New Graph Distance Measures and Matching of Topological Maps for Robotic Exploration

Fabio Morbidi

***Abstract*— Comparing graph-structured maps is a task of paramount importance in robotic exploration and cartography, but unfortunately the computational cost of the existing similarity measures, such as the *graph edit distance* (GED), is prohibitive for large graphs. In this paper, we introduce and characterize three new graph distance measures which satisfy the requirements for a metric. The first one, `LogEig`, computes the square root of the sum of the squared logarithms of the generalized eigenvalues of the shifted Laplacian matrices associated with the two graphs, while the second calculates the Bures distance between these positive definite matrices. The third distance, `Rank`, computes the rank of the difference of the graph shift operators associated with the two graphs, e.g. the adjacency or the Laplacian matrix. Examples and numerical experiments with graphs from a publicly-available dataset, show the accuracy and computational efficiency of the new metrics for 2D topological-map matching, compared to the GED. The effect of spectral sparsification on the new graph distance measures is examined as well.**

I. Introduction

Graphs are frequently used to encode high-level information and comparing graph-structured data (e.g. computer, electricity or biological networks) is relevant across a range of engineering applications. Multiple graph distance measures[1] exist in the literature: among them, we mention here the *graph edit distance* or GED (it is defined as the minimum number of operations that transform one graph into the other [2], [3], [4]), the *diffusion distance* [5], the distance based on the *maximum common subgraph* [6], and the *spectral distances* [7], [1]. The notion of cut similarity of graphs [8] (and its generalization, spectral similarity [9]), have also received special attention in the past. However, the algorithms to compute these distances generally have a high cost and struggle with graphs with hundreds to thousands of nodes. To speed up the graph-similarity evaluation, *learning-based* methods such as SimGNN [10], have recently emerged but they come with no guarantees of optimality.

Once a graph distance has been chosen, the so-called *graph matching problem* can be introduced. Two variants of this problem exist: in the exact (approximate) graph matching problem, the two graphs have the same (different) number of nodes. They have both been the subject of extensive research in graph theory, computer vision and pattern recognition, in the last forty years [11], [12], [13].

The author is with the MIS laboratory, University of Picardie Jules Verne, 33 rue Saint-Leu, 80039 Amiens Cedex, France. Email: `fabio.morbidi@u-picardie.fr`

[1]We refer to these tools as “distance measures”, since, although all are symmetric, they may fail one or more of the other requirements for a metric [1].

Comparing 2D or 3D maps built by a single or multiple robots using measurements collected during the exploration of an unknown environment, is a crucial task in robotics. In many instances, a simplified representation of the environment, called *topological map* (an undirected graph), can be considered [14], [15]. These maps provide a concise description of the navigable space within the environment, but they lack scale. However, the topological relationship between any pair of points of the map (up/down, left/right) is preserved.

Motivated by applications in mobile robotics, in this article we propose three new graph distance measures grounded in algebraic and spectral graph theory, which also satisfy the axioms for a metric. The first distance, `LogEig`, relies on the generalized eigenvalues of the shifted Laplacian matrices of the two graphs, while the second computes the Bures distance [16] between these two Laplacians. The third distance, `Rank`, is integer-valued as the GED, and it applies to any graph shift operator (GSO) [17] associated with the two graphs. The effectiveness of the new metrics in terms of accuracy and computational complexity, is shown for the 2D topological-map matching problem. The impact of *spectral sparsification* [18] of a large graph (one thousand nodes), on the new similarity measures is studied as well. Indeed, spectral sparsification, the approximation of an arbitrary graph by a sparse graph on the same set of nodes, has proved to be a useful tool not only in combinatorial optimization, machine learning and network analysis, but also in mobile robotics, e.g. for collaborative pose estimation [19].

The rest of this article is organized as follows. In Sect. II, we review some basic notions of graph theory and geometry. Sect. III presents the main theoretical results of the paper. In Sect. IV, the new graph distance measures are compared using a simple numerical example and 38 instances of the COLD-TopoMaps dataset [20], for the topological-map matching problem. We also quantify the effect of spectral sparsification on the proposed metrics. Finally, in Sect. V, the main contributions of the paper are summarized and some priority areas for future research are discussed.

Notation: Throughout this article, $\mathbf{I}_n$ denotes the $n \times n$ identity matrix, $\mathbb{1} = [1, 1, \dots, 1]^T$ the column vector of n ones, $\mathbf{J} = \frac{1}{n}\mathbb{1}\mathbb{1}^T$ the $n \times n$ averaging matrix, $[\mathbf{A}]_{ij}$ the (i, j) entry of matrix $\mathbf{A}$ and “$\succeq$” the matrix inequality in the positive semidefinite sense. ⋄

II. Preliminaries: Graph theory and geometry

Let $\mathcal{G} = (V, E)$ be an undirected graph, where $V = \{v_1, \dots, v_n\}$ is the set of nodes and $E \subseteq V \times V$ is the set of edges. We assume that $\mathcal{G}$ is finite, with no self-loops or multiple edges: we shall denote this by $\mathcal{G} \in \mathscr{F}$. Henceforth, we also assume that $\mathcal{G}$ is *connected*.

The following matrices associated with a graph $\mathcal{G}$ will be used in the rest of this paper.

***Definition** 1 (Adjacency matrix* $\mathbf{A}$*):* The adjacency matrix $\mathbf{A}$ of graph $\mathcal{G}$ is an $n \times n$ matrix defined as $[\mathbf{A}]_{ij} = 1$ if $\{i, j\} \in E$ and $[\mathbf{A}]_{ij} = 0$ otherwise. ⋄

***Definition** 2 (Laplacian matrix* $\mathbf{L}$*):* The Laplacian matrix of graph $\mathcal{G}$ is an $n \times n$ symmetric, positive semidefinite matrix defined as $\mathbf{L} = \mathbf{D} - \mathbf{A}$, where $\mathbf{D} = \mathrm{diag}(\mathbf{A}\mathbb{1})$ is the *degree matrix*. ⋄

Three useful variants of the Laplacian matrix $\mathbf{L}$ are the *symmetric normalized Laplacian*, the *reduced Laplacian*, and the *shifted Laplacian*.

***Definition** 3 (Symmetric normalized Laplacian* $\mathbf{L}^{\mathrm{s}}$*):* The symmetric normalized Laplacian is an $n \times n$ matrix defined as $\mathbf{L}^{\mathrm{s}} = \mathbf{D}^{-1/2}\,\mathbf{L}\,\mathbf{D}^{-1/2} = \mathbf{I}_n - \mathbf{D}^{-1/2}\mathbf{A}\,\mathbf{D}^{-1/2}$. ⋄

It is easy to very that $0 = \lambda_1(\mathbf{L}^{\mathrm{s}}) < \lambda_2(\mathbf{L}^{\mathrm{s}}) \leq \ldots \leq \lambda_n(\mathbf{L}^{\mathrm{s}}) \leq 2$ where $\lambda_k(\mathbf{L}^{\mathrm{s}})$ is the k-th eigenvalue of $\mathbf{L}^{\mathrm{s}}$. Therefore, $\mathbf{L}^{\mathrm{s}}$ as $\mathbf{L}$ is a positive semidefinite matrix.

***Definition** 4 (Reduced Laplacian* $\mathbf{L}^{\star}$*):* Let $\mathrm{Orth}(\mathbb{1}) = \{\mathbf{Q} \in \mathbb{R}^{n\times(n-1)} : \mathbf{Q}^T\mathbf{Q} = \mathbf{I}_{n-1}, \mathbf{Q}^T\mathbb{1} = \mathbf{0}\}$. Select a matrix $\mathbf{Q} \in \mathrm{Orth}(\mathbb{1})$. The reduced Laplacian is the $(n-1) \times (n-1)$ symmetric matrix $\mathbf{L}^{\star} = \mathbf{Q}^T\mathbf{L}\,\mathbf{Q}$. ⋄

Differently from $\mathbf{L}$ and $\mathbf{L}^{\mathrm{s}}$, the reduced Laplacian $\mathbf{L}^{\star}$ is a positive definite matrix.

***Definition** 5 (Shifted Laplacian* $\mathbf{L}'$*):* The shifted Laplacian is an $n \times n$ symmetric matrix defined as $\mathbf{L}' = \mathbf{L} + \mathbf{J}$. ⋄

The spectrum of $\mathbf{L}$ is identical to that of $\mathbf{L}'$ with the exception of the zero eigenvalue of $\mathbf{L}$, which becomes one in $\mathbf{L}'$. This clearly shows that $\mathbf{L}'$ is positive definite.

As a more general algebraic descriptor of network topology, the notion of graph shift operator (GSO) has been proposed in the literature [17], [21].

***Definition** 6 (Graph shift operator):* The GSO associated with a graph $\mathcal{G}$ is an $n \times n$ matrix $\mathbf{S}$, such that $[\mathbf{S}]_{ij} \neq 0$ if and only if $i = j$ or $\{i, j\} \in E$. ⋄

The sparsity pattern of matrix $\mathbf{S}$ captures the local structure of graph $\mathcal{G}$, but no specific assumptions are made on the values of its non-zero entries.

Motivated by problems in linear algebra and spectral graph theory, the notion of *spectral similarity* for two graphs was proposed in [18].

***Definition** 7 (Spectral similarity [9]):* Let $\mathbf{L}$ and $\widetilde{\mathbf{L}}$ be the $n \times n$ Laplacian matrices of graphs $\mathcal{G}$ and $\widetilde{\mathcal{G}}$, respectively. $\widetilde{\mathcal{G}}$ is a σ-spectral approximation of $\mathcal{G}$, if $\frac{1}{\sigma}\widetilde{\mathbf{L}} \preceq \mathbf{L} \preceq \sigma\widetilde{\mathbf{L}}$ where $\sigma > 0$. ⋄

Graph $\widetilde{\mathcal{G}}$ is also called a *spectral sparsifier* of graph $\mathcal{G}$. Matrices $\mathbf{L}$ and $\widetilde{\mathbf{L}}$ have similar eigenvalues, in fact

$$\frac{1}{\sigma}\lambda_i(\widetilde{\mathbf{L}}) \leq \lambda_i(\mathbf{L}) \leq \sigma\lambda_i(\widetilde{\mathbf{L}}),\ i \in \{1, \ldots, n\}. \quad (1)$$

In graph theory, an *isomorphism* of graphs $\mathcal{G}_a$ and $\mathcal{G}_b$ is a bijection between the sets of nodes of $\mathcal{G}_a$ and $\mathcal{G}_b$.

***Definition** 8 (Isomorphic graphs):* Let $\mathbf{A}_a$ and $\mathbf{A}_b$ be the $n \times n$ adjacency matrices of graphs $\mathcal{G}_a$ and $\mathcal{G}_b$, respectively. $\mathcal{G}_a$ and $\mathcal{G}_b$ are isomorphic, denoted by $\mathcal{G}_a \simeq \mathcal{G}_b$, if and only if there exists a permutation matrix $\mathbf{P}$ such that $\mathbf{A}_b = \mathbf{P}\,\mathbf{A}_a\,\mathbf{P}^{-1}$. ⋄

***Definition** 9 (Metric space):* A metric space is an ordered pair $(\mathscr{M}, d)$ where $\mathscr{M}$ is a set and d is a metric or distance function on $\mathscr{M}$, i.e. a function $d : \mathscr{M} \times \mathscr{M} \to \mathbb{R}$ satisfying the following three axioms $\forall\, x, y, z \in \mathscr{M}$:

1) (*Positivity*): $d(x, y) \geq 0$ and $d(x, y) = 0$ only if $x = y$,
2) (*Symmetry*): $d(x, y) = d(y, x)$,
3) (*Triangle inequality*): $d(x, z) \leq d(x, y) + d(y, z)$. ⋄

III. Graph distance measures

In this section, we review two existing graph similarity measures (Sect. III-A and Sect. III-B) that will be used in Sect. IV for a comparative analysis. We then define three new graph distances (see Table I for a summary of their properties).

A. Graph edit distance

The graph edit distance (GED) is commonly used to measure the dissimilarity between two labeled graphs. It corresponds to the number of edge/node changes needed to make two graphs isomorphic [22].

***Definition** 10 (Graph edit distance):* Let $\mathcal{G}_a = (V_a, E_a)$ and $\mathcal{G}_b = (V_b, E_b)$ be two graphs with n nodes. The graph edit distance between $\mathcal{G}_a$ and $\mathcal{G}_b$, $d_{\mathrm{GED}}(\mathcal{G}_a, \mathcal{G}_b) : \mathscr{F} \times \mathscr{F} \to \mathbb{Z}_{\geq 0}$ is defined as

$$d_{\mathrm{GED}}(\mathcal{G}_a, \mathcal{G}_b) = \min_{(o_1,\ldots,o_k)\,\in\,\Gamma(\mathcal{G}_a,\,\mathcal{G}_b)} \sum_{i=1}^{k} c(o_i),$$

where $\Gamma(\mathcal{G}_a, \mathcal{G}_b)$ denotes the set of all edit paths $(o_1, \ldots, o_k)$ that allow to transform $\mathcal{G}_a$ into (a graph isomorphic to) $\mathcal{G}_b$ and $c(o_i) \geq 0$ is the cost associated with the elementary edit operation o_i. Elementary edit operations include node substitution $(v_a \to v_b)$, edge substitution $(e_a \to e_b)$, node deletion $(v_a \to \eta)$, edge deletion $(e_a \to \eta)$, node insertion $(\eta \to v_b)$ and edge insertion $(\eta \to e_b)$, where $v_a \in V_a$, $v_b \in V_b$, $e_a \in E_a$, $e_b \in E_b$ and η denotes a dummy node or edge used to model the insertion or deletion. ⋄

The cost function $c(\,\cdot\,)$ plays an important role in the computation of GED and it has significant impact on the optimization problem. For the sake of simplicity, in the following we will assume that $c(o_i) = 1$ for all i. It has been shown in [3] that each elementary operation satisfies the three axioms in Definition 9: hence, the GED defines a metric between graphs. Finally, we recall that the *exact* computation of the GED between two graphs is NP-hard. For this reason, various heuristics have been proposed in the literature over the past two decades [22], [23], [4].

B. Frobenius norm

The following graph distance has been recently proposed in [24, Sect. IV-B] to measure the deviation of a swarm of aerial robots from the desired formation (see also [25]).

***Definition** 11 (Frobenius norm):* Let $\mathbf{L}_a^{\mathrm{s}}$ and $\mathbf{L}_b^{\mathrm{s}}$ be the symmetric normalized Laplacians of the connected graphs with n nodes, $\mathcal{G}_a$ and $\mathcal{G}_b$, respectively. Then, $d_{\mathrm{F}}(\mathcal{G}_a, \mathcal{G}_b) : \mathscr{F} \times \mathscr{F} \to \mathbb{R}_{\geq 0}$ is defined as

$$d_{\mathrm{F}}(\mathcal{G}_a, \mathcal{G}_b) = \left\|\mathbf{L}_a^{\mathrm{s}} - \mathbf{L}_b^{\mathrm{s}}\right\|_{\mathrm{F}} = \sqrt{\mathrm{trace}\big[(\mathbf{L}_a^{\mathrm{s}} - \mathbf{L}_b^{\mathrm{s}})^2\big]},$$

where $\|\cdot\|_{\mathrm{F}}$ denotes the Frobenius norm. ⋄

C. `LogEig` *distance*

Inspired by [26], where a metric for covariance matrices is proposed, we define the LogEig distance between two graphs as follows.

***Definition** 12 (*LogEig *distance):* Let $\mathbf{L}_a'$ and $\mathbf{L}_b'$ be the shifted Laplacians of the connected graphs with n nodes, $\mathcal{G}_a$ and $\mathcal{G}_b$, respectively. The LogEig distance between $\mathcal{G}_a$ and $\mathcal{G}_b$, $d_{\log}(\mathcal{G}_a, \mathcal{G}_b) : \mathscr{F} \times \mathscr{F} \to \mathbb{R}_{\geq 0}$ is defined as

$$d_{\log}(\mathcal{G}_a, \mathcal{G}_b) = \sqrt{\sum_{i=1}^{n} \ln^2\big(\lambda_i(\mathbf{L}_a', \mathbf{L}_b')\big)},$$

where $\lambda_i(\mathbf{L}_a', \mathbf{L}_b')$, $i \in \{1, \ldots, n\}$, are the generalized eigenvalues of $\mathbf{L}_a'$ and $\mathbf{L}_b'$, i.e. they satisfy $\mathbf{L}_a' \boldsymbol{x}_i = \lambda_i \mathbf{L}_b' \boldsymbol{x}_i$, where $\boldsymbol{x}_i$ is the generalized eigenvector associated with the generalized eigenvalue λ_i. ◇

The LogEig distance can be equivalently rewritten as $d_{\log}(\mathcal{G}_a, \mathcal{G}_b) = (\text{trace}\big[\text{logm}^2\big((\mathbf{L}_b')^{-1}\mathbf{L}_a'\big)\big])^{1/2}$ where $\text{logm}(\cdot)$ denotes the matrix logarithm [26]. The LogEig distance is a metric on the manifold of symmetric positive definite matrices [26], since it satisfies the three axioms in Definition 9. Moreover, it is invariant under *congruence transformations*, i.e. for any invertible matrix $\mathbf{U} \in \mathbb{R}^{n\times n}$

$$\sum_{i=1}^{n} \ln^2\big(\lambda_i(\mathbf{L}_a', \mathbf{L}_b')\big) = \sum_{i=1}^{n} \ln^2\big(\lambda_i(\mathbf{U}\,\mathbf{L}_a'\,\mathbf{U}^T, \mathbf{U}\,\mathbf{L}_b'\,\mathbf{U}^T)\big),$$

and it is invariant under *inversion*, i.e.

$$\sum_{i=1}^{n} \ln^2\big(\lambda_i(\mathbf{L}_a', \mathbf{L}_b')\big) = \sum_{i=1}^{n} \ln^2\big[\lambda_i((\mathbf{L}_a')^{-1}, (\mathbf{L}_b')^{-1})\big].$$

Note that with large graphs, one could focus only on the first k generalized eigenvalues of $\mathbf{L}_a'$ and $\mathbf{L}_b'$ with $k \ll n$. In this case, a *truncated* LogEig distance is obtained.

***Remark** 1 (*LogEig *distance: alternative definition):* The LogEig distance could be equivalently defined by using the reduced Laplacian $\mathbf{L}^\star$ instead of the shifted Laplacian $\mathbf{L}'$. The results being identical, the advantage of $\mathbf{L}^\star$ is that only $n-1$ generalized eigenvalues must be computed. ◇

The next proposition provides an upper bound on the LogEig distance between a graph and its spectral sparsifier.

***Proposition** 1:* Let $\mathbf{L}'$ and $\widetilde{\mathbf{L}}'$ be the $n \times n$ shifted Laplacians of graph $\mathcal{G}$ and its σ-spectral approximation $\widetilde{\mathcal{G}}$. Then

$$d_{\log}(\mathcal{G}, \widetilde{\mathcal{G}}) \leq \sqrt{\sum_{i=1}^{n} \ln^2\left(\frac{\sigma\,\lambda_n(\widetilde{\mathbf{L}}')}{\lambda_i(\widetilde{\mathbf{L}}')}\right)}.$$

Proof: By Definition 12,

$$\begin{aligned} d_{\log}^2(\mathcal{G}, \widetilde{\mathcal{G}}) &= \textstyle\sum_{i=1}^{n} \ln^2\big[\lambda_i((\widetilde{\mathbf{L}}')^{-1}\mathbf{L}')\big] \\ &\leq \textstyle\sum_{i=1}^{n} \ln^2\big[\lambda_i((\widetilde{\mathbf{L}}')^{-1})\,\lambda_n(\mathbf{L}')\big], \end{aligned}$$

where the inequality follows from [27, Fact 8.18.17] and the observation that $\lambda_i((\widetilde{\mathbf{L}}')^{-1}\mathbf{L}') \geq 1$ for all i. By applying the logarithmic identities and by recalling that the eigenvalues of an $n \times n$ matrix $\mathbf{B}^{-1}$ are $\lambda_1^{-1}(\mathbf{B}), \ldots, \lambda_n^{-1}(\mathbf{B})$, we obtain:

$$\sum_{i=1}^{n} \ln^2\big[\lambda_i((\widetilde{\mathbf{L}}')^{-1})\,\lambda_n(\mathbf{L}')\big] = \sum_{i=1}^{n} \ln^2\left(\frac{\lambda_n(\mathbf{L}')}{\lambda_i(\widetilde{\mathbf{L}}')}\right).$$

The statement follows from the application of inequality (1). ■

Measure	Representation of the graph	$\mathcal{G}_a$, $\mathcal{G}_b$ connected	If $\mathcal{G}_a \simeq \mathcal{G}_b$ $d_*(\cdot,\cdot) = \mathbf{0}$	Metric
GED	(V, E)	No	Yes	Yes
Frobenius	$\mathbf{L}^s$	Yes	No	Yes
LogEig	$\mathbf{L}'$ or $\mathbf{L}^\star$	Yes	No	Yes
Bures	$\mathbf{L}'$ or $\mathbf{L}^\star$	Yes	No	Yes
Rank	Any GSO	No	No	Yes

TABLE I: Comparison of the five distance measures considered in Sect. III. The 4th column of the table refers to the ability of a distance to discern between two isomorphic graphs. The computational complexity of the last four distances is $O(n^3)$, where n is the number of nodes.

By analogy with the standard L^p-norm, we finally introduce the generalized LogEig distance.

***Definition** 13 (Generalized *LogEig *distance):* Let $\mathbf{L}_a'$ and $\mathbf{L}_b'$ be the shifted Laplacians of the connected graphs with n nodes, $\mathcal{G}_a$ and $\mathcal{G}_b$, respectively. For a real number $p \geq 1$, the generalized LogEig distance between $\mathcal{G}_a$ and $\mathcal{G}_b$, $d_{\log,p}(\mathcal{G}_a, \mathcal{G}_b) : \mathscr{F} \times \mathscr{F} \to \mathbb{R}_{\geq 0}$ is defined as

$$d_{\log,p}(\mathcal{G}_a, \mathcal{G}_b) = \sqrt[p]{\sum_{i=1}^{n} \Big|\ln\big(\lambda_i(\mathbf{L}_a', \mathbf{L}_b')\big)\Big|^p}.$$

◇

It is easy to verify that $d_{\log,2}(\mathcal{G}_a, \mathcal{G}_b) = d_{\log}(\mathcal{G}_a, \mathcal{G}_b)$ and that $\lim_{p\to\infty} d_{\log,p}(\mathcal{G}_a, \mathcal{G}_b) = \max_i \big|\ln\big(\lambda_i(\mathbf{L}_a', \mathbf{L}_b')\big)\big|$ (cf. the standard L^∞-norm).

D. *Bures distance*

The *Bures metric* on the manifold of symmetric positive definite matrices arises in various optimization problems, in quantum information and in the theory of optimal transport [16]. We propose to use it to compare two graphs.

***Definition** 14 (Bures distance):* Let $\mathbf{L}_a'$ and $\mathbf{L}_b'$ be the shifted Laplacians of the connected graphs with n nodes, $\mathcal{G}_a$ and $\mathcal{G}_b$, respectively. The Bures distance between $\mathcal{G}_a$ and $\mathcal{G}_b$, $d_{\text{B}}(\mathcal{G}_a, \mathcal{G}_b) : \mathscr{F} \times \mathscr{F} \to \mathbb{R}_{\geq 0}$ is defined as

$$d_{\text{B}}(\mathcal{G}_a, \mathcal{G}_b) = \sqrt{\text{trace}\Big[\mathbf{L}_a' + \mathbf{L}_b' - 2\big((\mathbf{L}_a')^{1/2}\,\mathbf{L}_b'\,(\mathbf{L}_a')^{1/2}\big)^{1/2}\Big]}, \quad (2)$$

where $(\mathbf{L}_a')^{1/2}$ is the matrix square root of $\mathbf{L}_a'$. ◇

One can show that d_{B} is a distance on covariance matrices [28, Remark 2.31]. A formula similar to (2) was proposed in [25], [29] to measure the distance between two graphs. However, the authors do not compare the graphs *directly*, but look at the signal distributions which are governed by the graphs, and they consider the pseudoinverse of the Laplacian, instead of the shifted (or reduced) Laplacian.

E. `Rank` *distance*

The *rank metric* measures the distance between two matrices by the rank of their difference. It is widely used in information and code theory [30], [31], but it has found no application so far, in graph theory. In this paper, we propose the following definition, which applies to any GSO associated with a graph and it is not restricted to positive definite matrices, as the `LogEig` and Bures distances.

***Definition** 15 (`Rank` distance):* Let $\mathbf{S}_a$ and $\mathbf{S}_b$ be the GSOs of the graphs with n nodes, $\mathcal{G}_a$ and $\mathcal{G}_b$, respectively. The `Rank` distance between $\mathcal{G}_a$ and $\mathcal{G}_b$, $d_{\mathrm{rk}}(\mathcal{G}_a, \mathcal{G}_b) : \mathscr{F} \times \mathscr{F} \to \mathbb{Z}_{\geq 0}$ is defined as

$$d_{\mathrm{rk}}(\mathcal{G}_a, \mathcal{G}_b) = \mathrm{rank}(\mathbf{S}_a - \mathbf{S}_b).$$

◇

Note that as the previous measures, the `Rank` distance is a metric, since it satisfies the three axioms in Definition 9.

It is easy to verify that if $\mathbf{S}_a = \mathbf{L}_a$, $\mathbf{S}_b = \mathbf{L}_b$ (i.e. the GSO is the Laplacian matrix), $d_{\mathrm{rk},\mathbf{L}}(\mathcal{G}_a, \mathcal{G}_b) \in [0, n-1]$, while if $\mathbf{S}_a = \mathbf{A}_a$, $\mathbf{S}_b = \mathbf{A}_b$ then $d_{\mathrm{rk},\mathbf{A}}(\mathcal{G}_a, \mathcal{G}_b) \in [0, n]$. If two graphs are isomorphic, a tighter upper bound can be found.

***Proposition** 2 (`Rank` distance of isomorphic graphs):* If $\mathcal{G}_a \simeq \mathcal{G}_b$, then $d_{\mathrm{rk},\mathbf{A}}(\mathcal{G}_a, \mathcal{G}_b) \in [0, 2\,\mathrm{rank}(\mathbf{A}_a)]$.

Proof: If $\mathcal{G}_a$ and $\mathcal{G}_b$ are isomorphic, from Definitions 15 and 8, $d_{\mathrm{rk}}(\mathcal{G}_a, \mathcal{G}_b) = \mathrm{rank}(\mathbf{A}_a - \mathbf{A}_b) = \mathrm{rank}(\mathbf{A}_a - \mathbf{P}\,\mathbf{A}_a\mathbf{P}^{-1})$. By using [27, Fact 2.10.27] and by noticing that $\mathrm{rank}(\mathbf{P}\,\mathbf{A}_a\mathbf{P}^{-1}) = \mathrm{rank}(\mathbf{A}_a)$, the statement follows. ■

IV. Numerical experiments

In this section, we start by comparing the graph distance measures introduced in Sect. III, on a simple example. In Sect. IV-B, we apply them to the 2D topological-map matching problem in robotics, using a publicly-available dataset. Finally, in Sect. IV-C, we study the impact of spectral sparsification of a large graph, on the proposed metrics.

A. Example

Consider the seven graphs with four nodes shown in Fig. 1. Note that $\mathcal{G}_a = P_4$ (path graph), $\mathcal{G}_f = K_4$ (complete graph), and that two pairs of graphs are isomorphic: $\mathcal{G}_a \simeq \mathcal{G}_g$ and $\mathcal{G}_b \simeq \mathcal{G}_c$. Let $\mathbb{D}_{\mathrm{GED}}$, $\mathbb{D}_{\mathrm{F}}$, $\mathbb{D}_{\log}$, $\mathbb{D}_{\mathrm{B}}$, $\mathbb{D}_{\mathrm{rk},\mathbf{L}}$ and $\mathbb{D}_{\mathrm{rk},\mathbf{A}}$ be the 7×7 distance matrices corresponding to the graphs $\mathcal{G}_a$, $\mathcal{G}_b$, $\ldots$, $\mathcal{G}_g$, relative to the five metrics considered in Sect. III. We computed $\mathbb{D}_{\mathrm{GED}}$ with the `graph_edit_distance` function of NetworkX, a Python package for the analysis of complex networks [32]. This function implements the exact GED algorithm proposed in [33]. We obtained:

$$\mathbb{D}_{\mathrm{GED}} = \begin{bmatrix} 0 & 1 & 1 & 1 & 2 & 3 & 0 \\ \bullet & 0 & 0 & 2 & 1 & 2 & 1 \\ \bullet & \bullet & 0 & 2 & 1 & 2 & 1 \\ \bullet & \bullet & \bullet & 0 & 1 & 2 & 1 \\ \bullet & \bullet & \bullet & \bullet & 0 & 1 & 2 \\ \bullet & \bullet & \bullet & \bullet & \bullet & 0 & 3 \\ \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & 0 \end{bmatrix},$$

$$\mathbb{D}_{\mathrm{F}} = \begin{bmatrix} 0 & 0.685 & 0.685 & 0.820 & 1.039 & 1.132 & 2.000 \\ \bullet & 0 & 0.831 & 0.929 & 0.647 & 0.801 & 1.537 \\ \bullet & \bullet & 0 & 0.929 & 0.647 & 0.801 & 1.537 \\ \bullet & \bullet & \bullet & 0 & 1.121 & 0.817 & 1.871 \\ \bullet & \bullet & \bullet & \bullet & 0 & 0.517 & 1.039 \\ \bullet & \bullet & \bullet & \bullet & \bullet & 0 & 1.132 \\ \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & 0 \end{bmatrix},$$

$$\mathbb{D}_{\log} = \begin{bmatrix} 0 & 1.099 & 1.099 & 1.386 & 1.550 & 2.048 & 1.863 \\ \bullet & 0 & 1.361 & 1.087 & 0.981 & 1.416 & 1.633 \\ \bullet & \bullet & 0 & 1.087 & 0.981 & 1.416 & 1.633 \\ \bullet & \bullet & \bullet & 0 & 0.985 & 0.980 & 1.910 \\ \bullet & \bullet & \bullet & \bullet & 0 & 0.693 & 1.550 \\ \bullet & \bullet & \bullet & \bullet & \bullet & 0 & 2.049 \\ \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & 0 \end{bmatrix},$$

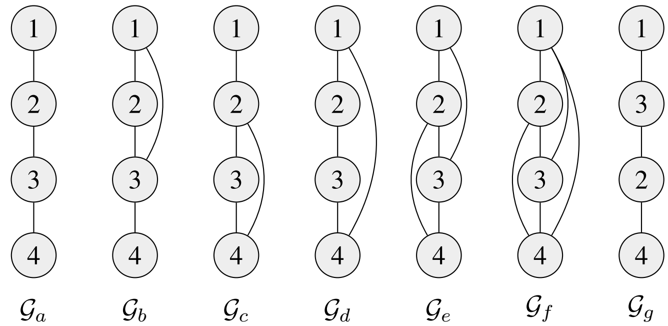


Fig. 1: Seven graphs with four nodes: $\mathcal{G}_b$, $\mathcal{G}_c$, $\ldots$, $\mathcal{G}_f$ are upgrowths of $\mathcal{G}_a$ obtained by edge insertion from P_4.

$$\mathbb{D}_{\mathrm{B}} = \begin{bmatrix} 0 & 0.680 & 0.680 & 0.787 & 0.981 & 1.375 & 1.035 \\ \bullet & 0 & 0.932 & 0.786 & 0.671 & 1.035 & 0.996 \\ \bullet & \bullet & 0 & 0.786 & 0.671 & 1.035 & 0.996 \\ \bullet & \bullet & \bullet & 0 & 0.825 & 0.828 & 1.231 \\ \bullet & \bullet & \bullet & \bullet & 0 & 0.586 & 0.981 \\ \bullet & \bullet & \bullet & \bullet & \bullet & 0 & 1.375 \\ \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & 0 \end{bmatrix},$$

$$\mathbb{D}_{\mathrm{rk},\mathbf{L}} = \begin{bmatrix} 0 & 1 & 1 & 1 & 2 & 3 & 2 \\ \bullet & 0 & 2 & 2 & 1 & 2 & 3 \\ \bullet & \bullet & 0 & 2 & 1 & 2 & 3 \\ \bullet & \bullet & \bullet & 0 & 3 & 2 & 2 \\ \bullet & \bullet & \bullet & \bullet & 0 & 1 & 2 \\ \bullet & \bullet & \bullet & \bullet & \bullet & 0 & 3 \\ \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & 0 \end{bmatrix},$$

$$\mathbb{D}_{\mathrm{rk},\mathbf{A}} = \begin{bmatrix} 0 & 2 & 2 & 2 & 4 & 4 & 2 \\ \bullet & 0 & 4 & 2 & 2 & 2 & 4 \\ \bullet & \bullet & 0 & 2 & 2 & 2 & 4 \\ \bullet & \bullet & \bullet & 0 & 4 & 4 & 3 \\ \bullet & \bullet & \bullet & \bullet & 0 & 2 & 4 \\ \bullet & \bullet & \bullet & \bullet & \bullet & 0 & 4 \\ \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & 0 \end{bmatrix},$$

where only the upper triangular part of the matrices is reported, since the matrices are symmetric by definition.

By comparing the first row of $\mathbb{D}_{\mathrm{GED}}$, $\mathbb{D}_{\mathrm{rk},\mathbf{L}}$ and $\mathbb{D}_{\mathrm{rk},\mathbf{A}}$, we can see that like the GED, the `Rank` distance is not able to discriminate between $(\mathcal{G}_a, \mathcal{G}_b)$, $(\mathcal{G}_a, \mathcal{G}_c)$ and $(\mathcal{G}_a, \mathcal{G}_d)$. If the adjacency matrix is considered, the `Rank` distance is not even able to tell $(\mathcal{G}_a, \mathcal{G}_e)$ and $(\mathcal{G}_a, \mathcal{G}_f)$ apart. In contrast, the other three metrics, Frobenius, `logEig` and Bures, cannot differentiate between $(\mathcal{G}_a, \mathcal{G}_b)$ and $(\mathcal{G}_a, \mathcal{G}_c)$, only. We can also notice that the largest Frobenius norm, $[\mathbb{D}_{\mathrm{F}}]_{1,7} = 2.000$, is obtained with the pair of isomorphic graphs $\mathcal{G}_a$ and $\mathcal{G}_g$. This does not happen with the other metrics.

For a more quantitative analysis, the *Mantel test*, a special type of randomization test, was used [34], [35]. This non-parametric test allows to compute the correlation between two distance matrices. More specifically, it calculates the significance of the correlation through M permutations of the rows and columns of one of the input distance matrices. In principle, any correlation coefficient could be considered, but the Pearson product-moment correlation coefficient $r \in [-1, 1]$ is typically chosen. A coefficient r being close to -1 $(+1)$ indicates strong negative (positive) correlation, while $r = 0$ means no correlation.

By taking the GED as our "baseline" and setting $M = 5000$, we obtained the following values for the Pearson coefficient r: $\mathrm{Mntl}(\mathbb{D}_{\mathrm{GED}}, \mathbb{D}_{\mathrm{F}}) = -0.1794$, $\mathrm{Mntl}(\mathbb{D}_{\mathrm{GED}}, \mathbb{D}_{\log}) = 0.2711$, $\mathrm{Mntl}(\mathbb{D}_{\mathrm{GED}}, \mathbb{D}_{\mathrm{B}}) = 0.4728$,

$\text{Mntl}(\mathbb{D}_{\text{GED}}, \mathbb{D}_{\text{rk},\mathbf{L}}) = 0.3691$ and $\text{Mntl}(\mathbb{D}_{\text{GED}}, \mathbb{D}_{\text{rk},\mathbf{A}}) = 0.2390$. These results indicate that $\mathbb{D}_{\text{GED}}$ has a strong relationship with $\mathbb{D}_{\text{B}}$ ($r = 0.4728$): the correlation between $\mathbb{D}_{\text{GED}}$ and $\mathbb{D}_{\text{rk},\mathbf{L}}$, $\mathbb{D}_{\log}$, $\mathbb{D}_{\text{rk},\mathbf{A}}$ (in decreasing order), is significant as well. The proposed graph distance measures are thus valid and easy-to-use alternatives to the GED. Moreover, the discriminating power of `logEig` and Bures distances is superior to that of the GED.

B. Matching of 2D topological maps in robotics

The proposed metrics have been also compared using the graphs from the COLD-TopoMaps dataset [36], [20]. The dataset contains 99 topological maps captured on 11 floors of 3 buildings in 3 cities in Germany and Sweden: Freiburg, Saarbrücken and Stockholm. In our study, we considered 38 pairs of 2D topological maps: 13 pairs with 90 and 92 nodes in Freiburg, 13 pairs with 52 and 53 nodes in Saarbrücken, and 12 pairs with 107, 122 and 191 nodes in Stockholm (see the table in Fig. 2). For illustration, in Fig. 2, the topological map `seq1_sunny4` in Freiburg (92 nodes and 161 edges) is displayed. For the color convention of the nodes, the reader is referred to [20] (e.g. orange and dark-green nodes refer to corridors and bathrooms, respectively). The table in Fig. 3 reports the Pearson coefficients of the Mantel test with $M = 5000$, between the distance matrix $\mathbb{D}_{\text{GED}}$ and $\mathbb{D}_{\text{F}}$, $\mathbb{D}_{\log}$, $\mathbb{D}_{\text{B}}$, $\mathbb{D}_{\text{rk},\mathbf{L}}$, $\mathbb{D}_{\text{rk},\mathbf{A}}$. As in Sect. IV-A, the GED was computed with NetworkX's `graph_edit_distance`. To ensure completion, a timeout of 180 seconds, reached in all trials, was considered (after timeout has expired, the current best GED is returned).

Mntl($\mathbb{D}_{\text{GED}}, \cdot$)	$\mathbb{D}_{\text{F}}$	$\mathbb{D}_{\log}$	$\mathbb{D}_{\text{B}}$	$\mathbb{D}_{\text{rk},\mathbf{L}}$	$\mathbb{D}_{\text{rk},\mathbf{A}}$
Freiburg, $n = 90$	0.9973	0.9827	0.9985	0.9848	0.9753
Freiburg, $n = 92$	0.9130	0.9588	0.9556	0.9395	0.9229
Saarbrücken, $n = 52$	0.9976	0.9813	0.9955	0.9970	0.9728
Saarbrücken, $n = 53$	0.8392	0.9545	0.9395	0.7285	0.7364
Stockholm, $n = 107$	0.9826	0.9366	0.9636	0.9658	0.9892
Stockholm, $n = 122$	0.0628	0.6496	0.5960	0.3785	0.3545
Stockholm, $n = 191$	0.9332	0.9284	0.9183	0.9793	0.9702
Mean value	0.8180	0.9131	0.9183	0.8520	0.8475

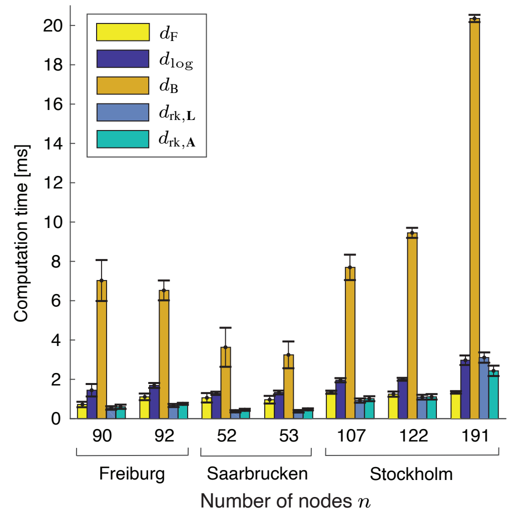


Fig. 3: (Top) Pearson coefficients r from the Mantel tests between $\mathbb{D}_{\text{GED}}$ and $\mathbb{D}_{\text{F}}$, $\mathbb{D}_{\log}$, $\mathbb{D}_{\text{B}}$, $\mathbb{D}_{\text{rk},\mathbf{L}}$, $\mathbb{D}_{\text{rk},\mathbf{A}}$ in the COLD-TopoMaps dataset. (Bottom) Mean and standard deviation of the computation time of the five metrics (milliseconds).

#	Map 1	Map 2	n
Freiburg			
1	seq1_sunny1	seq1_sunny3	90
2	seq1_sunny1	seq3_sunny1	90
3	seq1_sunny3	seq3_sunny1	90
4	seq1_cloudy2	seq1_night2	92
5	seq1_cloudy2	seq1_night3	92
6	seq1_cloudy2	seq1_sunny4	92
7	seq1_cloudy2	seq3_sunny3	92
8	seq1_night2	seq1_night3	92
9	seq1_night2	seq1_sunny4	92
10	seq1_night2	seq3_sunny3	92
11	seq1_night3	seq1_sunny4	92
12	seq1_night3	seq3_sunny3	92
13	seq1_sunny4	seq3_sunny3	92
Saarbrücken			
1	seq3_cloudy1	seq3_night1	52
2	seq3_cloudy1	seq3_night3	52
3	seq3_night1	seq3_night3	52
4	seq3_cloudy2	seq3_cloudy5	53
5	seq3_cloudy2	seq3_night2	53
6	seq3_cloudy2	seq3_sunny1	53
7	seq3_cloudy2	seq3_sunny3	53
8	seq3_cloudy5	seq3_night2	53
9	seq3_cloudy5	seq3_sunny1	53
10	seq3_cloudy5	seq3_sunny3	53
11	seq3_night2	seq3_sunny1	53
12	seq3_night2	seq3_sunny3	53
13	seq3_sunny1	seq3_sunny3	53
Stockholm			
1	floor6_base_cloudy_c	floor6_base_night_b	107
2	floor6_base_cloudy_c	floor6_base_sunny_a2	107
3	floor6_base_night_b	floor6_base_sunny_a2	107
4	floor4_cloudy_a1	floor4_night_a2	122
5	floor4_cloudy_a1	floor4_night_b	122
6	floor4_cloudy_a1	floor5_night_c	122
7	floor4_night_a2	floor4_night_b	122
8	floor4_night_a2	floor5_night_c	122
9	floor4_night_b	floor5_night_c	122
10	floor7_cloudy_a2	floor7_cloudy_b	191
11	floor7_cloudy_a2	floor7_night_a2	191
12	floor7_cloudy_b	floor7_night_a2	191

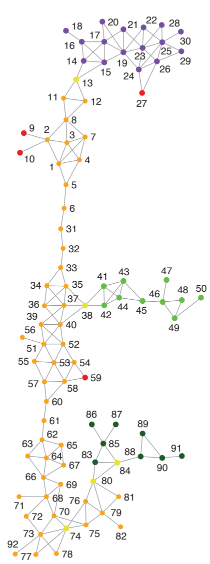


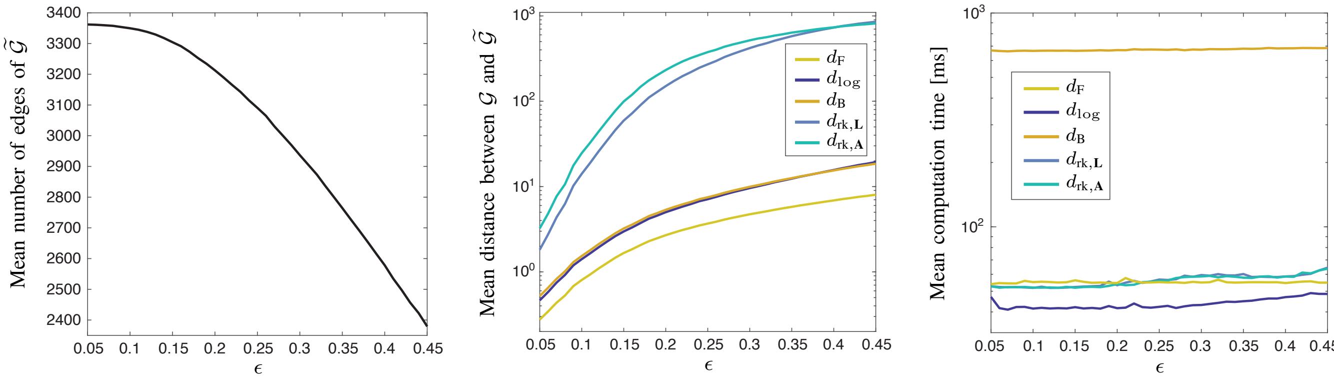
Fig. 2: (Left) Pairs of topological maps from the COLD-TopoMaps dataset, considered in Sect. IV-B. (Right) Example of a map with 92 nodes in Freiburg: `seq1_sunny4`.

In line with the results at the end of Sect. IV-A, the analysis of the mean values in the table in Fig. 3 shows that $\mathbb{D}_{\text{GED}}$ has a strong relationship with $\mathbb{D}_{\text{B}}$ and $\mathbb{D}_{\log}$: the correlation between $\mathbb{D}_{\text{GED}}$ and $\mathbb{D}_{\text{rk},\mathbf{L}}$, $\mathbb{D}_{\text{rk},\mathbf{A}}$ is strong as well, and superior to that between $\mathbb{D}_{\text{GED}}$ and $\mathbb{D}_{\text{F}}$.

The histogram in Fig. 3 reports the mean and standard deviation of the computation time of the five metrics (in milliseconds), over pairs of maps with the same number of nodes in Freiburg, Saarbrücken, and Stockholm. We ran our tests in Matlab R2017a, on a MacBook Pro with 2.3 GHz Quad-Core Intel Core i7 processor and 32 GB of RAM. For a fair comparison, the adjacency matrix $\mathbf{A}$ and the Laplacian $\mathbf{L}$ of all graphs, have been precomputed and stored in memory. From the histogram, we can see that the computation time of `LogEig` and `rank` is small (less than 4 ms), in all cases. In view of their speed and accuracy, these distances are thus well suited for complex robotic exploration and mapping tasks. Finally, unlike the GED, whose calculation is exceedingly slow, the Bures distance can still be computed in real time (for maps with up to 200 nodes).

C. Distance between a graph and its spectral sparsifier

To study the impact of spectral sparsification on the new distance measures, we considered a random graph $\mathcal{G}$ with $n = 1000$ nodes located in $[0,\,1] \times [0,\,1]$ m^2. This family of

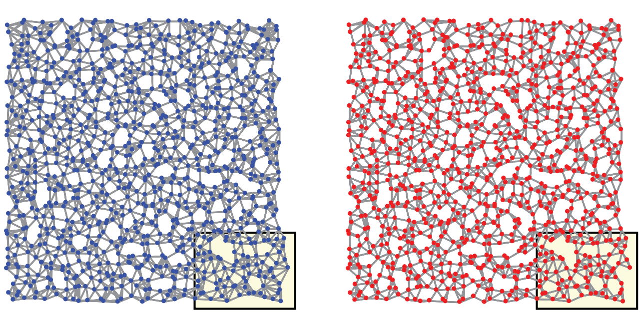


(a) (b) (c)

Fig. 4: Spectral sparsification of a random graph with 1000 nodes: the plots report statistics over 50 trials. (a) Mean number of edges of $\widetilde{\mathcal{G}}$ as ϵ grows; (b) Mean distance between $\mathcal{G}$ and $\widetilde{\mathcal{G}}$ as ϵ grows, according to the five metrics, and (c) corresponding mean computation time in milliseconds. In (b) and (c), a logarithmic scale is used on the vertical axis.

graphs is well known in robotics and it has been already used, e.g. for cooperative exploration [37]. We generated $\mathcal{G}$ with the `gsp_sensor` function (`param.distribute = 1`) of GSPBox toolbox [38] (see Fig. 5(a)). We sparsified $\mathcal{G}$ with Spielman & Srivastava's algorithm, implemented in the `gsp_graph_sparsify` function. According to [39, Th. 1], the sparsification parameter $\epsilon = \sigma - 1$ (cf. Definition 7) must satisfy the condition $0.0316 = 1/\sqrt{n} < \epsilon \leq 1$ (note that the larger ϵ, the sparser the graph). We varied ϵ between 0.05 and 0.45, with step size 0.01. An example of spectral sparsifier $\widetilde{\mathcal{G}}$ with $\epsilon = 0.45$ is shown in Fig. 5(b).

Fig. 4 reports statistical results over 50 trials (i.e. random graph $\mathcal{G}$ was generated 50 times). Graph $\mathcal{G}$ has 3364 ± 12 edges and Fig. 4(a) shows how the mean number of edges of the spectral sparsifier $\widetilde{\mathcal{G}}$ decreases as ϵ grows. Fig. 4(b) reports the mean distance between $\mathcal{G}$ and $\widetilde{\mathcal{G}}$ as ϵ grows, for the five metrics (a logarithmic scale is used on the vertical axis). Note that d_{GED} is not shown in Fig. 4(b), since NetworkX's `graph_edit_distance` failed to deliver a result for any ϵ. The mean computation time relative to Fig. 4(b), is shown in Fig. 4(c). As expected, in Fig. 4(b), the larger the value of ϵ, the larger the five graph distances.



(a) (b)

Fig. 5: Spectral sparsification of a random graph with 1000 nodes. (a) Original graph $\mathcal{G}$; (b) Example of spectral sparsifier $\widetilde{\mathcal{G}}$ with $\epsilon = 0.45$. The box in the bottom right corner highlights the difference between $\mathcal{G}$ and $\widetilde{\mathcal{G}}$.

In spite of the different scale, the five curves have a similar profile: in particular, $d_{\log}$ almost coincides with d_{B} (cf. the Definitions 12 and 14). On the other hand (see Fig. 4(c)), the mean computation time of the `Rank` distance and Frobenius norm is comparable (between 52 and 65 ms), while that of the `LogEig` distance ranges between 41 and 49 ms, thus delivering the best performance. As already noted in Sect. IV-B, the computational cost of the Bures distance is much higher: the runtime is between 662 and 686 ms, i.e. it is 10 times larger than that of the other four metrics.

V. Conclusion and Future Work

In this paper, we have proposed three new graph distance measures (`LogEig`, Bures and `Rank`), which satisfy the axioms for a metric. Topological-map matching in robotics served as a benchmark problem, revealing that the new measures compare favorably with other existing metrics (graph edit distance, Frobenius norm), for pairs of complex graphs in the COLD-TopoMaps dataset. The new distance measures have also been applied to graphs with up to one thousand nodes to study the effect of spectral sparsification. `LogEig` and `Rank` stood out, ensuring real-time performance on a standard laptop.

While the results presented in this article are promising, several open problems deserve further attention. A practical limitation of the proposed metrics is that the two graphs must have the same number of nodes and must be connected (with the exception of `Rank`). Unfortunately, if a topological map is compared to the one incrementally built by a robot exploring an unknown environment, this is rarely the case. In the future, we will then turn our attention to the more challenging *approximate* graph matching problem [11]. For time-varying maps, the *sum-rank metric* [31], a generalization of rank metric, represents an interesting option. The case of directed graphs will be investigated as well. Finally, to broaden the scope of this work, we plan to measure the dissimilarity between maps generated from 3D point clouds [15] and to take advantage of our metrics for fast loop-closure detection in (topological) SLAM.

References

[1] P. Wills and F.G Meyer. Metrics for graph comparison: A practitioner's guide. *PLoS One*, 15(2):e0228728, 2020.

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

[3] M. Neuhaus and H. Bunke. *Bridging the Gap between Graph Edit Distance and Kernel Machines*, volume 68 of *Machine Perception and Artificial Intelligence*. World Scientific, 2007.

[4] D.B. Blumenthal, N. Boria, J. Gamper, S. Bougleux, and L. Brun. Comparing heuristics for graph edit distance computation. *VLDB J.*, 29(1):419–458, 2020.

[5] D.K. Hammond, Y. Gur, and C.R. Johnson. Graph diffusion distance: A difference measure for weighted graphs based on the graph Laplacian exponential kernel. In *Proc. IEEE Global Conf. Signal Inf. Process.*, pages 419–422, 2013.

[6] H. Bunke and K. Shearer. A graph distance metric based on maximal common subgraph. *Pattern Recognit. Lett.*, 19(3–4):255–259, 1998.

[7] Y. Shimada, Y. Hirata, T. Ikeguchi, and K. Aihara. Graph distance for complex networks. *Sci. Rep.*, 6(1):34944, 2016.

[8] A.A. Benczúr and D.R. Karger. Approximating $s-t$ Minimum Cuts in $\tilde{O}(n^2)$ time. In *Proc. 28th Annual ACM Symp. Theory Comput.*, pages 47–55, 1996.

[9] J. Batson, D.A. Spielman, N. Srivastava, and S.H. Teng. Spectral Sparsification of Graphs: Theory and Algorithms. *Commun. ACM*, 56(8):87–94, 2013.

[10] Y. Bai, H. Ding, S. Bian, T. Chen, Y. Sun, and W. Wang. SimGNN: A Neural Network Approach to Fast Graph Similarity Computation. In *Proc. 12th ACM Int. Conf. Web Search Data Min.*, pages 384–392, 2019.

[11] D. Conte, P. Foggia, C. Sansone, and M. Vento. Thirty Years of Graph Matching in Pattern Recognition. *Int. J. Pattern Recognit. Artif. Intell.*, 18(3):265–298, 2004.

[12] M. Gori, M. Maggini, and L. Sarti. Exact and Approximate Graph Matching Using Random Walks. *IEEE Trans. Pattern Anal.*, 27(7):1100–1111, 2005.

[13] L. Livi and A. Rizzi. The graph matching problem. *Pattern Anal. Appl.*, 16:253–283, 2013.

[14] W.H. Huang and K.R. Beevers. Topological Map Merging. *Int. J. Robot. Res.*, 24(8):601–613, 2005.

[15] M. Spencer, R. Sawtell, and S. Kitchen. Sphere-Graph: A Compact 3D Topological Map for Robotic Navigation and Segmentation of Complex Environments. *IEEE Robot. Autom. Lett.*, 9(3):2567–2574, 2024.

[16] R. Bhatia, J. Tanvi, and L. Yongdo. On the Bures-Wasserstein distance between positive definite matrices. *Expo. Math.*, 37(2):165–191, 2019.

[17] G. Mateos, S. Segarra, A.G. Marques, and A. Ribeiro. Connecting the Dots: Identifying Network Structure via Graph Signal Processing. *IEEE Signal Process. Mag.*, 36(3):16–43, 2019.

[18] D.A. Spielman and S.-H. Teng. Spectral Sparsification of Graphs. *SIAM J. Comput.*, 40(4):981–1025, 2011.

[19] Y. Tian and J.P. How. Spectral Sparsification for Communication-Efficient Collaborative Rotation and Translation Estimation. *IEEE Trans. Robot.*, 40:257–276, 2024.

[20] K. Zheng, A. Pronobis, and R.P.N. Rao. Learning Graph-Structured Sum-Product Networks for Probabilistic Semantic Maps. In *Proc. 32nd AAAI Conf. Artif. Intell.*, pages 4547–4555, 2018.

[21] L. Ruiz, L.F.O. Chamon, and A. Ribeiro. Transferability Properties of Graph Neural Networks. *IEEE Trans. Signal Process.*, 71:3474–3489, 2023.

[22] J. Lerouge, Z. Abu-Aisheh, R. Raveaux, P. Héroux, and S. Adam. New binary linear programming formulation to compute the graph edit distance. *Pattern Recognit.*, 72:254–265, 2017.

[23] S. Bougleux, L. Brun, V. Carletti, P. Foggia, B. Gaüzère, and M. Vento. Graph edit distance as a quadratic assignment problem. *Pattern Recognit. Lett.*, 87:38–46, 2017.

[24] L. Quan, L. Yin, T. Zhang, M. Wang, R. Wang, S. Zhong, X. Zhou, Y. Cao, C. Xu, and F. Gao. Robust and Efficient Trajectory Planning for Formation Flight in Dense Environments. *IEEE Trans. Robot.*, 39(6):4785–4804, 2023.

[25] H. Petric Maretic, M. El Gheche, G. Chierchia, and P. Frossard. GOT: An Optimal Transport framework for Graph comparison. In *Proc. 33rd Adv. Neur. Inform. Proc. Syst.*, volume 32, 2019.

[26] W. Förstner and B. Moonen. A Metric for Covariance Matrices. In E.W. Grafarend, F.W. Krumm, and V.S. Schwarze, editors, *Geodesy-the Challenge of the 3rd Millennium*, pages 299–309. Springer, 2003.

[27] D.S. Bernstein. *Matrix Mathematics: Theory, Facts, and Formulas*. Princeton University Press, 2nd edition, 2009.

[28] G. Peyré and M. Cuturi. Computational Optimal Transport: With Applications to Data Science. *Foundations and Trends in Machine Learning*, 11(5–6):355–607, 2019.

[29] I. Haasler and P. Frossard. Bures-Wasserstein Means of Graphs. In *Proc. 27th Int. Conf. Artif. Intel. Stat.*, volume 238, pages 1873–1881, 2024.

[30] H. Bartz, L. Holzbaur, H. Liu, S. Puchinger, J. Renner, and A. Wachter-Zeh. Rank-Metric Codes and Their Applications. In *Foundations and Trends® in Communications and Information Theory*, volume 19, pages 390–546. Now publishers, 2022.

[31] U. Martínez-Peñas, M. Shehadeh, and F. R. Kschischang. Codes in the Sum-Rank Metric: Fundamentals and Applications. In *Foundations and Trends® in Communications and Information Theory*, volume 19, pages 814–1031. Now publishers, 2022.

[32] A.A. Hagberg, D.A. Schult, and P.J. Swart. Exploring network structure, dynamics, and function using NetworkX. In *Proc. 7th Python in Science Conf.*, 2008. [web] `https://networkx.org`.

[33] Z. Abu-Aisheh, R. Raveaux, J.-Y. Ramel, and P. Martineau. An Exact Graph Edit Distance Algorithm for Solving Pattern Recognition Problems. In *Proc. 4th Int. Conf. Pattern Recogn. Appl. Meth.*, pages 271–278, 2015.

[34] N. Mantel. The Detection of Disease Clustering and a Generalized Regression Approach. *Cancer Res*, 27(2_Part_1):209–220, 1967.

[35] R.R. Sokal and F.J. Rohlf. *Biometry*. W.H. Freeman, 4th edition, 2011.

[36] A. Pronobis and B. Caputo. COLD: The CoSy Localization Database. *Int. J. Robot. Res.*, 28(5):588–594, 2009.

[37] A. Franchi, L. Freda, G. Oriolo, and M. Vendittelli. The Sensor-based Random Graph Method for Cooperative Robot Exploration. *IEEE/ASME Trans. Mechatron.*, 14(2):163–175, 2009.

[38] N. Perraudin, J. Paratte, D. Shuman, L. Martin, V. Kalofolias, P. Vandergheynst, and D.K. Hammond. GSPBOX: A toolbox for signal processing on graphs. *arXiv:1408.5781*, 2014. [web]: `https://epfl-lts2.github.io/gspbox-html`.

[39] D.A. Spielman and N. Srivastava. Spectral Sparsification by Effective Resistances. *SIAM J. Comput.*, 40(6):1913–1926, 2011.