps in Higher Dimensions [Lie 1871]). *Every conformal mapping in a domain $\Omega \subset \mathbb{R}^N$ for $N \geq 3$ is a Möbius transformation*

The Möbius transform on $\hat{\mathbb{R}}^n$ is capable of matching 3 points with any other 3 points, since the Möbius group is 3 transitive.

Conformal transformations naturally exist in dimension greater than 3. These transformations form a group $O(n,1)$, which is exactly the special orthogonal group with signature $n, 1$. [Hall 03] gives the definition:

Definition 15 (Indefinite Orthogonal Group $O(n,1)$). *Define the bilinear form on $\mathbb{R}^{n+1}$ by*

$$\langle x, y \rangle = x_1 y_1 + x_2 y_2 + \cdots + x_n y_n - x_{n+1} y_{n+1}$$

The set of real matrices A that preserve this form so that $\langle Ax, Ay \rangle = \langle x, y \rangle$ is the special orthogonal group $O(n,1)$.

If A is an $(n+1) \times (n+1)$ matrix, let $A^{(i)}$ denote the ith column of A. Then A is in $O(n,1)$ if and only if the following conditions are satisfied

$$\langle A^{(i)}, A^{(j)} \rangle = 0 \qquad \text{if } i \neq j$$

$$\langle A^{(i)}, A^{(i)} \rangle = 1 \qquad \text{if } 1 \leq i \leq n$$

$$\langle A^{(j)}, A^{(j)} \rangle = -1 \qquad \text{if } j = n+1$$

The bilinear form $\langle x, y, \rangle$ is often called the Lorentz inner product. The group $O(n, 1)$ is sometimes called the general Möbius group, or general Lorentz group.

In higher dimensions, $O(n, 1)$ has dimension $\frac{n(n+1)}{2}$; while apparently a unique transformation can be determined by selecting such a number of points from each manifold, it is found that this transformation does not align more than three points.

Why is this? Part of the problem has to do with our perspective. It is convenient to say that Möbius transformations can match 3 points, but a more general statement is to say that conformal transformations preserve the cross ratio between points. This happens to be convenient for 3 points, since the cross ratio is between 4 points. With 3 points chosen and one left variable, it is possible to match this cross ratio with another while defining a map.

Note that the Möbius transformation hasn't "failed", it just doesn't align points more than 3 points in the manner that we would like. There is still some geometric invariant under this transformation, and have put this to use to simplify the matching problem.

2.6.2 Orthogonal Alignment

Consider the samples from the embedded meshes Σ_1 and Σ_2, containing s_1 and s_2 samples, respectively. Suppose the embedding space is $\mathbb{R}^N$ (we already know that the truncated space is Euclidean, we're just assuming the dimension is N.

First, note that likely $s >> N$. Again, see the truncation lemma. Here we see that the truncated dimension N depends on the geometry of the structure. We don't expect to see much complicated geometry with a few points. Many points are needed to describe a complicated structure on the manifold. The conclusion is that there are (or should be, if we sampled appropriately) far more vertices than the dimension.

The orthogonal group on $\mathbb{R}^N$ has dimension $\frac{N(N-1)}{2}$. We try to force N points on each manifold to be as close to each other as possible using some orthogonal transform A. As long as $N \geq 5$, $2N \leq \frac{N(N-1)}{2}$.

In the most straightforward case, we align N points. Consider a subset of sampled points $\{p_1, p_2, \ldots p_N\} \subset \Sigma_1$ and $\{q_1, q_2, \ldots, q_N\} \subset \Sigma_2$, with the points q_i translated so that $p_1 = q_1$. Then it is a simple matter to find $L_{N\times N}$ so that

$$L[p_1\ p_2\ \ldots\ p_N]^T = [q_1\ q_2\ \ldots\ q_N]^T$$

$$LP = Q$$

Since each point p_i and q_i has N coordinates, the matrices P and Q are square. As long as the matrices P and Q have full rank, these can be inverted so the solution

is given $L = QP^{-1}$. If desired, the deviation from isometry can be calculated

$$||L^T L - I||_F$$

where $|| \cdot ||_F$ is the standard matrix (Frobenius) norm, i.e. the square root of the summation of squares of all entries.

If the number of aligned points is less than the truncated dimension, the matrix solution will not be unique. Then it is worth it to consider the optimization which minimizes

$$||L^T L - I||_F$$

$$\text{Subject to } LP = Q$$

This is looser than N point alignment, which may be use when there are few orthogonal transforms between N points.

2.6.3 Spectral Graph Matching

A robust framework for matching graphs has been developed by [Sharma 12]. This method is based in the work of [Umeyama 88], who has greatly simplified the computation of graph similarity. We'll begin with a discussion of Umeyama's relaxation method.

Let G_1 and G_2 be graphs with weight matrices W_1 and W_2, degree matrices D_1 and D_2, and spectra $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$ and $\mu_1 \leq \mu_2 \leq \cdots \leq \mu_n$ respectively. Let $\mathcal{W}_1 = D_1^{-1}W_1$ and $\mathcal{W}_2 = D_2^{-1}W_2$ be the normalized adjacency matrices, and let P be a permutation matrix on the vertices.

The problem of finding an exact match for graphs can be described as finding a permutation matrix P^* which minimizes

$$||\mathcal{W}_1 - P\mathcal{W}_2P^T|| \tag{8}$$

The issue here is that there are too many choices for P (2^n in fact). To overcome this, begin by noting

$$\begin{aligned}\mathcal{W}_1 - P\mathcal{W}_2P^t &= I - \Delta_1 - P(I - \Delta_2)P^t \\ &= I - \Delta_1 - PIP^t + P\Delta_2P^t \\ &= -(\Delta_1 - P\Delta_2P^t)\end{aligned}$$

So solving (8) is the same as minimizing

$$||\Delta_1 - P\Delta_2P^t||$$

Consider the eigendecomposition

$$\mathcal{W}_1 = U_1 \Lambda_1 U_1^t$$

$$\mathcal{W}_2 = U_2 \Lambda_2 U_2^t$$

Then [Umeyama 88] finds the solution is

$$Q^* = U_1 S U_2^t$$

where S is the diagonal matrix with all entries either 1 or -1. To resolve the sign ambiguity in S, [Umeyama 88] proposes using the matrices $\overline{U}_1$ and $\overline{U}_2$, in which each entry of $\overline{U}$ is the absolute value of U. For the special case in which the graphs are exact, we have the following:

Theorem 12 ([Umeyama 88])**.** *If $\mathcal{W}_1$ and $\mathcal{W}_2$ are isomorphic, the following inequality holds:*

$$Tr(\overline{U}_1 \overline{U}_2 P^T) \leq n$$

and the optimal matching is found by solving the minimization problem

$$Tr(P \overline{U}_1 \overline{U}_2)$$

[Umeyama 88] suggests using the Hungarian Algorithm to find P^*.

In the event that the graphs are not exactly isomorphic (which is to say, most

cases), the approach is slightly different. [Sharma 12] shows that for graphs that are not isomorphic

$$\mathrm{Tr}(U_1 S U_2^T Q^T) \leq \mathrm{Tr}(\overline{U}_1 \overline{U}_2 Q^T) \leq n$$

where Q is a orthogonal matrix (contrast the isometric case, in which the inequality was based on permutation matrix). In the case where graphs are close to isometric, solving the optimization problem in the permutation case gets close to the optimal solution. [Umeyama 88] suggests using this as an initial estimate for a hill climbing algorithm applied to Q if further improvements on the estimate are necessary.

2.6.4 Subgraph Spectral Isomorphism

Graph isomorphism or near isomorphism is an acceptable standard for some situations, but is too rigid for some of the applications that we consider. In addition, solving the eigenvalue problem for large matrices is computationally intractable. As a final issue, the reader may have noticed that the method above relies on each graph having the same number of nodes (isomorphisms are between graphs of the same cardinality). These issues demand a workaround to be useful to us.

[KSMH 09] provides a method for reducing the dimension. Consider two graphs, G_1 and G_2, having m and n vertices respectively. We want to work with k eigenvectors with $k << m, n$. Define the matrix

$$U_x^k = [u_1 \ u_2 \ \ldots \ u_k]$$

i.e. the $N \times k$ truncated matrix where the k columns are k eigenvectors of x. First, ordering eigenvalues and working with the largest will not work due to the presence of multiplicity in the eigenvalues. For this reason, a permutation matrix is added to the minimization problem:

$$Q^* = U_x^k S^k P^k (U_y^k)^T$$

The elements of Q are given by

$$Q_{ij} = x_i^T y_j$$

where x_i is a row of U_x and y_i is a column of $S^k P^k (U_x^k)^T$. Then [KSMH 09] gives the expectation minimization problem, with missing variables α_{ij}, and R^k will be in the orthogonal group, i.e. finding R^k minimizing

$$\sum_{i,j} ||x_i - \alpha_{ij} R^k y_j||^2$$

Full details for implementing this expectation minimization problem are found in [MHKCB 08].

2.7 Measuring Error in Global Matching

The matrix L suggests a correspondence, since the points are chosen at random, there is the possibility that L aligns points that should not have been aligned in the first place. It is necessary to measure deformation of points outside of the sample sets.

The metric that we will use is the Hausdorff distance. Having embedded each shape into $\mathbb{R}^N$, we now consider each mesh in the same space. Recall that for our embedded shapes X and Y, each a subset of Z, the Hausdorff distance is computed

$$d_H^Z(X,Y) = \inf_{\epsilon}\{\epsilon \geq 0; X \subset N_\epsilon(Y), Y \subset N_\epsilon(X)\}$$

where $N_\epsilon(X)$ is the ϵ-tubular neighborhood of X in Z, likewise for $N_\epsilon(Y)$. That is, we want to find the smallest 'thickening' so that X is fully contained within ϵ of Y's boundary, and Y is fully contained within ϵ of X's boundary. One may wish to express the Hausdorff distance more explicitly in terms on the Euclidean distance. The following definition is equivalent to Definition 7 [Henrikson 99]

Definition 16 (Hausdorff Distance (Euclidean expression))**.**

$$d_H^Z(X,Y) = \max\{\sup_{x\in X}\inf_{y\in Y} d(x,y), \sup_{y\in Y}\inf_{x\in X} d(y,x)\}$$

Here, $d(x,y)$ indicates the Euclidean distance between the points x and y, both in the metric space Z, typically a Euclidean space.

While it appears that this is difficult to compute, it is important to keep in mind that the use of filters described in the previous sections will cut this number down significantly.

If two shapes are isometric, the best correspondence will have have Hausdorff error 0. Alternatively, we may say that if two shapes are isometric, the Gromov-Haudorff

distance will be 0. The existing work on Gromov-Hausdorff distance tells us that nearly isometric objects will have a small distance, and not-very isometric objects will have a large distance [Mémoli 07].

This is a good way of measuring isometry if this is what we are interested in. In the following section, we'll show a way to loosen this standard by use of a fuzzy matrix.

2.8 Voting

For shapes that have isometric regions, 'votes' are taken to find such regions. The process of voting rather than finding the minimum Hausdorff error allow for the setting of a threshold for closeness, rather than exact matches. This helps to loosen some of the rigidity of the isometry focused matching to some extent. The procedure here differs from Möbius voting as there is no conformal factor to consider, since our transformations have been nearly isometric.

2.8.1 Finding Correspondences

After aligning N points, we need to see how good this correspondence is. The idea is that there are enough points $x \in X$ and $y \in Y$ so that

$$|x - L(y)| < \epsilon$$

These neighbors give correspondences, ie $c(x) = L(y)$. If two points are not

within ϵ of each other, no correspondence exists for these points. If the number of these correspondences given by L is greater than some threshold K, we compute the 'cost' of this correspondence and add it to a matrix. The cost is given with the simple error calculation

$$E(c) = \sum_k |x - c(x)|$$

Setting a threshold loosens the rigidity in the following way. We may be comparing two shapes that share similar regions, but are not entirely similar. We would expect *some* of these shapes to be isometric, but not others. K is some percentage of the total points; this percentage can be adjusted based on the amount of isometry expected between the objects. In addition, if we use scalar curvature in our filter, we can make sure that the most complicated geometry of the objects is required to be similar, while the less complicated regions have more freedom.

In practice, the size of ϵ and K will be determined by heuristics, although ϵ should at least be as large as the largest error in matching $x - Ly$ for $x \in X$ and $y \in Y$.

2.8.2 Fuzzy Correspondence Matrix

The object which we will use to compare our objects is a matrix $C_{s_1 \times s_2}$, with s being the number of samples. The matrix C is initialized to 0. Each time L is recalculated, correspondences are found, and if the number of correspondences is above K, C is updated by the rule

$$C_{x,y} \leftarrow C_{x,y} + \frac{1}{e + E(c)/n}$$

Here, e is a small positive number.

After a number of iterations, the correspondence matrix provides, for each possible correspondence between x and y, a value predicting the strength of the correspondence. Strong correspondences are points that are most likely in an isometric region. The interested user might find it worthwhile to normalize the matrix to associate a probability for this notion. Note that this process is identical to that described in [LF 09].

3 Discussion

3.1 Sampling Methods

The sampling problem was resolved in our approach by considering the regions of high scalar curvature, corresponding to the importance of a particular point. This certainly has advantages, as we should be comparing more important regions, ie regions of high scalar curvature. On the other hand, this is unlikely to provide uniform sampling.

We also described a Vitali sampling approach that provides something near a uniform sampling. This provides regions between 2ϵ and 6ϵ apart. Obviously, the size of ϵ can be reduced so that the regions are sufficiently uniform, however, this introduces more samples, making this method more complicated and slow to compute. We'll spend some time here exploring several ideas related to the sampling problem.

3.1.1 Nodal Domains and the Extrema of Eigenfunctions

Here we explore the use of using the zeros of eigenfunctions, particularly of high frequency eigenfunctions as sample points.

Let ϕ_j be the eigenfunction of the Laplacian of $(\mathcal{M}, g)$ corresponding to the eigenvalue λ_j.

Definition 17 (Nodal Set [Chavel 84]). *The nodal set of a function $f : M \to \mathbb{R}$ is the set $f^{-1}(0)$.*

Nodal sets are defined for any C^∞ function on $\mathcal{M}$, however, our focus is only on the eigenfunctions. In essence, the Nodal sets for the eigenfunction ϕ are the values x so that $\phi(x) = 0$.

A connected component of the complement of a nodal set is referred to as a nodal domain, in other words, the nodal sets partition $\mathcal{M}$ into a set of nodal domains. For this we have the following

Theorem 13 (Courant's Nodal Domain Theorem [Chavel 84]). *The number n_j of nodal domains of the j^{th} eigenvalue satisfies $n_j \leq j$*

There are results on the size of these zero sets. The most well known is a conjecture from Yau.

Theorem 14 (Yau's conjecture on nodal set's sizes). *Given a compact manifold $(\mathcal{M}, g)$, there exist constants c, C so that*

$$c\sqrt{\lambda_j} \leq vol(\mathcal{N}_{\phi_j}) \leq C\sqrt{\lambda_j}$$

as $j \to \infty$

Note that Yau's conjecture refers to all manifolds. The case for an analytic metric has been proved by [DF 88]. In addition, [Logunov 18 1] proved the lower bound in the general case, as well as improving the previously known upper bound.

Theorem 15 (Logunov's Boundaries on the Size of Nodal Sets [Logunov 18 1], [Logunov 18 2])**.** *Let* $(\mathcal{M}, g)$ *be a compact smooth manifold. Then*

$$c\sqrt{\lambda_j} \leq vol(\mathcal{N}_{\phi_j}) \leq C\lambda_j^{\alpha}$$

where $\alpha > 1/2$ *depends only on* n*, and* $C > 0$*,* $c > 0$ *depend only on the metric on* $\mathcal{M}$*.*

The result above gives something that begins to look like a circle packing on $(\mathcal{M}, g)$, possibly not a very good one. Consider where we stand so far. We have an upper bound on the number of nodal domains, and we know that there are boundaries on the volume of the nodal sets, although we don't know where they are at. What we have is the vague hope that the nodal sets of high frequency eigenfunctions provide a good sampling.

In one particular case, the situation is very good. Ergodic flow has influenced the direction of research in the distribution of eigenfunctions [Zelditch 11], in particular the random wave conjecture [Zelditch 05]. The best case is for manifolds with constant, negative sectional curvature. These have ergodic geodesic flow, and so quantum ergodicity holds.

Theorem 16 (Quantum Ergodicity [Colin de Verdiere 85]). *Given a compact Riemannian manifold $(\mathcal{M}, g)$ with ergodic geodesic flow, there is a subsequence of eigenfunctions $\{\phi_{j_k}\}_k$ of density one, so that for any $D \subset M$,*

$$\lim_{k\to\infty} \int_D |\phi_{j_k}|^2 = \frac{Vol(D)}{Vol(M)}$$

This essentially tells us that high frequency eigenfunctions are equally distributed, if $(\mathcal{M}, g)$ has ergodic flow. As elegant as the result is, its application for our purposes is very limited, and questions pertaining to the distribution of high frequency eigenfunctions remain an area of interest. Realistically, we shouldn't expect eigenfunctions for non-ergodic manifolds to always tend toward uniform distribution. Eigenfunctions for 2 dimensional surfaces have known useful properties, but the study of critical points and critical values of eigenfunctions in higher dimensions remains active.

3.1.2 Sampling from the Truncated Space $\mathbb{R}^N$

Recall from Courant's nodal domain theorem that the number of nodal domains of the j^{th} eigenfunction is less than or equal to j. This tells us that the j^{th} eigenfunction changes sign no more than $j-1$ times. Our truncated embedding of the n-dimensional smooth manifold takes the form

$$\Phi_t^N(x) = \sqrt{2}(4\pi)^{n/4} t^{(n+2)/4} \{e^{-\lambda_j t/2}\phi_j(x)\}_{j=1}^N$$

Observe that the action on x is determined entirely by the eigenfunctions. Based

on Courant's nodal domain theorem, the total number of times that any eigenfunction changes sign is no more than $(n-1)!$. This gives us hope that a suitably small ϵ can be chosen for a uniform sampling from $\mathbb{R}^N$ that does not miss any important information from the embedded shape.

3.1.3 Scalar Curvature and the Second Fundamental Form

There is another approach relying on a special property of the embedding. [WZ 15] (Corollary 38) showed that under heat kernel embedding, the second fundamental form approaches uniform asymptotically with time.

Corollary 1 (WZ)**.** *for any $x \in M$, let $(x_1, \ldots, x_n)$ be the normal coordinates near x. The second fundamental form $A(x,y) = \sum_{1\le i\le j\le n} h_{ij}(x,t)dx^i dx^j$ of the submanifold $\Phi_t(M) \subset l^2$ can be written as*

$$A(x,y) = \frac{1}{\sqrt{2t}}\left(\sum_{i=1}^{n}\sqrt{3}a_{ii}(x,t)(dx^i)^2 + \sum_{1\le j<k\le n} 2a_{jk}(x,t)dx^j dx^k\right)$$

where $a_{jk}(x,t)$ $(1 \le j < k \le n)$ are vectors in l^2. Then as $t \to 0_+$, we have the following

1. *For any two subsets $\{i,j\}$ and $\{k,l\} \subset \{1,2,\ldots,n\}$,*

$$\langle a_{ij}, a_{ij} \rangle \to 1$$

$$\langle a_{ij}, a_{kl} \rangle \to 0 \text{ if } \{i,j\} \neq \{k,l\} \text{ and } \{i,k\} \neq \{j,l\}$$

$$\langle a_{ii}, a_{jj} \rangle \to \frac{1}{3} \text{ if } i \neq j$$

2. *The mean curvature vector* $H(x,t) = \frac{1}{n}\sum_{i=1}^{n} h_{ii}(x,t)$ *after scaling by a factor* $\sqrt{t}$*, converges to constant length:*

$$\sqrt{t}|H(x,t)| \to \sqrt{\frac{n+1}{2n}}$$

The convergence is uniform for all x *on* M *in the* C^r*-norm for any* $r \geq 0$

In this sense, we expect the mean curvature vector length to be close to uniform, and the embedded shape to have no significant regions of more or less geometric complexity, at least with regards to the ambient space. Therefore, we can sample points in the embedding space in a regular manner with little concern.

3.2 Reconstructing the Geometry From the Heat Kernel Embedding

The heat kernel method embeds a manifold into l^2 almost isometrically. Consider that the output of Φ is a set of points. When working with Riemannian manifolds however,

the structure of the points is very important, and here structure is referring to the Riemann curvature tensor. The work of [AT 07] and [Nicolaescu 12] have provided a probabilistic study of Riemannian manifolds, and [Zhu 13] has drawn the connection to the heat kernel embedding.

Definition 18 (Random Field [AT 07])**.** *Let* $(\Omega, \mathcal{F}, \mathbb{P})$ *be a complete probability space and* T *a topological space. Then a measurable mapping* $f : \Omega \to \mathbb{R}^T$ *is called a real-valued random field. Measurable mappings from* Ω *to* $(\mathbb{R}^T)^d$, $d > 1$ *are called vector-valued random fields. If* $T \subset \mathbb{R}^N$, *we call* f *an* (N, d) *random field, and if* $d = 1$, *simply an* N*-dimensional random field.*

The Riemannian metric and curvature tensor can be expressed as an expectation.

Lemma 3 ([AT 07])**.** *If* f *is a zero-mean,* C^2 *random field on a* C^3 *Riemannian manifold equipped with the metric induced by* f, *given as*

$$g(X, Y) = \boldsymbol{E}(Xf, Yf)$$

then the curvature tensor R *on* $\mathcal{M}$ *is given by*

$$-2R = \boldsymbol{E}\left\{\left(\nabla^2 f\right)^2\right\}$$

In general, the random field f used should be Gaussian, for which explicit com-

putation is possible. A random variable X is Gaussian if it has the density function

$$\varphi(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-(x-\mu)^2/2\sigma^2}$$

and a Gaussian field is a random field in which each component $(f_1, f_2, \ldots, f_n)$ is Gaussian.

For our purposes, it makes sense to consider the random function with consideration to a basis, in our case the basis consisting of eigenfunctions of the Laplacian, given independent Gaussian function φ_j, each with mean zero and variance 1 (standard independent Gaussian).

$$f = \sum_j \varphi_j \ \sqrt{2}(4\pi)^{n/4} t^{(n+1)/4} \phi_j$$

i.e., using the heat kernel embedding.

3.2.1 Sectional Curvature by Way of Random Morse Functions

[Nicolaescu 12] gives the following method for finding the sectional curvatures of a manifold.

Let φ_j be standard independent Gaussian random variables, ϕ_j and λ_j be the jth eigenfunction and eigenvalue, respectively. $w(s)$ is an even measurable function satisfying

$$\lim_{s\to\infty} s^n w(s) = 0$$

for all n.

Then random Morse functions are defined

$$u^t = \sum_{j \geq 1} \varphi_j \sqrt{w(t\sqrt{\lambda_j})}\phi_j$$

Define the tensor $\hat{g}^t$ satisfying.

$$\hat{g}^t(X,Y) = \frac{t^{m+2}}{d_m} \boldsymbol{E}(Xu^t, Yu^t), \quad p \in M, \quad X, Y \in \mathrm{Vect}(M)$$

where Xu denotes the derivative of u on the vector X, and $\boldsymbol{E}$ is the expectation. Then the following holds

Theorem 17 (Probabilistic Reconstruction of the Geometry [Nicolaescu 12], Theorem 1.7)**.** *The sectional curvatures on $\hat{g}^t$ converge to the corresponding sectional curvatures of g as $t \searrow 0$*

Note that $w(s)$ can be any fast-decaying function, but we can use $w(s) = e^{-s^2}$, satisfying the condition required for w, and more importantly for our purposes, since φ_j is standard, $w(s)$ has the property that $\sqrt{w(t\sqrt{\lambda_j})}$ gives the coefficients for the heat kernel embedding. Now, we may use $u = \Phi_t$.

The at this point, it is possible to recover the geometry from the heat kernel embedding.

3.2.2 Asymptotic Gauss Formula By Way of the Heat Kernel Embedding

Given a manifold $\mathcal{M}$ in a Euclidean space, with Riemann curvature tensor R and second fundamental form h, the Gauss formula for vector fields X, Y, Z and Y in the tangent space is as follows

$$R(X,Y,Z,W) = \langle h(X,W), h(Y,Z)\rangle - \langle h(X,Z), h(W,Y)\rangle$$

Given Φ_t is an almost isometric embedding into l^2, it is possible to think of l^2 as an ∞-dimensional Euclidean space, the analogue result follows:

Proposition 1 (Riemannian Curvature [Zhu 13])**.** *For any X, Y, Z, W of TM, we have*

$$R(X,Y,Z,W) = \lim_{t\to 0^+} [\langle \nabla_X \nabla_W \phi_t, \nabla_Y, \nabla_Z \phi_t\rangle - \langle \nabla_X \nabla_Z \phi_t, \nabla_Y \nabla_W \phi_t\rangle]$$

Here is clearly seen the relation between eigenfunctions and the Riemann curvature tensor. From here it is possible to recover the Ricci curvature and the scalar curvature.

3.3 The Conformal Model

The Möbius voting method favors the conformal model due to its ease of calculation. We can't this method in dimension higher than 3 as it does not align more than three points. The approach itself is very elegant however, and more so with the use of Clifford numbers.

Clifford algebras are quite broad. Here, we will take a less abstract approach than is common and consider the n dimensional Clifford algebra over $\mathbb{R}$, denoted as C_n. Several analogies can be put together before the general definition. The Clifford algebra C_2 is $\mathbb{C}$, with orthonormal basis $\{1, i\}$. The Clifford algebra C_3 is $\mathbb{H}$, the quaternions, with orthonormal basis $\{1, i, j, k\}$. We don't consider octonions, as the step from quaternions to octonions loses associativity.

Now we move to the definition.

Definition 19 (Clifford Algebra over $\mathbb{R}$, [LM 90])**.** *The Clifford algebra C_n is the associative algebra over $\mathbb{R}$ generated by elements $i_1, \ldots, i_{n-1}$ subject to the relations*

1. $i_j i_k = -i_k i_j \qquad j \neq k$

2. $i_j^2 = -1$

There are two structures that are considered under this algebra. The first we consider are vectors, $x \in \mathbb{R}^n$, of the form

$$x = x_0 + x_1 i_1 + x_2 i_2 + \cdots + x_{n-1} i_{n-1}, \quad x_0, x_1, \ldots, x_{n-1} \in \mathbb{R}$$

The second type are numbers $a \in C_n$, the Clifford numbers. Define the product $I = i_{j_1} i_{j_2} \ldots i_{j_p}$, with $0 < j_1 < j_2 < \cdots < j_p < n$. Then the Clifford number has the representation

$$a = \sum_I a_I I$$

3.3.1 The Clifford Group

We consider three involutions

1. **Main Involution** If $a \in C_n$, the main involution $\hat{a}$ of a is the action of replacing every i_k with $-i_k$.

2. **Second Involution**

   The second involution a^t of a is the action of reversing the order of terms in $I = i_{j_1} i_{j_2} \dots i_{jp}$

3. **Third Involution**

   The third involution is obtained by combining the action of the main and second involution $\overline{a} = \hat{a}^t$.

A nonzero vector can be inverted

$$x^{-1} = \frac{\overline{x}}{|x|^2}$$

The product of invertible vectors is invertible. With this notion we can classify an important group on the Clifford Algebra.

Definition 20 (Clifford-Lipshitz Group [VdR, 16])**.** *The subgroup $\Gamma_n < C_n$ consists of invertible $a \in C_n$:*

$$\Gamma_n = \{a \in \{ \text{ invertible elements of } C_n\} | axa^{-1}, \forall x \in \mathbb{R}^n\}$$

It is useful to note that if $a \in \Gamma_n$, then $|a|^2 = \sum a_I^2 = a\overline{a}$

Now we come to the main part. We can use the familiar form for Möbius transformations, where a, b, c, d are in the Clifford-Lipschitz group:

$$g = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

The action on $x \in \mathbb{R}^n$ is defined by

$$gx = (ax + b)(cx + d)^{-1}$$

The following theorem is given by Ahlfors.

Theorem 18 ([Ahlfors 84]). *1. $x \to gx = (ax + b)(cx + d)^{-1}$ with $a, b, c, d \in \Gamma_n \cup \{0\}$ is a bijective mapping $g : \overline{\mathbb{R}}^n \to \overline{\mathbb{R}}^n$ if and only if $a\hat{c}), b\hat{d} \in \mathbb{R}^n$ and $\det(g) = a\hat{d} - b\hat{c} \in \mathbb{R}$*

2. This mapping extends to a bijection $g' : \overline{\mathbb{R}}^{n+1} \to \overline{\mathbb{R}}^{n+1}$ by adjoining i_n to C_n. If $\det(g) > 0$, the upper half-space U^{n+1} is mapped on itself.

3. These mappings are conformal, and so are given by Möbius transformations in $M(\mathbb{R}^n)$ and $M(U^{n+1})$ respectively. Conversely, every sense preserving Möbius transformation can be expressed in the form $gx = (ax + b)(cx + d)^{-1}$ with $\det(g) > 0$.

In this way, the general Möbius transformations have the same simple representation as in the typical case in $\mathbb{C}$. However, it is important to consider that in spite

of the elegance of this form, computationally there are some difficulties, among them determining what falls in the Clifford group, as well as the combinatorial issue of expressing I for high values of n.

3.4 The Yamabe Problem

The Riemann mapping theorem posits the existence of a conformal map from a 2 dimensional surface to the unit sphere or unit disk. [Ahlfors 84] gives the theorem:

Theorem 19 (Riemann Mapping Theorem)**.** *Given any simply connected region* Ω *which is not the whole plane, and a point* $z_0 \in \Omega$*, there exists a unique analytic function* $f(z)$ *in* Ω *normalized by the conditions* $f(z_0) = 0$ *and* $f'(z_0) > 0$ *such that* $f(z)$ *defines a one-to-one mapping of* Ω *onto the disk* $|w| < 1$*.*

This is the theoretical foundation for the uniformization in Möbius voting, expressed by Pinkall's Harmonic maps. A more general result is the uniformization theorem

Theorem 20 (Uniformization Theorem [Poincaré 1907])**.** *Every simply connected Riemann surface is conformally equivalent to the Riemann sphere, the complex plane, or the unit disk*

As we've mentioned before, this is not possible in general for surfaces of higher genus. The question of whether a uniformization-type result exists in higher dimensions has been asked earlier.

The Yamabe Problem Given a compact Riemannian manifold $(\mathcal{M}, g)$ of dimension $n \geq 3$, is there a metric conformal to g so that the scalar curvature is constant?

Several individuals worked to attain this result. The approach concerns the Yamabe Invariant

Definition 21 (Yamabe Invariant)**.** *Given a metric g, and a metric conformal, $\tilde{g}$, denote the respective scalar curvatures S and $\tilde{S}$. Define*

$$Q(\tilde{g}) = \frac{\int_M \tilde{S} dV_{\tilde{g}}}{\left(\int_M dV_{\tilde{g}}\right)^{2/p}}$$

Then the Yamabe Invariant is defined

$$\lambda(M) = \inf\{Q(\tilde{g}) : \tilde{g} \text{ conformal to } g\}$$

[Yamabe 60], [Trudinger 68], and [Aubin 76] proved the result for compact manifolds with $\lambda(M) < \lambda(S^n)$.

[Aubin 76] showed that for manifolds of dimension greater than 6, $\lambda(M) < \lambda(S^n)$

[Schoen 84] completed the proof by showing that manifolds of dimension 3, 4 or 5 have $\lambda(M) < \lambda(S^n)$.

The final result gives:

Theorem 21 (Yamabe Problem)**.** *For any compact Riemannian manifold, there ex-*

ists a function f so that the metric

$$\tilde{g} = e^{2f} g$$

has constant scalar curvature.

This gives a theoretical foundation to a conformal approach for higher dimensions. Given a compact Riemannian manifold, it is possible to deform the metric conformally so that it has constant scalar curvature.

4 A Simple Example

Here we'll provide a small example of the discrete case. Consider the two graphs, G_2 and G_2, embedded in $\mathbb{R}^4$, with coordinates

G_1	
a	(.1, 2.4, -.3, 1.5)
b	(.6, 1.9, .1, 2.1)
c	(.3, 2.6, 0, 2)
d	(3.1, -5.7, .8, .1)
e	(2.6, -5.5, 1.2, -.3)
f	(2.5, -5.5, 1.2, .2)
g	(.5, -.3, .3, 1.9)
h	(-.1, -.1, .4, 2.2)
i	(3.9, -7.1, 1.3, -7)
j	(3.1, -6.9, 2.4, .7)
k	(-.1, 3, -.8, -2.4)
l	(-.5, 3.6, -1.1, -2.8)
m	(-5.2, 6.1, -2.8, -3.9)

G_2	
a	(4.1, 3.2, -5.6, 1)
b	(3.7, 2.3, -5.5, 1.3)
c	(3.5, 3, -5.7, 1.2)
d	(8.5, -2.2, -7.2, 5.3)
e	(8.9, -2.7, -8.8, 6.8)
f	(13.3, -7.1, -14.2, 9.7)
g	(5.1, 1.4, -6.1, 2.9)
h	(4.67, 1.02, -5.9, 2.5)
i	(14.2, -8.3, -15.6, 9.9)
j	(14.6, -6.9, -14.8, 8.6)
k	(3.9, 2.9, -5.5, 2.6)
l	(5.1, 1, -2.6, -3)
m	(6.1, .5, .1, -2.6)

Each having the configuration:

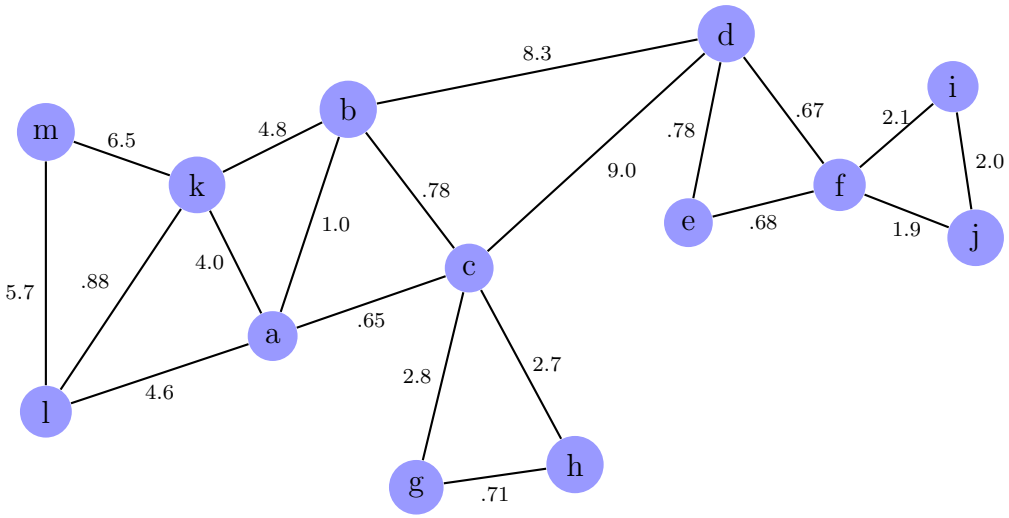

Figure 3: Edge Diagram of G_1

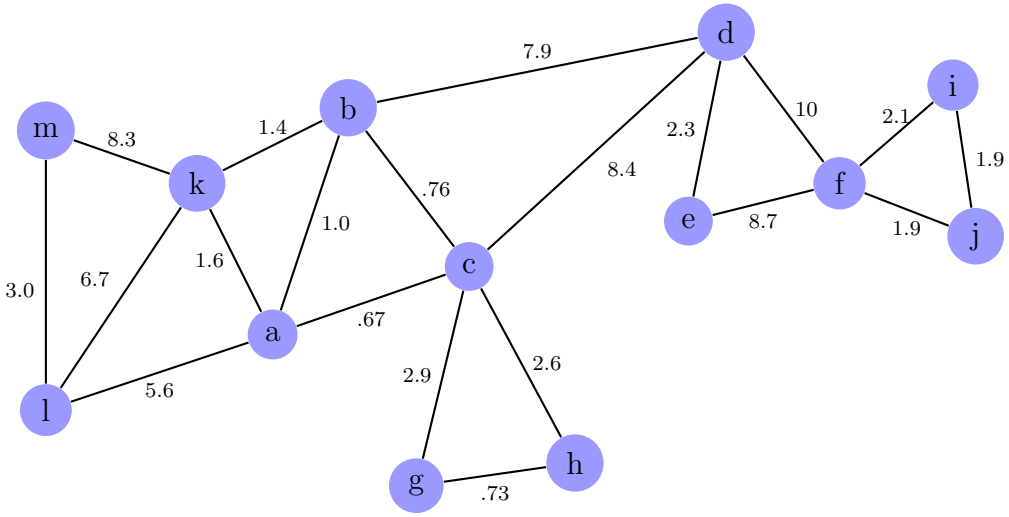

Figure 4: Edge Diagram of G_2

We first embed the graph in 3 dimensions, using the eigenmap framework. Voting is performed for all possible alignments of 3 vertices (3-alignment because that is the embedding dimension). We set the threshold K to be 6, and ϵ to be .3. The following is the fuzzy correspondence matrix

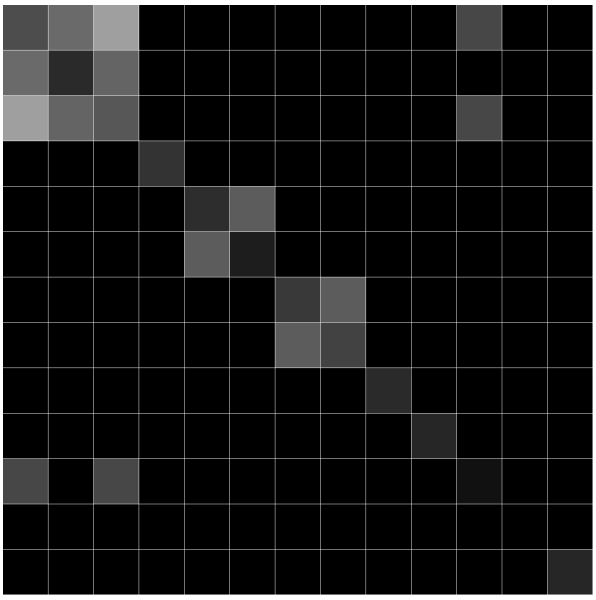

Figure 5: Fuzzy Correspondence Matrix

Here we see strong correspondences in the regions where the edge lengths in each graph are similar, in particular the a, b, c region and the g, h areas. The weakest (or nonexistent) correspondences are in the k, l, m region, in which the distortion in edge length is the greatest.

References

[Abdalla 12] Abdalla, H. (2012). *Embedding Riemannian Manifolds Via their Eigenfunctions and their Heat Kernel*. Korean Math. Society, B, 5.

[AT 07] Adler, R. Taylor, J. (2007). *Random Fields and Geometry*. Springer Monographs in Mathematics.

[Ahlfors 53] Ahlfors, L.V. (1953). *Complex Analysis*. International Series in Pure and Applied Mathematics.

[Ahlfors 84] Ahlfors, L. (1984). *Old and New in Möbius Groups*, Annales Academiae Scientaiarum Fennicae, 9.

[Aubin 76] Aubin, T. (1976). *Équations Différentielles Non Linéaires et Probléme de Yamabe Concernant la Courbure Scalaire*. J Math Pures Appl, 55, pp 66-72

[BN 02] Belkin, M. Niyogi, P. (2002). *Laplacian Eigenmaps and Spectral Techniques for Embedding and Clustering*. Advances in neural information processing systems, 14,6 pp 585-591.

[BBG 94] Bérard P. Besson, G. Gallot S. (1994). *Embedding Riemannian Manifolds By Their Heat Kernel*. Geometric and Functional Analysis, Vol. 4, No.4, pp 373-398.

[Chan 14] Chan, H.L, Lui, L.M, (2014). *Detection of n-Dimensional Shape Deformities Using n-Dimensional Quasi-Conformal Maps*. Geometry, Imaging and Computing, 1,4, pp 395-415.

[Chavel 84] Chavel, I. (1984). *Eigenvalues in Riemannian Geometry*. Academic Press Inc.

[Chung 97] Chung, F. (1997). textitSpectral Graph Theory. Regional Conference Series in Mathematics.

[Colin de Verdiere 85] Colin de Verdiere, Y. (1985). *Ergodicité et Functions Propres du Laplacien*. Communications in Mathematical Physics, 102 pp 497-502.

[CH 89] Courant, R. Hilbert, D. (1989). *Methods of Mathematical Physics*. Wiley, p132. (Original work published 1937).

[CW 17] Crane, K. Wardetsky, M. (2017) *A Glimpse into Discrete Differential Geometry*. Notices of the AMS, Nov 2017, pp 1153-1159.

[DF 88] Donnelly, H, Fefferman, C. (1988). *Nodal sets of Eigenfunctions on Riemannian Manifolds*. Inventiones Mathematicae, 93, pp 251-262.

[Forman 03] Forman, R. (2003). *Bochner's Method for Cell Complexes and Combinatorial Ricci Curvature*. Discrete and Compuational Geometry, 29, pp 323-374.

[Gromov 81] Gromov, M. (1981). *Groups of Polynomial Growth and Expanding Maps*. Publications Mathématiques de l/IHÉS, 53. pp 53-78.

[Hall 03] Hall, B. (2003). *Lie Groups, Lie Algebras and Representations, An Elementary Introduction*. Springer, Graduate Studies in Mathematics.

[Henrikson 99] Henrikson, J. (1999). *Completeness and Total Boundedness of the Hausdorff Metric*. MIT Undergraduate Journal of Mathematics.

[HW 53] Hoffman, A, Wielandt, H. (1953). *The Variation of the Spectrum of a Normal Matrix*. Duke Math. J, 20, pp 37-39

[KSMH 09] Knossow, D. Sharma, A. Mateus, D. Horaud, R. (2009) *Inexact Matching of Large and Sparse Graphs Using Laplacian Eigenvectors*. Graph-Based Representations in Pattern Recognition, 7th International Workshop, pp 144-153.

[LM 90] Lawson, B. Michelsohn, ML. (1990). *Spin Geometry*. Princeton Mathematical Series Vol 38.

[Lie 1871] Lie, S. (1871) *Ueber diejenige Theorie eines Raumes mit beliebig vielen Dimensionen, die der Krümmungs-Theorie Des gewöhnlichen Raumes entspricht*. Nachr. Akad. Wiss Götingen. Math-Phys. 1871, pp 191-209.

[LF 09] Lipman, Y. Funkhouser, T. (2009). *Möbius Voting for Surface Correspondence*. ACM Transactions on Graphics, 28,3 pp 1-12

[Logunov 18 1] Logunov, A. (2018). *Nodal Sets of Laplace Eigenfunctions: Proof of Nadirashvili's Conjecture and of the Lower Bound in Yau's Conjecture*. Annals of Mathematics, 187, 1, pp 241-262.

[Logunov 18 2] Logunov, A. (2018) *Nodal Sets of Laplace Eigenfunctions: Polynomial Upper Estimates of the Hausdorff Measure*. Annals of Mathematics, 187,1, pp 221-239.

[MHKCB 08] Mateus, D, Horaud, R. Knossow, D. Cuzzolin, F. Boyer, E. (2008) *Articulated Shape Matching Using Laplacian Eigenfunctions and Unsupervised Point Registration*. IEEE Conference CVPR.

[Mémoli 07] Mémoli, F. (2007). *On the use of Gromov-Hausdorff Distances for Shape Comparison*. Eurographics Symposium on Point-Based Graphics.

[Möbius 1855] Möbius, A.F. (1855). *Die Theorie der Kreisverwandtschaft in rein geometrischer Darstellung*. Abh. König. S achs. Ges Wiss. Math-Phys Kl 2, pp 529-595.

[Nicolaescu 12] Nicolaescu, L. (2012). *Random Morse Functions and Spectral Geometry*. Preprint, arXiv:1209.0639

[Ollivier 09] Ollivier, Y. (2009). *Ricci Curvature of Markov Chains on Metric Spaces*. Journal of Functional Analysis, 256, pp 810-864.

[Poincaré 1907] Poincaré, H. (1907). *Sur l'uniformisation des fonctions analytiques*. Acta Mathematica, 31, pp 1-36.

[Portegies 13] Portegies, J.W, (2013). *Embeddings of Riemannian manifolds with heat kernels and eigenfunctions*. Communications on Pure and Applied Mathematics, 69,3, pp 478-518.

[Ratcliffe 06] Ratcliffe, J. (2006). *Foundations of Hyperbolic Geometry*. Springer, Graduate Studies in Mathematics.

[Schoen 84] Schoen, R. (1984). *Conformal Deformation of a Riemannian Metric to Constant Scalar Curvature*. Differential Geometry, J, 20, pp 479-495.

[Schönemann 66] Schönemann, P. (1966). *A Generalized Solution of the Orthogonal Procrustes Problem*. Psychometrika, 31, pp 1-10.

[Schwertfeger 1962] Schwertfeger, H. (1962). *Geometry of Complex Numbers*. University of Toronto Press.

[Sharma 12] Sharma, A. (2012). *Representation, Segmentation and Matching of 3D Visual Shapes using Graph Laplacian and Heat Kernel*. https://tel.archives-ouvertes.fr/tel-00768768

[Tao 11] Tao, T. (2011). *An Introduction to Measure Theory*. American Mathematical Society.

[Trudinger 68] Trudinger, N. (1968). *Remarks Concerning the Conformal Deformation of Riemannian Structures on Compact Manifolds*. Annali della Scuola Noramale Superiore di Pisa, 3,22, pp 265-274.

[Uhlenbeck 72] Uhlenbeck, K. (1972). *Eigenfunctions of Laplace Operators*. Bulletin of the American Mathematical Society, 78,6 pp 1073-1076.

[Umeyama 88] Umeyama, S. (1988). *An Eigendecomposition Approach to Weighted Graph Matching Problems*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 10,5, pp 695-703.

[VdR, 16] Vaz, J. da Rocha, R. (2016). *An Introduction to Clifford Algebras and Spinors*. Oxford University Press.

[WZ 15] Wang, X. Zhu, K. (2015). *Isometric Embeddings Via Heat Kernel*. Differential Geometry, J , 99. pp 497-538.

[WSJ 16] Weber, M. Saucan, E. Jost, J. (2016). *Characterizing Complex Networks with Forman-Ricci Curvature and Associated Geometric Flows*. Complex Networks, J, 5, 4, pp 527-550.

[Yamabe 60] Yamabe, H. (1960). *On a Deformation of Riemannian Structures on Compact Manifolds*. Osaka Math, J, 12, 21-37.

[Ye 94] Ye, R. (1994). *Global Existence and Convergence of Yamabe Flow*. Differential Geometry, J, 39, 35-50.

[Zelditch 08] Zelditch, S. (2008). *Local and Global Analysis of Eigenfunctions on Riemannian Manifolds*. Handbook of Geometric Analysis, 1, pp 545-658.

[Zelditch 11] Zelditch, S. (2011). *Eigenfunctions and Nodal Sets*. Surveys in Differential Geometry, 18, pp 237-308.

[Zelditch 05] Zelditch, S. (2005). *Quantum Ergodicity and Mixing of Eigenfunctions* Encyclopedia of Mathematical Physics, 183-196.

[Zhu 13] Zhu, K. (2013). *High-Jet Relations of the Heat Kernel Embedding Map and Applications*. Preprint, arXiv:1308.0410