Learning Simplicial Complexes from Persistence Diagrams

Robin Lynne Belton* Brittany Terese Fasy*† Rostik Mertz† Samuel Micka† David L. Millman†
Daniel Salinas† Anna Schenfisch* Jordan Schupbach* Lucia Williams†

Abstract

Topological Data Analysis (TDA) studies the "shape" of data. A common topological descriptor is the persistence diagram, which encodes topological features in a topological space at different scales. Turner, Mukherjee, and Boyer showed that one can reconstruct a simplicial complex embedded in $\mathbb{R}^3$ using persistence diagrams generated from all possible height filtrations (an uncountably infinite number of directions). In this paper, we present an algorithm for reconstructing plane graphs $K = (V, E)$ in $\mathbb{R}^2$, i.e., a planar graph with vertices in general position and a straight-line embedding, from a quadratic number height filtrations and their respective persistence diagrams.

1 Introduction

Topological data analysis (TDA) is a promising tool in fields as varied as materials science, transcriptomics, and neuroscience [8, 11, 14]. Although TDA has been quite successful in the analysis of point cloud data [13], its purview extends to any data that can be encoded as a topological space. Topological spaces can be described in terms of their homology, e.g., connected components and "holes." Simplicial complexes, in particular, are the most common representation of topological spaces. In this work, we focus our attention on a subset of simplicial complexes, namely, plane graphs embedded in $\mathbb{R}^2$, with applications to shape reconstruction.

In this paper, we explore the question, *Can we reconstruct embedded simplicial complexes from a finite number of directional persistence diagrams?* Our work is motivated by [15], which proves that one can reconstruct simplicial complexes from an uncountably infinite number of diagrams. Here, we make the first step towards providing a polynomial-time reconstruction for simplicial complexes. In particular, the main contributions of this paper are to set a bound on the number of persistence diagrams required to reconstruct a plane graph and to provide a polynomial-time algorithm for reconstructing the graph.

*Depart. of Mathematical Sciences, Montana State U.
†School of Computing, Montana State U.
{robin.belton, brittany.fasy, david.millman, annaschenfisch, jordan.schupbach}@montana.edu {samuel.micka, daniel.salinas, lucia.williams}@msu.montana.edu rostik.mertz@student.montana.edu

2 Related Work

The problem of manifold and stratified space learning is an active research area in computational mathematics. For example, Zheng et al. study the 3D reconstruction of plant roots from multiple 2D images [16]. Their method uses persistent homology to ensure the resulting 3D root model is connected.

Map construction algorithms reconstruct street maps as an embedded graph from a set of input trajectories. Three common approaches are Point Clustering, Incremental Track Insertion, and Intersection Linking [1]. Ge, Safa, Belkin, and Wang develop a point clustering algorithm using Reeb graphs to extract the skeleton graph of a road from point-cloud data [6]. The original embedding can be reconstructed using a principal curve algorithm [10]. Karagiorgou and Pfoser give an incremental track insertion algorithm to reconstruct a road network from vehicle trajectory GPS data [9]. Ahmed et al. provide an incremental track insertion algorithm to reconstruct road networks from point could data [2]. The reconstruction is done incrementally, using a variant of the Fréchet distance to add curves to the current basis. Ahmed, Karagiorgou, Pfoser, and Wenk describe all these methods in [1]. Finally, Dey, Wang, and Wang use persistent homology to reconstruct embedded graphs. This research has also been applied to input trajectory data [4]. Dey et al. use persistence to guide the Morse cancellation of critical simplices. In contrast, the work presented here uses persistence to generate the diagrams that encode the underlying graph.

Our work extends previous work on the persistent homology transform (PHT) [15]. As detailed in Section 3, persistent homology summarizes the homological changes for a filtered topological space. When applied to a simplicial complex embedded in $\mathbb{R}^d$, we can compute a different filtration for every direction in $\mathbb{S}^{d-1}$; this family of persistence diagrams is referred to as the persistent homology transform (PHT). The map from a simplicial complex to PHT is injective [15]. Hence, knowing the PHT of a simplicial complex uniquely identifies that complex. The proof presented in [15] relies on the continuity of persistence diagrams as the direction of filtration varies *continuously*.

Our paper bounds the number of directions by presenting an algorithm for reconstructing the simplicial

complex, when we are able to obtain persistence diagrams for a given set of directions. Simultaneous to our investigation, others have also observed that the number of directions can be bounded using the Radon transform; see [3,7]. In the work presented in the current paper, we seek to reconstruct graphs from their respective persistence diagrams, using a geometric approach. We bound the number of directional persistence diagrams since computing the PHT, as presented in [15], requires the computation of filtrations from an infinite number of possible directions. Our work provides a theoretical guarantee of correctness for a finite subset of directions by providing the reconstruction algorithm.

3 Preliminary

In this paper, we explore the question, *Can we reconstruct embedded simplicial complexes from a finite number of directional persistence diagrams?* We begin by summarizing the necessary background information, but refer the reader to [5] for a more comprehensive overview of computational topology.

Simplices and Simplicial Complexes Intuitively, a k-simplex is a k-dimensional generalization of a triangle, i.e., a zero-simplex is a vertex, a one-simplex is an edge connecting two vertices, a two-simplex is a triangle, etc. In this paper, we focus on a subset of simplicial complexes embedded in $\mathbb{R}^2$ consisting of only vertices and edges. Specifically, we study *plane graphs* with straight-line embeddings (referred to simply as *plane graphs* throughout this paper). Furthermore, we assume that the embedded vertices are in general position, meaning that no three vertices are collinear and no two vertices share an x- or y-coordinate.

Height Filtration Let K be a plane graph and denote $\mathbb{S}^1$ as the unit sphere in $\mathbb{R}^2$. Consider $s \in \mathbb{S}^1$; we define the *lower star filtration* with respect to direction s in two steps. First, we let $h_s : K \to \mathbb{R}$ be defined for a simplex $\sigma \subseteq K$ by $h_s(\sigma) = \max_{v\in\sigma} v \cdot s$, where $x \cdot y$ is the inner (dot) product and measures height in the direction of y, if y is a unit vector. Intuitively, the height of σ from s is the maximum height of all vertices in σ. Then, for each $t \in \mathbb{R}$, the subcomplex $K_t := h_s^{-1}([-\infty, t))$ is composed of all simplices that lie entirely below or at the height t, with respect to the direction s. Notice $K_r \subseteq K_t$ for all $r \leq t$ and $K_r = K_t$ if no vertex has height in the interval $[r, t]$. The sequence of all such subcomplexes, indexed by $\mathbb{R}$, is the *height filtration* with respect to s, notated as $F_s(K)$. Often, we simplify notation and define $F_s := F_s(K)$.

Persistence Diagrams The persistence diagram is a summary of the homology groups $H_*(K_t)$ as the height parameter t ranges from $-\infty$ to ∞; in particular, the persistence diagram is a set of birth-death pairs (b_i, d_i). Each pair represents an interval $[b_i, d_i)$ corresponding to a homology generator. For example, a birth event may occur when the height filtration discovers a new vertex, representing a new component, and the corresponding death represents the vertex joining another connected component. By definition [5], all points in the diagonal $y = x$ are also included with infinite multiplicity. However, in this paper, we consider only those points on the diagonal that are explicitly computed in the persistence algorithm found in [5], which correspond to features with the same birth and death time. For a direction $s \in \mathbb{S}^1$, let the *directional persistence diagram* $\mathcal{D}_i(F_s(K))$ be the set of birth-death pairs for the i-th homology group from the height filtration $F_s(K)$. As with the height filtration, we simplify notation and define $\mathcal{D}_i(s) := \mathcal{D}_i(F_s(K))$ when the complex is clear from context. We conclude this section with a remark relating birth-death pairs in persistence diagrams to the simplices in K; a full discussion of this remark is found in [5, pp. 120–121 of §V.4].

Remark 1 (Adding a Simplex) *Let K be a simplicial complex and σ a k-simplex whose faces are all in K. Let β_i refer to the i-th Betti number, i.e., the rank of the i-th homology group. Then, the addition of σ to K will either increase β_k by one or decrease β_{k-1} by one.*

Thus, we can form a bijection between simplices of K and birth-death events in a persistence diagram. This observation is the crux of the proofs of Theorem 5 in Section 4 and Lemma 7 in Section 5.

4 Vertex Reconstruction

In this section, we present an algorithm for recovering the locations of vertices of a simplicial complex K using three directional persistence diagrams. Intuitively, for each direction, we identify the lines on which the vertices of K must lie. We show that by choosing the three directions such that they satisfy a simple property, we can identify all vertex locations by searching for points in the plane where three lines intersect. We call these lines *filtration lines*:

Definition 2 (Filtration Lines) *Given a direction vector $s \in \mathbb{S}^1$, and a height $h \in \mathbb{R}$ the* filtration line at height *h is the line, denoted $\ell(s, h)$, through point $h * s$ and perpendicular to direction s, where $*$ denotes scalar multiplication. Given a finite set of vertices $V \subset \mathbb{R}^2$, the* filtration lines of *V are the set of lines*

$$\mathbb{L}(s, V) = \{\ell(s, h_s(v))\}_{v\in V}.$$

Notice that all lines in $\mathbb{L}(s, V)$ are parallel. Intuitively, if v is a vertex in a simplicial complex K, then the

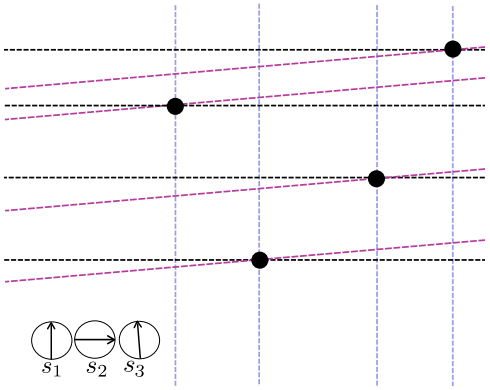


Figure 1: A vertex set, V, of size 4 with filtration lines that satisfy the Vertex Existence Lemma. Here, $s_1, s_2 \in \mathbb{S}^1$ are linearly independent and the filtration lines are colored so that $\mathbb{L}(s_1, V)$ are the black horizontal lines, $\mathbb{L}(s_2, V)$ are the blue vertical lines, and $\mathbb{L}(s_3, V)$ are the magenta diagonal lines. An intersection of three colored lines corresponds to the location of a vertex in V.

line $\ell(s, h_s(v))$ occurs at the height where the filtration $F_s(K)$ includes v for the first time. If the height is known but the complex is not, the line $\ell(s, h_s(v))$ defines all potential locations for v. By Remark 1, the births in the zero-dimensional persistence diagram are in one-to-one correspondence with the vertices of the simplex complex K. Thus, we can construct $\mathbb{L}(s, V)$ from a single directional diagram in $O(n)$ time. Given filtration lines for three carefully chosen directions, we next show a correspondence between intersections of three filtration lines and vertices in K.

In what follows, given a direction $s_i \in \mathbb{S}^1$ and a point $p \in \mathbb{R}^2$, define $\ell_i(p) := \ell(s_i, h_{s_i}(p))$ as a way to simplify notation.

Lemma 3 (Vertex Existence Lemma) *Let K be a simplicial complex with vertex set V of size n. Let $s_1, s_2 \in \mathbb{S}^1$ be linearly independent and further suppose that $\mathbb{L}(s_1, V)$ and $\mathbb{L}(s_2, V)$ each contain n lines. Let A be the collection of vertices at the intersections of lines in $\mathbb{L}(s_1, V) \cup \mathbb{L}(s_2, V)$. Let $s_3 \in \mathbb{S}^1$ such that for all $u, v \in A$, $\ell_3(u) = \ell_3(v) \iff u = v$. Then, the following two statements hold true:*

(1) *$v \in V \iff \ell_3(v) \in \mathbb{L}(s_3, V)$ and $A \cap \ell_3(v) = \{v\}$*
(2) *For all $\ell \in \mathbb{L}(s_3, V)$, $A \cap \ell \neq \emptyset$.*

Proof. First, we prove Part (1).

($\Rightarrow$) Let $v \in V$. Then, $\ell_i(v) \in \mathbb{L}(s_i, V)$ and $v \in \ell_i(v)$ for $i = \{1, 2, 3\}$. Hence, $\ell_1(v) \cap \ell_2(v) \cap \ell_3(v) = \{v\}$, as desired.

($\Leftarrow$) Assume, for the sake of contradiction, that $\ell_3(v) \in \mathbb{L}(s_3, V)$ and $A \cap \ell_3(v) = \{v\}$, yet $v \notin V$. Since $\ell_3(v) \in \mathbb{L}(s_3, V)$ and $v \notin V$, some other vertex $u \in V$ must have height $h_{s_3}(v)$. Since $u \in V$, we know $\ell_i(u) \in \mathbb{L}(s_i, V)$ for $i \in \{1, 2, 3\}$. And, by ($\Rightarrow$) applied to u, we know $u \in A$. Since $\ell_3(u) = \ell_3(v)$, both u and v are in A and on the line $\ell_3(v)$, but $u \neq v$, which is a contradiction.

Next, we prove Part (2) of the lemma. Assume, for contradiction, that there exists $\ell \in \mathbb{L}(s_3, V)$ such that $A \cap \ell = \emptyset$. As $\ell \in \mathbb{L}(s_3, V)$, a vertex $v \in V$ exists such that $\ell = \ell_3(v)$ and v lies on ℓ. However, $v \in \ell_1(v) \cap \ell_2(v) \subset A$, which is a contradiction. □

In the previous lemma, we needed to find a third direction with specific properties. If we use horizontal and vertical lines for our first two directions, then we can use the geometry of the boxes formed from these lines to pick the third direction. More specifically, we look at the box with the largest width and smallest height and pick the third direction so that if one of the corresponding lines intersects the bottom left corner of the box then it will also intersect the box somewhere along the right edge. In Figure 1, the third direction was computed using this procedure with the second box from the left in the top row. Next, we give a more precise description of the vertex localization procedure.

Lemma 4 (Vertex Localization) *Let L_H and L_V be n horizontal and n vertical lines, respectively. Let w (and h) be the largest (and smallest) distance between two lines of L_V (and L_H, respectively). Let B be the smallest axis-aligned bounding box containing the intersections of lines in $L_H \cup L_V$. For $0 < \varepsilon < h$, let $s = (w, h - \varepsilon)$. Any line parallel to s can intersect at most one line of L_H in B.*

Proof. Note that, by definition, s is a vector in the direction that is at a slightly smaller angle than the diagonal of the box of size w by h. Assume, by contradiction, that a line parallel to s may intersect two lines of L_H within B. Specifically, let $\ell_1, \ell_2 \in L_H$ and let ℓ_s be a line parallel to s such that the points $\ell_i \cap \ell_s = (x_i, y_i)$ for $i = \{1, 2\}$ are the two such intersection points within B. Notice since the lines of L_H are horizontal and by the definition of h, we observe that $|y_1 - y_2| \geq h$. Let $w' = |x_1 - x_2|$, and observe $w' \leq w$. Since the slope of ℓ_s is $(h - \varepsilon)/w$, we have $|y_1 - y_2| < h$, which is a contradiction. □

We conclude this section with an algorithm to determine the coordinates of the vertices of the original graph in $\mathbb{R}^2$, using only three height filtrations.

Theorem 5 (Vertex Reconstruction) *Let K be a plane graph. We can can compute the coordinates of all n vertices of K in $O(n \log n)$ time from three directional persistence diagrams.*

Proof. Let $s_1 = (1, 0)$ and $s_2 = (0, 1)$, which are linearly independent. We compute the filtration

lines $\mathbb{L}(s_i, V)$ for $i = 1, 2$ in $O(n)$ time by Remark 1. By our general position assumption, no two vertices of K share an x- or y-coordinate. Thus, the sets $\mathbb{L}(s_1, V)$ and $\mathbb{L}(s_2, V)$ each contain n distinct lines. Let A be the set of intersection points of the lines in $\mathbb{L}(s_1, V)$ and $\mathbb{L}(s_2, V)$. The next step is to identify a direction s_3 such that each line in $\mathbb{L}(s_3, V)$ intersects with only one point A, so that we can use Lemma 3.

Let w (and h) be the greatest (and least) distance between two adjacent lines in $\mathbb{L}(s_1, V)$ (and $\mathbb{L}(s_2, V)$, respectively). Let B be the smallest axis-aligned bounding box containing A, and let $s_* = (w, \frac{h}{2})$. By Lemma 4, any line parallel to s_* will intersect at most one line of $\mathbb{L}(s_2, V)$ in B. Thus, we choose $s_3 \in \mathbb{S}^1$ that is perpendicular to s_*. By the second part of Lemma 3, we now have that each line in $\mathbb{L}(s_3, V)$ intersects A. Thus, there are n intersections between $\mathbb{L}(s_2, V)$ and $\mathbb{L}(s_3, V)$ in B, each of which also intersects with $\mathbb{L}(s_1, V)$.

The previous paragraph leads us to a simple algorithm for finding the third direction and identifying all the triple intersections. In the analysis below, steps that do not mention a number of diagrams use no diagrams. First, we construct $\mathbb{L}(s_1, V)$ and $\mathbb{L}(s_2, V)$ in $O(n)$ time using two directional persistence diagrams. Second, we sort the lines of $\mathbb{L}(s_1, V)$ and $\mathbb{L}(s_2, V)$ by their x- and y-intercepts respectively in $O(n \log n)$ time. Third, we find s_3 by computing w and h from our sorted lines in $O(n)$ time. Fourth, we construct $\mathbb{L}(s_3, V)$ in $O(n)$ time using one directional persistence diagram. Fifth, we sort the lines in $\mathbb{L}(s_3, V)$ by their intersection with the leftmost line of $\mathbb{L}(s_1, V)$ in $O(n \log n)$ time. Finally, we compute coordinates of the n vertices by intersecting the i-th line of $\mathbb{L}(s_2, V)$ with the i-th line of $\mathbb{L}(s_3, V)$ in $O(n)$ time. (Observe, this last step works since the vertices correspond to the n intersections in B, as described above).

Therefore, we use three directional diagrams (two in the first step and one in the fourth step) and $O(n \log n)$ time (sorting of lines in the second and fifth steps) to reconstruct the vertices. □

5 Edge Reconstruction

Given the vertices constructed in Section 4, we describe how to reconstruct the edges in a plane graph using $n(n-1)$ persistence diagrams. The key to determining whether an edge exists or not is counting the degree of a vertex, for edges "below" the vertex with respect to a given direction. We begin this section by defining necessary terms, and then explicitly describing our algorithm for constructing edges.

Definition 6 (Indegree of Vertex) *Let K be a plane graph with vertex set V. Then, for every vertex $v \in V$ and every direction $s \in \mathbb{S}^1$, we define:*

$$\textsc{Indeg}(v, s) = |\{(v, v') \in E \mid s \cdot v' \leq s \cdot v\}|.$$

In other words, the indegree of v is the number of edges incident to v that lie below v, with respect to direction s; see Figure 2.

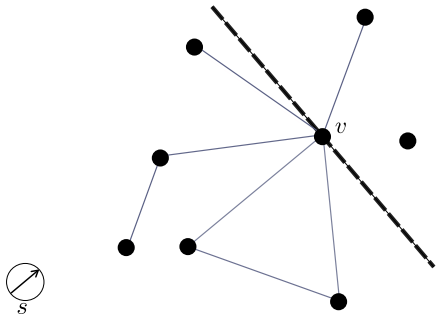


Figure 2: A plane graph with a dashed line drawn intersecting v in the direction perpendicular to s. Since four edges incident to v lie below v, with respect to direction s, $\textsc{Indeg}(v, s) = 4$.

Next, we prove that given a direction, we can determine the indegree of a vertex:

Lemma 7 (Indegree from Diagram) *Let K be a plane graph with vertex set V. Let $s \in \mathbb{S}^1$ be such that no two vertices are at the same height with respect to s, i.e., $|\mathbb{L}(s, V)| = n$. Let $\mathcal{D}_0(s)$ and $\mathcal{D}_1(s)$ be the zero- and one-dimensional persistence diagrams resulting from the height filtration F_s on K. Then, for all $v \in V$,*

$$\begin{aligned}\textsc{Indeg}(v, s) =&|\{(x, y) \in \mathcal{D}_0(s) \mid y = v \cdot s\}|+\\ &|\{(x, y) \in \mathcal{D}_1(s) \mid x = v \cdot s\}|.\end{aligned}$$

Proof. Let $v, v' \in V$ such that $s \cdot v' < s \cdot v$, i.e., the vertex v' is lower in direction s than v. Then, by Remark 1, if $(v, v') \in E$, it must be one of the following in the filter of K defined by s: (1) an edge that joins two disconnected components; or (2) an edge that creates a one-cycle. Since edges are added to a filtration at the height of the higher vertex, we see (1) as a death in $\mathcal{D}_0(s)$ and (2) as a birth in $\mathcal{D}_1(s)$, both at height $s \cdot v$. In addition, each finite death in $\mathcal{D}_0(s)$ and every birth in $\mathcal{D}_1(s)$ at time $s \cdot v$ must correspond to an edge, i.e., edges are the only simplices that can cause these events. Then, the set of edges of types (1) and (2) is $\{(x, y) \in \mathcal{D}_0(s) \mid y = v \cdot s\}$ and $\{(x, y) \in \mathcal{D}_1(s) \mid x = v \cdot s\}$, respectively. The size of the union of these two multi-sets is equal to the number of edges starting at v' lower than v in direction s and ending at v, as required. □

In order to decide whether an edge (v, v') exists between two vertices, we look at the degree of v as seen by two close directions such that v' is the only vertex in what we call a *bow tie at v*:

Definition 8 (Bow Tie) *Let $v \in V$, and choose $s_1, s_2 \in \mathbb{S}^1$. Then, a bow tie at v is the symmetric difference between the half planes below v in directions s_1*

and s_2. The width *of the bow tie is half of the angle between s_1 and s_2.*

Because no three vertices in our plane graph are collinear, for each pair of vertices $v, v' \in V$, we can always find a bow tie centered at v that contains the vertex v' and no other vertex in V; see Figure 3. We

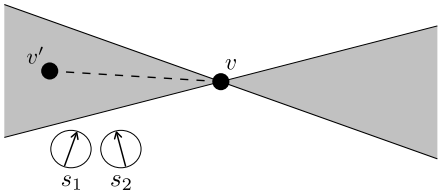


Figure 3: A bow tie B at v, denoted by the gray shaded area. B contains exactly one vertex, v', so the only potential edge in B is (v, v').

use bow tie regions that only contain one vertex, v' other than the center, v to determine if there exists an edge between v and v'; see Figure 4. We then use Lemma 9 to decide if the edge (v, v') exists in our plane graph.

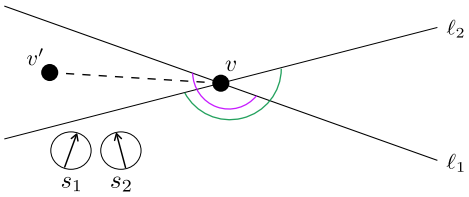


Figure 4: A bow tie at v that contains the vertex v' and no other vertices. In order to determine if there exists an edge between v and v', we compute $\textsc{Indeg}(v, s_1)$ and $\textsc{Indeg}(v, s_2)$, i.e., the number of edges incident to v in the purple and green arcs, respectively. An edge exists between v and v' if and only if $|\textsc{Indeg}(v, s_1) - \textsc{Indeg}(v, s_2)| = 1$.

Lemma 9 (Edge Existence) *Let K be a plane graph with vertex set V and edge set E. Let $v, v' \in V$. Let $s_1, s_2 \in \mathbb{S}^1$ such that the bow tie B at v defined by s_1 and s_2 satisfies: $B \cap V = v'$. Then,*

$$|\textsc{Indeg}(v, s_1) - \textsc{Indeg}(v, s_2)| = 1 \iff (v, v') \in E.$$

Proof. Since edges in K are straight lines, any edge incident to v will either fall in the bow tie region B or will be on the same side (above or below) of both lines. Let A be the set of edges incident on v and below both lines; that is $A = \{(v, v_*) \in E \mid s_i \cdot v_* < s_i \cdot v\}$. Furthermore, suppose we split the bowtie into the two infinite cones. Let B_1 be the set of edges in one cone and B_2 be the set of edges in the other cone. We note that $||B_1| - |B_2||$ is equal to one if there is an edge $(v, v') \in E$ with $v' \in B_1$ or $v' \in B_2$ and zero otherwise. Then, by definition of indegree,

$$\begin{aligned} |\textsc{Indeg}(v, s_1) - \textsc{Indeg}(v, s_2)| &= ||A| + |B_1| - |A| - |B_2|| \\ &= ||B_1| - |B_2|| \\ &= |V \cap B|, \end{aligned}$$

which holds if and only if $(v, v') \in E$. Then $|\textsc{Indeg}(v, s_1) - \textsc{Indeg}(v, s_2)| = 1 \iff (v, v') \in E$, as required. □

Next, we prove that we can find the embedding of the edges in the original graph using $O(n^2)$ directional persistence diagrams.

Theorem 10 (Edge Reconstruction) *Let K be a plane graph, with vertex set V and edge set E. If V is known, then we can compute E using $O(n^2)$ directional persistence diagrams.*

Proof. We prove this theorem constructively. Intuitively, we construct a bow tie for each potential edge and use Lemma 9 to determine if the edge exists or not. Our algorithm has three steps for each pair of vertices in V: Step 1 is to determine a global bow tie width, Step 2 is to construct suitable bow ties, and Step 3 is to compute indegrees. See Appendix A for an example of walking through the reconstruction.

Step 1: Determine bow tie width. For each vertex $v \in V$, we consider the cyclic ordering of the points in $V \setminus \{v\}$ around v. We define $\theta(v)$ to be the minimum angle between all adjacent pairs of lines through v; see Figure 5, where the angles between adjacent lines are denoted θ_i. Finally, we choose θ less than $\min_{v \in V} \theta(v)$. By Lemmas 1 and 2 of [12], we compute the cyclic orderings for all vertices in V in $O(n^2)$ time. Since computing each $\theta(v)$ is $O(n)$ time once we have the cyclic ordering, the runtime for this step is $O(n^2)$.

Step 2: Constuct bow ties. For each pair of vertices $(v, v') \in V \times V$ such that $v \neq v'$, let s be a unit vector perpendicular to vector $(v' - v)$, and let s_1, s_2 be the two unit vectors that form angles $\pm\theta$ with s. Let B be the bow tie between $\ell(s_1, h_{s_1}(v))$ and $\ell(s_2, h_{s_2}(v))$. Note that by the construction, B contains exactly one point from V, namely v'.

Step 3: Compute indegrees. Using B as the bow tie in Lemma 7, compute $\textsc{Indeg}(v, s_1)$ and $\textsc{Indeg}(v, s_2)$. Then, using Lemma 9, we determine whether (v, v') exists by checking if $|\textsc{Indeg}(v, s_1) - \textsc{Indeg}(v, s_2)| = 1$. If it does, the edge exists; if not, the edge does not.

Repeating for all vertex pairs requires $O(n^2)$ diagrams and discovers the edges of K. □

The implications of Theorem 5 and Theorem 10 lead to our primary result. We can find the embedding of the

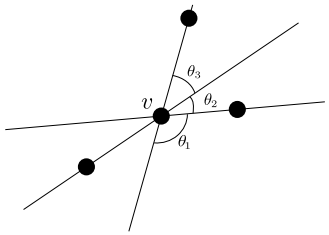


Figure 5: Using a vertex set of a plane graph to construct a bow tie at vertex, v. Lines are drawn through all vertices and then angles are computed between all adjacent pairs of lines. The smallest angle is chosen as $\theta(v)$. Here, $\theta(v) = \theta_2$.

vertices V by Theorem 5 using three directional persistence diagrams. Furthermore, we can discover edges E with $O(n^2)$ directional persistence diagrams by Theorem 10. Thus, we can reconstruct all edges and vertices of a one-dimensional simplicial complex:

Theorem 11 (Plane Graph Reconstruction) *Let K be a plane graph with vertex set V and edge set E. The vertices, edges, and exact embedding of K can be determined using persistence diagrams from $O(n^2)$ different directions.*

6 Discussion

In this paper, we provide an algorithm to reconstruct a plane graph with n vertices embedded in $\mathbb{R}^2$. Our method uses $O(n^2)$ persistence diagrams by first determining vertex locations using only three directions, and, second, determining edge existence based on height filtrations and vertex degrees. Moreover, if we have an oracle that can return a diagram given a direction in $O(T)$ time, then constructing the vertices takes $O(T+n\log n)$ and reconstructing the edges takes takes $O(Tn^2)$ time.

This approach extends to several avenues for future work. First, we plan to generalize these reconstruction results to higher dimensional simplicial complexes. We can show that the vertices of a simplicial complex K in $\mathbb{R}^d$ can be reconstructed in $O(dT + n^d)$ time using the complete arrangement of hyperplanes and $(d + 1)$ directional persistence diagrams. We conjecture that this bound can be improved to $O(dT + dn\log n)$ using the same observation that allows us to do the final step of the vertex reconstruction in linear time. We have a partial proof in this direction, and can likewise extend the bow tie idea to higher dimensions, but the number of directions grows quite quickly. Second, we conjecture that we can reconstruct these plane graphs with a subquadratic number of height filtrations by utilizing more information from each height filtration. Third, we suspect a similar approach can be used to infer other graph metrics, such as classifying vertices into connected components. Intuitively, determining such metrics should require fewer persistence diagrams than required for a complete reconstruction. Finally, we plan to provide an implementation for reconstruction that integrates with existing TDA software.

Acknowledgements This material is based upon work supported by the National Science Foundation under the following grants: CCF 1618605 (BTF, SM), DBI 1661530 (BTF, DLM, LW), DGE 1649608 (RLB), and DMS 1664858 (RLB, BTF, AS, JS). Additionally, RM thanks the Undergraduate Scholars Program. All authors thank the CompTaG club at Montana State University and the reviewers for their thoughtful feedback on this work.

References

[1] Mahmuda Ahmed, Sophia Karagiorgou, Dieter Pfoser, and Carola Wenk. Map construction algorithms. In *Map Construction Algorithms*, pages 1–14. Springer, 2015.

[2] Mahmuda Ahmed and Carola Wenk. Constructing street networks from GPS trajectories. In *European Symposium on Algorithms*, pages 60–71. Springer, 2012.

[3] Justin Curry, Sayan Mukherjee, and Katharine Turner. How many directions determine a shape and other sufficiency results for two topological transforms. arXiv:1805.09782, 2018.

[4] Tamal K. Dey, Jiayuan Wang, and Yusu Wang. Graph reconstruction by discrete Morse theory. arXiv:1803.05093, 2018.

[5] Herbert Edelsbrunner and John Harer. *Computational Topology: An Introduction*. American Mathematical Society, 2010.

[6] Xiaoyin Ge, Issam I Safa, Mikhail Belkin, and Yusu Wang. Data skeletonization via Reeb graphs. In *Advances in Neural Information Processing Systems*, pages 837–845, 2011.

[7] Robert Ghrist, Rachel Levanger, and Huy Mai. Persistent homology and Euler integral transforms. arXiv:1804.04740, 2018.

[8] Chad Giusti, Eva Pastalkova, Carina Curto, and Vladimir Itskov. Clique topology reveals intrinsic geometric structure in neural correlations. *Proceedings of the National Academy of Sciences*, 112(44):13455–13460, 2015.

[9] Sophia Karagiorgou and Dieter Pfoser. On vehicle tracking data-based road network generation.

In *SIGSPATIAL '12: Proceedings of the 20th International Conference on Advances in Geographic Information Systems*, pages 89–98. ACM, 2012.

[10] Balázs Kégl, Adam Krzyzak, Tamás Linder, and Kenneth Zeger. Learning and design of principal curves. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(3):281–297, 2000.

[11] Yongjin Lee, Senja D. Barthel, Paweł Dłotko, S. Mohamad Moosavi, Kathryn Hess, and Berend Smit. Quantifying similarity of pore-geometry in nanoporous materials. *Nature Communications*, 8:15396, 2017.

[12] David L. Millman and Vishal Verma. A slow algorithm for computing the Gabriel graph with double precision. *CCCG '11: Proceedings of the 23rd Annual Canadian Conference on Computational Geometry*, 2011.

[13] Monica Nicolau, Arnold J. Levine, and Gunnar Carlsson. Topology based data analysis identifies a subgroup of breast cancers with a unique mutational profile and excellent survival. *Proceedings of the National Academy of Sciences*, 108(17):7265–7270, 2011.

[14] Abbas H. Rizvi, Pablo G. Camara, Elena K. Kandror, Thomas J. Roberts, Ira Schieren, Tom Maniatis, and Raul Rabadan. Single-cell topological RNA-seq analysis reveals insights into cellular differentiation and development. *Nature Biotechnology*, 35(6):551, 2017.

[15] Katharine Turner, Sayan Mukherjee, and Doug M. Boyer. Persistent homology transform for modeling shapes and surfaces. *Information and Inference: A Journal of the IMA*, 3(4):310–344, 2014.

[16] Ying Zheng, Steve Gu, Herbert Edelsbrunner, Carlo Tomasi, and Philip Benfey. Detailed reconstruction of 3d plant root shape. *Proceedings of the IEEE International Conference on Computer Vision*, pages 2026–2033, 11 2011.

Appendix

A Example of Reconstructing a Plane Graph

We give an example of reconstructing a plane graph. Consider the complex given in Figure 6.

Vertex Reconstruction First, we find vertex locations using the algorithm described in Section 4. We need to choose pairwise linearly independent vectors s_1, s_2 and s_3 such that only n three-way intersections in $A = \mathbb{L}(s_1, V) \cup \mathbb{L}(s_2, V) \cup \mathbb{L}(s_3, V)$ exist; note that in this example, $n = 4$. Using the persistence diagrams from height filtrations in directions $s_1 = (0, 1)$ and $s_2 = (1, 0)$, we construct the set of lines $\mathbb{L}(s_1, V) \cup \mathbb{L}(s_2, V)$. This results in $n^2 = 16$ possible locations for the vertices at the intersections in A. We show these filtration lines and intersections in Figure 6b. Next, we compute the third direction s_3 using the algorithm outlined in Theorem 5. To do this, we need to find the greatest horizontal distance between two vertical lines, $d_1 = 2$ and the least vertical distance between two horizontal lines, $d_2 = 1$. Then, we use these to choose a direction s_3 perpendicular to $s_* = (d_1, \frac{d_2}{2}) = (2, \frac{1}{2})$ (e.g., $s_3 = (\frac{-1}{\sqrt{17}}, \frac{4}{\sqrt{17}}) \in \mathbb{S}^1$). Then, the four three-way intersections in $\mathbb{L}(s_1, V) \cup \mathbb{L}(s_2, V) \cup \mathbb{L}(s_3, V)$ identify all Cartesian coordinates of the original complex. We show filtration lines from all three directions in Figure 6c.

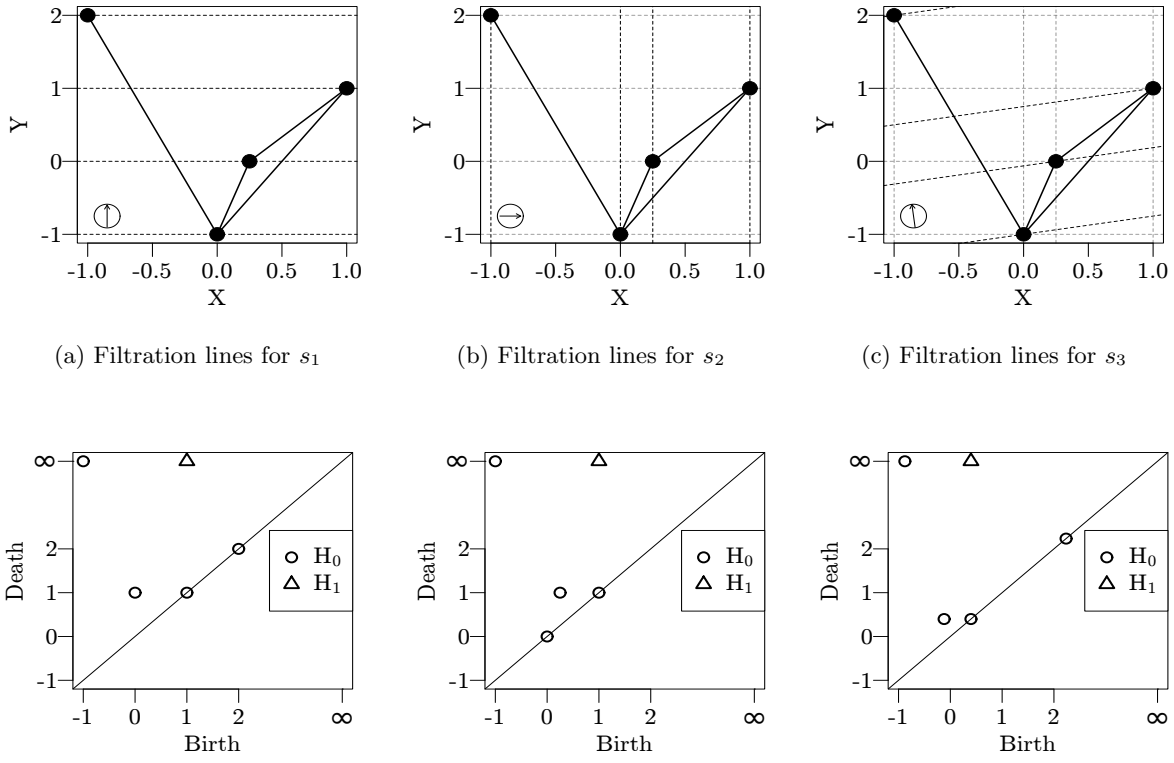
Edge Reconstruction Next, we reconstruct all edges as described in Section 5. In order to do so, we first find the θ we will use to construct bow ties. To do this, we examine each vertex v in turn, finding $\theta(v)$, the minimum angle between adjacent pairs of lines through v and $v' \in V - \{v\}$. Ordering v by increasing x-coordinate, we find $\theta(v)$ to be approximately $0.237, 0.219, 0.399$, and 0.180 radians, respectively. Then, we take θ to be less than the minimum of these, i.e. $< 0.180 radians$.

Now, for each of the $\frac{n(n-1)}{2}$ pairs of vertices $(v, v') \in V^2$, we construct a bow tie B and then use this bow tie to determine whether an edge exists between the two vertices. We go through two examples: one for a pair of vertices that does have an edge between, and one for a pair that does not. First, consider the pair $v = (0.25, 0)$ and $v' = (1, 1)$. To construct their bow tie, we first find the unit vector perpendicular to the vector that points from v to v', which is $s = (-0.8, 0.6)$. Now, we find s_1, s_2 such that they make angles θ with s. We choose $s_1 = (-0.956, 0.293)$ and $s_2 = (-0.433, 0.902)$. Now, by Lemma 7, we can use the persistence diagrams from these two directions to compute $\textsc{Indeg}(v, s_1)$ and $\textsc{Indeg}(v, s_2)$. We observe that $\mathcal{D}_0(s_1)$ contains exactly one birth-death pair (x, y) such that $y = v \cdot s_1$ and $\mathcal{D}_1(s_1)$ has one birth-death pair such that $x = v \cdot s_1$. Thus, $\textsc{Indeg}(v, s_1) = 2$. On the other hand, $\mathcal{D}_0(s_2)$ contains exactly one birth-death pair (x, y) such that $y = v \cdot s_2$, but $\mathcal{D}_1(s_2)$ contains no birth-death pair such that $x = v \cdot s_2$. So $\textsc{Indeg}(v, s_2) = 1$. Now, since $|\textsc{Indeg}(v, s_1) - \textsc{Indeg}(v, s_2)| = 1$, we know that $(v, v') \in E$, by Lemma 9.

For the second example, consider the pair of vertices $v = (0.25, 0)$ and $v' = (-1, 2)$. Again, we construct their bow tie by finding a unit vector perpendicular to the vector pointing from v to v'. We choose this $s = (0.848, 0.530)$. Then, the s_1 and s_2 which form angle $\theta < 0.180$ radians (e.g $\theta = .170$)

with s are $s_1 = (0.968, 0.248)$ and $s_2 = (0.472, 0.882)$. Again by Lemma 7, we examine the zero- and one-dimensional persistence diagrams from these two directions to compute the indegree from each direction for vertex v. In $\mathcal{D}_0(s_1)$, we have one pair (x, y) which dies at $y = v \cdot s_1$, but in $\mathcal{D}_1(s_1)$, no pair is born at $x = v \cdot s_1$. So $\textsc{Indeg}(v, s_1) = 1$. We see the exact same for s_2, which means that $|\textsc{Indeg}(v, s_1) - \textsc{Indeg}(v, s_2)| = 0$. Since Lemma 9 tells us that we have an edge between v and v' only if the absolute value of the difference of indegrees is one, we know that there is no edge between vertices $(0.25, 0)$ and $(-1, 2)$.

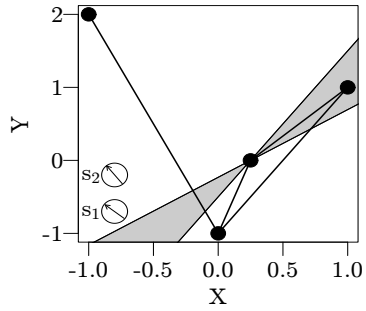
In order to reconstruct all edges, we perform the same computations for all pairs of vertices.



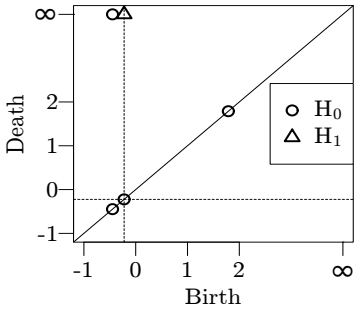
(a) Filtration lines for s_1 (b) Filtration lines for s_2 (c) Filtration lines for s_3

(d) Diagrams for s_1 (e) Diagrams for s_2 (f) Diagrams for s_3

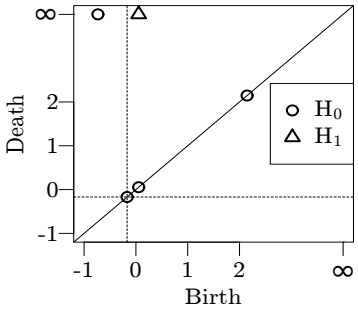
Figure 6: Example of vertex reconstruction from three directions, s_1, s_2 and s_3 with corresponding persistence diagrams built for height filtrations from these directions. The filtration lines are the dotted lines superimposed over the complex.



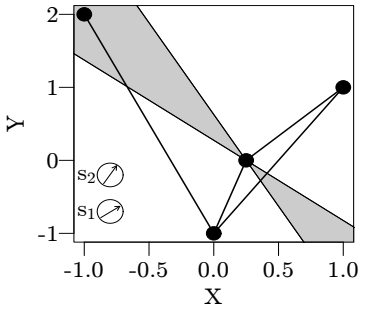
(a) Bow tie lines for s_1 and s_2



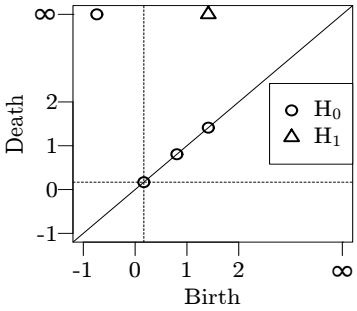
(b) Diagram for s_1



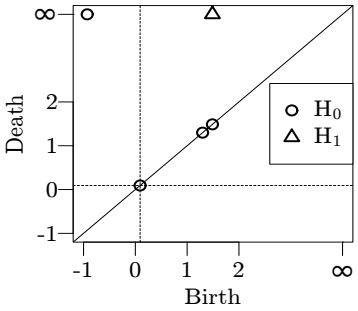
(c) Diagram for s_2



(d) Bow tie lines for s_1 and s_2



(e) Diagram for s_1



(f) Diagram for s_2

Figure 7: Example of edge reconstruction for two edges. The first edge (top row) exists while the second edge (bottom row) does not. The bow tie is given on the left while the persistence diagrams $\mathcal{D}_0(s_1)$ and $\mathcal{D}_1(s_1)$ are given in the middle and the persistence diagrams $\mathcal{D}_0(s_2)$ and $\mathcal{D}_1(s_2)$ are given on the right. The dotted lines indicate $v \cdot s_1$ and $v \cdot s_2$ in diagrams for s_1 and s_2 respectively.