Eurographics Workshop on 3D Object Retrieval (2016)
A. Ferreira, A. Giachetti, and D. Giorgi (Editors)

An edit distance for Reeb graphs

U. Bauer[1], B. Di Fabio[2] and C. Landi[2]

[1]Geometry & Visualization Group, TU München, Germany
[2]DISMI, Università di Modena e Reggio Emilia, Italy

Abstract

We consider the problem of assessing the similarity of 3D shapes using Reeb graphs from the standpoint of robustness under perturbations. For this purpose, 3D objects are viewed as spaces endowed with real-valued functions, while the similarity between the resulting Reeb graphs is addressed through a graph edit distance. The cases of smooth functions on manifolds and piecewise linear functions on polyhedra stand out as the most interesting ones. The main contribution of this paper is the introduction of a general edit distance suitable for comparing Reeb graphs in these settings. This edit distance promises to be useful for applications in 3D object retrieval because of its stability properties in the presence of noise.

Categories and Subject Descriptors (according to ACM CCS): I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—Curve, surface, solid, and object representations

1. Introduction

The significant increase of available 3D models, enabled by modern technology, strongly motivates 3D retrieval using content-based methods. 3D shape retrieval is generally the result of a pipeline of basic steps [TV08]. In a first step, shape features are computed from the 3D models, and encoded in shape signatures. Different types of shape signatures have been proposed in the literature for this task, the most common categories being graph-based, transform-based, statistics-based and view-based methods [BKS*05]. In a second step, the similarity between 3D models is assessed by evaluating the distance between the associated shape signatures: the smaller the distance, the more similar the shapes [CGK03]. In a third step, given a query model, the target models are sorted in order of increasing distance between their signature and that of the query model.

In this paper we focus on the second step of the shape retrieval pipeline, assuming that Reeb graphs have been chosen as shape signatures in the first step. The goal of this paper is to investigate theoretical aspects of the definition of the similarity concept for Reeb graphs.

The Reeb graph is defined for shapes modeled as spaces endowed with scalar functions. It is obtained by shrinking each connected component of a level set of the function to a single point [Ree46]. Often, vertices of the Reeb graph are labeled by the value of the function at the corresponding level set. If the function is constructed from geometric information, such as a height function or a distance function, the Reeb graph captures both topological and geometric features of a 3D model, thus combining global and local information on its shape. Reeb graphs have been used as an effective tool for shape analysis and description tasks since [SKK91, SK91]. Indeed, the Reeb graph has a number of characteristics that make it useful as a search query for 3D objects [BGSF08]. First, a Reeb graph always consists of a one-dimensional graph structure and does not have any higher dimension components such as the degenerate surface that can occur in a medial axis. Second, by defining the function appropriately, it is possible to construct a Reeb graph that is invariant to translation and rotation, or even more complicated isometries of the shape. Last but not least, as aforementioned, Reeb graphs allow for capturing global and local features.

One of the most important questions is the stability of the Reeb graph construction: whether Reeb graphs (the result of the construction) are robust against perturbations that may occur because of noise and approximation errors on the input, namely the spaces and the scalar functions.

Over the years, starting back with [HSKK01] until more recently with [BB13], a number of heuristics have been developed so that the Reeb graph turns out to be resistant to connectivity changes caused by simplification, subdivision and remesh, and robust against noise and certain changes due to deformation.

The problem of studying the stability of Reeb graphs from the theoretical standpoint has recently attracted significant interest in the area of Topological Data Analysis (TDA) and more broadly speaking in Computational Topology. Indeed, the success of TDA in applications is strongly connected with the stability properties of its tools such as persistence diagrams [CSEH07]. Therefore, it is natural to address the problem of stable comparison of Reeb graphs using techniques rooted in TDA, and in particular in Persistence Theory.

DOI: 10.2312/3dor.20161084

The stability problem is addressed by providing distances such that variations in Reeb graphs are bounded by variations in the input scalar functions. In other words, the map sending an input function to the associated Reeb graph is Lipschitz continuous. The first paper in this direction was [DFL12], where an edit distance between Reeb graphs of smooth curves endowed with Morse functions has been introduced and shown to yield stability with respect to function perturbations. More recently, in [DFL16], also Reeb graphs of smooth surfaces have been shown to satisfy stability in the same sense with respect to an appropriate edit distance. A drawback of this approach is that the set of admissible edit operations changes as we pass from curves to surfaces. Another result in the context of Reeb graph stability is the functional distortion distance between Reeb graphs proposed in [BGW14], with proven stable and discriminative properties. The functional distortion distance applies to a wider class of objects than the edit distances of [DFL12, DFL16] and is intrinsically continuous, whereas the edit distances are combinatorial. The authors of [dSMP15] address the question of a distance function stable under perturbations of the input data using methods from category theory, and propose to compare Reeb graphs through the interleaving distance. In [BMW15] it has been proved that the functional distortion distance and the interleaving distance on Reeb graphs are strongly equivalent on the space of Reeb graphs, in the mathematical sense. The paper [BYM*13] about a stable distance for merge trees is also pertinent to the stability problem for Reeb graphs: merge trees are known to determine contour trees, which are Reeb graphs for simple domains.

The first contribution of this paper is the definition of a set of edit operations that is general enough for defining an edit distance between Reeb graphs that applies to many different settings, from that of Morse functions on smooth curves and surfaces to that of piecewise linear functions on polyhedra. Indeed, the piecewise linear case is certainly the most relevant one in applications to 3D model retrieval. More precisely, we introduce a combinatorial dissimilarity measure, called an edit distance, between labeled graphs, applicable in particular to Reeb graphs. The basic idea is that labeled graphs of two shapes can be transformed into each other by a finite sequence of edit operations. Each such sequence has a cost that depends on how much we must vary the value of the label at the vertices of the graph during the transformation. Thus our edit distance between graphs belongs to the family of Graph Edit Distances [GXTL10], widely used in pattern analysis.

By the aforementioned generality of the edit operations introduced here, the edits defined in [DFL12] for curves, and in [DFL16] for surfaces, can be now uniformed. This allows for the second contribution of this paper, namely that the edit distance we propose, when applied to Reeb graphs of Morse functions of smooth curves or surfaces, yields the stability property with respect to function perturbations.

The paper is organized as follows. Section 2 focuses on mathematical aspects of Reeb graphs. Section 3 introduces our method, i.e. comparison of labeled graphs using the edit distance. In Section 4 we discuss the stability properties of this edit distance in the smooth case. A final discussion on the obtained results and the future related research concludes the paper.

2. Mathematical background on Reeb graphs

The more general definition of a Reeb graph is the topological definition. It applies to any topological space X endowed with any continuous function f.

Definition 2.1. The *topological Reeb graph* of f is the quotient space $X/\sim_f$ where, for every $x, x' \in X$, $x \sim_f x'$ if and only if x and x' belong to the same connected component of $f^{-1}(f(x))$.

Intuitively, this corresponds to shrinking each connected component of a level set of the function to a single point.

Appropriate assumptions on X and f ensure that the topological Reeb graph is well behaved. For example, it is sufficient that X is Hausdorff and compact to guarantee that $X/\sim_f$ is also Hausdorff and compact [Gov]. However, such general properties do not guarantee that $X/\sim_f$ has the structure of a finite connected cell complex of dimension 1, associating with $X/\sim_f$ the combinatorial structure of a graph. In order to obtain a combinatorial Reeb graph, more restrictive assumptions on the function are needed. Common choices are that f is Morse or piecewise linear. In view of this shift from the topological to the combinatorial definition of a Reeb graph, it is useful to introduce some notations.

In this paper, we define a *labeled graph* as a pair (Γ, ℓ) with Γ a finite graph, and $\ell : V(\Gamma) \to \mathbb{R}$ a function that endows each vertex of Γ with a scalar value. The graphs considered here are allowed to have multiple edges and loops. Moreover, for simplicity, we always suppose that they are connected. We denote by $V(\Gamma)$ and $E(\Gamma)$ the vertex and edge sets of Γ, respectively. If $e \in E(\Gamma)$ is an edge incident to the vertices $v_1, v_2 \in V(\Gamma)$, we say that v_1 and v_2 are adjacent and we write $e = e(v_1, v_2)$. As usual, we define the *degree of a vertex* $v \in V(\Gamma)$, denoted by $\deg(v)$, as the number of edges in $E(\Gamma)$ incident on v, each loop counting as two edges. Also we say that a *cycle*, if any, *has length m*, with $m \geq 2$, if it contains exactly m edges in the graph. Isomorphic graphs will be considered as the same graph. We review the definition of labeled graph isomorphism.

Definition 2.2. We say that two labeled graphs $(\Gamma, \ell), (\Gamma', \ell')$ are *isomorphic*, and we write $(\Gamma, \ell) \cong (\Gamma', \ell')$, if there exist a bijection $\Phi : V(\Gamma) \to V(\Gamma')$ and a bijection $\Psi : E(\Gamma) \to E(\Gamma')$ such that,

- $e = e(v_1, v_2)$ is in $E(\Gamma)$ if and only if $\Psi(e) = e(\Phi(v_1), \Phi(v_2))$ is in $E(\Gamma')$ (i.e. Φ preserves the edges), and
- for every $v \in V(\Gamma)$, $\ell(v) = \ell'(\Phi(v))$ (i.e. Φ preserves the labels).

When a labeled graph is obtained as the combinatorial Reeb graph of a function f, we denote it by writing (Γ_f, ℓ_f).

2.1. Reeb graphs of simple Morse functions

In the mathematical literature, the case of Reeb graphs of simple Morse functions on smooth compact manifolds appears as the most commonly studied (cf., e.g., [BF04]).

We recall that a smooth function $f : \mathcal{M} \to \mathbb{R}$ defined on a manifold is Morse if all of its critical points are non-degenerate, i.e. the Hessian at critical points is non-zero; moreover, it is said to be simple if it is injective on the set of its critical points.

Theorem 2.1 ([Ree46]). *Let $\mathcal{M}$ be a compact n-dimensional manifold and f a simple Morse function defined on $\mathcal{M}$. The quotient* *space $\mathcal{M}/\sim_f$ has the structure of a finite connected cell complex K of dimension 1, such that the set of 0-cells of K bijectively corresponds to the critical points of f.*

As a consequence of the previous result, we can identify $\mathcal{M}/\sim_f$ with a combinatorial Reeb graph Γ_f whose vertices correspond to the 0-cells and the edges to the 1-cells of K. Moreover, the vertices of Γ_f can be labeled by the function $\ell_f : V(\Gamma_f) \to \mathbb{R}$ induced by restricting $f : \mathcal{M} \to \mathbb{R}$ to its critical points. We call the pair (Γ_f, ℓ_f) the labeled Reeb graph of the manifold $\mathcal{M}$. An example of labeled Reeb graph is depicted in Figure 1.

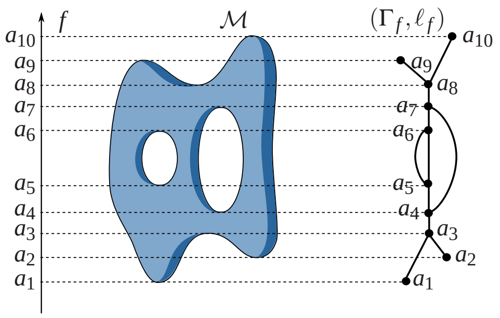


Figure 1: *Left: the height function $f : \mathcal{M} \to \mathbb{R}$; center: the surface $\mathcal{M}$; right: the associated labeled Reeb graph (Γ_f, ℓ_f).*

Let us focus on manifolds of dimension 1, i.e., curves, and dimension 2, i.e., surfaces. The stability of labeled Reeb graphs of curves via an edit distance has been proven in [DFL12], that of surfaces in [DFL16]. In both the cases, for any simple Morse functions f, g defined on the same manifold, the edit distance was defined as

$$d_E((\Gamma_f, \ell_f), (\Gamma_g, \ell_g)) = \inf_{S=(T_1,\dots,T_r)} \sum_{i=1}^{r} c(T_i),$$

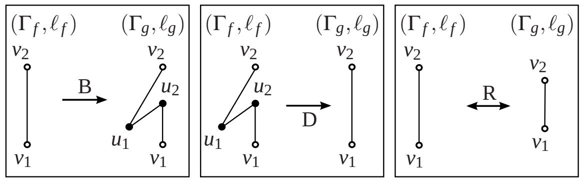
where S varies in a set of arbitrarily long sequences of elementary deformations $T_1, \dots, T_r$, necessary to transform (Γ_f, ℓ_f) into (Γ_g, ℓ_g), up to isomorphims. Each edit T_i has a cost $c(T_i)$ depending on its own type. What distinguishes the case of curves from that of surfaces is the type of admissible elementary deformations. In fact, the Reeb graph of a closed curve has only vertices of degree 2, while the Reeb graph of a surface has only vertices of degree 1 or 3. Figure 2 and Figure 3 illustrate the elementary deformations for curves and surfaces, respectively, together with their costs. In all the figures of this paper, black dots represent vertices whose degree needs to be exactly the same as it appears in the figure, whereas circled white dots represent vertices whose degree can be higher. Moreover, label values are represented by means of the height, and vertices are allowed to coincide whenever this makes sense.

For both curves and surfaces, the edit distance d_E yields the stability of Reeb graphs.

Theorem 2.2 ([DFL12, DFL16]). *For every $f, g : \mathcal{M} \to \mathbb{R}$, simple Morse functions defined on a connected, closed (i.e. compact and without boundary), orientable smooth manifold $\mathcal{M}$ of dimension 1 or 2, it holds that*

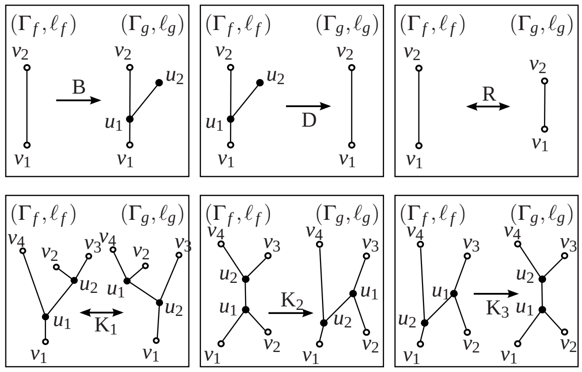
$$d_E((\Gamma_f, \ell_f), (\Gamma_g, \ell_g)) \le \|f - g\|_\infty,$$

with $\|f - g\|_\infty = \max_{p \in \mathcal{M}} |f(p) - g(p)|$.



$$c(B) = |\ell_g(u_1) - \ell_g(u_2)|/2,\ c(D) = |\ell_f(u_1) - \ell_f(u_2)|/2$$
$$c(R) = \max_{v \in V(\Gamma_f)} |\ell_f(v) - \ell_g(v)|$$

Figure 2: *Elementary deformations for Reeb graphs of curves and their costs.*



$$c(B) = |\ell_g(u_1) - \ell_g(u_2)|/2,\ c(D) = |\ell_f(u_1) - \ell_f(u_2)|/2$$
$$c(R) = \max_{v \in V(\Gamma_f)} |\ell_f(v) - \ell_g(v)|$$
$$c(K_i) = \max\{|\ell_f(u_1) - \ell_g(u_1)|, |\ell_f(u_2) - \ell_g(u_2)|\}$$

Figure 3: *Elementary deformations for Reeb graphs of surfaces and their costs.*

2.2. Reeb graphs of PL functions

Following [RS72], a polyhedron X is a subset of some $\mathbb{R}^n$, whose points $x \in X$ have cone neighborhoods in X, $N(x) = x * L(x) = \{\lambda \cdot x + \mu \cdot y : y \in L(x), \lambda, \mu \ge 0, \lambda + \mu = 1\}$, with $L(x)$ compact. Moreover, $f : X \to \mathbb{R}$ is a piecewise linear (briefly, PL) function if for each $x \in X$, $f(\lambda \cdot x + \mu \cdot y) = \lambda \cdot f(x) + \mu \cdot f(y)$ when $y \in L(x), \lambda, \mu \ge 0, \lambda + \mu = 1$.

Let X be a polyhedron and $f : X \to \mathbb{R}$ a PL function. It can be shown that $X/\sim_f$ is an abstract polyhedron of dimension not greater than 1. Hence, it embeds into a polyhedron R_f of dimension at most 1 in $\mathbb{R}^n$ for some n. Moreover, $f : X \to \mathbb{R}$ naturally induces a PL function $\hat{f} : R_f \to \mathbb{R}$.

For the sake of brevity, we postpone the proof of these facts to an extended version of this paper. However, we refer the reader to [EHP08] for a proof in the case when X is a manifold, and f is injective on the vertices of a simplicial complex triangulating X.

To define a combinatorial version of the Reeb graph that turns out to be a special instance of a labeled graph, for a cone neighborhood $N(x) = x * L(x)$ of x in X, we set

$$L^-(x) = \{y \in L(x) : f(y) < f(x)\}, L^+(x) = \{y \in L(x) : f(y) > f(x)\}.$$

As usual, we denote by β_0 the 0th Betti number, that is, the number of (arcwise) connected components.

Definition 2.3. We say that $x \in X$ is a *Reeb-regular point* of f if $\beta_0(L^-(x)) = 1$ and $\beta_0(L^+(x)) = 1$ for every cone neighborhood $N(x) = x * L(x)$ of x in X. Moreover, we say that $x \in X$ is a *Reeb-critical point* of f if it is not Reeb-regular.

Definition 2.4. We call the pair (Γ_f, ℓ_f) the *labeled Reeb graph* (briefly, *Reeb graph*) of the PL function $f : X \to \mathbb{R}$ if Γ_f is the graph whose vertex set $V(\Gamma_f)$ is the set of Reeb-critical points of $\hat{f} : R_f \to \mathbb{R}$, and whose edge set $E(\Gamma_f)$, if non-empty, is given by the set of 1-cells of the canonical cell decomposition of R_f; moreover, $\ell_f : V(\Gamma_f) \to \mathbb{R}$ is the function that coincides with $\hat{f}$ on the Reeb-critical points of $\hat{f}$.

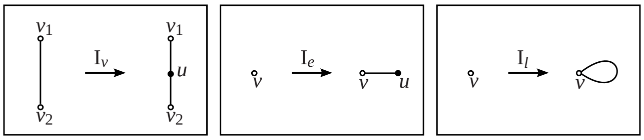
We observe that, by construction, if (Γ_f, ℓ_f) is a labeled Reeb graph, then Γ_f contains no loops, even though it may contain cycles, and if $v \in V(\Gamma_f)$, then $\beta_0(L^-(v)) \cdot \beta_0(L^+(v)) \neq 1$ for some cone neighborhood neighborhood $N(v) = v * L(v)$ of v in Γ_f. Moreover, ℓ_f takes different values on pairs of adjacent vertices: if v_1 and v_2 are adjacent in Γ_f, then $\ell_f(v_1) \neq \ell_f(v_2)$.

3. The edit distance for Reeb graphs comparison

In this section we define an edit distance for arbitrary labeled graphs to be used in particular to compare Reeb graphs. First, we introduce a set of *edit operations* on labeled graphs and prove that any two labeled graphs can be transformed into each other by a finite sequence of edit operations, called an *edit sequence*. Next, we define the cost of an edit sequence and our edit distance.

Edit operations on labeled graphs are of four types: 1. insertions, 2. deletions, 3. slidings, and 4. relabelings. These operations are formally defined in Definitions 3.1-3.4.

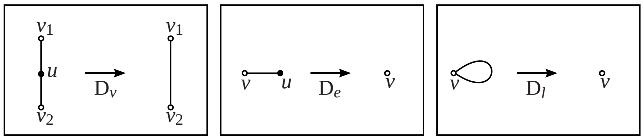
1. Insert operations:



Definition 3.1. Let (Γ, ℓ) be a labeled graph.

- We define a *vertex insertion* (I_v) to be any transformation T of (Γ, ℓ) such that, for a fixed edge $e(v_1, v_2) \in E(\Gamma)$, with $\ell(v_1) \geq \ell(v_2)$, $T(\Gamma, \ell)$ is a labeled graph (Γ', ℓ') defined as follows:
  - $V(\Gamma') = V(\Gamma) \cup \{u\}$;
  - $E(\Gamma') = (E(\Gamma) - \{e(v_1, v_2)\}) \cup \{e(v_1, u), e(u, v_2)\}$;
  - $\ell'_{|V(\Gamma)} = \ell$ and $\ell(v_1) \geq \ell'(u) \geq \ell(v_2)$.
- We define an *edge insertion* (I_e) to be any transformation T of (Γ, ℓ) such that, for a fixed vertex $v \in V(\Gamma)$, $T(\Gamma, \ell)$ is the labeled graph (Γ', ℓ') defined as follows:
  - $V(\Gamma') = V(\Gamma) \cup \{u\}$;
  - $E(\Gamma') = E(\Gamma) \cup \{e(v, u)\}$;
  - $\ell'_{|V(\Gamma)} = \ell$ and $\ell'(u) = \ell(v)$.
- We define a *loop insertion* (I_l) to be any transformation T of (Γ, ℓ) such that, for a fixed vertex $v \in V(\Gamma)$, $T(\Gamma, \ell)$ is the labeled graph (Γ', ℓ') defined as follows:
  - $V(\Gamma') = V(\Gamma)$;
  - $E(\Gamma') = E(\Gamma) \cup \{e(v, v)\}$;
  - $\ell' = \ell$.

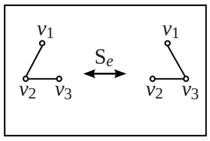
2. Delete operations:



Definition 3.2. Let (Γ, ℓ) be a labeled graph.

- We define a *vertex deletion* (D_v) to be any transformation T of (Γ, ℓ) such that, for fixed edges $e(v_1, u), e(u, v_2) \in E(\Gamma)$, with u a vertex of degree 2, and $\ell(v_1) \geq \ell(u) \geq \ell(v_2)$, $T(\Gamma, \ell)$ is the labeled graph (Γ', ℓ') defined as follows:
  - $V(\Gamma') = V(\Gamma) - \{u\}$;
  - $E(\Gamma') = (E(\Gamma) - \{e(v_1, u), e(u, v_2)\}) \cup \{e(v_1, v_2)\}$;
  - $\ell' = \ell_{|V(\Gamma) - \{u\}}$.
- We define an *edge deletion* (D_e) to be any transformation T of (Γ, ℓ) such that, for a fixed edge $e(v, u) \in E(\Gamma)$, with u a vertex of degree 1, and $\ell(v) = \ell(u)$, $T(\Gamma, \ell)$ is the labeled graph (Γ', ℓ') defined as follows:
  - $V(\Gamma') = V(\Gamma) - \{u\}$;
  - $E(\Gamma') = E(\Gamma) - \{e(v, u)\}$;
  - $\ell' = \ell_{|V(\Gamma) - \{u\}}$.
- We define a *loop deletion* (D_l) to be any transformation T of (Γ, ℓ) such that, for a fixed edge $e(v, v) \in E(\Gamma)$, $T(\Gamma, \ell)$ is the labeled graph (Γ', ℓ') defined as follows:
  - $V(\Gamma') = V(\Gamma)$;
  - $E(\Gamma') = E(\Gamma) - \{e(v, v)\}$;
  - $\ell' = \ell$.

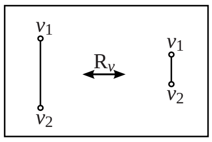
3. Slide operation:



Definition 3.3. Let (Γ, ℓ) be a labeled graph.

- We define an *edge sliding* (S_e) to be any transformation T of (Γ, ℓ) such that, for fixed edges $e(v_1, v_2), e(v_2, v_3) \in E(\Gamma)$, with either $\ell(v_1) > \ell(v_2) = \ell(v_3)$, or $\ell(v_1) < \ell(v_2) = \ell(v_3)$, $T(\Gamma, \ell)$ is the labeled graph (Γ', ℓ') defined as follows:
  - $V(\Gamma') = V(\Gamma)$;
  - $E(\Gamma') = (E(\Gamma) - \{e(v_1, v_2)\}) \cup \{e(v_1, v_3)\}$;
  - $\ell' = \ell$.

4. Relabel operation:



Definition 3.4. Let (Γ, ℓ) be a labeled graph.

- We define a *relabeling* (R_v) to be any transformation T of (Γ,ℓ) such that $T(\Gamma,\ell)$ is a labeled graph (Γ',ℓ') defined as follows:
  - $V(\Gamma') = V(\Gamma)$;
  - $E(\Gamma') = E(\Gamma)$;
  - For every $u,v \in V(\Gamma)$, if $\ell(u) \le \ell(v)$, then $\ell'(u) \le \ell'(v)$.

We now introduce the concept of inverse of an edit operation.

Definition 3.5. Let T be an edit operation such that $T(\Gamma,\ell) \cong (\Gamma',\ell')$. Let us identify $T(\Gamma,\ell)$ with (Γ',ℓ') via the pair of bijections (Φ,Ψ) inducing the isomorphism. We define the *inverse operation* of T, denoted by T^{-1}, as the edit operation that acts on the vertices, edges, and labels of (Γ',ℓ') as follows:

- if T is a delete operation that removes one vertex (edge, loop, resp.), then T^{-1} is an insert operation that adds the same vertex, with the same label (edge, loop, resp.), and vice versa if T is an insert operation;
- if T is a slide operation that changes adjacencies among three vertices, then T^{-1} is a slide operation that changes adjacencies among the same three vertices in the inverse way;
- if T is a relabel operation that changes labels to the vertices of Γ, then T^{-1} is again a relabel operation that changes labels to the same vertices in the inverse way.

Remark 1. Definition 3.5 implies that, if $T(\Gamma,\ell) \cong (\Gamma',\ell')$, then $T^{-1}(\Gamma',\ell') \cong (\Gamma,\ell)$.

Applying an edit operation to a labeled graph produces again a labeled graph. Thus, we can apply edit operations iteratively. We use this fact in the next Definition 3.6. Given an edit operation T of (Γ,ℓ) and an edit operation T' of $T(\Gamma,\ell)$, the composition $T'T$ means applying first T and then T'.

Definition 3.6. We call an *edit sequence* of the labeled graph (Γ,ℓ) any finite ordered sequence $S = (T_1,T_2,\ldots,T_n)$ of edit operations such that T_1 is an edit operation acting on (Γ,ℓ), and for every $2 \le k \le n$, T_k is an edit operation acting on $T_{k-1}T_{k-2}\cdots T_1(\Gamma,\ell)$. We denote by $S(\Gamma,\ell)$ the result of the editings $T_nT_{n-1}\cdots T_1$ applied to (Γ,ℓ). Moreover, if $S = (T_1,\ldots,T_n)$ is such that $S(\Gamma,\ell) \cong (\Gamma',\ell')$, then the *inverse sequence* of S is $S^{-1}(\Gamma',\ell') \cong (\Gamma,\ell)$, where $S^{-1} = (T_n^{-1},\ldots,T_1^{-1})$.

In what follows, we write $\mathcal{S}((\Gamma,\ell),(\Gamma',\ell'))$ to denote the set of edit sequences turning the labeled graph (Γ,ℓ) into the labeled graph (Γ',ℓ') up to isomorphisms:

$$\mathcal{S}((\Gamma,\ell),(\Gamma',\ell')) = \{S = (T_1,\ldots,T_n), n \ge 1 : S(\Gamma,\ell) \cong (\Gamma',\ell')\}.$$

In the following part of the section we prove that, for any pair of labeled graphs, $(\Gamma,\ell),(\Gamma',\ell')$, the set $\mathcal{S}((\Gamma,\ell),(\Gamma',\ell'))$ is nonempty. To do so, only with the aim of simplifying the proof, we reduce our problem to the similar one treated in [DFL16], where labeled graphs have only vertices of degree 1 or 3, the vertices of degree 3 are only up- or down-forks, and there are neither loops nor vertices with equal labels.

We recall that in a labeled graph, a vertex v of degree 3 is called an *up-fork* (resp., *down-fork*), if two of its adjacent vertices (possibly coincident), say v_1,v_2, are such that $\ell(v_1),\ell(v_2) > \ell(v)$ (resp., $\ell(v_1),\ell(v_2) < \ell(v)$), and the third, say v_3 is such that $\ell(v_3) < \ell(v)$ (resp., $\ell(v_3) > \ell(v)$). Hence, in both the cases, there exists at least one vertex adjacent to v with a label higher than $\ell(v)$ and at least one vertex adjacent to v with a label lower than $\ell(v)$.

Lemma 3.1. *For any labeled graph (Γ,ℓ), there exists an edit sequence S such that $S(\Gamma,\ell)$ is a labeled graph (Γ',ℓ') with the following properties:*

- *the vertices of Γ' are either of degree 1, or up- or down-forks of degree 3. In particular, Γ' has no loops.*
- *ℓ' is injective on $V(\Gamma')$.*

Proof. Without loss of generality, we can assume that ℓ is injective on $V(\Gamma)$, otherwise we apply a relabel operation to Γ to achieve injectivity. Moreover, we can assume that Γ has no loops, after applying appropriate loop deletions.

Now we show that each vertex v of Γ that is neither of degree 1, nor an up- or down-fork of degree 3, can be removed or transformed into a vertex with the claimed properties. More precisely, let v be any vertex in Γ. We consider the following cases.

- Case $\deg(v) = 0$: We can take the edit sequence $S = (T_1,T_2)$, with T_1 the edge insertion that inserts a vertex u, with the same label as v, and the edge $e(u,v)$, and T_2 a relabeling R_v that changes the label of u. In $S(\Gamma,\ell)$ the vertices u and v are of degree 1.
- Case $\deg(v) = 2$: We observe that either $\#\{w \in V(\Gamma) : e(w,v) \in E(\Gamma), \ell(w) < \ell(v)\} < 2$ or $\#\{w \in V(\Gamma) : e(w,v) \in E(\Gamma), \ell(w) > \ell(v)\} < 2$. Hence, we can take the edit sequence $S = (T_1,T_2)$, with T_1 the edge insertion that inserts a vertex u, with the same label as v, and the edge $e(u,v)$. Thus, in $T_1(\Gamma,\ell)$, the vertex v is of degree 3. Moreover, if $\#\{w \in V(\Gamma) : e(w,v) \in E(\Gamma), \ell(w) < \ell(v)\} < 2$, then we choose T_2 to be a relabeling such that $\ell(u) < \ell(v)$, otherwise, we choose T_2 such that $\ell(u) > \ell(v)$. As a result, v has turned into an up- or down-fork and the other vertices of Γ have changed neither adjacencies nor labels.
- Case $\deg(v) \ge 3$: Possibly after a relabeling, we can suppose that at least two of the vertices adjacent to v, say v_1,v_2, are such that $\ell(v_1),\ell(v_2) > \ell(v)$ or $\ell(v_1),\ell(v_2) < \ell(v)$. Let us consider the first case, the other being analogous. Let $w_1,\ldots,w_{k-2}$ the other vertices adjacent to v. We transform (Γ,ℓ) through the edit sequence $S = (T_1,T_2,T_3,T_4)$, where the edits T_i's , with $i = 1,\ldots,4$, are sequences taken as follows (see also Figure 4): $T_1 = S_e^1 I_e^1$, where I_e^1 is the edge insertion that inserts a vertex u_1 of degree 1, with the same label as v, and the edge $e(u_1,v)$, while S_e^1 is the edge sliding that removes the edge $e(v,w_1)$ and inserts the edge $e(u_1,w_1)$; $T_2 = R_v I_e$, where I_e is the edge insertion that inserts a vertex w of degree 1, with the same label as u_1, and the edge $e(u_1,w)$, while R_v is the relabeling that relabels w in such a way that, if $\ell(w_1) > \ell(u_1)$ before, then $\ell(w) < \ell(u_1)$ after, while if $\ell(w_1) < \ell(u_1)$ before, then $\ell(w) > \ell(u_1)$ after; $T_3 = S_e^{k-2} I_v^{k-2} \cdots S_e^2 I_v^2$, where I_v^j is the vertex insertion that inserts a vertex u_j of degree 2 between v and u_{j-1}, with the same label as v, thus removing the edge $e(v,u_{j-1})$, and inserting the edges $e(v,u_j), e(u_j,u_{j-1})$, while S_e^j is the elementary deformation that removes the edge $e(v,w_j)$ and inserts the edge $e(u_j,w_j)$; $T_4 = R_v'$, where R_v' is the relabeling that relabels the vertices $u_1,\ldots,u_{k-2}$ in such a way that in $S(\Gamma,\ell)$, $\ell(v) > \ell(u_{k-2}) > \ldots > \ell(u_1)$. Recalling that $\ell(v_1),\ell(v_2) > \ell(v)$, the vertex v is of degree 3 and an up-fork in $S(\Gamma,\ell)$, while $u_1,\ldots,u_{k-2}$ are of degree 3 and up- or down-forks in $S(\Gamma,\ell)$, depending on the labels of $w_1,\ldots,w_{k-2}$. Also in this

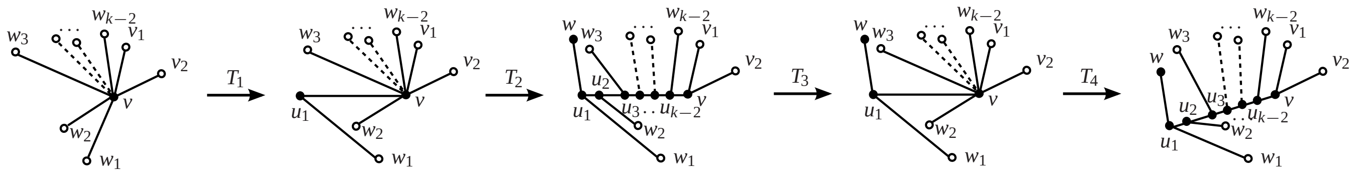


Figure 4: *The edit sequence that splits a vertex of degree greater than 3 into a number of up- or down-forks*

case, all these operations do not change neither the labels nor the adjacencies of vertices of the original graph different from v and its adjacent vertices.

□

Proposition 3.2. *Let $(\Gamma,\ell),(\Gamma',\ell')$ be two labeled graphs. The set $\mathcal{S}((\Gamma,\ell),(\Gamma',\ell'))$ is non-empty.*

Proof. Let us apply Lemma 3.1 to both (Γ,ℓ) and (Γ',ℓ'), for example starting from the lowest to the highest vertex, and call S and S' the edit sequences such that $S(\Gamma,\ell)$ and $S'(\Gamma,\ell)$ are labeled graphs whose vertices have different labels, and degree 1 or 3, in this case being up- or down-forks.

Under these assumptions, [DFL16, Prop. 23] applies to $S(\Gamma,\ell)$ and $S'(\Gamma',\ell')$. More precisely, letting $n,m \geq 0$ be the number of linearly independent cycles of $S(\Gamma,\ell)$ and $S'(\Gamma,\ell)$, respectively, $S(\Gamma,\ell)$ can be transformed into a labeled graph (Γ_1,ℓ_1) with exactly two vertices of degree 1, and n cycles of length 2, while $S'(\Gamma',\ell')$ can be transformed into a labeled graph (Γ'_1,ℓ'_1) with exactly two vertices of degree 1, and m cycles of length 2. It is sufficient to apply a finite sequence of elementary deformations of birth-, death-, relabeling-, K_1-, K_2- and K_3-types (see Figure 3).

To prove our claim, we start by showing in Figure 5 (rows 1-3) that each elementary deformation of birth, death, or K_i-type can be obtained also by applying a finite sequences of the edit operations introduced in Definitions 3.1-3.4. The deformation of relabeling type is already a particular case of the relabeling operation defined here.

As a consequence, it holds that $\mathcal{S}(S(\Gamma,\ell),(\Gamma_1,\ell_1))$ and $\mathcal{S}(S'(\Gamma',\ell'),(\Gamma'_1,\ell'_1))$ are non-empty.

Now, to show that $\mathcal{S}((\Gamma,\ell),(\Gamma',\ell'))$ is also non-empty, we consider the following two cases: (i) the case when $m=n$, and (ii) the case when $m \neq n$.

(i) If $m=n$, then there is a bijection $\Phi: V(\Gamma_1) \to V(\Gamma'_1)$ preserving adjacencies. Hence, it is sufficient to take the relabeling S'' of the vertices that, for every $v \in V(\Gamma_1)$, changes the label $\ell_1(v)$ into the label $\ell'_1(\Phi(v))$.

(ii) If $m \neq n$, then we can assume that $n > m$. Let $V(\Gamma_1) = \{v_0, v_1, v'_1, \ldots, v_n, v'_n, v_{n+1}\}$ and $V(\Gamma'_1) = \{u_0, u_1, u'_1, \ldots, u_m, u'_m, u_{m+1}\}$ as in Figure 6 (leftmost and rightmost graphs). We consider the sequence $S'' = (T_1,\ldots,T_6) \in \mathcal{S}((\Gamma_1,\ell_1),(\Gamma'_1,\ell'_1))$ defined as follows: T_1 is the relabel operation that relabels the up-forks $v'_{m+1},\ldots,v'_n \in V(\Gamma)$ in such a way that their labels in $T_1(\Gamma_1,\ell_1)$ are the same as $v_{m+1},\ldots,v_n$, respectively; T_2 is the sequence of edge slidings that delete the edges $e(v'_j,v_{j+1})$, with $j=m+1,\ldots,n$, and add the edges $e(v_j,v_{j+1})$, with $j=m+1,\ldots,n$; T_3 is the sequence of vertex deletions that remove the vertices $v'_{m+1},\ldots,v'_n$; T_4 is the sequence of loop deletions that remove the edges $e(v_j,v_j)$, with $j=m+1,\ldots,n$; T_5 is the sequence of vertex deletions that remove the vertices $v_{m+1},\ldots,v_n$; finally, T_6 is a relabel operation analogous to that used in the case (i).

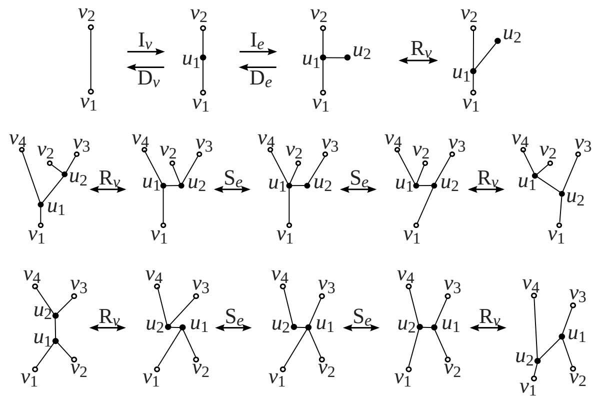


Figure 5: *The elementary deformations in Figure 3 can be obtained as sequences of edit operations. Top row: elementary deformation of B- and D- types. Center row: elementary deformation of K_1-type. Bottom row: elementary deformation of K_2- and K_3-types.*

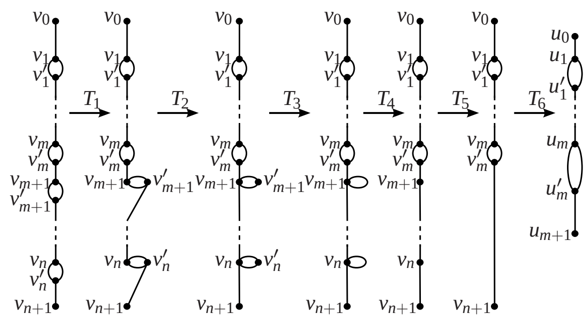


Figure 6: *How to transform the leftmost Reeb graph into the rightmost one.*

In conclusion, $\mathcal{S}((\Gamma_f,\ell_f),(\Gamma_g,\ell_g))$ contains at least the edit sequence (S,S'',S'^{-1}), proving that it is non-empty. □

The rest of the section is devoted to define our edit distance. We start introducing the cost of an edit sequence.

Definition 3.7. Let $S = (T_1, \dots, T_n) \in \mathcal{S}((\Gamma,\ell),(\Gamma',\ell'))$. Set $(\Gamma,\ell) = (\Gamma_1,\ell_1)$, $(\Gamma',\ell') = (\Gamma_{n+1},\ell_{n+1})$, and $(\Gamma_{i+1},\ell_{i+1}) = T_i(\Gamma_i,\ell_i)$ for $i = 1,\dots,n$. Setting $J_S(v) = \{i \in \mathbb{N}_{n+1} : v \in V(\Gamma_i)\}$, the *cost* of S is taken to be

$$c(S) = \max_{v \in \bigcup_i^{n+1} V(\Gamma_i)} \left(\max_{i \in J_S(v)} \ell_i(v) - \min_{i \in J_S(v)} \ell_i(v) \right).$$

Remark 2. By Definition 3.7, we have:

(i) if $T \in \mathcal{S}((\Gamma,\ell),(\Gamma',\ell'))$ is an insert, deletion, or slide operation, then $c(T) = 0$;
(ii) if $T \in \mathcal{S}((\Gamma,\ell),(\Gamma',\ell'))$ is a relabel operation, then $c(T) \geq 0$;
(iii) for every edit sequence $S \in \mathcal{S}((\Gamma,\ell),(\Gamma',\ell'))$, $c(S^{-1}) = c(S)$.

The following example illustrates how to compute the cost of an edit sequence.

Example 3.1. Let us consider the sequence $S = (I_v, I_e, R_v)$ displayed in the first row of Figure 5 that takes the leftmost graph (Γ_1,ℓ_1) to the rightmost graph (Γ_4,ℓ_4). By Remark 2, we get $c(S) = c(R_v)$, with $R_v(\Gamma_3,\ell_3) \cong (\Gamma_4,\ell_4)$. Hence, $c(S)$ is the maximum between $\max\{\ell_3(u_1),\ell_4(u_1)\} - \min\{\ell_3(u_1),\ell_4(u_1)\}$ and $\max\{\ell_3(u_2),\ell_4(u_2)\} - \min\{\ell_3(u_2),\ell_4(u_2)\}$, that is

$$c(S) = \max\{\ell_3(u_1) - \ell_4(u_1), \ell_4(u_2) - \ell_3(u_2)\}.$$

Definition 3.8. The *edit distance* between any two labeled graphs (Γ,ℓ) and (Γ',ℓ') is defined to be

$$\delta_E((\Gamma,\ell),(\Gamma',\ell')) = \inf_{S \in \mathcal{S}((\Gamma,\ell),(\Gamma',\ell'))} c(S).$$

Proposition 3.3. *The edit distance δ_E is a pseudo-metric on isomorphism classes of labeled graphs.*

Proof. By Proposition 3.2, δ_E is a real number. The coincidence property can be verified by observing that the relabel operation T that does not change any label, i.e. $T(\Gamma_f,\ell_f) = (\Gamma_f,\ell_f)$, has cost $c(T) = 0$, yielding $\delta_E = 0$. The symmetry property holds because, for every edit sequence $S \in \mathcal{S}((\Gamma,\ell),(\Gamma',\ell'))$, $c(S^{-1}) = c(S)$ and $S(\Gamma,\ell) \cong (\Gamma',\ell')$ if and only if $(\Gamma,\ell) \cong S^{-1}(\Gamma',\ell')$. Finally, the triangle inequality can be proved in the standard way. □

The edit distance is not a metric because different labeled graphs (for example, two graphs connected by an editing sequence involving no relabel operations) can have zero distance.

4. The stability property in the case of Morse functions

The goal of this section is to show the robustness of Reeb graphs with respect to perturbations of the function. In this work, we only consider the case of Morse functions on curves or surfaces. We do not face with the same problem in the case of manifolds with a dimension higher than 2, while the case of PL functions on polyhedra is postponed to an extended version of this paper.

As mentioned before (see Theorem 2.2), the edit distance d_E between Reeb graphs of curves or surfaces endowed with simple Morse functions implies the stability of Reeb graphs with respect to function perturbations. Now we show that the general edit distance δ_E inherits the same stability property from that of d_E.

In this section, to avoid confusion, we add the superscript M to the edits defined for Reeb graphs of Morse functions to distinguish them from those introduced for general labeled graphs. The cost of an edit will be always denoted by c, the presence of the superscript M in the considered edit signaling that the cost must be computed as explained in Figures 2, 3.

Proposition 4.1. *Let $\mathcal{M}$ be a connected, closed, orientable, smooth manifold of dimension 1 or 2. Let $f : \mathcal{M} \to \mathbb{R}$ be a simple Morse function and (Γ_f,ℓ_f) the associated labeled Reeb graph. The following statements hold:*

(i) For every elementary deformation T^M, there exists an edit sequence S such that $S(\Gamma_f,\ell_f) \cong T^M(\Gamma_f,\ell_f)$ and $c(S) \leq c(T^M)$.
(ii) For every deformation S^M, there exists an edit sequence S such that $S(\Gamma_f,\ell_f) \cong S^M(\Gamma_f,\ell_f)$ and $c(S) \leq c(S^M)$.

Proof. Let us prove statement (i) in the case when $\mathcal{M}$ is a closed curve. Let T^M be an elementary deformation of birth-type. The case when T^M is of death-type can be shown analogously. Let us call $(\Gamma_f,\ell_f) = (\Gamma_1,\ell_1)$ and $T^M(\Gamma_f,\ell_f) = (\Gamma_5,\ell_5)$ the leftmost and the rightmost graph in Figure 7, respectively. As recalled under Figure 2, the cost of this deformation is $c(T^M) = \frac{\ell_5(u_2)-\ell_5(u_1)}{2}$.

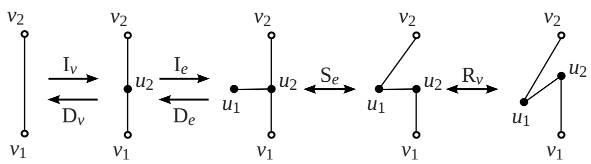


Figure 7: *The elementary deformations in Figure 2 (left) can be obtained as sequences of edit operations introduced in Definitions 3.1-3.4.*

Let $S = (I_v, I_e, S_e, R_v)$ be the edit sequence displayed in Figure 7 such that $S(\Gamma_1,\ell_1) \cong (\Gamma_5,\ell_5)$, with $I_v(\Gamma_1,\ell_1) \cong (\Gamma_2,\ell_2)$, $I_e(\Gamma_2,\ell_2) \cong (\Gamma_3,\ell_3)$, $S_e(\Gamma_3,\ell_3) \cong (\Gamma_4,\ell_4)$ $R_v(\Gamma_4,\ell_4) \cong (\Gamma_5,\ell_5)$. The cost of S is $c(S) = c(R_v)$ because of Remark 2. Hence, by an argument analogous to that used in Example 3.1, $c(S) = \max(\ell_4(u_1) - \ell_5(u_1), \ell_4(u_2) - \ell_5(u_2))$. Setting $\ell_4(u_1) = \ell_4(u_2) = \frac{\ell_5(u_1)+\ell_5(u_2)}{2}$, we get $c(S) = c(T^M)$.

The proof of statement (i) in the case when $\mathcal{M}$ is a surface is based on a similar argument to the one considered above. The different types of elementary deformations T^M are displayed and their costs are recalled in Figure 3. For each of these elementary deformations, Figure 5 shows an edit sequence such that $S(\Gamma_f,\ell_f) \cong T^M(\Gamma_f,\ell_f)$. In particular, if we set $\ell_4(u_1) < \ell_3(u_1) = \ell_3(u_2) = \frac{\ell_4(u_1)+\ell_4(u_2)}{2} < \ell_4(u_2)$ in the first row, and $\ell_1(u_1), \ell_5(u_2) < \ell_2(u_1) = \ell_2(u_2) = \frac{\ell_1(u_1)-\ell_1(u_2)}{2} = \frac{\ell_5(u_1)-\ell_5(u_2)}{2} < \ell_1(u_2), \ell_5(u_1)$ in the second and third row, in all the cases, we obtain $c(S) = c(T^M)$.

Let us now consider a deformation $S_n^M = (T_1^M, \dots, T_n^M)$ acting on the Reeb graph (Γ_f,ℓ_f) of a manifold $\mathcal{M}$ of dimension 1 or 2, and recall that the cost of S_n^M is $c(S_n^M) = \sum_{i=1}^n c(T_i^M)$. We prove statement (ii) by induction on n. If $n = 1$, i.e. the deformation S_n^M reduces to the elementary deformation T_1^M, then the claim follows from statement (i). Let us assume that, for any $n \geq 1$, there exists an edit sequence S' such that $S'(\Gamma_f,\ell_f) \cong S_n^M(\Gamma_f,\ell_f)$ and $c(S') \leq$

$c(S_n^M)$. We consider a deformation $S_{n+1}^M = (T_1^M, \ldots, T_n^M, T_{n+1}^M)$, where $S_n^M = (T_1^M, \ldots, T_n^M)$, and T_{n+1}^M is a certain elementary deformation. By the inductive assumption and statement (i), the edit sequence $S = (S', S'')$ is such that $S(\Gamma_f, \ell_f) \cong S_{n+1}^M(\Gamma_f, \ell_f)$, whenever S'' is an edit sequence such that $S''(S'(\Gamma_f, \ell_f)) \cong T_{n+1}^M(S'(\Gamma_f, \ell_f))$ that we can take with a cost $c(S'') \le c(T_{n+1}^M)$. It is sufficient to show that $c(S) \le c(S_{n+1}^M)$. Let us call $(\Gamma_f, \ell_f) = (\Gamma_1, \ell_1)$ and $T_k^M \cdots T_1^M(\Gamma_1, \ell_1) = (\Gamma_{k+1}, \ell_{k+1})$, with $k = 1, \ldots, n+1$, and assume that $c(S) = |\ell_i(v) - \ell_j(v)|$, with $v \in \bigcup_{k=1}^{n+2} V(\Gamma_k)$, $i, j \in J_S(v)$. If $i, j \in \{1, \ldots, n+1\}$ or $i, j \in \{n+1, n+2\}$, then $c(S) = c(S')$ or $c(S) = c(S'')$, respectively, and the claim follows; if $i \in \{1, \ldots, n+1\}$ and $j = n+2$, then we can state that $v \in V(\Gamma_{n+1})$ and $c(S) = |\ell_i(v) - \ell_{n+2}(v)| = |\ell_i(v) - \ell_{n+1}(v) + \ell_{n+1}(v) - \ell_{n+2}(v)|$. Hence, $c(S) \le |\ell_i(v) - \ell_{n+1}(v)| + |\ell_{n+1}(v) - \ell_{n+2}(v)| \le c(S') + c(S'') \le c(S_{n+1}^M)$. □

Corollary 4.2. *Let $\mathcal{M}$ be a connected, closed, orientable, smooth manifold of dimension 1 or 2. For every simple Morse functions $f, g : \mathcal{M} \to \mathbb{R}$, we have*

$$\delta_E((\Gamma_f, \ell_f), (\Gamma_g, \ell_g)) \le \|f - g\|_\infty .$$

Proof. The claim follows from Proposition 4.1 and Theorem 2.2. □

5. Conclusions

In this paper we presented a general edit distance between labeled graphs that can be applied to compare Reeb graphs. In particular, it allows for comparison of Reeb graphs of Morse functions and PL functions. We also proved that, in the case of Morse functions of curves or surfaces, this comparison is stable with respect to noise in the functions.

The proof of the stability property for manifolds of dimension higher than 2 and for PL functions on polyhedra requires further investigation and will be the subject of our future research. In particular, considering our strong interest in producing an algorithm able to test the proposed framework, the problem of stability in the piecewise linear context actually represents our main priority. For example, we believe that the inequality

$$\delta_E((\Gamma_f, \ell_f), (\Gamma_g, \ell_g)) \le \|f - g\|_\infty$$

holds true in the case when the considered PL maps $f, g : X \to \mathbb{R}$ are defined by extending injective functions defined on the vertices of a fixed simplicial complex K such that $X = |K|$, while requires much more effort without fixing any simplicial complex or when the assumption of injectivity is removed. Moreover, further investigations will concern also the case of robustness with respect to perturbations of the underlying space X.

References

[BB13] Barra V., Biasotti S.: 3D shape retrieval using kernels on Extended Reeb Graphs. *Pattern Recognition 46*, 11 (2013), 2985 – 2999. 1

[BF04] Bolsinov A. V., Fomenko A. T.: *Integrable Hamiltonian Systems: Geometry, Topology, Classification*. CRC Press, Boca Raton, FL, 2004. 2

[BGSF08] Biasotti S., Giorgi D., Spagnuolo M., Falcidieno B.: Reeb graphs for shape analysis and applications. *Theoretical Computer Science 392*, 1âĂŞ3 (2008), 5 – 22. Computational Algebraic Geometry and Applications. 1

[BGW14] Bauer U., Ge X., Wang Y.: Measuring distance between Reeb graphs. In *Proceedings of the Thirtieth Annual Symposium on Computational Geometry* (2014), SOCG'14, pp. 464–473. 2

[BKS*05] Bustos B., Keim D. A., Saupe D., Schreck T., Vranic D. V.: Feature-based similarity search in 3D object databases. *ACM Computing Surveys 37*, 4 (2005), 345–387. 1

[BMW15] Bauer U., Munch E., Wang Y.: Strong equivalence of the interleaving and functional distortion metrics for Reeb graphs. In *31st International Symposium on Computational Geometry (SoCG 2015)* (Dagstuhl, Germany, 2015), Arge L., Pach J., (Eds.), vol. 34 of *Leibniz International Proceedings in Informatics (LIPIcs)*, Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, pp. 461–475. 2

[BYM*13] Beketayev K., Yeliussizov D., Morozov D., Weber G. H., Hamann B.: Measuring the distance between merge trees. In *Topological Methods in Data Analysis and Visualization V (TopoInVis 2013)* (2013). 2

[CGK03] Cardone A., Gupta R. K., Karnik M.: A survey of shape similarity assessment algorithms for product design and manufacturing applications. *Journal of Computing and Information Science in Engineering 3* (2003), 109–118. 1

[CSEH07] Cohen-Steiner D., Edelsbrunner H., Harer J.: Stability of persistence diagrams. *Discrete Comput. Geom. 37*, 1 (Jan. 2007), 103–120. 1

[DFL12] Di Fabio B., Landi C.: Reeb graphs of curves are stable under function perturbations. *Mathematical Methods in the Applied Sciences 35*, 12 (2012), 1456–1471. 2, 3

[DFL16] Di Fabio B., Landi C.: The edit distance for Reeb graphs of surfaces. *Discrete & Computational Geometry 55*, 2 (2016), 423–461. 2, 3, 5, 6

[dSMP15] de Silva V., Munch E., Patel A.: Categorified Reeb graphs. *CoRR abs/1501.04147* (2015). 2

[EHP08] Edelsbrunner H., Harer J., Patel A. K.: Reeb spaces of piecewise linear mappings. In *Proceedings of the Twenty-fourth Annual Symposium on Computational Geometry* (2008), SCG '08, pp. 242–250. 3

[Gov] Govc D.:. Private communication. 2

[GXTL10] Gao X., Xiao B., Tao D., Li X.: A survey of graph edit distance. *Pattern Anal. Appl. 13*, 1 (Jan. 2010), 113–129. 2

[HSKK01] Hilaga M., Shinagawa Y., Kohmura T., Kunii T. L.: Topology matching for fully automatic similarity estimation of 3D shapes. In *ACM Computer Graphics, (Proc. SIGGRAPH 2001)* (Los Angeles, CA, August 2001), ACM Press, pp. 203–212. 1

[Ree46] Reeb G.: Sur les points singuliers d'une forme de Pfaff complétement intégrable ou d'une fonction numérique. *Comptes Rendus de L'Académie des Sciences 222* (1946), 847–849. 1, 2

[RS72] Rourke C. P., Sanderson B. J.: *Introduction to piecewise-linear topology*. Springer-Verlag, New York, 1972. Ergebnisse der Mathematik und ihrer Grenzgebiete, Band 69. 3

[SK91] Shinagawa Y., Kunii T.: Constructing a Reeb graph automatically from cross sections. *IEEE Computer Graphics and Applications 11*, 6 (nov 1991), 44–51. 1

[SKK91] Shinagawa Y., Kunii T. L., Kergosien Y. L.: Surface coding based on Morse theory. *IIEEE Computer Graphics and Applications 11*, 5 (1991), 66–78. 1

[TV08] Tangelder J. W., Veltkamp R. C.: A survey of content based 3D shape retrieval methods. *Multimedia Tools Appl. 39*, 3 (Sept. 2008), 441–471. 1