Preliminary Examination

Discovering Metrics and Scale Space

26 July 2010

Brittany Terese Fasy
Dr. Herbert Edelsbrunner, Adviser
Dr. Hubert L. Bray, Committee Member
Dr. John Harer, Committee Member
Dr. Carlo Tomasi, Committee Member

Contents

1. Introduction

Many researchers are interested in quantifying and comparing shapes in space. In this context, we consider metrics and scale space. Specifically, we focus on the Fréchet Distance between curves and the Vineyard Metric between images. When considering the distance between two images, we define the *heat equation homotopy*, which uses the scale space of the difference between two images in order to create a discrete homotopy between the images.

In Section 3.5, we present an inequality that relates the Fréchet distance, the total length, and the total curvature of curves. The inequality bounds the difference of the length of closed curves by $4/\pi$ times the sum of the total curvatures multiplied by the Fréchet distance between the curves (Theorem 3.10). This bound is independent of the dimension of the ambient Euclidean space, it improves upon a bound by Cohen-Steiner and Edelsbrunner [24], and it generalizes a result by Fáry [40] and Chakerian [22]. The contribution of this inequality is that it removes the dependence on the dimension in which the curves are embedded, a factor on which the previous results depended. To conclude Section 3, we raise several interesting open questions regarding the Fréchet distance and the Vineyard Metric.

In Section 4, we explore the deep structure of images and the heat equation homotopy. The deep structure, or scale space, of an image f is a family of images, parameterized by t, that describe f at different scales. With $t = 0$, we have the original image f, and as t increases, f blurs. The blurring is governed by the heat equation, computed by convolving the image with the Gaussian kernel. We extend the study of scale space of one image to the study of the difference between two images f and g. In particular, we define the heat equation homotopy, which uses the scale space of the difference between images in order to create a discrete homotopy between the images. We can use the Vineyard Metric to measure the difference between f and g. Traditionally, the scale space is restricted to the investigation of the structure of grayscale images. As an open question, we ask how to extend the concept of scale space to color images. We wish to study the set of scale spaces obtained from the intensities of each color. Individually, the scale spaces may witness different structures present in the image. However, it is also interesting to consider the interaction between different colors as the scale parameter t increases.

2. Topology

Observing patterns and features in data sets is a common goal in many disciplines. However, extracting the key features from a noisy data set can be an ambiguous task, and often involves simplifying and finding the best view of the data. Computational topology, and more specifically persistent homology, is a set of tools used for data analysis. Here, we give a brief review of the necessary background of computational topology, but refer to [34], [35] and [46] for more details.

2.1. **Background.** This prelim focuses on curves in Euclidean space and the heat equation acting on manifolds. Before defining persistence homology, we will define manifolds and simplicial complexes, which are used to approximate manifolds in computations.

Manifolds. A *paracompact* space is a topological space such that every open cover has a locally finite refinement. A *Hausdorff* space is a topological space such that for every two points $a \neq b$, there exist open sets $a \in A$ and $b \in B$ such that $A \cap B = \emptyset$. A smooth (real) m-manifold $\mathbb{M}$ is a paracompact, Hausdorff space such that each point is contained in a neighborhood homeomorphic to an open subspace of $\mathbb{R}^m$. We sometimes use $\mathbb{M}^m$ to specify that $\mathbb{M}$ is an m-manifold[1]. Most real-world objects can be modeled by manifolds (or manifolds with boundaries). Although manifolds give a clean and concise characterization of a space (such as a 3D object), it is often difficult to obtain a mathematical description of the manifold, yet alone to perform computations on it. Thus, we use simplicial complexes to approximate the sometimes unknown manifolds.

Simplicial Complexes. An *n-simplex* is the convex hull of $n+1$ vertices such that no vertex can be removed without changing the convex hull. For example, a 1-simplex is a line segment and a 2-simplex is a triangle. If σ is an n-simplex, then any subset of the vertices of σ defines a *face* of σ. We use $\tau \leq \sigma$ to denote that τ is a face of σ. If the simplex contains exactly n vertices, then it is a *co-dimension* 1 *face*. A *simplicial complex* K is a collection of simplices such that each face of simplex $\sigma \in K$ is also in K and the intersection of two simplices in K is either a common face or empty. The dimension of a simplicial complex is the largest size of any simplex contained in it. In order to approximate a manifold with a simplicial complex, we choose sample points on the manifold and create the complex from that set of points.

We define the *mesh* of a simplicial complex K as the maximum distance between any two points that belong to the same simplex. We write $N(r)$ for the minimum number of simplices needed to triangulate K with mesh at most r.

Functions. Often, the space that we are interested in is equipped with a function $f\colon \mathbb{M} \to \mathbb{R}$. If $\mathbb{M} = \mathbb{R}$, we can think of f as a height function of a graph, as depicted in Figure 1. In the case of images, the function is an intensity function.

There are three special types of functions that we will use: Lipschitz functions, Morse functions, and tame functions. If there exists a real number λ such that for all x, y in $\mathbb{M}$,

$$|f(x) - f(y)| \leq \lambda d_{\mathbb{M}}(x, y),$$

[1] Appendix A contains a table summarizing the notation used in this document.

where $d_{\mathbb{M}}(x-y)$ is the distance between the two points in $\mathbb{M}$, then f is said to be *Lipschitz*. In this case, λ is the *Lipschitz constant* of f.

If f is a smooth real-valued function over $\mathbb{M}$, then a point $p \in \mathbb{M}$ is a *critical point* if the derivative of f in every direction vanishes. The value $f(p)$ is then a *critical value* of f. A *Morse function* is a smooth function, such that no two critical points share a critical value, and the Hessian matrix, which is the matrix of second derivatives in local coordinates, is non-singular at every critical point [52].

The *sublevel set* of a function is defined as:

$$\mathbb{M}_s^f = f^{-1}((-\infty, s]).$$

We sometimes write $\mathbb{M}_s$, if it is understood which function we are using. Similarly, we define the *superlevel set* as $\mathbb{M}^s = f^{-1}([s, \infty))$. The function f is said to be a *tame function* if the homology groups of $\mathbb{M}_s$ have finite rank for all s, and only a finite number of homology groups (1) are realized as $H_p(\mathbb{M}_s^f)$ [34].

We use a 2D image as an example of a function from $\mathbb{M}$ to $\mathbb{Z}_{256}^c$, where $\mathbb{M}$ is a rectangular subset of $\mathbb{R}^2$ and c is the number of colors used to represent the image. We use $c = 1$ for grayscale images and $c = 3$ for typical color images. The three values of a color image usually represent the red, green, and blue intensities. To represent the image digitally, we impose a regular grid over $\mathbb{M}$. We add diagonal edges in order to triangulate the domain for the persistence computation, although we ignore the diagonal edges when computing the scale space of an image.

Filtrations. Suppose that we have a simplicial complex K and there exists a function f on the vertices of the simplicial complex. A simplex σ is assigned a value $f(\sigma)$ by taking the maximum function value of the vertices that define σ. Now, we order the simplices by the following two rules:

1. If $f(\sigma_1) < f(\sigma_2)$, then σ_1 appears before σ_2.
2. If τ is a face of σ, then τ appears before σ.

We observe here that the second rule does not contradict the first, because $f(\tau) \leq f(\sigma)$ whenever τ is a face of σ. The resulting ordering of the simplices is called a *filter*. The two rules imply that every initial subsequence of the filter defines a subcomplex of K. Growing this initial subsequence until it equals the entire filter gives a sequence of simplicial complexes called the induced *filtration*. As with any sorting algorithm, to create a filter of m simplices has time complexity $O(m \log m)$. We also notice that the filtration is not necessarily unique, as it is possible for two simplices to have equal values. We will discuss later the implications of changing the order of the simplices in the filter.

For example, we define a Ripps filtration over a point set $\mathcal{S}$. A point set has *diameter* 2α if the radius of the smallest enclosing sphere is α. For a given real number α, the Ripps Complex $\mathcal{R}_\alpha(\mathcal{S})$ is the set of simplices defined by subsets of $\mathcal{S}$ with diameter less than 2α. The Ripps filtration is the sequence of Ripps complexes $\{\mathcal{R}_\alpha(\mathcal{S})\}$ created by increasing α from 0 to infinity.

2.2. **Homology.** Let X be a simplicial complex of dimension d. For $p \in \mathbb{N}$ and $p \leq d$, the symbol X_p denotes the power set of all p-simplices in X. Each set of X_p is called a *p-chain.* The *chain group* C_p is defined by the set X_p under the disjoint union, or symmetric difference, operation. This operation can be interpreted as addition modulo two. Consider the boundary homomorphism :

$$\partial_p : C_p \to C_{p-1},$$

which maps the p-chain $\alpha \in C_p$ to the boundary of α, a chain in C_{p-1} [46]. The *kernel* of ∂_p is the set of elements in the domain that are evaluated to zero (the empty set) and the image of ∂_{p+1} is the set of elements of the form $\partial_{p+1}(x)$, where x is in the domain C_{p+1}:

$$\mathrm{Ker}(\partial_p) = \{\alpha \in C_p | \ \partial_p(\alpha) = \emptyset\} \text{ and}$$
$$\mathrm{Im}(\partial_{p+1}) = \{\alpha \in C_p | \ \exists \alpha' \in C_{p+1} \ni \partial_{p+1}(\alpha') = \alpha\}.$$

The p^{th} homology group of X, denoted $H_p(X)$, is the kernel of ∂_p modulo the image of ∂_{p+1}:

$$H_p(X) = \mathrm{Ker}(\partial_p)/\mathrm{Im}(\partial_{p+1}). \tag{1}$$

The p^{th} *Betti number*, β_p, is the rank of the p^{th} homology group of X. By definition, the *rank* of a group is the (smallest) number of generators needed to define the group up to isomorphism. Since we are concerned with groups with $\mathbb{Z}_2$ coefficients, the rank uniquely defines the group up to isomorphism. For example, the group with three generators is $\mathbb{Z}_2^3 = \mathbb{Z}_2 \oplus \mathbb{Z}_2 \oplus \mathbb{Z}_2$ and the group with m generators is $\mathbb{Z}_2^m$.

2.3. **Persistent Homology.** Now, we define persistent homology for functions from $\mathbb{R}$ to $\mathbb{R}$. The complete discussion of the extension of these ideas to higher dimensions is found in [34] and [35]. We present a simplified setting to focus on the relevant concepts, while avoiding the complications that arise in the general setting.

Let f be a real valued function $f \colon \mathbb{M} \to \mathbb{R}$, defined on a manifold $\mathbb{M}$. We characterize the topology of the sublevel set $\mathbb{R}_s^f$, and we monitor how the homology groups change as the parameter s increases from negative infinity to positive infinity. The zeroth persistent homology group, denoted $H_0(\mathbb{R}_s^f)$, will change whenever s is a local maximum or a local minimum of the function f. The maxima and the minima are where the Betti numbers as well as the homology classes change for functions from $\mathbb{R}$ to $\mathbb{R}$. For a Morse function f, the critical points are the set of $r \in \mathbb{R}$ with $s := f(r)$, such that a Betti number changes by exactly one from $\mathbb{R}_{s-\epsilon}^f$ to $\mathbb{R}_{s+\epsilon}^f$ for every sufficiently small value of $\epsilon > 0$. If the sum of the Betti numbers increases, we call r a positive critical point. If the sum decreases, then r is a negative critical point. As s increases from negative infinity, we label each new component

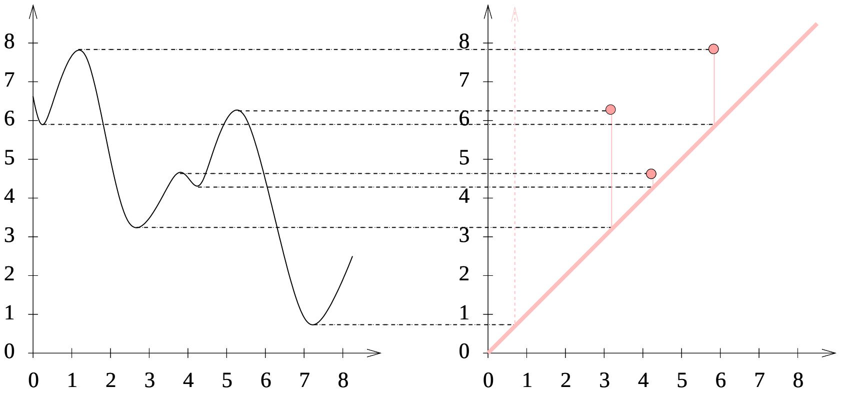


Figure 1: On the left, we see the graph of a function f in $\mathbb{R}^2$. On the right is the corresponding persistence diagram, $\mathrm{Dgm}_0(f)$. Each point is drawn with multiplicity one. The birth at the critical point $(7.1, 0.8)$ remains unpaired.

with the value of the positive critical point that introduces the component. We pair each negative critical point with value s with the most recently discovered unpaired positive critical point representing the components joined at s. This will lead to the diagram $\mathrm{Dgm}_0(f)$ as demonstrated in Figure 1. Consider one pair: the positive critical value that was introduced at time $s = 3.2$ and the negative critical value that was introduced at $s = 6.2$. This pair is represented in $\mathrm{Dgm}_0(f)$ as the point $(3.2, 6.2)$. The *persistence* of that pair is equal to the difference in function values: $6.2 - 3.2 = 3$. In Figure 1, there are three pairs of points and one positive critical value that remains unmatched. This value represents an *essential* homology class. An essential homology class is nontrivial in the topological space $\mathbb{M}$. It would be paired if we were to consider the extended persistence diagram as presented in [26].

Total Persistence. We have just discussed the construction of the 0^{th} persistence diagram, which records the births and deaths of $H_0(\mathbb{M}_s)$ as s increases. The births and deaths of $H_k(\mathbb{M}_s)$ are recorded in the k^{th} persistence diagram, $\mathrm{Dgm}_k(f)$. We let $Dgm(f)$ be the overlay of all persistence diagrams for f. The degree k total persistence is the sum of the k^{th} powers of persistence over all points in the persistence diagram:

$$\mathrm{Pers}_k(f) = \sum_{(x,y)\in\mathrm{Dgm(f)}} (y-x)^k \tag{2}$$

Sometimes we are interested in the total persistence over all points with persistence at least ϵ, for some $\epsilon \geq 0$. In this case, we use $\mathrm{Pers}_k(f, \epsilon)$ to denote the sum in (2) restricted to the points with persistence greater than ϵ. We justify this simplification by interpreting points with persistence less than ϵ as noise.

Recall that $N(\epsilon)$ is equal to the minimum number of simplices needed to triangulate $\mathbb{M}$ with mesh at most ϵ. This number and the Lipschitz constant λ are used to prove an upper bound for the number of points in the persistence diagram of f that have persistence at least $\epsilon > 0$.

Lemma 2.1. (Persistent Cycle [27]). *The number of points in* $\mathrm{Dgm}(f)$ *whose persistence exceeds* ϵ *is at most* $N(\epsilon/\lambda)$*, where* λ *is the Lipschitz constant.*

The amplitude of f is the maximum difference in function values, $\text{Amp}\ (f) = max_{(x,y)\in(\mathbb{M}\times\mathbb{M})}(f(x) - f(y))$. We use the Persistent Cycle Lemma to obtain an upper bound on the degree k total persistence.

Lemma 2.2. (Moment [27]).

$$Pers_k(f, r) \leq r^k N\left(\tfrac{r}{\lambda}\right) + k \int_{\epsilon=r}^{Amp(f)} N\left(\tfrac{\epsilon}{\lambda}\right) \epsilon^{k-1}\, d\epsilon.$$

Now suppose that the size of the triangulation grows only polynomially with the reciprocal of the mesh; that is, there are constants C_0 and M such that $N(r) \leq C_0/r^M$. We can use the Moment Lemma to show

$$\text{Pers}_k(f) \leq C_0\lambda^M \text{Amp}(f)\frac{M + 2\delta}{\delta}. \tag{3}$$

As r approaches zero, the Lipschitz constant λ also approaches zero. Thus, the right hand side of (3) goes to zero like λ^M.

Algorithm 2.1 Left-Right Persistence Algorithm

```
for $j = 1$ to $n$ do
  while $\exists j' < j$ with $\text{low}(j') = \text{low}(j) \neq 0$ do
    add column $j'$ to column $j$
  end while
end for
```

Left-Right Algorithm. We will conclude this section by giving two algorithms to compute persistence. The first one is the standard persistence algorithm, originally presented in [36], and summarized in [34]. Given a filtration, we compute the persistence diagrams, just as we did for the sublevel sets above.

We explain the persistence algorithm in terms of matrix reduction. We create a filter to order the simplices of the simplicial complex. Then, we label the columns and rows with the simplices in the order in which they appear in the filter. We use σ_i to represent the simplex that is represented in column (and row) i. We create the *boundary matrix* D defined by $D(i, j) = 1$ iff simplex σ_i is on the boundary of (is a codimension one face of) simplex σ_j. All other entries are zero. We reduce D until we have a unique lowest non-zero entry in each column.

In Algorithm 2.1, we scan the columns from left to right, incrementing through the filter. Let low(j) denote the index of the lowest one in column j. If column j is a zero column, then low(j) $= -1$. If the lowest one in the column has only zero entries to the left, then we have found a persistence pairing. If there exists a column to the left with a lowest one on the same row, then we add the second column to the first using $\mathbb{Z}_2$ coefficients. Then, we check to the left again. We repeat this until either we find a lowest one that is the first one in the row, or we have a column with zero entries. A zero column corresponds to a simplex that represents the birth of a homology class.

Algorithm 2.2 Spectral Sequence Persistence Algorithm

```
for $r = 1$ to $m - 1$ do
  for all $j \in [r + 1, m]$ do
    if $D(j - r, j) = 1$ then
      if $\exists i' < i$ with low($i'$) =low($i$) $\neq 0$ then
        add column $i'$ to column $i$
      end if
    end if
  end for
end for
```

Spectral Sequence Algorithm. In Algorithm 2.2, we scan parallel to the diagonal of the matrix instead of left to right. At stage r, we consider all entries such that the index of the column is equal to r plus the index of the row. For each of these entries, we look to see if the value is zero or one. We will ignore zero entries. If it is a one and it is the lowest one in the column containing it, then we check to see if this entry corresponds to a death. If the entry has only zeros to the left, then we have found a persistence pairing. Otherwise, we add the second column to the first using $\mathbb{Z}_2$ coefficients. Since we have just moved the lowest one up (or removed all ones), we do not repeat this process of column additions as we will revisit this column in a subsequent iteration. Although this algorithm will result in the same reduced matrix as Algorithm 2.1, it has the advantage that it can be parallelized.

2.4. **Homotopy.** We now turn to looking at two functions, $f, g\colon \mathbb{M} \to \mathbb{R}$. We say that f and g are homotopic if there exists a continuous deformation of the first function into the second. We are interested in creating a homotopy between f and g in order to observe how the corresponding persistence diagram changes through time. More importantly, we use the homotopy when matching points in the persistence diagrams. Before proceeding, let us formally define a homotopy and its discrete analog.

Continuous Homotopy. We say that $f, g\colon \mathbb{M} \to \mathbb{R}$ are *homotopic* if there exists a continuous function $H\colon \mathbb{M} \times I \to \mathbb{R}$ such that I is the closed interval $[0,1]$, $H(x,0) = f(x)$, and $H(x,1) = g(x)$, for all $x \in \mathbb{M}$. To simplify notation, we will denote the homotopy $H(x,t)$ by $h_t(x)$. The constant homotopy $h_t(x) = f(x)$ for all $t \in I$ is the simplest homotopy, although it is not too useful. Often, the straight line homotopy is used to continuously interpolate between two functions.

Example 2.3 (The Straight Line Homotopy)**.** The straight line homotopy interpolates linearly from each point in the continuous function f to the corresponding point in the continuous function g. We can write $h_t(x) = (1-t)f(x) + tg(x)$ for all $t \in [0,1]$ and all $x \in M$.

In Section 4.3, we define the Heat Equation Homotopy, which uses the heat equation to diffuse the difference between two functions. We use a regular grid decomposition to discretize the domain. Thus, the domain $\mathbb{M}$ that we are interested in is a graph, not a rectangular subset of $\mathbb{R}^m$. In addition, we will only define the homotopy values at a finite number of times. In order to bridge the gap between the continuous definition of a homotopy and our discrete spatial domain, we define a discrete homotopy.

Discrete Homotopy. A discrete analog of homotopy comes from A-theory [18]. There are two versions of the discrete homotopy, one for simplicial complexes and one for graphs [19]. We will focus on the A-theory of graphs.

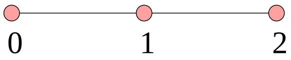


Figure 2: The graph of the discrete interval I_2.

The discrete interval, I_τ, is the polygonal chain with $\tau+1$ vertices. I_τ as a set is equal to $\{0, \ldots, \tau\}$ and the edges of I_τ, edges(I_τ), are created between consecutive vertices: edges$(I_\tau) = \{(i-1, i)|\ i = 1, \ldots \tau\}$. Figure 2 shows the discrete interval for $\tau = 2$, and Figure 3 shows $\Gamma \times I_2$ for a 3-cycle Γ. Let $\Gamma_1 = (V_1, E_1)$ and $\Gamma_2 = (V_2, E_2)$ be simple graphs. Now, suppose $\alpha_1, \alpha_2\colon (\Gamma_1, v_1) \to (\Gamma_2, v_2)$ are *based graph maps*; this means that $\alpha_i(v_1) = v_2$ and if $uv \in E_1$, then either $\alpha_i(u) = \alpha_i(v)$ or $\alpha_i(u)\alpha_i(v) \in E_2$. Let the vertex set V be the set of pairs (u,v) in $V_1 \times I_\tau$ and let the edge set E be equal to the set of pairs $\{(u,v),(a,b)\}$ in V such that either $u = a$ and $vb \in$ edges(I_τ), or $(v = b)$ and $ua \in E_1$. Then a graph map $\varphi\colon \Gamma_1 \times I_\tau \to \Gamma_2$ is a *G-homotopy* between α_1 and α_2 if:

1. $\varphi(*, 0) = \alpha_1$,
2. $\varphi(*, \tau) = \alpha_2$,
3. $\varphi(v_1, i) = v_2$ for all i, and
4. If (u,i) and (v,j) are in the edge set E, then either $\varphi(u,i) = \varphi(v,j)$ or $\varphi(u,i)\varphi(v,j) \in E_2$.

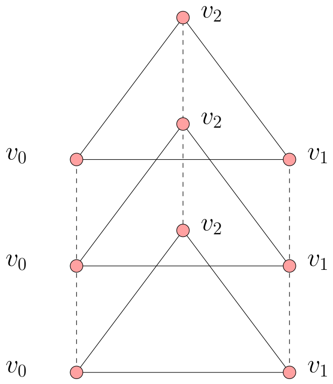
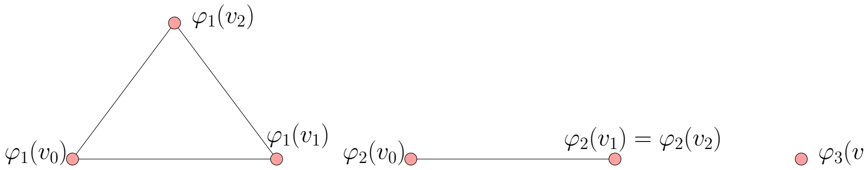


Figure 3: The graph of $\Gamma \times I_2$, for a 3-cycle Γ. The solid lines show the edges for a fixed vertex of I_2, and the dashed lines show the edges corresponding to a fixed vertex in Γ.



(a) $\varphi(\Gamma, 1)$ (b) $\varphi(\Gamma, 2)$ (c) $\varphi(\Gamma, 3)$

Figure 4: A triangle is G-homotopy equivalent to a point. This is not true in the traditional homotopy theory.

In this case, we say that α_1 and α_2 are *G-homotopic*. Figure 4 shows that the triangle is G-homotopic to a point. Both 3- and 4-cycles are G-homotopic to a point, but 5-cycles are not. This means that the G-homotopy is sensitive to the discretization of the domain and differs from the continuous definition of homotopy, since a 3-cycle and a 5-cycle are homotopic and not homotopic to a point, according to the traditional definition.

We generalize upon this idea, making our own definition that will be used later to define the Heat Equation Homotopy. This definition is compatible with the continuous definition of homotopy, unlike the definition of the G-homotopy discussed above. Let f, g be two functions from the graph $\Gamma = (V, E)$ to $\mathbb{R}$. A *valued discrete homotopy of functions*, $\varphi \colon \Gamma \times I_\tau \to \mathbb{R}$ associates a (real) value for each pair $(u, i) \in V \times \mathrm{vert}(I_\tau)$ such that $\varphi(*, 0) = f$ and $\varphi(*, \tau) = g$, and linearly interpolates the values along the edges of the form $((v, t)(v, t+1)) \in \Gamma \times I_\tau$. We can think of this as a time-varying function on Γ. The homotopy is *based* if for some $u \in V$, $\varphi(u, t) = c$ for all

t. For any given t, the image $\varphi(uv)$ of the edge $uv \in E$ is determined by linearly interpolating between $\varphi(u)$ and $\varphi(v)$. In this way, we can entirely define the homotopy φ by defining it at the vertices only, $\widehat{\varphi}\colon \Gamma \times I_\tau \to \mathbb{R}$.

2.5. **Vineyards.** For each function $f, g\colon \mathbb{M}^m \to \mathbb{R}$, we have a persistence diagram for each integer $p \in \{0, 1, 2, \ldots, m\}$. Now, choose a value of p and assume that we have a generic homotopy from f to g. The homotopy gives us a time varying persistence diagram. We stack the persistence diagrams associated with the homotopy.

Vines. Choose τ time-steps of the homotopy, $0 = t_0 < t_1 < \ldots < t_\tau = 1$. Notice that $\mathrm{Dgm}_p(f_0)$ $=\mathrm{Dgm}_p(f)$ is the initial persistence diagram. At each time t_j, we have a persistence diagram $\mathrm{Dgm}_p(f_{t_j})$. Moreover, given $\mathrm{Dgm}_p(f_{t_j})$, we can compute $\mathrm{Dgm}_p(f_{t_{j+1}})$ for t_{j+1} close enough to t_j in time linear in the number of simplices of the filtration by using a straight line homotopy between f_{t_j} and $f_{t_{j+1}}$, as described in [28]. We track the transpositions of simplices obtained from this computation in order to pair points in the persistence diagrams. Then, we stack the diagrams so that $\mathrm{Dgm}_p(f_t)$ is drawn at height t in $\mathbb{R}^3$ and connect the paired points in consecutive diagrams by line segments. The result is a collection of piecewise linear paths between points in $\mathrm{Dgm}_p(f_0)$ and points in $\mathrm{Dgm}_p(f_1)$. If we let the time difference between any two consecutive diagrams approach zero, the piecewise linear paths become a set of continuous curves by the stability result for the straight line homotopy, presented in Theorem 3.7. Each curve that traces the path of an off-diagonal point through time is called a *vine*. The collection of vines is called a *vineyard* [28, 35].

Create a filtration for $f_0(x)$ as prescribed in Section 2.1, and compute the persistence diagram $\mathrm{Dgm}_p(f_{t_0})$ using one of the two algorithms in Section 2.3. To compute $\mathrm{Dgm}_p(f_{t_1})$, we could repeat the same process. However, if we use $\mathrm{Dgm}_p(f_{t_0})$, we can compute the new diagram in linear time (assuming a constant number of transpositions between t_0 and t_1) and we can match points in the two diagrams [28]. We pair the endpoints of each vine to obtain a matching of the persistence points in $\mathrm{Dgm}_p(f)$ with the points in $\mathrm{Dgm}_p(g)$. In Section 3.3, we term this the *vineyard matching*. If $a \in \mathrm{Dgm}_p(f_{t_0})$ and $b \in \mathrm{Dgm}_p(f_{t_1})$ are paired using this algorithm, then there exists a vine $s : I \to \mathbb{R}^3$ connecting a and b. When a point $a \in \mathrm{Dgm}_p(f_{t_0})$ enters the diagonal at $t < 1$, we pair a with the corresponding diagonal point, since the diagonal points can only occur as endpoints of a vine.

Transpositions. Suppose we have two similar functions on m simplices, such that the filters we create are identical except that two adjacent simplices have swapped order. Then, we can write the first filter:

$$F1 : \sigma_1, \sigma_2, \ldots, \sigma_i, \sigma_{i+1}, \ldots, \sigma_m$$

and the second filter:

$$F2 : \sigma_1, \sigma_2, \ldots, \sigma_{i+1}, \sigma_i, \ldots, \sigma_m.$$

If we have the persistence diagram for the filter $F1$, then we can compute the persistence diagram for the the filter $F2$ by performing one *transposition*. A transposition updates the persistence diagram by changing the pairings of two consecutive simplices, if necessary. In the case that $\dim(\sigma_i) \neq \dim(\sigma_{i+1})$, the transposition does not affect the pairs in the persistence diagram. Thus, only the transposition of simplices of the same dimension can result in a pairing swap. One transposition can swap the births and deaths of at most two points in the persistence diagram.

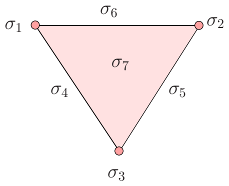


Figure 5: The filter of a simplicial complex is an ordering on the vertices, edges, and faces. The simplices are originally ordered σ_1 through σ_7. Swapping σ_1 and σ_2 results in a pairing swap of type T1. Swapping σ_4 and σ_5 results in a pairing swap of type T2. Swapping σ_5 and σ_6 results in a pairing swap of type T3.

We must distinguish between nested and unnested persistence pairings. We say that two persistence points, (b_1, d_1) and (b_2, d_2), are *nested* if the birth at b_2 and the death at d_2 occur after b_1 and before d_1. Assuming all events happen at distinct moments of time, this is equivalent to $b_1 < b_2 < d_2 < d_1$. If we transpose σ_i and σ_{i+1}, then the three types of pair-swapping transpositions are:

T1. The births of two nested pairs are transposed. In this case, σ_i is associated with b_1 in $\mathrm{Dgm}_p(F1)$, and with b_2 in $\mathrm{Dgm}_p(F2)$.

T2. The deaths of two nested pairs are transposed. In this case, σ_i is associated with d_2 in $\mathrm{Dgm}_p(F1)$, and with d_1 in $\mathrm{Dgm}_p(F2)$.

T3. The birth of one pair and the death of another, unnested, pair are transposed. In this case, the addition of σ_i to the filtration created a death in $\mathrm{Dgm}_p(F1)$, and a birth in $\mathrm{Dgm}_p(F2)$.

Figure 5 illustrates the three types of transpositions that can be made. Although pair swaps of types T1 and T2 involve two persistence points in the same diagram, pair swaps of type T3 involve persistence points in diagrams of two consecutive dimensions. In each of the cases, the transposition results in a swap only if changing the order in which the simplices are added changes the persistence pairing that is made. For a complete algorithm to compute the transpositions, please refer to [28].

From these transpositions, we can create the matching needed in a later result Theorem 3.7. Specifically, suppose the transposition of σ_i and of σ_{i+1}

resulted in a pairing swap. Then, every persistence point is paired with itself in the matching, except for (b_1, d_1) and (b_2, d_2), whose pairings are swapped.

Sweep Algorithm. Changing one filtration into another may require more than one transposition. In order to compute the persistence diagram, we first create a topological arrangement and then use a sweep algorithm. In the Cartesian plane, we write the filtration of f_{t_j} horizontally. Below that, we write the filtration of $f_{t_{j+1}}$ using the same simplex names. Then, we connect like simplices with a single curve (does not need to be straight). An example of this process is given in Figure 6. After this arrangement has been created, an ordering on the transpositions can be found by topologically sweeping the arrangement, as presented in [33]. We then compute $\mathrm{Dgm}_p(f_{t_{j+1}})$ and keep track of the matching by progressing one transposition at a time in the order dictated by the sweep algorithm.

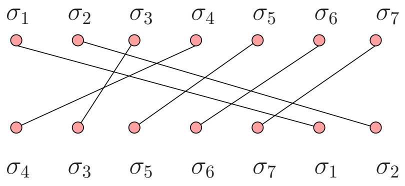


Figure 6: We create a topological arrangement by connecting drawing a line between the vertices that represent the same simplex. By doing this, we find a finite number of crossings. Each crossing represents one transposition in the filter.

3. Metrics

The focus of this section is a result that bounds the difference between the lengths of curves by a function of the total curvature of the two curves and the Fréchet distance between them. This section begins by discussing several metrics, focusing on the Fréchet distance. Then, we give an overview of the result in Section 3.5 and present related open questions in Section 3.6.

3.1. Background.

We begin this section with the definition and properties of a subclass of 1-manifolds, curves, and their discrete analog, polygons. Then, we formally define a metric and give examples. One of these metrics, the Fréchet distance, plays a central role in the proof presented in Section 3.5.

Curves and Polygons. A *closed curve* in Euclidean space is a continuous map from the unit circle $\mathbb{S}^1$ to $\mathbb{R}^n$:

$$\gamma\colon \mathbb{S}^1 \to \mathbb{R}^n.$$

In general, we think of a closed curve as the parameterized image of the circle, but we use the unit interval whenever it is more convenient to parameterize with $I = [0, 1]$. This allows for us to define an *open curve* as a map $\gamma\colon I \to \mathbb{R}^n$ where $\gamma(0) \neq \gamma(1)$.

A *polygonal chain* is a piecewise linear open curve. We can define a polygonal chain as a mapping from the discrete interval of length m to Euclidean n-dimensional space:

$$P\colon I_m \to \mathbb{R}^n.$$

This polygonal chain comprises $m + 1$ vertices and m edges. Each edge e_i defines a linear interpolation between $P(i)$ and $P(i + 1)$. A *polygon* is a closed polygonal chain, i.e, a polygonal chain such that $P(1) = P(m)$.

We say that a curve γ is *k-bounded* if for every $0 < \tau_i < \tau_j < 1$, the subcurve from $p_i = \gamma(\tau_i)$ to $p_j = \gamma(\tau_j)$ is contained within the union of the balls centered at p_i and p_j with radius k times one half the distance between p_i and p_j:

$$\gamma_{p_i,p_j} \subseteq B_{\frac{k}{2}||p_i-p_j||}(p_i) \cup B_{\frac{k}{2}||p_i-p_j||}(p_j).$$

We will see in Section 3.3 that the Fréchet distance between two k-bounded curves is upper bounded by $(k + 1)$ times the Hausdorff distance between the two vertex sets.

A curve γ is called *c-packed* if the total length of the sub-curves contained inside any ball is bounded by c times the radius of that ball. Unlike k-bounded curves, c-packed curves can self intersect and are closed under concatenation. A k-bounded curve does not need to be c-bounded, as a k-bounded curve can be arbitrarily long in a ball of a given diameter.

Metrics. The function $d_{\mathbb{M}}\colon \mathbb{M} \times \mathbb{M} \to [0, \infty]$ is a *distance function* or *metric* on a set $\mathbb{M}$ if for all $x, y, z \in \mathbb{M}$, the following three properties hold:

1. Definiteness: $d_{\mathbb{M}}(x, y) = 0$ if and only if $x = y$.
2. Symmetry: $d_{\mathbb{M}}(x, y) = d_{\mathbb{M}}(y, x)$.
3. Triangle Inequality: $d_{\mathbb{M}}(x, y) \leq d_{\mathbb{M}}(x, z) + d_{\mathbb{M}}(z, y)$.

A *metric space* is a set $\mathbb{M}$ with a distance function $d_{\mathbb{M}}$ defined between any pair of elements, denoted as $(\mathbb{M}, d_{\mathbb{M}})$ [54].

Example 3.1 (Riemannian Manifold)**.** A *Riemannian Manifold* is a manifold $\mathbb{M}$ equipped with a Riemannian metric $\langle\cdot,\cdot\rangle_{\mathbb{M}}$, an inner product in the tangent space $T_p\mathbb{M}$ at $p \in \mathbb{M}$. This metric is used to find lengths, areas, and angles on the manifold. For example, the squared length of a tangent vector $v \in T_p\mathbb{M}$ is $\langle v, v\rangle_{\mathbb{M}}$. The Riemannian metric provides a measure (the length) of any continuously differentiable curve, $\gamma\colon [a, b] \to \mathbb{M}$. The length of γ is found by integrating the length of the velocity vector along the curve:

$$\begin{aligned} \ell(\gamma) &= \int_a^b ||\gamma'(t)||\, dt, \text{ where} \\ ||\gamma'(t)||^2 &= \langle\gamma'(t), \gamma'(t)\rangle_{\mathbb{M}}. \end{aligned}$$

The *geodesic distance* between any two points $x, y \in \mathbb{M}$, denoted $d_{\mathbb{M}}(x, y)$, is defined as the infimum of the lengths of all curves connecting x to y. With this metric, every connected Riemannian Manifold $\mathbb{M}$ becomes a metric space, $(\mathbb{M}, d_{\mathbb{M}})$.

Length in the limit. Given a closed curve $\gamma : I \to \mathbb{R}^n$ and a choice of m parameter values $0 = t_0 < t_1 < t_2 < \cdots < t_m = 1$, we let P be the *inscribed polygon* consisting of the vertices $v_i = \gamma(t_i)$ and the edges $v_i v_{i+1}$, where we take indices modulo m. We write $P < \gamma$ to denote that P is inscribed in γ. Figure 7 shows an example of an inscribed polygon. If γ is smooth, then we can get good polygonal approximations by choosing the values without leaving large gaps. To make this precise, we define the *mesh* of a polygon P as the maximum distance between two consecutive values in the parameterizing interval: $\text{mesh}(P) = \max_{0 \le i < m}\{t_{i+1} - t_i\}$. We then have the following result, proven in [56]:

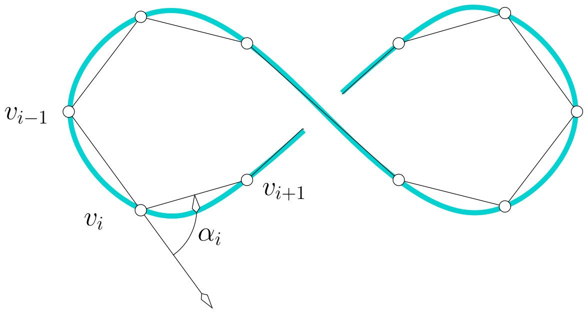


Figure 7: A closed curve in space, an inscribed polygon, and the turning angle at the vertex v_i.

Lemma 3.2 (Length in the Limit)**.** *If P^k is a sequence of polygons inscribed in a smooth closed curve γ such that* $\text{mesh}(P^k)$ *goes to zero, then* $\ell(\gamma) = \lim_{k\to\infty} \ell(P^k)$.

Total curvature in the limit. As shown in Figure 7, the *turning angle*, α_i, at the vertex v_i is the angle between the incoming edge, $v_i - v_{i-1}$, and the outgoing edge, $v_{i+1} - v_i$. Note that $0 \le \alpha_i \le \pi$. Milnor [52] defines the *total curvature* of a closed polygon P as the sum of the turning angles: $\kappa(P) = \sum_{i=0}^{m-1} \alpha_i$. The total curvature of the smooth closed curve γ is the supremum over all inscribed polygons:

$$\kappa(\gamma) \;=\; \sup_{P<\gamma} \kappa(P).$$

Another notion of total curvature was studied by Fenchel [42]. Letting $\kappa(s)$ at a point s on a smooth curve be one over the radius of the locally best approximating circle, he defines the *total curvature* by integration:

$$\kappa(\gamma) \;=\; \int_{s\in\gamma} \kappa(s)\,\mathrm{d}s.$$

Milnor proved that these two definitions of total curvature are equivalent, and using his definition, we find the total curvature of a smooth curve as the limit of the total curvature of inscribed polygons [39]:

Lemma 3.3 (Curvature in the Limit)**.** *If P^k is a sequence of polygons inscribed in a smooth closed curve γ such that* $\mathrm{mesh}(P^k)$ *goes to zero, then* $\kappa(\gamma) = \lim_{k\to\infty} \kappa(P^k)$.

3.2. **Direct Metrics.** We discuss three metrics: the Hausdorff Distance, the Gromov-Hausdorff Distance, and the Fréchet Distance. We consider these metrics to be *direct* because they measure the distance based on the inherent shapes. In Section 3.3, we look at distance metrics that measure the distance between two shapes based on associated persistence diagrams.

Hausdorff Distance. Let Z be a metric space, with $\mathcal{S}_1, \mathcal{S}_2 \subseteq Z$. One way to measure the distance between these sets is to take the *Hausdorff Distance*, denoted $\mathcal{H}(\mathcal{S}_1, \mathcal{S}_2)$, which is the maximum distance (measured in Z) from a point in one set to the closest point in the other set:

$$\mathcal{H}(\mathcal{S}_1, \mathcal{S}_2) \;=\; \max\{\max_{p\in\mathcal{S}_1} \min_{q\in\mathcal{S}_2} d_Z(p,q),\; \max_{p\in\mathcal{S}_2} \min_{q\in\mathcal{S}_1} d_Z(p,q)\}. \tag{4}$$

When the ambient space Z is not obvious from the context, we will note it in a subscript: $\mathcal{H}_Z(\mathcal{S}_1, \mathcal{S}_2)$. Intuitively, this is the maximum distance between a point in one set to the closest point in the other set when $\mathcal{S}_1$ and $\mathcal{S}_2$ are embedded in Z.

Gromov-Hausdorff Distance. The Hausdorff Distance is limited, because we must choose our embedding space Z before computing the distance. The *Gromov-Hausdorff Distance* (G-Hd) finds the minimum Hausdorff Distance over all possible embedding spaces:

$$d_{GH}(\mathcal{S}_1, \mathcal{S}_2) = \inf_{\substack{Z \\ f\colon \mathcal{S}_1\to Z \\ g\colon \mathcal{S}_2\to Z}} \mathcal{H}(f(\mathcal{S}_1), g(\mathcal{S}_2)) \tag{5}$$

where f, g are isometric embeddings of $\mathcal{S}_1$ and $\mathcal{S}_2$, and Z is a metric space. This allows the distance to be computed between metric spaces $\mathcal{S}_1$ and $\mathcal{S}_2$ up to embedding.

Fréchet Distance. Let $\mathcal{S}_1$ and $\mathcal{S}_2$ be two homeomorphic spaces embedded in a metric space Z with parameterizations $\mathcal{S}_1(t)$ and $\mathcal{S}_2(s)$. The *Fréchet distance* between $\mathcal{S}_1$ and $\mathcal{S}_2$ is the minimum of the maximum distances between two corresponding points over all homeomorphisms:

$$\mathcal{F}(\mathcal{S}_1, \mathcal{S}_2) \;=\; \inf_{f:\mathcal{S}_1\to\mathcal{S}_2} \max_{t\in\mathcal{S}_1} \|\mathcal{S}_1(t) - \mathcal{S}_2(f(t))\|. \tag{6}$$

Although the Hausdorff distance does not require two spaces to be homeomorphic in order to compute a distance between them, the Fréchet distance seems to be a more natural way to describe the distance between shapes, as it takes not only the position of the shapes within a metric space, but also the structure of the shapes, into account. For example, Figure 8 shows two polygonal chains with a small Hausdorff Distance but a large Fréchet Distance. This indicates that, in some cases, Fréchet Distance is a more appropriate way to measure the similarity between curves.

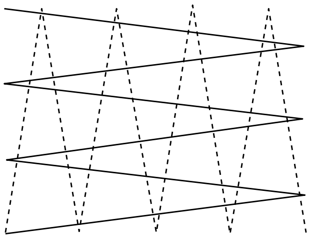


Figure 8: These two polygonal chains have a small Hausdorff distance but a large Fréchet distance.

3.3. **Metrics on Persistence Diagrams.** As in the previous section, let $\mathcal{S}_1$ and $\mathcal{S}_2$ be two metric spaces. Suppose further that there are functions $f\colon \mathcal{S}_1 \to \mathbb{R}$ and $g\colon \mathcal{S}_2 \to \mathbb{R}$. Now, we will use the persistence diagrams to define three distance metrics to measure the distance between $\mathcal{S}_1$ and $\mathcal{S}_2$. We say that these metrics are *indirect* since they use only the information in the persistence diagrams (and vineyard) in order to compute a distance.

For the first two metrics, we simply compute the cost of matching the points in $A = \mathrm{Dgm}_p(f)$ and $B = \mathrm{Dgm}_p(g)$. This setting is easily described in terms of the general matching problem: we must match the elements of set $A \subseteq \mathbb{R}^2$ with elements of a second set $B \subseteq \mathbb{R}^2$, where there is a cost associated with each pair $(a, b) \in A \times B$. The cost $d^\infty(a, b)$ between a and b is the L_∞ distance:

$$d^\infty(a, b) = \max\{|a_x - b_x|, |a_y - b_y|\}. \tag{7}$$

The two metrics that we consider are the Wasserstein Metric and the Bottleneck Metric. The Wasserstein Distance minimizes the sum of the pairwise costs; whereas, the Bottleneck Distance minimizes the maximum pairwise cost.

Wasserstein Matching of Persistence Diagrams. We define the Wasserstein Cost function as follows:

Definition 3.4 (Wasserstein Cost)**.** The *degree q Wasserstein Cost $C_q(P)$* of a perfect matching P is the q^{th} root of the sum of the q^{th} powers of the edge costs over all edges in the matching:

$$C_q(P) = \left(\sum_{(a,b)\in P} d^\infty(a,b)^q \right)^{1/q}.$$

The *Wasserstein Distance* $W_q(A,B)$ is the maximum degree q Wasserstein Cost over all perfect matchings:

$$W_q(A,B) = \min_P C_q(P).$$

The matching that attains the Wasserstein Distance is called the *Wasserstein Matching*.

The Hungarian method computes the Wasserstein Matching in $O(n^4)$ computational complexity [50], where $|A| = |B| = n$. In [59], Vaidya maintains weighted Voronoi diagrams for an $O(n^{2.5} \log n)$ computation of this matching. Further improvements were made by Agarwal, Efrat, and Sharir [9]. They utilize a data structure that improves the running time to $O(n^{2+\epsilon})$.

Bottleneck Matching of Persistence Diagrams. Analogous to finding the Wasserstein Matching, the Bottleneck Matching minimizes the Bottleneck Cost function, which is defined as follows:

Definition 3.5 (Bottleneck Cost)**.** The bottleneck cost of a perfect matching P is the maximum edge cost in the matching:

$$C_\infty(P) = \max_{(a,b)\in P} d^\infty(a,b).$$

We minimize this cost over all perfect matchings of A and B to obtain the Bottleneck Distance between the sets:

$$W_\infty(A,B) = \min_P C_\infty(P).$$

The matching P that attains the Bottleneck Distance is called the *Bottleneck Matching*. Although for small values of q, the Bottleneck and the Wasserstein Matchings may differ, the Wasserstein Cost approaches the Bottleneck Cost as q approaches infinity.

The computational complexity of the Bottleneck Matching is slightly greater than that of the Wasserstein Matching. A maximal matching can be found in $O(n^{5/2})$ using the Hopcroft-Karp algorithm [47]. If we mimic the thresholding approach of the Hungarian method [50], then the bottleneck solution can be found in $O(n^{5/2} \log n)$. Since A and B are sets of points in the plane, we can improve the computational complexity of determining the matching under the bottleneck distance. Efrat, Itai, and Katz developed a geometric improvement to the Hopcroft-Karp algorithm with a running time of $O(n^{1.5} \log^2 n)$ [37].

Since the running time of computing the Bottleneck Distance between finite point sets is polynomial, approximating computationally intensive metrics by the Bottleneck Distance is often desired. For example, we can lower bound d_{GH} of Equation (5) by the Bottleneck Distance W_∞ of persistence diagrams corresponding to Ripps filtrations.

Theorem 3.6 (Persistence Based Lower Bound for G-Hd Distance [23])**.** *For all finite metric spaces $\mathcal{S}_1$ and $\mathcal{S}_2$ and for all $k \in \mathbb{N}$, the bottleneck distance between diagrams of Ripps filtrations is upper bounded by the G-Hd Distance of the vertex sets:*

$$W_\infty\left(Dgm_k(\mathcal{R}(\mathcal{S}_1)), Dgm_k(\mathcal{R}(\mathcal{S}_2))\right) \leq d_{GH}(\mathcal{S}_1, \mathcal{S}_2).$$

This theorem says that the Bottleneck Distance is stable with respect to the G-Hd Distance. Since computing the Bottleneck Distance is more manageable than computing the NP-Hard G-Hd Distance, Chazal et al. use the Bottleneck Distance as a shape signature [23].

Degree k Vineyard Distance. Neither the Wasserstein nor the Bottleneck Distance measures the cost of transforming the function f into the function g. Both measure the difference between the two fixed functions. The Vineyard Distance, however, uses a homotopy between functions f and g in the distance computation. The corresponding vineyard is used to measure the distance between f and g. Each point a in $A = \mathrm{Dgm}_p(f)$ is connected with a point b in $B = \mathrm{Dgm}_p(g)$ by a vine in the vineyard. We use these pairings as our matching and define a distance metric for it, measuring the length of the path the persistence points traversed in the time varying persistence diagram.

Assume a and b are connected by the vine $s : I \to \mathbb{R}^3$, as described in Section 2.5. Since the vine was created from a homotopy, we will use the index t to emphasize that $s(t)$ is a persistence point for the function $f_t(x)$. The velocity of the vine $\partial s/\partial t$ will be integrated over the unit interval in order to measure the distance traveled between a and b :

$$D_s = \int_0^1 \left|\left|\frac{\partial s(t)}{\partial t}\right|\right| dt.$$

If $s(t)$ is only defined for a discrete set of times t_i with $0 \leq i \leq \tau$, then we obtain a discretized version of D_s:

$$D_s^k = \sum_{i \in \{1,\ldots,\tau\}} ||s(t_i) - s(t_{i-1})||_\infty^k.$$

In order to obtain a distance between persistence diagrams we sum the k^{th} powers of these distances over all vines in the vineyard V:

$$V_k(H) = \sum_{s \in V} D_s^k \tag{8}$$

When a point $a \in A$ enters the diagonal at $t < 1$, we pair a with the corresponding diagonal point, since the diagonal points can only occur as endpoints of a vine. Then, we measure the distance over the interval in which the vine is defined, $[0, t)$. Symmetrically, we can have a diagonal point in A paired with an off-diagonal point in B.

We are interested in understanding how this distance metric compares with the Bottleneck and Wasserstein Distances, as well as to understand the properties of this distance metric and corresponding matching. We note here that the Vineyard Distance can be thought of as an editing distance. An *editing distance* sums the cost of edit operations necessary to transform one object into another [57]. In the original context, the objects are trees; however, for us, the objects are functions on simplicial complexes. In [31], the authors define a (stable) edit distance between Reeb graphs that closely relates to the results of [25].

Stability Theorems. A distance metric (and the corresponding matching) is stable if a small change in the input sets A and B produces a small change in the measured distance between the sets. The property of stability is not easy to verify and sometimes not even true.

Let f, g be tame functions. The L_∞ difference $||f-g||_\infty$ is the maximum difference between the function values:

$$||f-g||_\infty = \sup_{x \in \mathcal{S}_1} |f(x) - g(x)|.$$

The Stability Theorem for Tame Functions states that the bottleneck distance is bounded above by the L_∞ distance between the functions [25]:

$$W_\infty(\mathrm{Dgm}_p(f), \mathrm{Dgm}_p(g)) \leq ||f-g||_\infty. \tag{9}$$

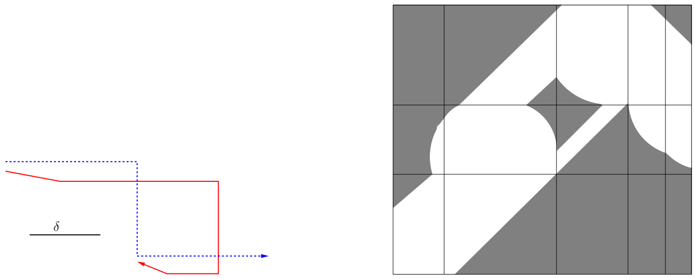
The proof of this theorem presented in [28] uses the following result:

Theorem 3.7 (Stability of the Straight Line Homotopy)**.** *Given the straight line homotopy from f to g, there exists a perfect matching P of the persistence diagrams for f and g such that the bottleneck cost of P is upper bounded by the distance between f and g:*

$$\max_{(a,b) \in P} C(a,b) \leq ||f-g||_\infty.$$

Although (9) does not hold for the Wasserstein Distance, the Wasserstein Distance is stable for Lipschitz functions with bounded degree k total persistence [27]. If we relax either of these two conditions (Lipschitz or bounded total persistence), then the Wasserstein distance becomes unstable for two functions f and g where $||f-g||_\infty \leq \epsilon$ as ϵ approaches zero. In Open Question 3.14, we ask if there exists a stability result for the degree k Vineyard Distance. To answer this question, we must first define a suitable measure on a homotopy.

3.4. **Fréchet Distance.** The Fréchet Distance measures the distance between homeomorphic metric subspaces. Equation (6) defined the Fréchet Distance for two subspaces of a metric space. In this section, we restrict our attention to closed curves in Euclidean space.



(a) Two Curves

(b) Free Space Diagram

Figure 9: The free space diagram for the two open polygons. In this diagram, the curves are parameterized by archlength. The horizontal and vertical lines correspond to the vertices of the polygons. The regions that the lines create are called the *grid cells* of the free space diagram.

Fréchet Distance for Closed Space Curves. The *Fréchet Distance* between closed curves γ_1 and γ_2 is defined as the infimum over all homeomorphisms of the maximum distance between corresponding points. Recalling that a closed curve is a map from the unit circle $\mathbb{S}^1$ to $\mathbb{R}^n$, we can take the infimum over homeomorphisms of $\mathbb{S}^1$ onto itself.

$$\mathcal{F}(\gamma_1, \gamma_2) \;=\; \inf_{f:\mathbb{S}^1\to\mathbb{S}^1} \max_{t\in\mathbb{S}^1} \|\gamma_1(t) - \gamma_2(f(t))\|.$$

The Fréchet Distance has earned the name *dog-leash distance*. If we imagine a man walking on the path γ_1 and his dog walking along γ_2, then the Fréchet distance is the length of the shortest leash needed so that the man can traverse γ_1 and the dog traverse γ_2 continuously and without backtracking. If we allow the dog and/or the man to walk backward as well as forward, then the length of the shortest leash is called the *weak Fréchet Distance*.

Computation. The fundamental tool used when computing–and, as we will see next, approximating–the Fréchet Distance is the *free space diagram* [14]. Given a parameterization of two curves $\gamma_1, \gamma_2 \colon I \to \mathbb{R}^n$ and a choice of $\delta > 0$, we create a black and white image with domain $D = I \times I$. This is the free space diagram, as illustrated in Figure 9, in which a point $(x, y) \in D$ is white if $d(\gamma_1(x), \gamma_2(y)) \le \delta$ and is black otherwise. All white points are *feasible* or *free* in the domain; that is, $\gamma_1(x)$ and $\gamma_2(y)$ have distance less than δ, indicating that they could be paired by a homeomorphism that realizes the Fréchet distance. The set of feasible points is called the *free space*, denoted $F_\delta(\gamma_1, \gamma_2)$:

$$F_\delta(\gamma_1, \gamma_2) = \left\{(s, t) \in [0, 1]^2 | d(\gamma_1(s), \gamma_2(t)) \le \delta\right\} \tag{10}$$

On the other hand, the points that are marked black represent *infeasible* pairings of points, since the Euclidean distance between the points is greater than δ.

The Fréchet distance is less than or equal to δ if there exists a bimonotone path within the white (feasible) region connecting $(0,0)$ and $(1,1)$. Thus, in Figure 9, the Fréchet Distance is greater than the δ used to compute the free space diagram. It turns out that if the curves are open polygons with p and q points respectively, then there are a finite number of possible values for the Fréchet Distance. Furthermore, the complexity of computing the exact Fréchet Distance between open polygonal chains is $\mathrm{O}(pq\log(pq))$ by the use of parametric search [14], improving upon a previous $\mathrm{O}((pq^2 + p^2q)\log(pq))$ result [43]. Rote extended the result of Alt et al. to work for piecewise smooth curves [53]. However, parametric search is a cumbersome tool to use for this computation. There are two alternatives to computing the Fréchet distance using parametric search: use a randomized algorithm or approximate the distance. Cook and Wenk [29] explore a randomized algorithm for the Fréchet distance between two bounded planar curves. We will look into the possibility of approximating the Fréchet Distance.

Approximation. Recall from Section 3.1 that a curve is c-packed if the total length of the curve contained inside any ball is bounded by c times the radius of the ball. In [32], Driemel et al. present an algorithm to approximate the Fréchet distance that runs in near linear time for c-packed curves. We give a brief overview of this algorithm.

Let P_1 and P_2 be two c-packed polygons. The first step of the algorithm is to find a μ-simplifications of the polygons; that is, create a polygon $\mathrm{simp}_\mu(P_i)$ from a subset of the vertices of P_i such that the Fréchet distance between P_i and $\mathrm{simp}_\mu(P_i)$ is at most μ. The basic idea of their algorithm is to first search over a (finite) set of approximate Fréchet distances in order to define an interval $[\alpha, \beta]$ that contains the exact Fréchet distance between the simplified polygons. Then, they do a binary search on the interval until an appropriate approximation of the actual Fréchet distance is found.

Given $\delta > 0$, the *reachable free space* $\mathcal{R}_\delta(P_1, P_2)$ is the set of all points in the free space diagram $F_\delta(P_1, P_2)$ that are reachable from the origin by a bi-monotone path. The Fréchet distance is at most δ if $(1,1) \in \mathcal{R}_\delta(P_1, P_2)$. We let $\mathcal{N}_\delta(P_1, P_2)$ be the maximum number of grid cells with non-empty intersection with $\mathcal{R}_\delta(P_1, P_2)$ over all μ-simplifications of P_1 and P_2. This is called the *complexity* of the reachable free space. The algorithm described in [32] is $O(\mathcal{N}_\delta(P_1, P_2)\log n)$. Since $\mathcal{N}_\delta(P_1, P_2)$ is linear for c-packed curves, the approximation algorithm for c-packed curves is $O(n \log n)$.

The relationship between Hausdorff and the Fréchet distance for planar curves is explored in [15]. We can use the Hausdorff Distance as an upper bound for the Fréchet distance: for k-bounded planar polygons P_1 and P_2,

$$\mathcal{F}(P_1, P_2) \leq (k+1)\mathcal{H}(P_1, P_2).$$

Furthermore, the Hausdorff and Fréchet metric coincide for two convex curves. This, however, is not true in general, as was demonstrated in Figure 8.

Fréchet distance in the limit. Suppose that we have two smooth closed curves, γ_1 and γ_2 in $\mathbb{R}^n$, and two inscribed polygons, $P_1 < \gamma_1$ and $P_2 < \gamma_2$. Since compositions of homeomorphisms are again homeomorphisms, the Fréchet distance satisfies the triangle inequality, and therefore

$$\mathcal{F}(P_1, P_2) \quad \leq \quad \mathcal{F}(P_1, \gamma_1) + \mathcal{F}(\gamma_1, \gamma_2) + \mathcal{F}(\gamma_2, P_2).$$

In words, if the Fréchet distance between γ_j and P_j is small, for $j = 1, 2$, then the Fréchet distance between the smooth curves is similar to that between the two polygons. Letting $P_j^k < \gamma_j$ be a sequence of inscribed polygons indexed by k whose mesh goes to zero, the Fréchet distance between P_j^k and γ_j also goes to zero. This implies that we can use polygons to approximate the Fréchet distance between curves γ_1 and γ_2.

Lemma 3.8 (Fréchet Distance in the Limit). *If P_1^k and P_2^k are sequences of polygons inscribed in the smooth closed curves γ_1 and γ_2 in $\mathbb{R}^n$ such that* $\mathrm{mesh}(P_1^k)$ *and* $\mathrm{mesh}(P_2^k)$ *both go to zero, then $\mathcal{F}(\gamma_1, \gamma_2) = \lim_{k\to\infty} \mathcal{F}(P_1^k, P_2^k)$.*

Application. We now describe how the Fréchet distance can be used to measure the distance between the motion of jointed objects. As an object with n joints moves in time, we keep track of the angle at each joint. Then, we represent the motion of the object as a curve in $\mathbb{R}^n$:

$$\gamma(t) = (\alpha_1(t), \alpha_2(t), \ldots, \alpha_n(t)).$$

Each function $\alpha_i(t)$ for $1 \leq i \leq n$ represents the continuous motion of the i^{th} joint by recording the angle. This function is the *joint-angle trajectory*, and has been explored in [58] to classify the gaits of people walking. Given two such curves, γ_1 and γ_2, we can use the Fréchet Distance between them as a way to measure how similarity (or difference) between the two motions. The Fréchet Distance can measure the distance up to speed invariance. The challenge with this approach is that the start time of the motion must be well chosen. In addition, care must be taken when interpreting the angles, since an angle of 2π is equal to an angle of zero.

3.5. **A New Inequality.** In [24], Cohen-Steiner and Edelsbrunner show that given two closed curves γ_1 and γ_2 in $\mathbb{R}^n$, the difference of lengths $|\ell_1 - \ell_2|$ is upper bounded by a function of the total curvature and Fréchet Distance between the curves:

Theorem 3.9 (Length Bound). *If γ_1 and γ_2 are two closed curves in $\mathbb{R}^n$, then $|\ell_1 - \ell_2| \leq \frac{2\mathrm{vol}(\mathbb{S}^{n-1})}{\mathrm{vol}(\mathbb{S}^n)}(\kappa_1 + \kappa_2 - 2\pi)\mathcal{F}(\gamma_1, \gamma_2)$.*

In this theorem, ℓ_j and κ_j are the length and the total curvature of γ_j. The ratio of sphere volumes $\mathrm{vol}(\mathbb{S}^{n-1})/\mathrm{vol}(\mathbb{S}^n)$ is an increasing function that reaches $2/\pi$ at $n = 3$ and diverges as n goes to infinity. More precisely, the ratio grows like $\sqrt{n}$. Thus, an upper bound that avoids this factor is an improvement on the Length Bound. Such a bound is the result that we present here:

Theorem 3.10 (Improved Length Bound [**?**])**.** *If γ_1 and γ_2 are closed curves in $\mathbb{R}^n$, then $|\ell_1 - \ell_2| \leq \frac{4}{\pi}(\kappa_1 + \kappa_2)\mathcal{F}(\gamma_1, \gamma_2)$.*

The remainder of this section presents the proof of this theorem for polygonal chains. To prove this result, we first map the polygonal chains in $\mathbb{R}^n$ into similar curves in $\mathbb{R}^3$, namely curves that have the same lengths and whose Fréchet distance and total curvatures are at most that of the corresponding curves in $\mathbb{R}^n$. Finally, we apply the case $n = 3$ of Theorem 3.9 in which twice the volume ratio is $4/\pi$.

Angles and Distances. In Section 3.1, we saw that the length and the total curvature of inscribed polygons approach the length and total curvature of the original curve as the mesh of the inscribed polygons decreases. In Section 3.4, we saw that this is true of the Fréchet distance as well. Now, we present one more lemma that is crucial to the construction in the proof.

Let $\triangle abc$ and $\triangle ABC$ be triangles with two congruent edges: $\|a - c\| = \|A - C\|$ and $\|b - c\| = \|B - C\|$. Proposition 24 of Euclid's Elements says that $\|a - b\| \leq \|A - B\|$ implies that the angle at c is less than or equal to the angle at C, where all angles are measured between 0 and π [38, p. 296]. This is also known as the Caliper Lemma [55].

Next consider two triangles, $\triangle axy$ and $\triangle xyb$ in $\mathbb{R}^n$. Since the configuration is determined by four points, we can place it isometrically into $\mathbb{R}^3$. Suppose that xy is a hinge so that we can change the distance between a and b without distorting the two triangles. Then $\|a - b\|$ is maximized when the two triangles lie in a common plane, with a (straight) dihedral angle of π at the hinge. To see this, we fix $\triangle axy$ and rotate b about the hinge. It sweeps out a circle that intersects the plane of axy in two points, b_1 when the dihedral angle is straight and b_0 when the two triangle overlap and the dihedral angle vanishes. Since b_0 is the reflection of b_1 across the line of xy and it lies on the same side as a, we have $\|a - b_0\| < \|a - b_1\|$. It follows that the entire circle swept out by b lies inside the ball with center a and radius $\|a - b_1\|$. The point b_1 lies on the boundary of that ball, implying that the distance between a and b is maximized when $b = b_1$, as claimed. We generalize this result to tetrahedra hinged on a face: $axyz$ and $xyzb$; see Figure 10 with $a = p_{k-1}$, $b = p_{k+1}$, $x = p_k$, $y = q_k$, $z = q_{k-1}$.

Lemma 3.11 (3D Caliper)**.** *The distance between a and b is maximized when $axyz$ and $xyzb$ lie in a three-dimensional plane, and a and b are on opposite sides of the plane containing $\triangle xyz$.*

Proof. We begin by isometrically embedding $axyzb$ in $\mathbb{R}^4$. Let the orbit of b be the set of points b' such that $axyzb'$ is isometric to $axyzb$. The set of points with constant distance to three points in $\mathbb{R}^4$ is a circle, and the two-plane spanned by those three points is the orthogonal axis of rotation. Hence, the orbit of b is a circle orthogonal to the hyperplane H defined by $axyz$. There are two points where this circle intersects H. Let b_0 be the point that causes the two tetrahedra $axyz$ and $xyzb$ to overlap, and b_1

the reflection of b_0 across the plane xyz. Consider the line segment that connects a and b_1. Let c be the intersection of the line and the plane xyz. By construction, the distance cb is constant throughout the orbit of b. Since $b = b_1$ corresponds to the angle at c being straight, we know by the Caliper Lemma that $b = b_1$ maximizes the distance between a and b. □

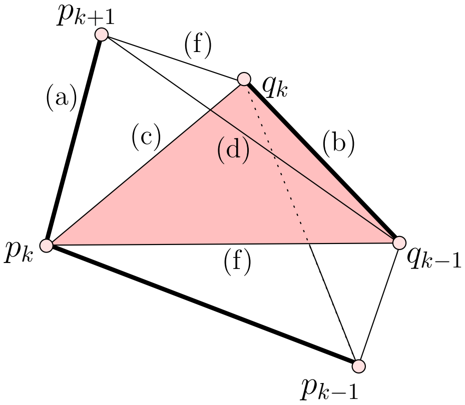


Figure 10: The first half of the inductive step in which we add p_{k+1} to the construction. In the next step, we will use $\triangle p_k q_k p_{k+1}$ to place q_{k+1} at the furthest distance from q_{k-1}.

Polygonal Chains. Now, we return to the main result of this section, the proof of Theorem 3.10. We begin with the case in which the two curves are (open) polygonal chains in $\mathbb{R}^n$, which we denote as P_1 and P_2. We recall that the main idea of the proof is to turn the two curves in $\mathbb{R}^n$ into curves in $\mathbb{R}^3$. The objectives in this transformation are to preserve lengths while not to increase the Fréchet distance and the total curvatures. Letting $\epsilon > 0$, there exists a piecewise linear homeomorphism $f : P_1 \to P_2$ such that $\|x - f(x)\| \leq \mathcal{F}(P_1, P_2) + \epsilon$ for all $x \in P_1$. Writing V_j for the set of vertices of P_j, we are interested in $V = V_1 \cup f^{-1}(V_2)$. Assuming $m = |V|$, we write $v_0, v_1, \ldots, v_{m-1}$ for the m points in order along P_1. Since f is a homeomorphism, $f(v_0), f(v_1), \ldots, f(v_{m-1})$ are the corresponding points in order along P_2.

We now map V and $f(V)$ into $\mathbb{R}^3$. We start by isometrically embedding the first two vertices of both polygons, mapping v_0 to p_0, v_1 to p_1, $f(v_0)$ to q_0, and $f(v_1)$ to q_1. We map the remaining vertices inductively. Suppose v_0 through v_k and $f(v_0)$ through $f(v_k)$ have been mapped to $\mathbb{R}^3$ such that the following eight conditions are satisfied:

(a) $\|v_i - v_{i-1}\| = \|p_i - p_{i-1}\|,$ for $1 \leq i \leq k,$

(b) $\|f(v_i) - f(v_{i-1})\| = \|q_i - q_{i-1}\|,$ for $1 \leq i \leq k,$

(c) $\|v_i - f(v_i)\| = \|p_i - q_i\|,$ for $0 \leq i \leq k,$

(d) $\|v_i - v_{i-2}\| \leq \|p_i - p_{i-2}\|,$ for $2 \leq i \leq k,$

(e) $\|f(v_i) - f(v_{i-2})\| \leq \|q_i - q_{i-2}\|,$ for $2 \leq i \leq k.$

(f) $\|v_i - f(v_{i-1})\| = \|p_i - q_{i-1}\|,$ for $1 \leq i \leq k,$

(g) $\|v_{i-1} - f(v_i)\| = \|p_{i-1} - q_i\|,$ for $1 \leq i \leq k,$

(h) $\|v_i - f(v_{i-2})\| = \|p_i - q_{i-2}\|,$ for $2 \leq i \leq k,$

The first two equalities ensure that the total length of each curve is preserved. Equation (c) ensures that the Fréchet distance at the vertices, and hence the edges, does not increase. The two inequalities guarantee that turning angles do not increase. Along with (a), (b), and (c), the final three equalities are needed for the inductive arguments.

Next, we explain how to construct p_{k+1} and q_{k+1}. Generally speaking, we map p_{k+1} (and q_{k+1}) to a vertex that maximizes its distance to p_{k-1} by choosing a point on the opposite side of $\triangle p_k q_k q_{k-1}$ ($\triangle p_{k+1} q_k p_k$). We use (b), (c), and (f) for $i = k$, which we get by inductive assumption. Thus, we have $\triangle p_k q_k q_{k-1}$. We can therefore map v_{k+1} to a point p_{k+1} such that (a), (f), and (h) are satisfied for $i = k + 1$. There are two such points p_{k+1}, and we choose the one such that the new tetrahedron, $p_{k+1} q_k p_k q_{k-1}$, and the old tetrahedron, $q_k p_k q_{k-1} p_{k-1}$, overlap only on the shared face defined by $q_k p_k q_{k-1}$. By Lemma 3.11, this is also the configuration that maximizes the distance between p_{k-1} and p_{k+1} without distorting the either tetrahedra. This gives us (d) for $i = k+1$. Finally, we use the same method to construct q_{k+1} using (a) and (f) for $i = k+1$ and (c) for $i = k$. We choose q_{k+1} so that (b), (c), (e), and (g) are satisfied for $i = k+1$. This completes the inductive step and thus the construction of the two open polygonal curves in $\mathbb{R}^3$.

Using the construction described above to map polygonal chains into $\mathbb{R}^3$, we can use Lemmas 3.2, 3.3, and 3.8 to prove Theorem 3.10 for closed smooth curves.

3.6. **Open Questions.** Now, we focus on two areas where we define specific open questions: the approximation of the Fréchet distance and the computation of the degree k Vineyard metric.

We recall from Section 3.2 that the relationship between the Fréchet distance (6) and the Hausdorff distance (4) for k-bounded polygonal chains in the Euclidean plane was quantified by Alt et al. in [15]:

$$\mathcal{F}(P_1, P_2) \leq (k+1)\mathcal{H}(P_1, P_2). \tag{11}$$

It has not been shown that (11) is a tight bound, but we do know that $\mathcal{H}(P_1, P_2) \leq \mathcal{F}(P_1, P_2)$ is a tight lower bound for the Fréchet distance. For closed convex curves in the plane, the Fréchet distance and the Hausdorff distance coincide. By the example given in Figure 8, we know that the Fréchet distance and the Hausdorff distance are not the same metric.

To prove Equation (11), one must examine the free space diagram $F_\delta(P_1, P_2)$, where δ is the Hausdorff distance $\mathcal{H}(P_1, P_2)$. Given the free space diagram $F_\delta(P_1, P_2)$, Alt et al. attain a bimonotone path φ_δ from $(0, 0)$ to $(1, 1)$. Unlike before, when we required that the path be restricted to the feasible space in the diagram, this path may go through both infeasible and feasible space. This curve can be parameterized by $\varphi_\delta(t) = (\rho_1(t), \rho_2(t))$ for $t \in I$. Alt et al. show that the reparameterizations ρ_1 and ρ_2 of P_1 and P_2 respectfully yield a Fréchet distance of at most $(k + 1)\delta$ by examining the geometric situations that can arise in the planar curves P_1 and P_2 given $F_\delta(P_1, P_2)$. We ask how to generalize this result:

Open Question 3.12 (Relating Metrics in Higher Dimensions)**.** *What is the relationship between the Fréchet and Hausdorff distances in higher dimensions?*

I conjecture that (11) holds for curves in $\mathbb{R}^n$ for any n. The crux of the proof presented in [15] resides in choosing a curve in $F_\delta(P_1, P_2)$, then determining the Fréchet distance between corresponding parameterizations of the curves in $\mathbb{R}^2$. A generalization of this result must examine the potential curves in $\mathbb{R}^n$. In addition, we ask under which circumstances does the Fréchet metric and the Hausdorff metric coincide for non-planar curves?

As we have discussed, the Fréchet distance is not easy to compute. Given two polygonal chains with $O(n)$ vertices each, the fastest known algorithm to compute the Fréchet distance is $O(n^2 \log n)$ using parametric search [14]; however, parametric search requires a complicated and delicate data structure. This algorithm was successfully implemented by van Oostrum and Veltkamp for the Shape Matching Environment (SHAME) library [60], but it is a complicated library. A slightly more understandable algorithm uses binary search with a $\mathrm{O}(n^3 \log n)$ time complexity [43]. This is an undesirable running time. Thus, we turn to approximating the Fréchet distance. For example, inequality (11) is particularly interesting since the Hausdorff distance is easier to compute than the Fréchet distance: Alt et al. show that the Hausdorff distance for simple polygonal chains can be computed in $\mathrm{O}(n \log n)$ [12]. In Section 3.4, we described a near-linear time approximation algorithm for the Fréchet distance between c-packed polygonal curves. This algorithm uses a curve simplification that decreases the complexity of the free space diagram, and an efficient construction algorithm for the new free space diagram.

Open Question 3.13 (Approximating General Curves)**.** *Can the algorithm for approximating the Fréchet distance, presented in* [32]*, be extended to non-polygonal curves?*

For example, can we find a $(1 + \epsilon)$ approximation of the Fréchet distance between piecewise smooth curves? Moreover, we ask how to approximate the Fréchet distance between higher dimensional surfaces. In general, computing the Fréchet distance between surfaces is NP-Hard [44], but it is semi-computable, i.e., there exists " a non-halting Turing machine which produces a monotone decreasing sequence of rationals converging to the result" [13].

Next, we ask a question regarding the vineyard metric. In Equation (8), we defined the degree k vineyard distance, V_k, of a Homotopy H:

$$V_k(H) = \sum_{s \in V} \int_0^1 \frac{\partial s(t)}{\partial t}\, dt.$$

Looking to justify the usefulness of this metric, we ask:

Open Question 3.14 (Vineyard Metric Stability)**.** *Does there exist a good stability result for the degree k vineyard metric?*

The vineyard metric is sensitive to the homotopy, not just to the initial and the final states. For the constant homotopy at f, the vineyard metric is zero. However, we can conceive of a non-constant homotopy H such that $H(x, 0) = H(x, 1) = f(x)$. The distance that the associated vineyard measures is nonzero under the vineyard metric. We desire to find a good stability result, which will require defining a metric on the space of homotopies. If we can do this, perhaps we can bound the degree k vineyard metric by a function of the measure of the homotopy.

4. Scale Space

The heat equation is a mathematical description of the dissipation of heat through a region in space. We start with the initial reading or an estimate of the temperature throughout an adiabatically enclosed space, such as an empty room. After infinite time and given no external changes, the temperature at each point in the space will (usually) converge to the average of the initial temperatures. The scale space of an image uses the heat equation to study the image at all scale levels. At a small scale, the scale space preserves individual pixel values, but at a large scale, the values blur together.

4.1. Background.

A grayscale image h_0 is a function from a rectangular domain $\mathbb{M} \subset \mathbb{R}^n$ to $\mathbb{Z}_{256} = \{0, 1, 2, \ldots, 255\}$. Later, we will discuss color images, maps $\mathbb{M} \to \mathbb{Z}_{256}^3$, but in this section we will limit our discussion to images with range $\mathbb{Z}_{256}$. Thinking of the value $h_0(x)$ at a point $x \in \mathbb{M}$ as *heat*, we consider the diffusion of heat over time. We call this the study of the *deep structure* or *scale space* of the image h_0. This process is used to create a family of functions parameterized by t that is used to understand the image. There are two main challenges in the area of shape identification: extracting invariant shape descriptors and deciding if two shape descriptors represent the same shape. A multitude of challenges arises in both areas, although there are many examples where progress has been made. Still, there exists a gap between human perception and computer vision. We use scale space to provide a mathematical description of an image at all scales, attempting to quantify the natural way that humans view the world.

The Heat Equation. We describe the heat equation as it applies to a continuous function. A more detailed account can be found in [16, 21]. In Section 4.2, we modify these equations to approximate the solution in the discrete case and, in Section 4.3, we introduce the heat equation homotopy.

Let $\{b_1, b_2, \ldots, b_m\}$ be an orthonormal basis for $\mathbb{M}$, a closed cuboid in $\mathbb{R}^m$. Let f be a real-valued, twice differentiable function on $\mathbb{M}$. We define the Laplace-Beltrami operator as the following equation:

$$\Delta_{\mathbb{M}}(f) = \operatorname{div}(\triangledown(f)). \tag{12}$$

In Euclidean space, the Laplace-Beltrami operator is often referred to as simply the *Laplace* operator, and we can work out an explicit formula:

$$\triangle\, f(x) = \sum_{i=1}^{m} \frac{\partial^2 f}{\partial b_i^2}(x).$$

The general form of the heat equation satisfies the following two conditions:

$$\frac{\partial u}{\partial t}(x,t) - \triangle\, u(x,t) = 0 \tag{13}$$

and

$$u(x,0) = f(x), x \in \mathbb{M}. \tag{14}$$

If we can solve this partial differential equation (PDE), we obtain $u(x,t)$ defined for all $x \in \mathbb{M}$ and $t \geq 0$. Typically, the heat equation has the additional constraint that $u(x,t)$ is constant with respect to t for all $x \in \partial\mathbb{M}$. This is an example of the *Dirichlet boundary condition* that fixes the values in $\partial\mathbb{M}$. However, we are interested in the case where $u(x,t)$ is *heat conserving*; that is, we would like avg(u_{t_1}) =avg(u_{t_2}) for all t_1, t_2. As we will show, in order to obtain this goal, the values on the boundary will depend upon the interior values.

Equation (13) describes how the temperature changes with respect to time and space. We note that if we impose the condition $\frac{\partial u}{\partial t}(x,t) = 0$, then the heat equation will not change with respect to time, and Equation (13) becomes Laplace's equation, $\Delta u(x,t) = 0$. This is known as the steady-state heat equation and has a unique solution. The iterative methods that we look at in Section 4.2 aim at finding an approximation of $u(x,t)$ for this problem. The final solution will be constant with respect to time. We are interested in studying the behavior of heat equation as it approaches the solution to the steady-state heat equation.

The heat equation has several important properties. First, it is invariant under translation. To see why, suppose $u(x,t)$ is a solution to the heat equation. Now, let $w(x,t) = u(x,t) + c$, for some $c \in \mathbb{R}^n$. Since $\partial w/\partial t = \partial u/\partial t$ and $\partial w/\partial x_i = \partial u/\partial x_i$, we know that $w(x,t)$ also satisfies (13) and is thus a solution to the heat equation, with initial value $w(x,0) = u(x,0) + c$. As we will see next, the heat equation can be explained by convolving a function with the Gaussian kernel in Equation (15). Thus, for all $t > 0$, the function $u(x,t)$ is smooth and singularities are isolated. In addition, the heat equation, which describes the physical phenomena of heat dissipating through space, can be applied to many areas. We are interested in the application of the heat equation to the two main challenges in shape identification.

Gaussian Blurring. The concept of scale space has been introduced by Koenderink [49] and Witkin [61]. They constructed it by blurring an image with a one-parameter family of Gaussian kernels. The n-dimensional isotropic Gaussian kernel at a point x with standard deviation t is:

$$G_n(x,t) = \frac{1}{(\sqrt{2\pi}t)^n} e^{-\frac{||x||^2}{2t^2}}. \tag{15}$$

We call this kernel *isotropic* since it is uniform in all directions originating at x. The normalization constant, $-1/(\sqrt{2\pi}t)^n$, ensures that the integral of the Gaussian kernel over the domain is unity:

$$\int_{x \in \mathbb{R}^n} G_n(x,t)\, dx = 1. \tag{16}$$

We then blur the image $h_0 \colon \mathbb{R}^n \to [0,255]$ at a point $y \in \mathbb{R}^n$ by integrating the values of h_0, weighting each point by its value under the Gaussian kernel centered at y:

$$\mathrm{Blur}(y,t) = \int_{x \in \mathbb{R}^n} G_n(x-y,t) h_0(x)\, dx. \tag{17}$$

This process is called *convolution.* In this case, the standard deviation t is the *scale parameter* of the scale space.

We note that changing the standard deviation of the Gaussian kernel has a simple geometric interpretation. Specifically, we go between $t_0 \leq t_1$ by first stretching the distribution along each basis direction b_i of the domain by a factor of t_1/t_0, and second shrinking it along the range, by t_0/t_1. The result is an isotropic function such that the integral over the entire domain is one, as in (16). Indeed, we have

$$G_n(x,t_1) = \frac{t_0}{t_1} G_n(\frac{x}{t_1/t_0}, t_0).$$

We now wish to verify that $G(x,t) = G_n(x, \sqrt{2t})$ is a solution to the heat equation; that is, we will show $\partial G/\partial t - \bigtriangleup G = 0$. First, we compute the derivative of $G(x,t)$ with respect to t, then we will compute the Laplacian operator.

$$
\begin{aligned}
\frac{\partial G}{\partial t}(x,t) &= -\frac{1}{(\sqrt{4\pi t})^n}\frac{\partial\left(e^{-\frac{|x|^2}{4t}}\right)}{\partial t} - \frac{\partial\left((\sqrt{4\pi t})^{-n}\right)}{\partial t}e^{-\frac{|x|^2}{4t}} \\
&= -\frac{1}{(\sqrt{4\pi t})^n}\left(e^{-\frac{|x|^2}{4t}}\left(\frac{|x|^2}{4t}\right)\right) \\
&\quad -\left((-n)(\sqrt{4\pi t})^{-n-1}(4\pi t)^{-1/2}(2\pi)\right)e^{-\frac{|x|^2}{4t}} \\
&= G(x,t)\left(\frac{|x|^2}{4t^2}\right) \\
&\quad +\left((-n)(\sqrt{4\pi t})^{-1}(4\pi t)^{-1/2}(2\pi)\right)G(x,t) \\
&= G(x,t)\left(\frac{|x|^2}{4t^2}\right) - \left(\frac{2n}{4t}\right)G(x,t)
\end{aligned}
$$

$$
\frac{\partial G}{\partial t}(x,t) = G(x,t)\left(\frac{|x|^2-2tn}{4t^2}\right) \tag{18}
$$

Now, we take the first partial derivative of $G(x,t)$ with respect to a basis vector b_i:

$$
\begin{aligned}
\frac{\partial G}{\partial b_i}(x,t) &= -\frac{1}{(\sqrt{4\pi t})^n}\frac{\partial\left(e^{-\frac{|x|^2}{4t}}\right)}{\partial b_i} - \frac{\partial\left((\sqrt{4\pi t})^{-n}\right)}{\partial b_i}e^{-\frac{|x|^2}{4t}} \\
&= -\frac{1}{(\sqrt{4\pi t})^n}e^{-\frac{|x|^2}{4t}}\cdot\frac{\partial\left(-\frac{|x|^2}{4t}\right)}{\partial b_i} \\
&= G(x,t)\cdot\frac{-2|x|}{4t}\frac{\partial|x|}{\partial b_i} \\
&= G(x,t)\cdot\frac{-|x|}{2t}\frac{x_i}{|x|} \\
&= G(x,t)\cdot\frac{-x_i}{2t}.
\end{aligned}
$$

We take the derivative of this result to obtain one summand of (12):

$$\begin{aligned}
\frac{\partial^2 G}{\partial b_i^2}(x,t) &= \frac{\partial\left(G(x,t)\cdot\frac{-x_i}{2t}\right)}{\partial b_i} \\
&= G(x,t)\frac{\partial\left(\frac{-x_i}{2t}\right)}{\partial b_i} + \frac{-x_i}{2t}\frac{\partial G(x,t)}{\partial b_i} \\
&= G(x,t)\frac{-1}{2t} + \frac{-x_i}{2t}G(x,t)\cdot\frac{-x_i}{2t} \\
\frac{\partial^2 G}{\partial b_i^2}(x,t) &= G(x,t)\cdot\frac{x_i^2-2t}{4t^2}. \qquad (19)
\end{aligned}$$

We use (18) and (19) for $i = 1, 2, \dots, m$ to obtain the following equation:

$$\frac{\partial G}{\partial t}(x,t) - \sum_{i=1}^{m}\frac{\partial^2 u}{\partial b_i^2}(x,t) = 0 \qquad (20)$$

Comparing (20) with (13), we see that the heat equation is satisfied. In fact, the Gaussian kernel is the *fundamental solution* to the heat equation. General solutions to the heat equation with a specified initial condition are found by convolution with the fundamental solution, as shown in Equation (17).

Intuitively, we hope that increasing the scale parameter t will blur away any unimportant details of the image. Since heat diffuses, one might expect regions where the heat exceeds a particular threshold would expand. When two regions meet, it seems plausible the regions would merge and further expand. This intuition, however, is false [30]. During the course of the heat equation, regions (of a particular value or range of values) can merge and split. Furthermore, new regions can be created, as demonstrated in the following example:

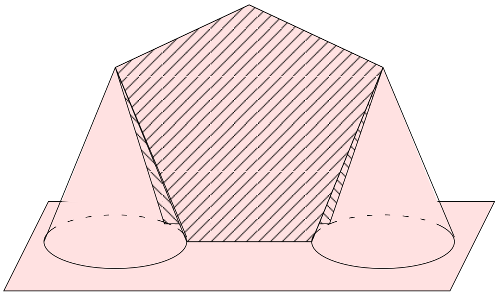


Figure 11: The function used in Example 4.1.

Example 4.1 (Mountain and Bridge)**.** Suppose we have two cones connected by two pentagons, as shown in Figure 11. We will call this surface S. Now, consider the height $h\colon \mathbb{R}^2 \to \mathbb{R}$ defined to be the height of the surface at (x, y) below S and zero elsewhere. We assume that the base of the cones and the pentagons is in the xy-plane. There is one dimension-one critical value, the pinnacle of the pentagons. If we apply the heat equation to this surface, there will soon be three local maxima: one corresponding to the original critical value and two from the cone tips. In fact, Damon [30] proves that the creation of singularities, points where the gradient vector field vanishes, is a generic event in the heat equation. A *singularity* for a fixed time t is a point x in the spatial domain such that the gradient vector field vanishes at x. Kalitzin et al. proved that although the number of singularities can change in scale space, the sum of the topological numbers (defined in their paper) of the singularities remains constant [48].

4.2. **Iterative Algorithm for the Heat Equation.** Solving a PDE is not a simple task. The first step is to compute $u_t(x) = u(x, t)$ over the discretized domain. There are two issues that can arise when using the continuous formulation of the heat equation described in Section 4.1:

1. We need to solve the PDE presented in Equations (13) and (14).
2. The partial derivatives $\frac{\partial^2 u_t}{\partial b_i^2}(x)$ and $\frac{\partial u_t}{\partial t}(x)$ are not well defined over a discrete domain.

In order to resolve these issues, we defer to numerical methods to estimate this solution, which require spatial and temporal discretization [21]. The goal is to obtain a discretization of the heat equation solution $u(x, t)$.

Spatial Discretization. In the following computations, we use a regular grid decomposition of $\mathbb{M} = [0, 1]^2$, writing $x_i = (i - 1)h$ and $y_j = (j - 1)h$, where $h = 1/(m - 1)$ for some fixed integer $m \geq 2$. There are m^2 vertices in this decomposition. The vertices are connected by vertical and horizontal edges. The neighborhood of a point, $\mathrm{Nhd}(i, j)$, is defined to be the set of

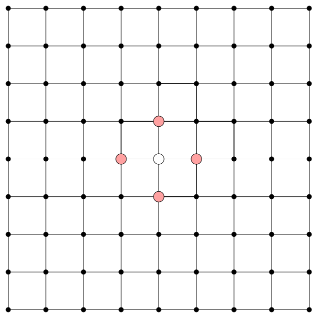


Figure 12: In the center, the white dot represents the vertex $x = (i, j)$. The vertices highlighted in pink are those whose values contribute to the estimate of the heat equation at x.

points directly connected to $x_{i,j}$:

$$\mathrm{Nhd}(i,j) = \{(i, j \pm 1), (i \pm 1, j)\}. \tag{21}$$

The set on the right hand side of (21) is specific to a regular grid decomposition of the domain. For any graph, it is possible to define a neighborhood of a vertex, $\mathrm{Nhd}_G(v)$, to be the set of vertices connected to v by an edge in the graph G.

Mathematical Description. Let us focus on the steady-state heat equation over $\mathbb{R}^2$:

$$\frac{\partial^2 u}{\partial b_1^2}(x,t) + \frac{\partial^2 u}{\partial b_2^2}(x,t) = 0. \tag{22}$$

In this equation, we are using $\{b_1, b_2\}$ as the standard basis vectors for $\mathbb{R}^2$. At each mesh point $x = (i,j)$, we employ the Taylor polynomial in the variable b_1 to obtain an approximation of the second derivative with respect to b_1:

$$\frac{\partial^2}{\partial b_1^2} u((i,j),t) = \frac{u((i+1,j),t) - 2u((i,j),t) + u((i-1,j),t)}{h^2}, \tag{23}$$

where h is the spatial step size. Similarly, we also have an approximation for the second derivative with respect to b_2:

$$\frac{\partial^2}{\partial b_2^2} u((i,j),t) = \frac{u((i,j+1),t) - 2u((i,j),t) + u((i,j-1),t)}{h^2}. \tag{24}$$

To simplify notation, we will now use $x_{i,j}$ to denote vertex (i,j). If we plug (23) and (24) into (22), then we obtain the following equation:

$$4u(x_{i,j},t) - u(x_{i+1,j},t) - u(x_{i-1,j},t) - u(x_{i,j+1},t) - u(x_{i,j-1},t) = 0 \tag{25}$$

As we will see below, the approximation of the heat equation is made by looking at local neighborhoods for each vertex $x_{i,j}$. Figure 12 highlights the four neighbors of the mesh needed to compute an approximation to the heat equation. We have m^2 equations of the form presented in (25), one for each point in the $m \times m$ grid. We relabel the grid in column-major order in order to use one index instead of two: $v_{\{(j-1)n+i\}} := u(x_{i,j},t)$. Then, we may express the m^2 linear equations in matrix-vector form, $Av = 0$.

Algorithm. As we have shown above, solving the discrete heat equation finds a solution v to the linear system of equations $Av = 0$. We have described how to construct the matrix A, as the coefficient matrix for the system of linear equations. In fact, A is equal to L_m, the *Poisson matrix* of order m. We note here that this matrix is $m^2 \times m^2$. We can write $L_m = D - N$, where D is the diagonal matrix $4 \cdot I$ and N is a matrix with 0's on the diagonal and with only 1 as the non-zero entries of the matrix. Sometimes we refer to D as the *valency matrix*, since it expresses the degree of each mesh point. The matrix N is symmetric and we call it the *neighborhood matrix* since the non-zero entries in row i correspond to the neighbors of the mesh point v_i [17]. That is, $N(i,j) = 1$ iff v_i and v_j are adjacent.

The iterative algorithm can be defined by these matrices. We want a solution of the form $Av = 0$, which means $(D - N)v = 0$. We can re-write this so that $Dv = Nv$. And, the iterative algorithm can immediately be seen:

$$v_{\text{new}} = (D^{-1}N)v.$$

In the original formulation, this translates to:

$$u_{t+1}(x) = \frac{1}{4} \sum_{y \in \text{Nhd}(x)} u_t(x),$$

where the neighborhood $\text{Nhd}(x)$ is the neighborhood of x defined by Equation (21). This iterative method is known as *Jacobi iteration* [21]. We can generalize this technique to non-regular meshes by computing u_t at a vertex by averaging the values of u_{t-1} at the neighbors of that vertex, where the neighbors are the vertices directly connected by edges.

4.3. **Heat Equation Homotopy.** We use the heat equation to define a homotopy between f and g that disperses the difference $f - g$. As described in Section 2.5, we make an informed matching of the points in the persistence diagrams for f and g using the vines in the corresponding vineyard of the homotopy. As a result, the matching and vineyard distance obtained will be based on the underlying functions f and g. This section describes how to create the discrete heat equation homotopy.

Dispersing the Difference. Let the manifold $\mathbb{M}$ be a rectangular subset of $\mathbb{R}^2$. Then, we can think of the functions $f, g \colon \mathbb{M} \to \mathbb{R}$ as surfaces in $\mathbb{R}^3$. Let u_0 be the difference $f - g$. If $u_0(x) = 0$ for all $x \in M$, then we have $f = g$. Define the *average* of a function as the integral divided by the area:

$$\text{avg}(f) = \frac{\int_{\mathbb{M}} f(x)dx}{\text{area}(\mathbb{M})}$$

and

$$\text{avg}(g) = \frac{\int_{\mathbb{M}} g(x)dx}{\text{area}(\mathbb{M})}.$$

Then, we calculate the average value of u_0 over the domain $\mathbb{M}$ by subtraction:

$$\text{avg}(u_0) = \text{avg}(f) - \text{avg}(g).$$

We apply the heat equation to u_0 and obtain $u(x,t)$ where $u(x,0) = u_0(x)$ and $\lim_{t\to\infty} u(x,t) = \text{avg}(u_0)$. For simplicity, we assume that the average of u_0 is zero. If this is not the case, we impose this condition by subtracting $\text{avg}(u_0)$ from f.

Now, we observe u disperse through time until u becomes the zero function. Similarly, we know that $g(x) + u(x,t)$ smoothly transitions from f to g. Although the value of $u(x,t)$ approaches zero for all x as t increases, we will stop at a time τ when $u(x,\tau)$ is in $(-\epsilon, +\epsilon)$ for all $x \in \mathbb{M}$ and for some $\epsilon > 0$. Then, the function $g + u$ goes from f to a function ϵ-close to g. Furthermore, the manner in which the function $g + u$ changes is dictated by the heat equation.

Creating the Discrete Homotopy. After applying the discrete heat equation to $u(x, 0)$, we have a function $u : \mathbb{M} \times I_\tau \to \mathbb{R}$, where I_τ is the discrete interval discussed in Section 2.4. The discrete heat equation homotopy $f_t : \mathbb{M} \times I_\tau \to \mathbb{R}$ is then defined by the equation:

$$f_t(x) = f(x) + u(x, t).$$

Notice that $f_0(x) = f(x)$ and $f_1(x) \approx g(x)$ by the choice of τ. By using this homotopy, the initial difference between f and g disperses. Although $u(x, t)$ approaches the constant zero function as t increases, interesting things can happen along the way. For example, critical values can be created or destroyed, as demonstrated in Section 4.1.

4.4. **Open Questions.** We present two open questions regarding scale space. The first question is theoretical and aims to relate scale space with the total persistence (2). The second question is more experimental in nature, but can lead to some interesting results and has many applications.

In Section 4.1, we saw that critical points can be created by the Heat Equation. We have observed, however, that the degree k total persistence for $k = 2$ and $k = 3$ approaches zero as the scale parameter increases [41].

Open Question 4.2 (Creation of Critical Points in Scale Space)**.** *Can we use persistence to verify the observation that the creation of critical points is rare and minor?*

Moreover, can we quantify the relationship between persistence and the deep structure of an image? The Moment Lemma (Lemma 2.2), which bounds the degree k total persistence, requires that the domain is compact. If we were able to apply the Moment Lemma, then we would have a vanishing total persistence as the scale parameter t increases. Unfortunately, the domains of interest, $\mathbb{R}^2$ and $\mathbb{R}^3$, are not compact. Since applying the heat equation to an infinite domain is infeasible, implementations and applications of the heat equation do use compact domains. We ask how and when can we apply the Moment Lemma to bound the total persistence in the limit of the heat equation?

Scale space, as we have described it in Section 4.1, is defined for a real-valued intensity function over a subset of Euclidean space. Often, the images (2D or 3D) that we are interested in have three color values, not just one grayscale value as this setting assumes. Scale space has been used to analyze the color histogram associated with an image [51]. The *color histogram* partitions the domain (intensities of Red, Green, and Blue values) into bins and records the number of pixels in each bin. This approach does not take into consideration the location and adjacencies of colors. We wish to analyze and compare the deep structure of each color intensity:

Open Question 4.3 (Multiple Heat Equations)**.** *Do we gain information by separating out the color values instead of considering only one intensity (the grayscale value)?*

In other words, we ask what happens if we apply the heat operator independently to each color. In particular, it may be interesting to watch how the areas of equal intensity change with respect to time. For image segmentation, areas of equal intensity are boundaries of regions where one intensity is more dominant than the another. Moreover, we can apply this method to other areas, such as cell biology. The development of an axon, the long slender part of a nerve cell that carries electrical signals away from a neuron's core, senses and responds to chemical gradients in the environment. For example, a calcium gradient in the environment will result in axonal elongation [45]. There are many chemical gradients of interest in the development of cells, including calcium and sodium. Knowing how the proportions of each chemical as each disperses through time might lead to more accurate simulations of cell development. Open Question 4.3 can be generalized to ask, *what can we learn from both the dispersion of one agent and the interaction among multiple agents as they diffuse through time?*

5. Conclusion

Throughout the next year, I will investigate the open problems presented in Sections 3.6 and 4.4. In Spring 2011, I will meet with Herbert Edelsbrunner in order to develop a clear plan for my dissertation. By August 2011, I will email my committee with a summary of the progress that I have made as well as the problem or problems that I have identified that will comprise my dissertation. By that time, I will have a concrete outline of the milestones leading up to my defense. Although I will not schedule a formal committee meeting, I will discuss these plans with each committee member, preferably in person. I plan to graduate in Spring or Summer 2012.

References

[1] *Proc. of the Eighth Annual Symp. Theoret. Aspects Comput. Sci.* (1991), vol. 480 of *Lect. Notes Comput. Sci.*, Springer Berlin-Heidelberg. Symposium held in Hamburg.

[2] *Proc. of the Eleventh Annu. Symp. Comput. Geom.* (New York, Jun. 1995), ACM. Symposium held in Vancouver.

[3] *Proc. of the 2001 IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognition* (Dec. 2001), IEEE Comput. Soc. Conference held in Kauai, HI.

[4] *Proc. of the Eighteenth Annu. Symp. Comput. Geom.* (New York, Jun. 2002), ACM. Symposium held in Barcelona.

[5] *Proc. of the* 21^{st} *Annu. Symp. Comput. Geom.* (New York, Jun. 2005), ACM. Symposium held in Pisa.

[6] *Proc. of the* 21^{st} *Eur. Wksp. Comput. Geom.* (Mar. 2005). Workshop held in Eindhoven, Netherlands.

[7] *Proc. of the* 22^{nd} *Annu. Symp. Comput. Geom.* (New York, Jun. 2006), ACM. Symposium held in Sedona, AZ.

[8] *Proc. of the* 26^{th} *Annu. Symp. Comput. Geom.* (New York, Jun. 2010), ACM. Symposium held in Snowbird, UT.

[9] Agarwal, P. K., Efrat, A., and Sharir, M. Vertical Decomposition of Shallow Levels in 3-Dimensional Arrangements and Its Applications. In *Proc. of the Eleventh Annu. Symp. Comput. Geom.* [2], pp. 39–50. 1995.

[10] Albers, S., and Weil, P., Eds. *Proc. of the* 25^{th} *Annual Symp. Theoret. Aspects Comput. Sci.* (Dagstuhl, Germany, Feb. 2008), vol. 1 of *Leibniz Internat. Proc. Inform.*, Schloss Dagstuhl–Leibniz-Zentrum für Informatik. Symposium held in Bordeaux.

[11] Alexa, M., Kazdan, M., and Polthier, K., Eds. *Proc. of the Seventh Eurographics Symp. Geom. Processing* (2009), vol. 28, Eurographics. Symposium held in Berlin.

[12] Alt, H., Behrends, B., and Blömer, J. Approximate Matching of Polygonal Shapes. *Ann. Math. Artif. Intell. 13*, 3 (Sept. 1995), 251–265.

[13] Alt, H., and Buchin, M. Semi-Computability of the Fréchet Distance Between Surfaces. In *Proc. of the* 21^{st} *Eur. Wksp. Comput. Geom.* [6], pp. 45–48. 2005.

[14] Alt, H., and Godau, M. Computing the Fréchet Distance Between Two Polygonal Curves. *Internat. J. Comput. Geom. Appl. 5*, 1 (Mar. 1995), 75–91.

[15] Alt, H., Knauer, C., and Wenk, C. Comparison of Distance Measures for Planar Curves. *Algorithmica 38*, 1 (Oct. 2003), 45–58.

[16] Alt, H. W. Analysis III. Online., 2007. Lecture Winter Semester 2001–2002 at the Inst. of Applied Math., Univ. of Bonn.

[17] Babić, D., Klein, D. J., Lukovits, I., Nikolić, S., and Trinajstić, N. Resistance-Distance Matrix: A Computational Algorithm and Its Application. *Internat. J. Quantum Chem. 90*, 1 (2002), 166–176.

[18] Barcelo, H., Kramer, X., Laubenbacher, R., and Weaver, C. Foundations of a Connectivity Theory for Simplicial Complexes 1. *Adv. in Appl. Math. 26*, 2 (2001), 97–128.

[19] Barcelo, H., and Laubenbacher, R. Perspectives on A-Homotopy Theory and Its Applications. *Discrete Math. 298*, 1–3 (Aug. 2005), 39–61.

[20] Bundy, A., Ed. *Proc. of the Eighth Internat. Joint Conf. Artif. Intell.* (Aug. 1983), vol. 2, Morgan Kaufmann Publishers. Conference held in Karlsruhe, West Germany.

[21] Burden, R. L., and Faires, J. D. *Numerical Analysis*, eighth ed. Thompson, 2005.

[22] Chakerian, G. D. An Inequality for Closed Space Curves. *Pacific J. Math. 12*, 1 (1962), 53–57.

[23] Chazal, F., Cohen-Steiner, D., Guibas, L. J., Mémoli, F., and Oudot, S. Y. Gromov-Hausdorff Stable Signatures for Shapes using Persistence. In Alexa et al. [11], pp. 1393–1403. 2009.

[24] Cohen-Steiner, D., and Edelsbrunner, H. Inequalities for the Curvature of Curves and Surfaces. *Found. Comput. Math. 7*, 4 (Nov. 2007), 391–404.

[25] Cohen-Steiner, D., Edelsbrunner, H., and Harer, J. Stability of persistence diagrams. In *Proc. of the* 21^{st} *Annu. Symp. Comput. Geom.* [5], pp. 263–271. 2005.

[26] Cohen-Steiner, D., Edelsbrunner, H., and Harer, J. Extending Persistence Using Poincare and Lefschetz Duality. *Found. Comput. Math. 9*, 1 (Feb. 2009), 79–103.

[27] Cohen-Steiner, D., Edelsbrunner, H., Harer, J., and Mileyko, Y. Lipschitz Functions have L_p-Stable Persistence. *Found. Comput. Math. 10*, 2 (April 2010), 127–139.

[28] Cohen-Steiner, D., Edelsbrunner, H., and Morozov, D. Vines and Vineyards by Updating Persistence in Linear Time. In *Proc. of the* 22^{nd} *Annu. Symp. Comput. Geom.* [7], pp. 119–126. 2006.

[29] Cook IV, A. F., and Wenk, C. Geodesic Fréchet Distance Inside a Simple Polygon. In Albers and Weil [10], pp. 193–204. 2008.

[30] Damon, J. Local Morse Theory for Solutions to the Heat Equation and Gaussian Blurring. *J. Differential Equations 115*, 2 (Jan. 1995), 368–401.

[31] Di Fabio, B., and Landi, C. Stability of Reeb Graphs under Function Perturbations: the Case of Closed Curves. http://arxiv.org/abs/1003.4610, April 2010. University of Bologna.

[32] Driemel, A., Har-Peled, S., and Wenk, C. Approximating the Fréchet Distance for Realistic Curves in Near Linear Time. [8]. Arxiv preprint arXiv:1003.0460.

[33] Edelsbrunner, H., and Guibas, L. J. Topologically Sweeping an Arrangement. *J. Comput. System Sci. 38*, 1 (Feb. 1989), 165–194. Corrigendum appears in J. Comput. System Sci.42, 2 (April 1991), 249–251.

[34] Edelsbrunner, H., and Harer, J. Persistent Homology–A survey. In *Surveys on Discrete and Computational Geometry–Twenty Years Later*, vol. 453. Amer. Math. Soc., 2006, pp. 257–282.

[35] Edelsbrunner, H., and Harer, J. *Computational Topology. An Introduction.* Amer. Math. Soc., Providence, RI, 2010.

[36] Edelsbrunner, H., Letscher, D., and Zomorodian, A. Topological Persistence and Simplification. *Discrete Comput. Geom. 28*, 4 (Jul. 2002), 511–533.

[37] Efrat, A., Itai, A., and Katz, M. J. Geometry Helps in Bottleneck Matching and Related Problems. *Algorithmica 31*, 1 (Dec. 2001), 1–28.

[38] Euclid. *The Thirteen Books of the Elements*, second ed., vol. 1. Dover, New York, 1956. Books I and II, unabridged.

[39] Fáry, I. Sur la Courbure Totale dune Courbe Gauche Faisant un Noeud. *Bull. Soc. Math. France 77* (1949), 128–138.

[40] Fáry, I. Sur Certaines Inégalités Géométriques. *Acta Sci. Math. (Szeged) 12*, aa (1950), 117–124.

[41] Fasy, B. T. Persistence Diagrams and the Heat Equation Homotopy. http://arxiv.org/abs/1002.1937, 2009. Research Initiation Project, Duke University Comput. Sci. Dept.

[42] Fenchel, W. Über Krümmung und Windung Geschlossener Raumkurven. *Math. Ann. 101*, 1 (Dec. 1929), 238–252.

[43] Godau, M. A Natural Metric for Curves–Computing the Distance for Polygonal Chains and Approximation Algorithms. In *Proc. of the Eighth Annual Symp. Theoret. Aspects Comput. Sci.* [1], pp. 127–136. 1991.

[44] Godau, M. *On the Complexity of Measuring the Similarity between Geometric Objects in Higher Dimensions.* PhD thesis, Freie Univ. Berlin, 1998.

[45] Graham, B., and van Ooyen, A. Mathematical Modelling and Numerical Simulation of the Morphological Development of Neurons. *BMC Neurosci. 7*, 1 (Oct. 2006), S9.

[46] Hatcher, A. *Algebraic Topology.* Cambridge Univ. P., 2002. Electronic Version.

[47] Hopcroft, J. E., and Karp, R. M. An $n^{5/2}$ Algorithm for Maximum Matchings in Bipartite Graphs. *SIAM J. Comput. 2*, 4 (1973), 225–231.

[48] Kalitzin, S. N., Ter Haar Romeny, B. M., Salden, A. H., Nacken, P. F., and Viergever, M. A. Topological Numbers and Singularities in Scalar Images: Scale-Space Evolution Properties. *J. Math. Imaging Vision 9*, 3 (Nov. 1998), 253–269.

[49] Koenderink, J. The Structure of Images. *Biol. Cybernet. 50*, 5 (Aug. 1984), 363–370.

[50] Kuhn, H. W. The Hungarian Method for the Assignment Problem. *Naval Res. Logist. Quarterly 2*, 1 (Mar. 1955), 83–97.

[51] Liu, J., and Yang, Y. H. Multiresolution Color Image Segmentation. *IEEE Trans. Pattern Anal. Mach. Intell. 16*, 7 (Jul. 1994), 689–700.

[52] Milnor, J. *Morse Theory.* No. 51 in Ann. of Math. Stud. Princeton Univ. P., Princeton, 1963.

[53] Rote, G. Computing the Fréchet Distance Between Piecewise Smooth Curves. *Comput. Geom. 37*, 3 (Aug. 2007), 162–174.

[54] Schilling, R. L. *Measures, Integrals and Martingales.* Cambridge Univ. P., New York, 2005.

[55] Singer, D. A. *Geometry: Plane and Fancy.* Springer-Verlag, New York, 1998.

[56] Sullivan, J. Curves of Finite Total Curvature. In *Discrete Differential Geometry*, A. I. Bobenko, P. Shröder, J. M. Sullivan, and G. M. Ziegler, Eds., vol. 38 of *Oberwolfach Seminars.* Birkhäuser Verlag Basel / Switzerland, Boston, 2008, pp. 137–161. In Part II: Curvatures of Discrete Curves and Surfaces.

[57] Tai, K.-C. The Tree-to-Tree Correction Problem. *J. ACM 26*, 3 (Jul. 1979), 422–433.

[58] Tanawongsuwan, R., and Bobick, A. F. Gait Recognition from Time-Normalized Joint-Angle Trajectories in the Walking Plane. In *Proc. of the 2001 IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognition* [3], pp. 726–731. Conference held in Kauai, HI.

[59] Vaidya, P. Geometry Helps in Matching. *SIAM J. Comput. 18*, 6 (1989), 1201–1225.

[60] Van Oostrum, R., and Veltkamp, R. C. Parametric Search Made Practical. In *Proc. of the Eighteenth Annu. Symp. Comput. Geom.* [4], pp. 1–9. 2002.

[61] Witkin, A. P. Scale-Space Filtering. In Bundy [20], pp. 1019–1022. 1983.

Appendix A. Notations

In this appendix, we have a table summarizing the commonly used mathematical symbols throughout this paper. Where appropriate, equation numbers are given for definitions.

Table 1: Mathematical symbols used throughout this document.

	Definition
$\text{Amp}(f)$	The maximum difference in function values of f.
$\text{Dgm}_p(\mathbb{M})$	The p^{th} persistence diagram.
$d^{\infty}(a, b)$	The L_{∞} distance between points a and b (7).
$F_{\delta}(P, Q))$	The free space for given parameterizations of P and Q (10).
$\mathcal{F}(P_1, P_2)$	Fréchet distance between two curves (6).
$G_n(x, t)$	The n-dimensional Gaussian kernel centered at x, with standard deviation t (15).
$H(x, t)$	Homotopy between $H(x, 0)$ and $H(x, 1)$.
$\mathcal{H}(\gamma_1, \gamma_2)$	Hausdorff distance between two curves (4).
$H_p(\mathbb{M})$	The p^{th} homology group of $\mathbb{M}$.
I_m	The discrete interval of length m.
λ	Lipschitz constant.
$u(x, t) = u_t(x)$	A solution to the heat equation.
$\mathbb{M}^m$	An m-manifold.
$\mathbb{M}_s^f$	A sublevel set of the function f.
$N(r)$	The size of the triangulation with mesh size at most r.
$\text{pers}_k(f)$	The degree q total persistence (2).
σ	A simplex.
τ	The number of steps in the heat equation homotopy.
V_k	The k-vineyard metric (8).

Appendix B. Code

Here, I briefly describe the code that has been used in my research thus far. I indicate the original source, the enhancements I have made, and the potential for future development.

Fréchet Distance. As noted in Section 3.4, we can compute the Fréchet distance between two polygonal curves in $O(n \log n)$ time using parametric search. The Polyline Matching Library[2] provides a framework for implementing applications of the parametric search technique, including an example of computing the Fréchet distance between two random polygonal chains in $\mathbb{R}^2$.

I have made two enhancements to the code. First, updates were made (with the help of David L. Millman) to make it compatible with g++ version 4, since this code was written for version 3 of the compiler. Second, I allow the input to the algorithm to be a polygonal chains in $\mathbb{R}^3$ instead of only in $\mathbb{R}^2$. Next, I plan to allow this code to work with polygonal chains and closed polygons in $\mathbb{R}^n$. In addition, I am currently looking into using BVH files (a file format that captures the motion of a jointed object moving) as input to investigate movement, as described in Section 3.4.

Persistence. I use the persistence library implemented by Dmitriy Morozov.[3] I wrote two parsers so that I can use OBJ files and text files (produced in my Matlab code) as input to compute persistence. I also implemented the parallel spectral sequence algorithm [34, Sec. 5] using boost::threads[4]. Interestingly enough, I did not find that the parallel code ran significantly faster than the regular left-right persistence algorithm when using 16 processors.

Heat Equation. I have developed Matlab code that uses Jacobi iteration to compute the heat equation. Although Jacobi iteration has slow convergence, I found it useful for observing the behavior of the heat equation. In addition, this code computes the Heat Equation Homotopy between two curves. This code was written during the course of my Research Initiation Project in the 2008-09 academic year. Since the Persistence code is written in C++, I plan on porting the Heat Equation code for consistency (as well as to handle larger examples more efficiently). In Matlab, I provide a visualization of the homotopy and the associated vineyard as a time-varying persistence diagram (after the computations have been made using the C++ Persistence code), and am able to compute the Wasserstein distance between diagrams using the Hungarian method [50].

[2] http://www.cs.uu.nl/groups/AA/multimedia/matching/shame.html

[3] http://www.mrzv.org/software/dionysus

[4] http://www.boost.org/