

A Functional Representation for Graph Matching

Fu-Dong Wang, Nan Xue, *Student Member, IEEE*, Yipeng Zhang,
Gui-Song Xia, *Senior Member, IEEE*, and Marcello Pelillo, *Fellow, IEEE*

Abstract—Graph matching is an important and persistent problem in computer vision and pattern recognition for finding node-to-node correspondence between graphs. However, graph matching that incorporates pairwise constraints can be formulated as a quadratic assignment problem (QAP), which is NP-complete and results in intrinsic computational difficulties. This paper presents a functional representation for graph matching (FRGM) that aims to provide more geometric insights on the problem and reduce the space and time complexities. To achieve these goals, we represent each graph by a linear function space equipped with a functional such as inner product or metric, that has an explicit geometric meaning. Consequently, the correspondence matrix between graphs can be represented as a linear representation map. Furthermore, this map can be reformulated as a new parameterization for matching graphs in Euclidean space such that it is consistent with graphs under rigid or nonrigid deformations. This allows us to estimate the correspondence matrix and geometric deformations simultaneously. We use the representation of edge-attributes rather than the affinity matrix to reduce the space complexity and propose an efficient optimization strategy to reduce the time complexity. The experimental results on both synthetic and real-world datasets show that the FRGM can achieve state-of-the-art performance.

Index Terms—Graph matching, functional representation, Frank-Wolfe method, geometric deformation.

1 INTRODUCTION

Graph matching (GM) is widely used to find node-to-node correspondence [1], [2] between graph-structured data in many computer vision and pattern recognition tasks, such as shape matching and retrieval [3], [4], object categorization [5], person re-identification [6], [7], action recognition [8], and structure from motion [9], to name a few. In these applications, real-world data are generally represented as abstract graphs equipped with node attributes (*e.g.*, *SIFT descriptor* [10], *shape context* [3]) and edge attributes (*e.g.*, *relationships between nodes*). In this way, many GM methods have been proposed based on the assumption that nodes or edges with more similar attributes are more likely to be matched. Generally, GM methods construct objective functions w.r.t. the varying correspondence to measure similarities (or dissimilarities) between nodes and edges. Then, they maximize (or minimize) the objective functions to pursue an optimal correspondence that achieves maximal (or minimal) total similarities (or dissimilarities) between two graphs. In the literature, an objective function is generally composed of unary [3], pairwise [11], [12] or higher-order [13], [14] potentials. In practice, matching graphs using only unary potential (node attributes) might lead to undesirable results due to the insufficient discriminability of node attributes. Therefore, pairwise or higher-order potentials are often integrated to better preserve the structural alignments between graphs.

Although the past decades have witnessed remarkable progresses in GM [1], there are still many challenges with respect to both computational difficulty and formulation expression. Specifically, as widely used, GM that incorporates pairwise constraints

can be formulated as a quadratic assignment problem (QAP) [15], among which Lawler's QAP [16] and Koopmans-Beckmann's QAP [17] are two common formulations. However, due to the NP-complete [18] nature of QAP, only approximate solutions are available in polynomial time. In practice, solving GM problems with pairwise constraints often encounters intrinsic difficulties due to the high computational complexity in space or time. For GM methods that apply Lawler's QAP, the affinity matrix results in high space complexity $O(m^2n^2)$ w.r.t. the numbers of nodes in two graphs (m, n). For GM methods that aim to solve the objective functions with discrete binary solutions through a gradually convex-concave continuous optimization strategy, the verbose iterations result in high time complexity. Restricted by these limitations, only graphs with dozens of nodes can be handled by these methods in practice.

In addition to the computational difficulties, how to formulate the GM model for real applications is also important. Representing real-world data in the conventional graph model can provide some generalities for the general GM methods mentioned above. However, their formulations can neither reflect the geometric nature of real-world data nor handle graphs with geometric deformations (rigid or nonrigid). For example, when the edge attributes of graphs are computed as distances [19], [20], [21] on some explicit or implicit spaces that contain the real-world data, the formulations of the original GM methods that define objective functions in the form of Lawler's or Koopmans-Beckmann's QAP ignore the geometric properties behind these data. They can only achieve generality and ignore the geometric nature of real-world data. For graphs with rigid or nonrigid geometric deformations [19], the original GM methods cannot compatibly handle the two tasks that estimate both correspondence and deformation parameters because they can hardly provide the correspondence a geometric interpretation that is naturally contained in the deformation.

Facing these issues, this paper introduces a new functional

F.-D. Wang, G.-S. Xia, N. Xue are with LIESMARS, Wuhan University, Wuhan, 430079, China. E-mail: {fudong-wang, guisong.xia, xuenan}@whu.edu.cn.
Y. Zhang is with the Department of Computer Science, Wuhan University, Wuhan, 430072, China. E-mail: zyp91@whu.edu.cn.
M. Pelillo is with Computer Vision Lab., Ca' Foscari University of Venice, 30172 Venezia Mestre, Italy. Email: pelillo@unive.it.
Corresponding author: Gui-Song Xia (guisong.xia@whu.edu.cn).

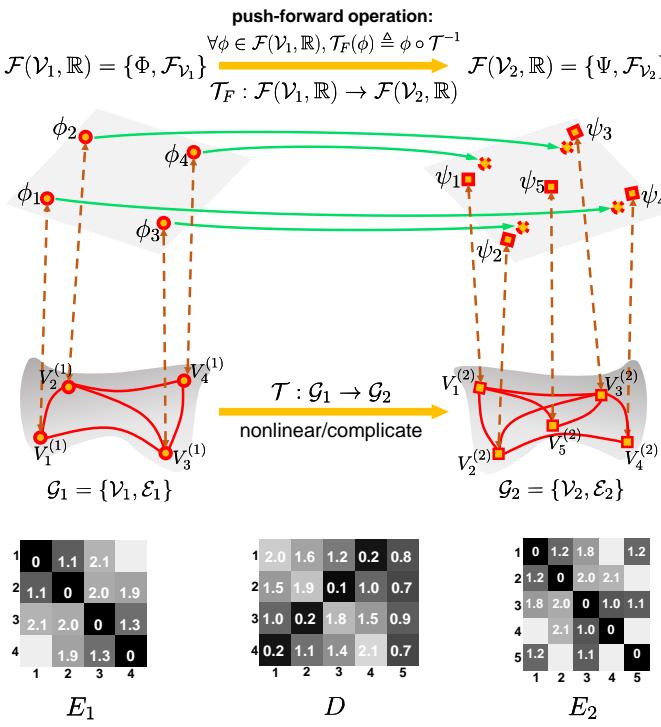


Fig. 1. FRGM: given two graphs G_1 and G_2 , we construct two function spaces $\mathcal{F}(\mathcal{V}_1, \mathbb{R})$ and $\mathcal{F}(\mathcal{V}_2, \mathbb{R})$ as representations, where Φ and Ψ are two sets of basis functions that represent the nodes \mathcal{V}_1 and \mathcal{V}_2 , and $\mathcal{F}_{\mathcal{V}_1}$ and $\mathcal{F}_{\mathcal{V}_2}$ are the inner product or metric that represent the edge attributes E_1 and E_2 . The matching between two graphs can be viewed as a transformation $\mathcal{T} : G_1 \rightarrow G_2$, which may be nonlinear and complicated. Fortunately, \mathcal{T} can be recovered from a linear functional: $\mathcal{T}_F : \mathcal{F}(\mathcal{V}_1, \mathbb{R}) \rightarrow \mathcal{F}(\mathcal{V}_2, \mathbb{R})$, which is induced from \mathcal{T} by the push-forward operation and represented by a linear functional representation map $\mathbf{P} \in \mathbb{R}^{m \times n}$. \mathbf{P} is exactly a correspondence between graphs. Based on the inner product or metric defined as $\mathcal{F}_{\mathcal{V}_2}$, each transformed node will lie closer to its correct match, as shown in matrix D . This property is helpful for improving the matching performance.

representation for graph matching (FRGM). The main idea is to represent the graphs and the node-to-node correspondence in linear functional representations for both general and Euclidean GM models. Specifically, for general GM, as shown in Fig. 1, given two undirected graphs, we can identically represent the node sets as linear function spaces, on which some specified functionals \mathcal{F}_V (e.g., inner product or metric) can be compatibly constructed to represent the edge attributes. Then, between the two function spaces, a functional \mathcal{T}_F induced by the push-forward operation is represented by a linear representation map \mathbf{P} , which is exactly the correspondence between graphs. With these concepts, our general GM algorithm is proposed by minimizing the objective function w.r.t. \mathbf{P} that measures the difference of graph attributes between graph G_1 and its transformed graph $\mathcal{T}(G_1)$. Namely, we want an optimal functional \mathcal{T}_F in the sense of preserving the inner product or metric. For the Euclidean GM in which the graphs are embedded in Euclidean space \mathbb{R}^d , the functional \mathcal{T}_F that plays the role of correspondence between graphs can be directly deduced on the background space \mathbb{R}^d . Due to the natural linearity of \mathbb{R}^d , \mathcal{T}_F can also be represented by a linear representation map \mathbf{P} , which is not only a parameterization for GM but also associative with geometric parameters for graphs under geometric deformations. A preliminary version of this work was presented in [22].

FRGM only needs to compute and store the edge attributes of graphs; thus its space complexity is $\mathbf{O}(n^2)$ (with $m \leq n$).

To reduce the time complexity, we first propose an optimization algorithm with time complexity $\mathbf{O}(n^3)$ based on the Frank-Wolfe method. Then, by taking advantage of the specified property of the relaxed feasible field, we improve the Frank-Wolfe method by an approximation that has a lower time complexity of $\mathbf{O}(mn)$.

The contributions of this paper can be distinguished in the following aspects:

- We introduce a new functional representation perspective that can bridge the gap between the formulation of general GM and the geometric nature behind the real-world data. This guides us in constructing more efficient objective functions and algorithms for general GM problem.
- For graphs embedded in Euclidean space, we extend the linear functional representation map as a new geometric parameterization that achieves compatibility with the geometric parameters of graphs. This helps to globally handle graphs with or without geometric deformations.
- We propose GM algorithms with low space complexity and time complexity by avoiding the use of an affinity matrix and by improving the optimization strategy. The proposed algorithms outperform the state-of-the-art methods in terms of both efficiency and accuracy.

The remainder of this paper is organized as follows. Sec. 2 presents the mathematical formulation and related work of GM. In Sec. 3, we demonstrate the functional representation for GM with general settings (FRGM-G) and the resulting algorithm. In Sec. 4–Sec. 5, we extend FRGM for matching graphs in Euclidean space (FRGM-E) with and without geometric deformations (FRGM-D), respectively. In Sec. 6, we present a numerical analysis of optimization strategy. Finally, we report the experimental results and analysis in Sec. 7 and conclude this paper in Sec. 8.

2 BACKGROUND AND RELATED WORK

This section first introduces the preliminaries and basic notations of GM and then it discusses some related works on GM. Tab. 1 summarizes some frequently used notations through this paper.

TABLE 1
Frequently used notations in our paper.

m, n	size of graph
$k = 1, 2$	index of graphs
$\mathcal{G}_k = \{\mathcal{V}_k, \mathcal{E}_k\}$	graph \mathcal{G}_k with nodes \mathcal{V}_k and edges \mathcal{E}_k
$\mathbf{v}_i^{(k)} \in \mathbb{R}^{d_v}$	node attributes vector of \mathcal{G}_k
$\mathbf{E}_k \in \mathbb{R}^{m \times m}$ (or $\mathbb{R}^{n \times n}$)	edge attributes matrix of \mathcal{G}_k
$\mathbf{K} \in \mathbb{R}^{mn \times mn}$	affinity matrix
$\mathbf{U} \in \mathbb{R}^{m \times n}$	node dissimilarity
\mathcal{P} (or $\hat{\mathcal{P}}$)	discrete (or relaxed) feasible field
$\mathbf{P} \in \mathcal{P}$ (or $\hat{\mathcal{P}}$)	correspondence matrix
$\mathcal{F}(\mathcal{V}_k, \mathbb{R})$	function space on \mathcal{G}_k
$\Phi = \{\phi_i\}_{i=1}^m$	basis functions of $\mathcal{F}(\mathcal{V}_1, \mathbb{R})$
$\Psi = \{\psi_j\}_{j=1}^n$	basis functions of $\mathcal{F}(\mathcal{V}_2, \mathbb{R})$
$\mathcal{F}_{\mathcal{V}_k}(\cdot, \cdot)$	functional on $\mathcal{F}(\mathcal{V}_k, \mathbb{R}) \times \mathcal{F}(\mathcal{V}_k, \mathbb{R})$
\mathcal{T}	mapping from \mathcal{G}_1 to \mathcal{G}_2
\mathcal{T}_F	functional from $\mathcal{F}(\mathcal{V}_1, \mathbb{R})$ to $\mathcal{F}(\mathcal{V}_2, \mathbb{R})$

2.1 Definition of GM Problem

An undirected graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ of size m is defined by a discrete set of nodes $\mathcal{V} = \{V_i\}_{i=1}^m$ and a set of undirected edges $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ such that $(V_{i_1}, V_{i_2}) = (V_{i_2}, V_{i_1})$. Generally, the edge of

graph \mathcal{G} is written as a symmetric edge indicator matrix (also denoted as) $\mathcal{E} \in \mathbb{R}^{m \times m}$, where $\mathcal{E}_{i_1 i_2} = 1$ if there is an edge between V_{i_1} and V_{i_2} , and $\mathcal{E}_{i_1 i_2} = 0$ otherwise. An important generalization is the weighted graph defined by the association of non-negative real values $\mathcal{E}_{i_1 i_2}$ to graph edges, and \mathcal{E} is called adjacency weight matrix. We assume graphs with no self-loop in this paper, i.e., $\mathcal{E}_{ii} = 0$.

In many real applications, graph \mathcal{G} is associated with node and edge attributes expressed as scalars or vectors. For an attribute graph \mathcal{G} , we denote $\mathbf{v}_i \in \mathbb{R}^{d_v}$ as the node attribute of V_i and $\mathbf{e}_{i_1 i_2} \in \mathbb{R}^{d_e}$ as the edge attribute of $\mathcal{E}_{i_1 i_2}$. Typically, an edge attribute matrix $\mathbf{E} \in \mathbb{R}^{m \times m}$ will be calculated by some user-specified functions such as $\mathbf{E}_{i_1 i_2} = \phi(\mathbf{v}_{i_1}, \mathbf{v}_{i_2})$ or $\mathbf{E}_{i_1 i_2} = \phi(\mathbf{e}_{i_1 i_2})$.

Given two graphs $\mathcal{G}_1 = \{\mathcal{V}_1, \mathcal{E}_1\}, \mathcal{G}_2 = \{\mathcal{V}_2, \mathcal{E}_2\}$ of size m and n ($m \leq n$) respectively, the GM problem is to find an optimal node-to-node correspondence $\mathbf{P} \in \{0, 1\}^{m \times n}$, where $\mathbf{P}_{ij} = 1$ when the nodes $V_i^{(1)} \in \mathcal{V}_1$ and $V_j^{(2)} \in \mathcal{V}_2$ are matched and $\mathbf{P}_{ij} = 0$ otherwise. It is clear that any possible correspondence \mathbf{P} equals a (partial) permutation matrix when GM imposes the one-to-(at most)-one constraints. Therefore, the feasible field of \mathbf{P} can be defined as:

$$\mathcal{P} \triangleq \left\{ \mathbf{P} \in \{0, 1\}^{m \times n}; \mathbf{P}\mathbf{1}_n = \mathbf{1}_m, \mathbf{P}^T \mathbf{1}_m \leq \mathbf{1}_n \right\}, \quad (1)$$

where $\mathbf{1}_m$ is a unit vector. When $m = n$, $\mathbf{P} \in \mathcal{P}$ is orthogonal: $\mathbf{P}\mathbf{P}^T = \mathbf{I}_m$, where \mathbf{I}_m is a unit matrix.

To find the optimal correspondence, GM methods that incorporate pairwise constraints generally minimize or maximize their objective functions w.r.t. \mathbf{P} upon the feasible field \mathcal{P} . There are two main typical objective functions: Lawler's QAP [16] and Koopmans-Beckmann's QAP [17].

The main idea behind Lawler's QAP [11], [12], [16], [19], [23] is to maximize the sum of the node and edge similarities:

$$\max_{\mathbf{P} \in \mathcal{P}} \mathbf{P}_v^T \mathbf{K} \mathbf{P}_v = \sum_{ij} \mathbf{P}_{ij} \mathbf{K}_{ij;ij} + \sum_{\substack{(i_1, i_2) \\ (j_1, j_2)}} \mathbf{P}_{i_1 j_1} \mathbf{K}_{i_1 j_1; i_2 j_2} \mathbf{P}_{i_2 j_2}, \quad (2)$$

where \mathbf{P}_v is the columnwise vectorized replica of \mathbf{P} . The diagonal element $\mathbf{K}_{ij;ij}$ measures the node affinity calculated with node attributes as $\Phi_v(\mathbf{v}_i^{(1)}, \mathbf{v}_j^{(2)})$, and $\mathbf{K}_{i_1 j_1; i_2 j_2}$ measures the edge affinity calculated with edge attributes as $\Phi_e(\mathbf{e}_{i_1 i_2}^{(1)}, \mathbf{e}_{j_1 j_2}^{(2)})$. $\mathbf{K} \in \mathbb{R}^{mn \times mn}$ is called the affinity matrix of \mathcal{G}_1 and \mathcal{G}_2 .

Koopmans-Beckmann's QAP [17], [24] formulates GM as

$$\max_{\mathbf{P} \in \mathcal{P}} -\text{tr}(\mathbf{U}^T \mathbf{P}) + \lambda \text{tr}(\mathcal{E}_1 \mathbf{P} \mathcal{E}_2 \mathbf{P}^T), \quad (3)$$

where $\{\mathbf{U}_{ij}\} \in \mathbb{R}^{m \times n}$ measures the dissimilarity between node $V_i^{(1)}$ and $V_j^{(2)}$, and $\mathcal{E}_1, \mathcal{E}_2$ are the adjacency weight matrices of \mathcal{G}_1 and \mathcal{G}_2 . $\lambda \geq 0$ is a weight between the unary and pairwise terms. This formulation differs from Eqn. (2) mainly in the pairwise term which measures the edge compatibility as the linear similarity of adjacency matrices \mathcal{E}_1 and \mathcal{E}_2 . In fact, Eqn. (3) can be regarded as a special case of Lawler's QAP (Eqn. (2)) if $\mathbf{K} = \mathcal{E}_1 \otimes \mathcal{E}_2$, where \otimes denotes the Kronecker product. With this formulation, the space complexity of GM is $O(n^2)$ and much lower than that $O(m^2 n^2)$ of Eqn. (2).

The Eqn. (3) has another approximation, which aims to minimize the node and edge dissimilarity between two graphs:

$$\min_{\mathbf{P} \in \mathcal{P}} \langle \mathbf{P}, \mathbf{U} \rangle_F + \frac{\lambda}{2} \|\mathcal{E}_1 - \mathbf{P} \mathcal{E}_2 \mathbf{P}^T\|_F^2, \quad (4)$$

where $\langle \cdot, \cdot \rangle_F$ is the Frobenius dot-product defined as $\langle A, B \rangle_F = \sum_{ij} A_{ij} B_{ij}$ and $\|\cdot\|_F^2$ is the Frobenius matrix norm defined as $\|A\|_F^2 = \langle A, A \rangle_F$. The conversion from Eqn. (4) to Eqn. (3) holds equally under the fact that $\forall \mathbf{P} \in \mathcal{P}$ is an orthogonal matrix.

Due to the NP-complete nature of the above formulations, GM methods generally approximate the discrete feasible field \mathcal{P} by a continuous relaxation $\hat{\mathcal{P}} : \mathbf{P}_{ij} \in [0, 1]$, which is known as the *doubly stochastic relaxation*. Then the objective functions can be approximately solved by applying constrained optimization methods and employing a post-discretization step such the Hungarian algorithm [25] to obtain a discrete binary solution.

2.2 Related Work

Over the past decades, the GM problem of finding node-to-node correspondence between graphs has been extensively studied [1], [2]. Earlier works (exact GM) [26], [27] tended to regard GM as (sub)graph isomorphism. However, this assumption is too strict and leads to less flexibility for real applications. Therefore, later works on GM (inexact/error-tolerant GM) [12], [19], [21], [23] focused more on finding inexact matching between weighted graphs via optimizing more flexible objective functions.

Among the inexact GM methods, some of them aim to reduce the considerable space complexity caused by the affinity matrix \mathbf{K} in Eqn. (2). A typical work is the factorized graph matching (FGM) [19], which factorized \mathbf{K} as a Kronecker product of several smaller matrices. An efficient sampling heuristic was proposed in [28] to avoid storing the whole \mathbf{K} at once. Some works [24], [29] constructed objective functions similar to Eqn. (4) or Eqn. (3) to avoid using matrix \mathbf{K} . Our work use the representation of edge attributes rather than \mathbf{K} .

Since exactly solving the objective functions upon discrete feasible field \mathcal{P} is NP-complete, most GM methods relax \mathcal{P} for approximation purpose in several ways. The first typical relaxation is spectral relaxation, as proposed in [11], [30], by forcing $\|\mathbf{P}\|_2 = 1$; then, the solution is computed as the leading eigenvector of \mathbf{K} . The second relaxation [24] is to consider \mathcal{P} as a subset of an orthogonal matrices set such that $\mathbf{P}\mathbf{P}^T = \mathbf{I}_m$, which is the basis of converting Eqn. (4) into Eqn. (3). Semidefinite-programming (SDP) was also applied to approximately solve the GM problem in [31], [32] by introducing a new variable $\mathbf{X} = \mathbf{P}_v \mathbf{P}_v^T$ under the convex semidefinite constraint $\mathbf{X} - \mathbf{P}_v \mathbf{P}_v^T \succeq 0$. Then, \mathbf{P} is approximately recovered by \mathbf{X} .

The most widely used relaxation approach is the doubly stochastic relaxation $\hat{\mathcal{P}}$, which is the convex hull of \mathcal{P} . Since $\hat{\mathcal{P}}$ is a convex set defined in a linear form, it allows the GM objectives functions to be solved by more flexible convex or nonconvex optimization algorithms. To find more global optimal solutions with a binary property, the algorithms proposed in [19], [24], [33], [34] constructed objective functions in both convex and concave relaxations controlled by a continuation parameter, and then they developed a path-following-based strategy for optimization. These approaches are generally time consuming, particularly for matching graphs with more than dozens of nodes. The graduated assignment method [35] iteratively solved a series of first-order approximations of the objective function. Its improvement [36] provided more convergence analysis. The decomposition-based work in [37] developed its optimization technique by referring to dual decomposition. Additionally, another method in [21] decomposed the matching constraints and then used an optimization strategy based on the alternating direction method of multipliers.

To ensure binary solutions, some methods such as the integer-projected fixed point algorithm [23] and discrete tabu search [38], have been proposed to search in the discrete feasible domain. Although our method also adopt the doubly stochastic relaxation, we can solve our objective functions with nearly binary solutions, which helps to reduce the effect of the post-discretization step.

In addition to approximating the objective functions, some works also intended to provide more interpretations of the GM problem. The probability-based works [28], [39] solved the GM problem from a maximum likelihood estimation perspective. Some learning-based works [40], [41] went further to explore how to improve the affinity matrix \mathbf{K} by considering rotations and scales of real data. A pioneering work [42] presented an end-to-end deep learning framework for GM. A random walk view [12] was introduced by simulating random walks with reweighting jumps. Some works [19], [43] addressed matching graphs with geometric deformations. A max-pooling-based strategy was proposed in [44] to address the presence of outliers. Compared to these works, our proposed FRGM provides more geometric insights for GM with general settings by using a functional representation to interpret the geometric nature of real-world data, and it then matches graphs embedded in Euclidean space by providing a new parameterization view to handle graphs under geometric deformations.

3 FUNCTIONAL REPRESENTATION FOR GM

This section presents the functional representation for general GM (FRGM-G) that incorporates pairwise constraints. In Sec. 3.1, we introduce the function space of a graph, on which functionals can be defined as the inner product or metric to compatibly represent the edge attributes. In Sec. 3.2, we discuss how to represent the correspondence between graphs as a linear functional representation map between function spaces. Finally, the correspondence is an optimal functional map obtained by the algorithm in Sec. 3.3.

3.1 Function Space on Graph

Given an undirected graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ with edge attribute matrix $\mathbf{E} \in \mathbb{R}^{m \times m}$, we aim to establish function space $\mathcal{F}(\mathcal{V}, \mathbb{R})$ of \mathcal{G} , on which some geometric structures, such as inner product or metric can be defined. This is especially meaningful when graphs are embedded in explicit or hidden manifolds.

Let $\mathcal{F}(\mathcal{V}, \mathbb{R})$ denote the function space of all real-valued functions on $\mathcal{V} = \{V_i\}_{i=1}^m$. Since \mathcal{V} is finite discrete, we can choose a finite set of basis functions $\Phi = \{\phi_i\}_{i=1}^m$ to explicitly construct $\mathcal{F}(\mathcal{V}, \mathbb{R})$.

Definition 3.1. The *function space* $\mathcal{F}(\mathcal{V}, \mathbb{R})$ on graph \mathcal{G} can be defined as:

$$\mathcal{F}(\mathcal{V}, \mathbb{R}) \triangleq \left\{ \phi_{\mathbf{a}} = \sum_i a_i \phi_i, \mathbf{a} \triangleq (a_1, \dots, a_m)^T \in \mathbb{R}^m \right\}. \quad (5)$$

For example, ϕ_i can be chosen as the indicator of V_i :

$$\phi_i : \mathcal{V} \rightarrow \mathbb{R}, \quad \phi_i(V_j) = \begin{cases} 1, & j = i \\ 0, & j \neq i \end{cases}. \quad (6)$$

Considering the fact that the correspondence matrix $\mathbf{P} \in \hat{\mathcal{P}}$ is positive, *i.e.*, $\mathbf{P}_{ij} \in [0, 1]$, a typical subset of $\mathcal{F}(\mathcal{V}, \mathbb{R})$ can be defined as follows, which is the convex hull of $\{\phi_i\}_{i=1}^m$:

$$\mathcal{C}(\mathcal{V}, \mathbb{R}) \triangleq \left\{ \phi_{\mathbf{a}} = \sum_i a_i \phi_i; \sum_i a_i = 1, \mathbf{a} \in \mathbb{R}_+^m \right\}. \quad (7)$$

Once the function space $\mathcal{F}(\mathcal{V}, \mathbb{R})$ is built, some trivial operations can be defined, *e.g.*, inner product $\langle \phi_{\mathbf{a}}, \phi_{\mathbf{b}} \rangle = \sum_i a_i b_i$ and metric $d(\phi_{\mathbf{a}}, \phi_{\mathbf{b}}) = (\sum_i (a_i - b_i)^2)^{1/2}$. However, these definitions cannot express the edge attribute $\mathbf{E}_{i_1 i_2}$. Therefore, we aim to define some other operations to represent $\mathbf{E} \in \mathbb{R}^{m \times m}$ based on $\mathcal{F}(\mathcal{V}, \mathbb{R})$. An available approach is to define functionals on the product space $\mathcal{F}(\mathcal{V}, \mathbb{R}) \times \mathcal{F}(\mathcal{V}, \mathbb{R})$. Moreover, the functionals should (1) be compatible with \mathbf{E} and (2) have geometric structures such as inner product or metric, as demonstrated in the following:

Definition 3.2. A functional $\mathbb{F}_{\mathcal{V}} : \mathcal{F}(\mathcal{V}, \mathbb{R}) \times \mathcal{F}(\mathcal{V}, \mathbb{R}) \rightarrow \mathbb{R}$ is *compatible* with \mathbf{E} if it satisfies $\mathbb{F}_{\mathcal{V}}(\phi_{i_1}, \phi_{i_2}) = \mathbf{E}_{i_1 i_2}$.

Among all the compatible functionals, there are some specified ones that can be defined as the inner product or metric on the function space $\mathcal{F}(\mathcal{V}, \mathbb{R})$ or its subset $\mathcal{C}(\mathcal{V}, \mathbb{R})$, as follows.

Definition 3.3. The *inner product* on the function space $\mathcal{F}(\mathcal{V}, \mathbb{R})$ can be defined in an explicit form: $\forall \phi_{\mathbf{a}}, \phi_{\mathbf{b}} \in \mathcal{F}(\mathcal{V}, \mathbb{R})$,

$$\mathbb{F}_{\mathcal{V}}(\phi_{\mathbf{a}}, \phi_{\mathbf{b}}) \triangleq \sum_{i_1, i_2} a_{i_1} b_{i_2} \mathbb{F}_{\mathcal{V}}(\phi_{i_1}, \phi_{i_2}) = \sum_{i_1, i_2} a_{i_1} b_{i_2} \mathbf{E}_{i_1 i_2}. \quad (8)$$

For the given edge attribute matrix \mathbf{E} that is symmetric, $\mathbb{F}_{\mathcal{V}}(\cdot, \cdot)$ satisfies the first two inner product axioms: symmetry and linearity. To satisfy the third axiom, positive-definiteness, we need more knowledge about \mathbf{E} , *e.g.*, \mathbf{E} is positive-definite. However, if the positive-definiteness is too strong, we can relax it to a weaker condition.

Proposition 1. Assume that \mathbf{E} satisfies $\mathbf{E}_{i_1 i_2} = 0$ iff $i_1 = i_2$. Then, the functional $\mathbb{F}_{\mathcal{V}}(\cdot, \cdot)$ in Eqn. (8) satisfies all three axioms on $\mathcal{F}(\mathcal{V}, \mathbb{R})$ by replacing \mathbf{E} with $\exp(-\mathbf{E}^2/\sigma^2)$ with $\sigma > 0$ small enough. Here, \mathbf{E}^2 is a pointwise product.

This proposition holds because when $\sigma > 0$ is sufficiently small, all the eigenvalues of matrix $\exp(-\mathbf{E}^2/\sigma^2)$ will be positive. In particular, when \mathbf{E} is computed as a metric (distance) matrix on an explicit or hidden manifold, it satisfies that $\mathbf{E}_{ii} = 0$ and $\exp(-\mathbf{E}^2/\sigma^2)$ is positive-definite. Moreover, σ can be used to adjust the eigenspace of $\exp(-\mathbf{E}^2/\sigma^2)$. Fig. 2 illustrates an empirical study on thousands of \mathbf{E} 's extracted from both realistic and synthetic datasets used in Sec. 7. The edge attribute \mathbf{E} of each graph is computed in a metric form (either Euclidean distance or geodesic distance) and then normalized to $[0, 1]$ divided by the maximum element. We can see that

- 1) All the eigenvalues of $\exp(-\mathbf{E}^2/\sigma^2)$ are positive.
- 2) The ratio between the minimum and maximum eigenvalues has a similar tendency when σ varies from 0 to 1.

It shows that $\exp(-\mathbf{E}^2/\sigma^2)$ will become indistinguishable if σ is too small or unbalanced if σ is too large. We can choose a suitable σ to adjust the eigenspace of $\exp(-\mathbf{E}^2/\sigma^2)$ to achieve better matching performance.

The inner product $\mathbb{F}_{\mathcal{V}}(\cdot, \cdot)$ can induce a metric by definition $d(\phi_{\mathbf{a}}, \phi_{\mathbf{b}}) \triangleq \mathbb{F}_{\mathcal{V}}(\phi_{\mathbf{a}} - \phi_{\mathbf{b}}, \phi_{\mathbf{a}} - \phi_{\mathbf{b}})^{1/2}$. Moreover, we can also define another metric on the subset $\mathcal{C}(\mathcal{V}, \mathbb{R})$ based on \mathbf{E} itself.

Definition 3.4. The *metric* on the convex hull $\mathcal{C}(\mathcal{V}, \mathbb{R})$ can be defined in an implicit form: $\forall \phi_{\mathbf{a}}, \phi_{\mathbf{b}} \in \mathcal{C}(\mathcal{V}, \mathbb{R})$,

$$\mathbb{F}_{\mathcal{V}}(\phi_{\mathbf{a}}, \phi_{\mathbf{b}}) = \min_{\pi \in \mathcal{P}(\mathbf{a}, \mathbf{b})} \pi_{i_1 i_2} \mathbb{F}_{\mathcal{V}}(\phi_{i_1}, \phi_{i_2}) \quad (9)$$

$$= \min_{\pi \in \mathcal{P}(\mathbf{a}, \mathbf{b})} \pi_{i_1 i_2} \mathbf{E}_{i_1 i_2}, \quad (10)$$

where $\mathcal{P}(\mathbf{a}, \mathbf{b}) = \left\{ \pi \in \mathbb{R}_+^{m \times m}; \sum_i \pi_{ij} = b_j, \sum_j \pi_{ij} = a_i \right\}$.

SUBMITTED TO IEEE T-PAMI, 2019

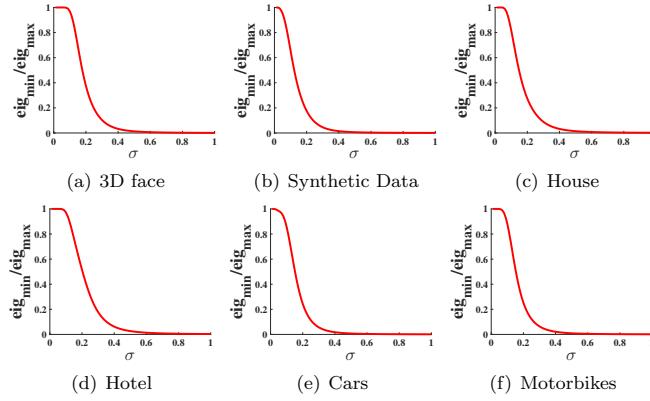


Fig. 2. Empirical statistics of $\exp(-\mathbf{E}^2/\sigma^2)$ extracted from the realistic and synthetic datasets used in the experimental section. For thousands of graphs in all six datasets, as σ varies from 0 to 1, the ratio between the minimum and maximum eigenvalues of $\exp(-\mathbf{E}^2/\sigma^2)$ changes with a similar tendency.

When \mathbf{E} is computed as a metric, $\mathbb{F}_{\mathcal{V}}(\cdot, \cdot)$ satisfies all three distance axioms on $\mathcal{C}(\mathcal{V}, \mathbb{R})$, and it is a typical Wasserstein distance (or Sinkhorn distance) [45]. The definition in Eqn. (10) is not differentiable w.r.t. **a**, **b**; one can use the entropy-regularized Wasserstein distance [45] to achieve differentiability.

With the function space equipped with inner product or metric, each graph is assigned with explicit geometric structures that are compatible with the edge attribute. Next, we demonstrate the idea of using the functional map representation to formulate the correspondence between graphs as a functional \mathcal{T}_F between two function spaces $\mathcal{T}_F : \mathcal{F}(\mathcal{V}_1, \mathbb{R}) \rightarrow \mathcal{F}(\mathcal{V}_2, \mathbb{R})$.

3.2 Functional Map Representation for GM

The matching between two graphs $\mathcal{G}_1 = \{\mathcal{V}_1, \mathcal{E}_1\}$ and $\mathcal{G}_2 = \{\mathcal{V}_2, \mathcal{E}_2\}$ can be viewed as a mapping \mathcal{T} from \mathcal{V}_1 to \mathcal{V}_2 , which may be nonlinear and complicated. Therefore, we use the push-forward operation to induce a linear functional \mathcal{T}_F rather than \mathcal{T} to equally represent the matching between graphs.

Due to the one-to-(at most)-one constraint of GM, we strict that $\mathcal{T} : \mathcal{V}_1 \rightarrow \mathcal{V}_2$ is an injective mapping, *i.e.*, each node in \mathcal{G}_2 is mapped to by at most one node in \mathcal{G}_1 . Then, the constrained mapping $\mathcal{T}|_{\mathcal{T}(\mathcal{V}_1)} : \mathcal{V}_1 \rightarrow \mathcal{T}(\mathcal{V}_1) \subseteq \mathcal{V}_2$ is bijective, *i.e.*, exactly one-to-one correspondence. Therefore, without ambiguity, we can assume that \mathcal{T} is bijective and invertible between \mathcal{V}_1 and $\mathcal{T}(\mathcal{V}_1) \subseteq \mathcal{V}_2$. Each \mathcal{T} induces a natural transformation $\mathcal{T}_F : \mathcal{F}(\mathcal{V}_1, \mathbb{R}) \rightarrow \mathcal{F}(\mathcal{V}_2, \mathbb{R})$ via the push-forward operation, which is widely used in functional analysis [46] and real applications [47] [48]:

Definition 3.5. The functional $\mathcal{T}_F : \mathcal{F}(\mathcal{V}_1, \mathbb{R}) \rightarrow \mathcal{F}(\mathcal{V}_2, \mathbb{R})$ induced from \mathcal{T} is defined as: $\forall \phi \in \mathcal{F}(\mathcal{V}_1, \mathbb{R})$, the image of ϕ is $\mathcal{T}_F(\phi) \triangleq \phi \circ \mathcal{T}^{-1} \in \mathcal{F}(\mathcal{V}_2, \mathbb{R})$.

Proposition 2. *The original \mathcal{T} can be recovered from \mathcal{T}_F .*

For each point $V_i^{(1)} \in \mathcal{V}_1$, it can be associated with an indicator function ϕ_i as Eqn. (6). To recover the image $\mathcal{T}(V_i^{(1)})$ from \mathcal{T}_F , we utilize the function $\psi = \mathcal{T}_F(\phi_i) : \mathcal{V}_2 \rightarrow R$, which satisfies $\forall V \in \mathcal{V}_2$,

$$\psi(V) \triangleq \phi \circ \mathcal{T}^{-1}(V) = \begin{cases} 1, & \mathcal{T}^{-1}(V) = V_i^{(1)}, \\ 0, & \mathcal{T}^{-1}(V) \neq V_i^{(1)}. \end{cases} \quad (11)$$

Since \mathcal{T} is bijective and invertible, a unique $V \in \mathcal{V}_2$ exists s.t. $\mathcal{T}^{-1}(V) = V_i^{(1)}$. Then, once we find $\psi(V_j^{(2)}) = 1$, we have

$\mathcal{T}^{-1}(V_j^{(2)}) = V_i^{(1)}$, and $V_j^{(2)}$ must equal the image $\mathcal{T}(V_i^{(1)})$ of $V_i^{(1)}$: $\mathcal{T}(V_i^{(1)}) = V_j^{(2)}$. Thus, the functional \mathcal{T}_F can be used to equally represent \mathcal{T} .

Proposition 3. *\mathcal{T}_F is a linear mapping from function spaces $\mathcal{F}(\mathcal{V}_1, \mathbb{R})$ to $\mathcal{F}(\mathcal{V}_2, \mathbb{R})$.*

It holds because $\forall f_1, f_2 \in \mathcal{F}(\mathcal{V}_1, \mathbb{R}), \alpha_1, \alpha_2 \in \mathbb{R}$,

$$\begin{aligned} \mathcal{T}_F(\alpha_1 f_1 + \alpha_2 f_2) &= (\alpha_1 f_1 + \alpha_2 f_2) \circ \mathcal{T}^{-1} \\ &= \alpha_1 f_1 \circ \mathcal{T}^{-1} + \alpha_2 f_2 \circ \mathcal{T}^{-1} = \alpha_1 \mathcal{T}_F(f_1) + \alpha_2 \mathcal{T}_F(f_2). \end{aligned} \quad (12)$$

Although \mathcal{T} may be complicated, \mathcal{T}_F is linear and simple.

With function spaces $\mathcal{F}(\mathcal{V}_1, \mathbb{R})$ and $\mathcal{F}(\mathcal{V}_2, \mathbb{R})$ defined by basis functions $\Phi = \{\phi_i\}_{i=1}^m$ and $\Psi = \{\psi_j\}_{j=1}^n$ respectively, each basis function ϕ_i can be transformed into $\mathcal{F}(\mathcal{V}_2, \mathbb{R})$ and represented in a linear form as $\mathcal{T}_F(\phi_i) = \sum_{j=1}^n \mathbf{P}_{ij} \psi_j$. Whenever \mathbf{P} reaches an extreme point of the feasible field $\hat{\mathcal{P}}$, it is a binary correspondence between graphs, and consequently, ϕ_i is transformed into (*i.e.*, matches) a ψ_j' , where $\mathbf{P}_{ij'} = 1, \mathbf{P}_{i,j' \neq j'} = 0$.

To find an optimal correspondence between two graphs with edge attributes $\mathbf{E}_1 \in \mathbb{R}^{m \times m}$ and $\mathbf{E}_2 \in \mathbb{R}^{n \times n}$, we declare that the induced functional \mathcal{T}_F should be able to preserve the geometric structures defined on function spaces. Namely, \mathcal{T}_F should be the inner product or metric preserving. More precisely, for each pair (ϕ_{i_1}, ϕ_{i_2}) , the functional value $\mathbb{F}_{\mathcal{V}_1}(\phi_{i_1}, \phi_{i_2})$ should be similar to the functional value of the transformed pair $(\mathcal{T}_F(\phi_{i_1}), \mathcal{T}_F(\phi_{i_2}))$, which is calculated as

$$\mathbb{F}_{\mathcal{V}_2}(\mathcal{T}_F(\phi_{i_1}), \mathcal{T}_F(\phi_{i_2})) = \mathbb{F}_{\mathcal{V}_2}\left(\sum_{j=1}^n \mathbf{P}_{i_1 j} \psi_j, \sum_{j=1}^n \mathbf{P}_{i_2 j} \psi_j\right). \quad (13)$$

The functionals defined in Definition 3.3 or Definition 3.4 can be used to calculate it. Finally, to incorporate the pairwise constraints, we aim to minimize the total sum as follows:

$$\begin{aligned} &\sum_{(i_1, i_2)} \mathcal{E}_{1_{i_1 i_2}} [\mathbb{F}_{\mathcal{V}_1}(\phi_{i_1}, \phi_{i_2}) - \mathbb{F}_{\mathcal{V}_2}(\mathcal{T}_F(\phi_{i_1}), \mathcal{T}_F(\phi_{i_2}))]^2 \\ &\triangleq \sum_{(i_1, i_2)} \mathcal{E}_{1_{i_1 i_2}} [\mathbf{E}_{1_{i_1 i_2}} - \mathbf{F}(\mathbf{P})_{i_1 i_2}]^2 \triangleq \|\mathbf{E}_1 - \mathbf{F}(\mathbf{P})\|_{F, \mathcal{E}_1}^2, \end{aligned} \quad (14)$$

where $\mathbf{F}(\mathbf{P})_{i_1 i_2} \triangleq \mathbb{F}_{\mathcal{V}_2}\left(\sum_{j=1}^n \mathbf{P}_{i_1 j} \psi_j, \sum_{j=1}^n \mathbf{P}_{i_2 j} \psi_j\right)$ is computed based on the edge attributes matrix \mathbf{E}_2 . Note that the affinity matrix \mathbf{K} with size $\mathbf{O}(m^2 n^2)$ is replaced here by the edge attributes matrix \mathbf{E}_1 with size $\mathbf{O}(m^2)$ and \mathbf{E}_2 with size $\mathbf{O}(n^2)$.

3.3 FRGM-G: matching graphs with general settings

Here, we propose our FRGM-G algorithm for matching graphs with general settings, *i.e.* without knowledge on the geometrical structures of the graphs. To find an optimal correspondence, *i.e.*, functional map \mathbf{P} mentioned above, we first minimize an objective function as

$$J_{ori}(\mathbf{P}) = (1 - \alpha_1) \langle \mathbf{P}, \mathbf{U} \rangle_F + \alpha_1 \|\mathbf{E}_1 - \mathbf{F}(\mathbf{P})\|_{F, \mathcal{E}_1}^2, \quad (15)$$

where $\alpha_1 \in [0, 1]$ balances the weights of the unary term and pairwise term. In general, $J_{ori}(\mathbf{P})$ is nonconvex and minimizing $J_{ori}(\mathbf{P})$ upon the feasible field $\hat{\mathcal{P}}$ results in a local minimum. The minimizer \mathbf{P}_1^* may be not binary, and the post-discretization of \mathbf{P}_1^* may reduce the matching accuracy. Therefore, we next construct another objective function to find a better solution based on the obtained \mathbf{P}_1^* .

According to the definition $\mathcal{T}_F(\phi_i) = \sum_{j=1}^n \mathbf{P}_{ij} \psi_j$, each $\mathcal{T}_F(\phi_i)$ lies in the convex set $\mathcal{C}(\mathcal{V}_2, \mathbb{R})$, which is the convex hull of $\{\psi_j\}_{j=1}^n$. Therefore, the transformed functions $\{\mathcal{T}_F(\phi_i)\}_{i=1}^m$ lies in the same function space spanned by $\{\psi_j\}_{j=1}^n$, and the

offset between $\{\mathcal{T}_F(\phi_i)\}_{i=1}^m$ and $\{\psi_j\}_{j=1}^n$ can be controlled. Moreover, since \mathbf{P}_1^* indeed preserves the pairwise geometric structure between two graphs, $\mathcal{T}_F(\phi_i)$ will lie closer to the correct matching ψ_{δ_i} . This means that, based on the metric defined on the function spaces, the distance $d(\mathcal{T}_F(\phi_i), \psi_{\delta_i})$ makes sense and will be smaller than $d(\mathcal{T}_F(\phi_i), \psi_{j \neq \delta_i})$. Therefore, we define the second objective function as:

$$J_{int}(\mathbf{P}) = (1 - \alpha_2) \langle \mathbf{P}, \mathbf{D} \rangle_F + \alpha_2 \|\mathbf{F}(\mathbf{P}_1^*) - \mathbf{F}(\mathbf{P})\|_{F, \mathcal{E}_1}^2, \quad (16)$$

where $\mathbf{D}_{ij} = d(\mathcal{T}_F(\phi_i), \psi_j)$ is the distance between $\mathcal{T}_F(\phi_i)$ and ψ_j computed by the metric functional defined on $\mathcal{F}(\mathcal{V}_2, \mathbb{R})$ or $\mathcal{C}(\mathcal{V}_2, \mathbb{R})$. The minimizer \mathbf{P}_2^* can be viewed as a displacement interpolation: to minimize $\langle \mathbf{P}, \mathbf{D} \rangle_F$, we obtain a solution \mathbf{P}_0^* that is an extreme point (thus, binary) of the feasible field $\hat{\mathcal{P}}$; to minimize $\|\mathbf{F}(\mathbf{P}_1^*) - \mathbf{F}(\mathbf{P})\|_{F, \mathcal{E}_1}^2$, we obtain a solution that equals $\mathbf{P}_1^* \in \hat{\mathcal{P}}$. Then, \mathbf{P}_2^* is an interpolation between \mathbf{P}_0^* and \mathbf{P}_1^* controlled by $\alpha_2 \in [0, 1]$. Finally, we use the Hungarian method to discretize \mathbf{P}_2^* into being binary.

4 FRGM IN EUCLIDEAN SPACE

In many computer vision applications, graphs are often embedded in explicit or implicit manifolds \mathcal{M} , e.g., Euclidean space \mathbb{R}^d and surface \mathcal{S} , where graphs with nodes $\mathcal{V} \in \mathcal{M}$ are naturally associated with some specific geometric properties. For example, the node attributes can be computed as SIFT [10], shape context [3], HKS [49], etc; the edge attribute matrix \mathbf{E} can be computed as Euclidean distance on \mathbb{R}^d or geodesic distance on \mathcal{S} .

The proposed method FRGM-G in Sec. 3 can be used to match graphs in these cases. Furthermore, for graphs embedded in \mathbb{R}^d , we can extend FRGM to address Euclidean GM (FRGM-E) based on the fact that the functional representation of \mathcal{T}_F between abstract function spaces can be deduced into the concrete Euclidean space \mathbb{R}^d with explicit geometric interpretations. Since each node can be represented as a vector $V_j^{(2)} \in \mathbb{R}^d$, the expression $\mathbf{P}_{ij} V_j^{(2)}$ naturally makes sense. Consequently, we can directly define the unknown transformation $\mathcal{T} : \mathcal{V}_1 \rightarrow \mathcal{V}_2$ in a linear form:

$$\mathcal{T} : \mathcal{V}_1 \rightarrow \mathcal{V}_2, \quad (17)$$

$$V_i^{(1)} \mapsto \mathcal{T}(V_i^{(1)}) = \sum_{j=1}^n \mathbf{P}_{ij} V_j^{(2)}. \quad (18)$$

The transformed nodes can be rewritten in a matrix notation $\mathcal{T}(\mathcal{V}_1) \triangleq \mathbf{P}\mathcal{V}_2$ and $\mathcal{T}(V_i^{(1)}) \triangleq (\mathbf{P}\mathcal{V}_2)_i$. Now, $\mathbf{P} \in \mathbb{R}^{m \times n}$ is a linear representation map of the unknown transformation \mathcal{T} . With the constraint that $\mathbf{P} \in \hat{\mathcal{P}}$, each node $\mathcal{T}(V_i^{(1)})$ lies in the convex hull of $\mathcal{V}_2 \in \mathbb{R}^d$. Once \mathbf{P} reaches a binary correspondence matrix, $V_i^{(1)}$ is transformed into $V_{j'}^{(2)}$, where $\mathbf{P}_{ij'} = 1, \mathbf{P}_{i,j \neq j'} = 0$.

For graphs embedded in Euclidean spaces, the edge attributes, such as edge length and edge orientation, are widely used. The edge attributes of the transformed graph $\mathcal{T}(\mathcal{V}_1) = \mathbf{P}\mathcal{V}_2$ can be computed as a function w.r.t. \mathbf{P} as:

- **edge length** computed as the Euclidean distance

$$\|\mathcal{T}(V_{i_1}^{(1)}) - \mathcal{T}(V_{i_2}^{(1)})\| = \|(\mathbf{P}\mathcal{V}_2)_{i_1} - (\mathbf{P}\mathcal{V}_2)_{i_2}\|,$$

- **edge orientation** computed as the vector between nodes

$$\overrightarrow{\mathcal{T}(V_{i_1}^{(1)}) - \mathcal{T}(V_{i_2}^{(1)})} = \overrightarrow{(\mathbf{P}\mathcal{V}_2)_{i_1} - (\mathbf{P}\mathcal{V}_2)_{i_2}},$$

where $\|\cdot\|$ is the Euclidean L_2 norm. We propose our algorithm for matching graphs in Euclidean space, i.e. **FRGM-E**, in the following sections.

4.1 Preserving edge-length

Given two graphs with visually similar structures, a general constraint is to preserve the edge length between the original edge $V_{i_1 i_2}^{(1)} \triangleq (V_{i_1}^{(1)}, V_{i_2}^{(1)})$ and its corresponding edge $\mathcal{T}(V_{i_1 i_2}^{(1)}) \triangleq (\mathcal{T}(V_{i_1}^{(1)}), \mathcal{T}(V_{i_2}^{(1)})) \triangleq (\mathbf{P}\mathcal{V}_2)_{i_1 i_2}$. Thus, the pairwise potential of the first objective function can be defined as follows:

$$J_{non}(\mathbf{P}) = \sum_{(i_1, i_2)} \mathcal{E}_{1_{i_1 i_2}} (\|V_{i_1 i_2}^{(1)}\| - \|\mathcal{T}(V_{i_1 i_2}^{(1)})\|)^2 \quad (19)$$

$$= \sum_{(i_1, i_2)} \mathcal{E}_{1_{i_1 i_2}} (\|V_{i_1 i_2}^{(1)}\| - \|(\mathbf{P}\mathcal{V}_2)_{i_1 i_2}\|)^2. \quad (20)$$

We can add a unary term $\langle \mathbf{P}, \mathbf{U} \rangle_F$ computed with node attributes to this pairwise term as follows:

$$J_{non}(\mathbf{P}) = (1 - \lambda_1) \langle \mathbf{P}, \mathbf{U} \rangle_F \quad (21)$$

$$+ \lambda_1 \sum_{(i_1, i_2)} \mathcal{E}_{1_{i_1 i_2}} (\|V_{i_1 i_2}^{(1)}\| - \|(\mathbf{P}\mathcal{V}_2)_{i_1 i_2}\|)^2. \quad (22)$$

Due to the nonconvexity of $J_{non}(\mathbf{P})$, its solution $\mathbf{P}_1^* \in \hat{\mathcal{P}}$ often reaches a local minimum and is not binary, and the post-discretization procedure will result in low accuracy; see Fig. 3 (b) for illustration. Consequently, the transformed node $\mathcal{T}(V_i^{(1)})$ is not exactly equal to a $V_j^{(2)} \in \mathcal{V}_2$, and there is often an offset between $\mathcal{T}(V_i^{(1)})$ and its correct match $V_{\delta_i}^{(2)}$. Fig. 3 (a) shows this phenomenon, where each $\mathcal{T}(V_i^{(1)})$ shifts from the correct match $V_{\delta_i}^{(2)}$ to some degree.

4.2 Reducing node offset

Benefiting from the property of the solution \mathbf{P}_1^* that preserves the edge length of \mathcal{G}_1 , the offset vectors of adjacent transformed nodes in $\{(\mathbf{P}_1^*\mathcal{V}_2)_i\}_{i=1}^m$ have similar directions and norms, as shown in Fig. 3 (a). To reduce the node offset from $(\mathbf{P}_1^*\mathcal{V}_2)_i$ to the corresponding correct match $V_{\delta_i}^{(2)}$ denoted by

$$\overrightarrow{(\mathbf{P}_1^*\mathcal{V}_2)_i V_{\delta_i}^{(2)}} = V_{\delta_i}^{(2)} - (\mathbf{P}_1^*\mathcal{V}_2)_i,$$

we aim to minimize the differences between adjacent offset vectors,

$$J_{con}(\mathbf{P}) = \sum_{(i_1, i_2)} \mathbf{S}_{i_1 i_2} \|((\mathbf{P}\mathcal{V}_2)_{i_1} - (\mathbf{P}_1^*\mathcal{V}_2)_{i_1}) - ((\mathbf{P}\mathcal{V}_2)_{i_2} - (\mathbf{P}_1^*\mathcal{V}_2)_{i_2})\|^2 \quad (23)$$

$$= \text{Tr} \left((\mathbf{P}\mathcal{V}_2 - \mathbf{P}_1^*\mathcal{V}_2)^T \mathbf{L}_S (\mathbf{P}\mathcal{V}_2 - \mathbf{P}_1^*\mathcal{V}_2) \right), \quad (24)$$

where $\mathbf{L}_S = \text{diag}(\mathbf{S}) - \mathbf{S}$ and $\mathbf{S} \in \mathbb{R}_{+}^{m \times m}$ is computed to indicate the adjacency relation of node pair $(\mathcal{T}(V_{i_1}^{(1)}), \mathcal{T}(V_{i_2}^{(1)}))$. The undirected graph here will result in a symmetric \mathbf{S} ; therefore \mathbf{L}_S is positive-definite and $J_{con}(\mathbf{P})$ is convex.

Compared to the algorithm proposed for general GM in Sec. 3.3, the distance matrix \mathbf{D} here can be computed with an explicit geometric interpretation: $\mathbf{D}_{ij} = \|\mathcal{T}(V_i^{(1)}) - V_j^{(2)}\|$ is the Euclidean distance between the transformed node $\mathcal{T}(V_i^{(1)})$ and $V_j^{(2)}$. As shown in Fig. 3 (c), $\|\mathcal{T}(V_i^{(1)}) - V_{\delta_i}^{(2)}\|$ is smaller than $\|\mathcal{T}(V_i^{(1)}) - V_{j \neq \delta_i}^{(2)}\|$, where $V_{\delta_i}^{(2)}$ denotes the correct matching of $V_i^{(1)}$. Therefore, the unary term $\langle \mathbf{P}, \mathbf{D} \rangle_F$ can be added as a useful constraint during matching. Finally, $J_{con}(\mathbf{P})$ is summarized as:

$$J_{con}(\mathbf{P}) = (1 - \lambda_2) \langle \mathbf{P}, \mathbf{D} \rangle_F + \lambda_2 \text{Tr} \left((\mathbf{P}\mathcal{V}_2 - \mathbf{P}_1^*\mathcal{V}_2)^T \mathbf{L}_S (\mathbf{P}\mathcal{V}_2 - \mathbf{P}_1^*\mathcal{V}_2) \right). \quad (25)$$

In general, this objective function $J_{con}(\mathbf{P})$ is solved by a (nearly) binary solution \mathbf{P}_2^* if $\lambda_2 \in [0, 1]$ is small. This significantly improves the matching accuracy. See Fig. 3 (d) as an example.

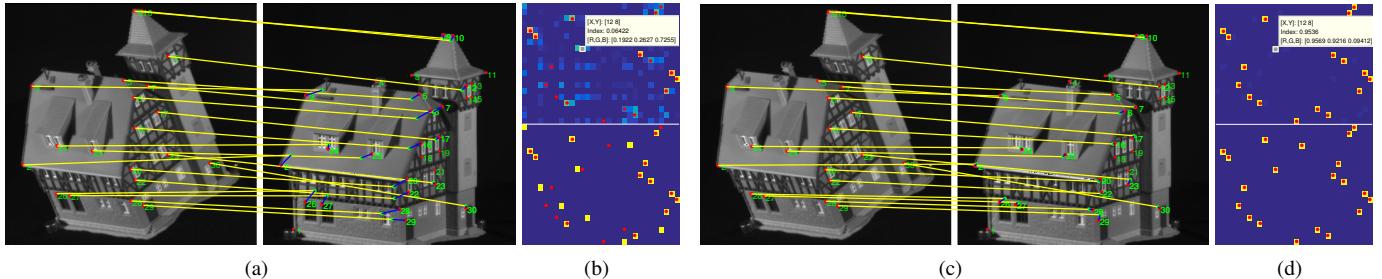


Fig. 3. (a) Nodes shift after being transformed by minimizing $J_{non}(\mathbf{P})$ in a 20-vs-30 case. The lines in blue are the offset vectors, and the points in green are transformed nodes $\{\mathcal{T}(V_i^{(1)})\}_{i=1}^m$. (b) Representation map \mathbf{P}_1^* (top) and the post-discretization (bottom) corresponding to (a). (c) Nodes transformed by minimizing $J_{con}(\mathbf{P})$ with almost no offset. (d) Representation map \mathbf{P}_2^* (top) and the post-discretization (bottom) corresponding to (c). In (b) and (d), red points mark the ground-truth correspondence.

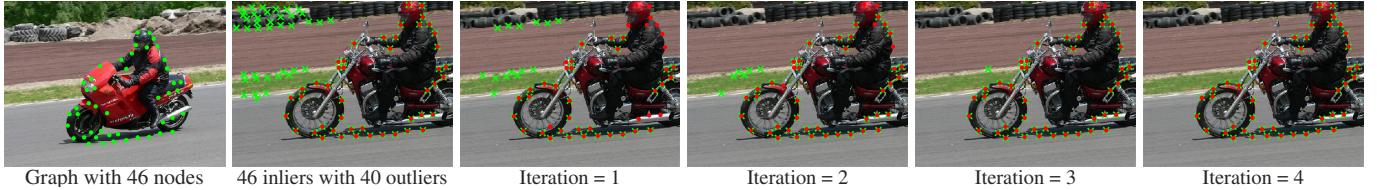


Fig. 4. Outlier removal with transformation map \mathbf{P}^* obtained by alternately minimizing $J_{non}(\mathbf{P})$ and $J_{con}(\mathbf{P})$. In each iteration, the red dots are inliers, and the green plus signs are the nodes remaining after removal.

4.3 Explicit outlier-removal strategy

In practice, outliers generally occur in graphs and affect the matching accuracy. Based on the ability of the optimal representation map \mathbf{P}_1^* and \mathbf{P}_2^* that preserves the geometric structure between \mathcal{V}_1 and the transformed graph $\mathbf{P}_1^*\mathcal{V}_2$ or $\mathbf{P}_2^*\mathcal{V}_2$, we can propose an explicit outlier-removal strategy.

The transformed graph $\mathcal{G}_{\mathcal{T}(\mathcal{V}_1)}$ with nodes $\mathcal{T}(\mathcal{V}_1) = \mathbf{P}^*\mathcal{V}_2$ lies in the convex hull of \mathcal{V}_2 . In some sense, the operation $\mathcal{T}(\mathcal{V}_1) = \mathbf{P}^*\mathcal{V}_2$ can be viewed as a domain adaptation [50] from the source domain \mathcal{V}_1 to the target domain \mathcal{V}_2 . The graph $\mathcal{G}_{\mathcal{T}(\mathcal{V}_1)}$ has a geometric structure similar to the original graph \mathcal{G}_1 and lies in the same space of \mathcal{G}_2 with a relatively small offset. Then, we can remove outliers adaptively using a ratio test technique. Given two point sets $\mathcal{T}(\mathcal{V}_1)$ and \mathcal{V}_2 , we compute the Euclidean distance d_{ij} of all the pairs $(\mathcal{T}(V_i^{(1)}), V_j^{(2)})$. For each node $\mathcal{T}(V_i^{(1)})$, we find the closest node $V_{j^*}^{(2)}$ and remove all the nodes $V_j^{(2)}$ when $d_{ij} > k \cdot d_{ij^*}$ for a given $k > 0$. If the number of remaining nodes l is less than m , $m - l$ nodes are selected from the removed ones that are closer to $\mathcal{T}(\mathcal{V}_1)$ and added. See Fig. 4 as an example, where after several iterations most outliers are removed. More experimental results are reported in the experimental section.

5 FRGM WITH GEOMETRIC DEFORMATION

For Euclidean GM, rigid or nonrigid geometric deformations may exist between graphs, where we need to estimate both the correspondence and deformation parameters. This section demonstrates that the FRGM can provide a new parameterization of transformation between graphs. Due to the associative law of matrix multiplication, this parameterization is associative with the deformation parameters. Theoretically, this allows us to extend FRGM for geometrically deformed graphs (FRGM-D) by estimating the correspondence and deformation parameters alternately.

5.1 Geometric deformation

Given two point sets $\mathcal{V}_1 = \{V_i^{(1)}\}_{i=1}^m$, $\mathcal{V}_2 = \{V_j^{(2)}\}_{j=1}^n \subseteq \mathbb{R}^d$ with geometric transformation $\tau : \mathcal{V}_1 \rightarrow \mathcal{V}_2$, the task to estimate both the correspondence \mathbf{P} and parameters of τ is generally formulated as minimizing the sum of residuals:

$$\min_{\mathbf{P} \in \mathcal{P}, \tau \in \chi} J(\mathbf{P}, \tau) = \sum_{i,j} \mathbf{P}_{ij} \|V_i^{(1)} - \tau(V_j^{(2)})\|^2 + \Upsilon(\tau), \quad (26)$$

where Υ is a regularization term of τ . On the one hand, most of the state-of-the-art registration algorithms such as [51], [52], [53] do not explicitly recover the correspondence \mathbf{P} as a binary solution. Rather, they estimate \mathbf{P} in a soft way as $\mathbf{P}_{ij} \in [0, 1]$ to give \mathbf{P}_{ij} a probability interpretation: \mathbf{P}_{ij} stands for the correspondence probability between $V_i^{(1)}$ and $V_j^{(2)}$. On the other hand, some methods [19], [54], [55] have also been proposed to find the binary correspondence by general GM algorithms. However, these GM-based methods are not consistent with the geometric nature behind the real data. Therefore, they can only handle point sets with simple geometric deformations.

Given a finite point set $\mathcal{V} \subseteq \mathbb{R}^{m \times d}$, the rigid or nonrigid geometric deformation is generally expressed as follows.

- **Similarity transformation:** $\tau(\mathcal{V}) = s\mathcal{V}\mathbf{R} + \mathbf{1}_m\mathbf{t}$, where $s \in \mathbb{R}_+$, $\mathbf{R} \in \mathbb{R}^{d \times d}$, and $\mathbf{t} \in \mathbb{R}^{1 \times d}$ denote the scaling factor, the rotation matrix and the translation vector, respectively. Naturally, \mathbf{R} should satisfy the constraint: $\mathbf{R}^T\mathbf{R} = \mathbf{I}_d, \det(\mathbf{R}) = 1$.
- **Affine transformation:** $\tau(\mathcal{V}) = \mathcal{V}\mathbf{A} + \mathbf{1}_m\mathbf{t}$, where $\mathbf{A} \in \mathbb{R}^{d \times d}$ and $\mathbf{t} \in \mathbb{R}^{1 \times d}$ denote the affine matrix and the translation vector, respectively.
- **Non-rigid transformation:** $\tau(\mathcal{V}) = \mathcal{V} + \mathbf{K}_e\mathbf{W}$, where $\mathbf{K}_e \in \mathbb{R}^{m \times m}$ is a kernel determined by the basis points $\{V_i\}_i$ and displacement functions $\{\varphi_i\}_i$, and $\mathbf{W} \in \mathbb{R}^{m \times d}$ is a weight matrix that measures the degree of deformation. This definition is based on the radial basis function (RBF) method, which is widely used to parameterize nonrigid transformation. This formulation means that the nonrigid transformation is assumed to be a displacement shifted from its initial position. In this paper, we utilize the Gaussian RBF, i.e., $\varphi_i(V) \triangleq \exp(-\|V - V_i\|_2^2/\sigma_w^2)$, where σ_w is the bandwidth dependent on the degree of deformation. Then, the kernel is computed as $\mathbf{K}_e(i, j) = \varphi_i(V_j)$. Following some previous works, we set the regularization term as $\Upsilon(\tau) = \text{Tr}(\mathbf{W}^T\mathbf{K}_e\mathbf{W})$ to penalize the nonsmoothness of nonrigid deformation.

Next, We demonstrate our FRGM-D method in the following.

5.2 Function composition-based method

With the geometric deformation $\tau : \mathcal{V}_1 \rightarrow \mathcal{V}_2$, we aim to find a linear representation map \mathbf{P} of matching $\mathcal{T} : \mathcal{V}_1 \rightarrow \mathcal{V}_2$, which remains

consistent with τ such that the composition of \mathcal{T} and τ^{-1} is an identity function \mathbf{I}_d :

$$\tau^{-1} \circ \mathcal{T} = \mathbf{I}_d : \mathcal{V}_1 \rightarrow \mathcal{V}_1. \quad (27)$$

According to the associative property of matrix multiplication, the composition $\tau^{-1} \circ \mathcal{T}$ can be rewritten as follows:

- For similarity transformation

$$\begin{aligned} \tau^{-1} \circ \mathcal{T}(\mathcal{V}_1) &= \frac{1}{s} (\mathbf{P}\mathcal{V}_2)\mathbf{R}^{-1} - \frac{1}{s} \mathbf{1}_m \mathbf{t} \mathbf{R}^{-1} \\ &= \frac{1}{s} \mathbf{P}(\mathcal{V}_2 \mathbf{R}^{-1}) - \frac{1}{s} \mathbf{1}_m \mathbf{t} \mathbf{R}^{-1} \end{aligned} \quad (28)$$

- For affine transformation

$$\begin{aligned} \tau^{-1} \circ \mathcal{T}(\mathcal{V}_1) &= (\mathbf{P}\mathcal{V}_2)\mathbf{A}^{-1} - \mathbf{1}_m \mathbf{t} \mathbf{A}^{-1} \\ &= \mathbf{P}(\mathcal{V}_2 \mathbf{A}^{-1}) - \mathbf{1}_m \mathbf{t} \mathbf{A}^{-1} \end{aligned} \quad (29)$$

- For nonrigid transformation

$$\tau^{-1} \circ \mathcal{T}(\mathcal{V}_1) = (\mathbf{P}\mathcal{V}_2) - \mathbf{K}_e \mathbf{W}. \quad (30)$$

Consequently, this associative property allows us to estimate \mathbf{P} and the parameters of τ alternately.

The alternating estimations of \mathcal{T} and τ are as follows. First, we use the identity function \mathbf{I}_d for the initialization of τ . In the alternating steps, once τ is given, we update the graph as $\mathcal{V}_1 \leftarrow \tau(\mathcal{V}_1)$ and then apply our algorithm FRGM-E to find the correspondence between \mathcal{V}_1 and \mathcal{V}_2 . After \mathbf{P} is given, we recover the parameters of τ by minimizing the objective function as:

$$\begin{aligned} J(\tau) &= \sum_i \|V_i^{(1)} - \tau^{-1}((\mathbf{P}\mathcal{V}_2)_i)\|^2 \\ &+ \lambda \sum_{(i_1, i_2)} \mathcal{E}_{i_1 i_2} \|(V_{i_1}^{(1)} - V_{i_2}^{(1)}) - \tau^{-1}((\mathbf{P}\mathcal{V}_2)_{i_1} - (\mathbf{P}\mathcal{V}_2)_{i_2})\|^2. \end{aligned} \quad (31)$$

Given the correspondence \mathbf{P} , the parameters of τ can be computed in closed form as follows for different transformations:

- For similarity transformation: The optimal translation vector \mathbf{t}^* can be represented as a function of \mathbf{R}^* and s^* as

$$\mathbf{t}^* = \mathbf{P}\bar{\mathcal{V}}_2 - s^*\bar{\mathcal{V}}_1\mathbf{R}^*, \quad \bar{\mathcal{V}}_1 = \frac{\mathbf{1}_m^T \mathcal{V}_1}{m}, \quad \bar{\mathcal{V}}_2 = \frac{\mathbf{1}_n^T \mathcal{V}_2}{n}. \quad (32)$$

By the centralization of points, $\mathcal{V}_1 \leftarrow \mathcal{V}_1 - \mathbf{1}_m \bar{\mathcal{V}}_1$ and $\mathcal{V}_2 \leftarrow \mathcal{V}_2 - \mathbf{1}_n \bar{\mathcal{V}}_2$, we have:

$$\mathbf{R}^* = (\text{Udiag}(1, \dots, |\mathbf{U}\mathbf{V}^T|) \mathbf{V}^T)^{-1}. \quad (33)$$

$$s^* = \frac{\text{Tr}[(\mathbf{P}\mathcal{V}_2)^T (\mathbf{I}_m + \lambda \mathbf{L}_1)(\mathbf{P}\mathcal{V}_2)]}{\text{Tr}[\mathcal{V}_1^T (\mathbf{I}_m + \lambda \mathbf{L}_1)(\mathbf{P}\mathcal{V}_2)\mathbf{R}^*]}, \quad (34)$$

where $\mathbf{U}\Sigma\mathbf{V}^T = \text{svd}(\mathcal{V}_1^T (\mathbf{I}_m + \lambda \mathbf{L}_1)(\mathbf{P}\mathcal{V}_2))$ and $\mathbf{L}_1 = \text{diag}(\mathcal{E}_1 \mathbf{I}) - \mathcal{E}_1$.

- For affine transformation: The parameters can be computed as:

$$\mathbf{t}^* = \mathbf{P}\bar{\mathcal{V}}_2 - \bar{\mathcal{V}}_1\mathbf{A}^*, \quad (35)$$

$$\mathbf{A}^* = \frac{(\mathbf{P}\mathcal{V}_2)^T (\mathbf{I}_m + \lambda \mathbf{L}_1)(\mathbf{P}\mathcal{V}_2)}{\mathcal{V}_1^T (\mathbf{I}_m + \lambda \mathbf{L}_1)(\mathbf{P}\mathcal{V}_2)}, \quad (36)$$

with centralized points $\mathcal{V}_1 \leftarrow \mathcal{V}_1 - \mathbf{1}_m \bar{\mathcal{V}}_1$ and $\mathcal{V}_2 \leftarrow \mathcal{V}_2 - \mathbf{1}_n \bar{\mathcal{V}}_2$.

- For nonrigid transformation: We choose points $\mathcal{V}_1 = \{V_i^{(1)}\}_{i=1}^m$ as the basis points to compute the kernel matrix \mathbf{K}_e . Note that, a regularization term $\sigma^2 \text{Tr}(\mathbf{W}^T \mathbf{K}_e \mathbf{W})$ is added to $J(\tau)$. After centralizing the points, the optimal solution \mathbf{W}^* is

$$\mathbf{W}^* = -\frac{(\mathcal{V}_1 - \mathbf{P}\mathcal{V}_2)^T (\mathbf{I}_m + \lambda \mathbf{L}_1)}{\mathbf{K}_e (\mathbf{I}_m + \lambda \mathbf{L}_1) + \sigma^2}, \quad (37)$$

where $\sigma^2 = \frac{1}{mn} \sum_{ij} \mathbf{P}_{ij} \|V_i^{(1)} - V_j^{(2)}\|^2$ is used to avoid the singularity of matrix division in Eqn. (37).

6 NUMERICAL ANALYSIS

In this section, we discuss the optimization strategy for solving the proposed algorithms. We first introduce an efficient optimization algorithm based on the Frank-Wolfe method. Then, we propose an entropy regularization-based approximation of the Frank-Wolfe method to further reduce the time complexity.

6.1 The Frank-Wolfe method

The objective functions proposed in the previous sections may be either convex or nonconvex, and the feasible field $\hat{\mathcal{P}}$ is a convex and compact set. The Frank-Wolfe (FW) method (also known as conditional gradient) [56], [57] has been well studied for solving constraint convex or nonconvex optimization problems with at least a sublinear convergence rate.

Given that $f(\mathbf{P})$ is differentiable with an L -Lipschitz gradient and $\hat{\mathcal{P}}$ is a convex and compact set, the FW method iterates the following steps until it converges:

$$\tilde{\mathbf{P}}^{(k+1)} \in \underset{\mathbf{P} \in \hat{\mathcal{P}}}{\operatorname{argmin}} f^{(k)}(\mathbf{P}) \triangleq \langle \nabla f(\mathbf{P}^{(k)}), \mathbf{P} \rangle, \quad (38)$$

$$\mathbf{P}^{(k+1)} = \mathbf{P}^{(k)} + \alpha^{(k)} (\tilde{\mathbf{P}}^{(k+1)} - \mathbf{P}^{(k)}), \quad (39)$$

where $\alpha^{(k)}$ is the step size obtained by exact or inexact line search [58], and $\nabla f(\mathbf{P}^{(k)})$ is the gradient of f at $\mathbf{P}^{(k)}$.

In Eqn. (38), the minimizer $\tilde{\mathbf{P}}^{(k+1)} \in \hat{\mathcal{P}}$ is theoretically an extreme point of $\hat{\mathcal{P}}$ (thus, it is binary). This means that $\tilde{\mathbf{P}}^{(k+1)} \in \mathcal{P}$. Therefore, Eqn.(38) is a linear assignment problem (LAP) that can be efficiently solved by approaches such as the Hungarian [25] and LAPJV [59] algorithms. Moreover, since $\tilde{\mathbf{P}}^{(k+1)}$ is binary in each iteration, the final solution \mathbf{P}^* can be (nearly) binary.

Time complexity of the FW method. The time complexity can be roughly calculated as $\mathbf{O}(T(\tau_f + \tau_s + \tau_h))$, where $\tau_s = \mathbf{O}(mn + m^2 + n^2)$ is the cost of the unary term and edge attributes for graphs, $\tau_h = \mathbf{O}(n^3)$ is the cost of the Hungarian algorithm used as a post-discretization step, T is the number of iterations. In each iteration, $\tau_f = \mathbf{O}(m^2 n)$ is the cost to compute the gradient, function value and step size at $\mathbf{P}^{(k)}$, and $\tau_l = \mathbf{O}(n^3)$ is the cost to compute LAP using the Hungarian or LAPJV algorithm. Note that since $\tau_f = \mathbf{O}(m^2 n)$ is computed in closed form, it takes much less time compared to $\tau_l = \mathbf{O}(n^3)$. Since $m \leq n$, the time complexity approximately equals $\mathbf{O}(Tn^3)$ with the maximum number of iterations T .

6.2 A fast approximated FW method

Similar to the above analysis, the time complexity of applying the FW method to solve FRGM-D is roughly $\mathbf{O}(kTn^3)$, where k is the number of alternations to estimate the correspondence \mathbf{P} and transformation τ . Note that we can neglect the computation for the parameters of τ because it is calculated in closed form and is much less than the cost of computing \mathbf{P} . Therefore, the time complexity $\mathbf{O}(kTn^3)$ is mainly caused by solving the LAP. To achieve faster execution and proper approximation, we approximate the original Frank-Wolfe method based on the generalized conditional gradient algorithm [60] by adding a convex entropy regularization term in each iteration of solving LAP. The approximated Frank-Wolfe method (AFW) is defined as follows:

$$\hat{\mathbf{P}}^{(k+1)} \in \underset{\mathbf{P} \in \hat{\mathcal{P}}}{\operatorname{argmin}} f_\epsilon^{(k)}(\mathbf{P}) \triangleq \langle \nabla f(\mathbf{P}^{(k)}), \mathbf{P} \rangle - \epsilon H(\mathbf{P}), \quad (40)$$

$$\mathbf{P}_\epsilon^{(k+1)} = \mathbf{P}^{(k)} + \hat{\alpha}^{(k)} (\hat{\mathbf{P}}^{(k+1)} - \mathbf{P}^{(k)}), \quad (41)$$

where $H(\mathbf{P}) = -\sum_{ij} \mathbf{P}_{ij} \ln(\mathbf{P}_{ij})$ is the entropy of \mathbf{P} . To minimize Eqn. (40), we can use the Sinkhorn method [45] as a fast implementation, which has a time complexity of $\mathbf{O}(mn)$.

With the entropy regularization $H(\mathbf{P})$, we can approximate the original FW method well within a given tolerance:

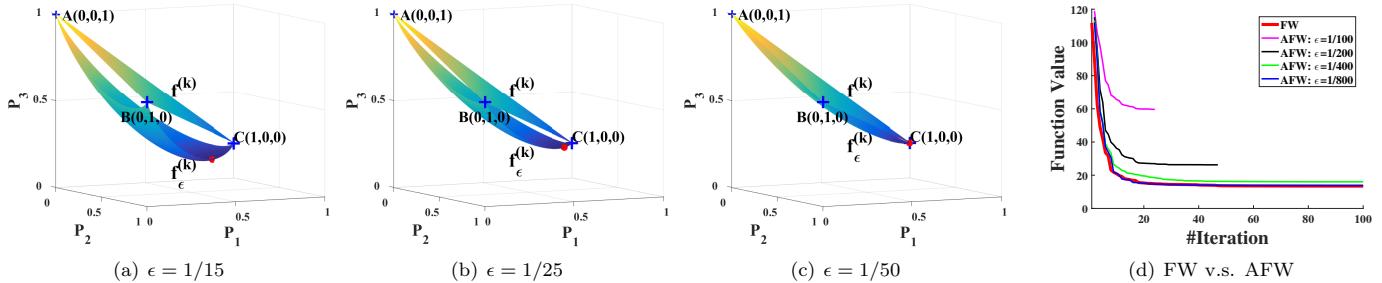


Fig. 5. Comparison between FW and AFW. A toy example in (a)–(c), where $\hat{\mathcal{P}} \subseteq \mathbb{R}_+^{1 \times 3}$, $\mathbf{P} = (\mathbf{P}_1, \mathbf{P}_2, \mathbf{P}_3)$, and the hyperplane $\nabla f(\mathbf{P}^{(k)}) = (0.3, 0.4, 0.5)$. There are three extreme points of $\hat{\mathcal{P}}$: $\mathbf{A} = (0, 0, 1)$, $\mathbf{B} = (0, 1, 0)$, $\mathbf{C} = (1, 0, 0)$. From (a) to (b), $\epsilon = \frac{1}{15}, \frac{1}{25}, \frac{1}{50}$, respectively. $f^{(k)}(\mathbf{P})$ reaches its minimum at $\mathbf{C} = (1, 0, 0)$, and $\{f_\epsilon^{(k)}(\mathbf{P})\}_\epsilon$ reach their minima at the red dots. A cool color means a small value. It shows that during the iterations, the solution of $f_\epsilon^{(k)}(\mathbf{P})$ gradually approximates the solution of $f^{(k)}(\mathbf{P})$ with ϵ tending to be smaller. In (d), we show a real example from Sec. 7.4. The function values of objective functions calculated by AFW tend to be equal to the values obtained by FW when ϵ becomes smaller.

Proposition 4. The solution $\hat{\mathbf{P}}^{(k+1)}$ tends to $\tilde{\mathbf{P}}^{(k+1)}$ as $\epsilon \rightarrow 0$.

$$\|\hat{\mathbf{P}}^{(k+1)} - \tilde{\mathbf{P}}^{(k+1)}\| \leq \frac{\sqrt{mn}}{\epsilon} e^{-\frac{c}{\epsilon}}, \quad (42)$$

where $c \in [0, 1]$ is a constant dependent on m, n and $\nabla f(\mathbf{P}^{(k)})$.

The proof can be given based on the primal-dual analysis for linear programming [61], see appendix ?? for more details. Therefore, with $\epsilon > 0$ small enough, we can obtain a good approximation. We can prove that the AFW method achieves at least a sublinear convergence rate inspired by [56]. And a short proof can be found in appendix ??.

Proposition 5. Assume that $f(\mathbf{P})$ is differentiable with an L -Lipschitz gradient; by choosing a series of $\epsilon_k \leq \frac{1}{k+1}$, the AFW method ensures at least a sublinear convergence rate.

$$0 \leq f(\mathbf{P}_\epsilon^{(k+1)}) - f(\mathbf{P}^*) \leq \frac{2(LC^2 + mln(m))}{k+2}, \quad (43)$$

where \mathbf{P}^* is the ideal solution of $f(\mathbf{P})$ and $C = \text{diam}(\hat{\mathcal{P}})$ is the diameter of the feasible field $\hat{\mathcal{P}}$.

Another reason for choosing $H(\mathbf{P})$ is that $H(\mathbf{P}) = 0$ when \mathbf{P} is an extreme point of $\hat{\mathcal{P}}$, i.e., a binary correspondence between graphs. This means that $f_\epsilon^{(k)}(\mathbf{P})$ has the same function value as $f^{(k)}(\mathbf{P})$ at any extreme point. See Fig. 5 for an illustration. Note that, in the implementation of optimization, the iteration procedure will stop once it reaches the maximal number of iterations ($T = 100$) or the absolute difference between two successive function values is less than 10^{-5} .

Time complexity of the AFW method Similar to the FW method, the time complexity of the AFW method can be roughly calculated as $\mathbf{O}(T(\tau_f + \tau_l') + \tau_s + \tau_h)$, where $\tau_l' = \mathbf{O}(mn)$ is the cost to compute Eqn. (40) using the Sinkhorn method.

7 EXPERIMENTAL ANALYSIS

In this section, we evaluate our functional-representation-based GM methods, i.e. the general GM (FRGM-G), Euclidean GM (FRGM-E) and deformable GM (FRGM-D) algorithms.

In Sec. 7.1–Sec. 7.3, we compare FRGM-G and FRGM-E to several state-of-the-art GM algorithms, including GA [35], PM [28], SM [11], SMAC [30], IPFP-S [23], RRWM [12], FGM-D [19], MPM [44] and DTA [38]. In Sec. 7.4, we compare FRGM-D with several state-of-the-art point registration algorithms, including GLS [53], GMM [52], CPD [51]. Note that in Sec. 7.4 where we evaluate our FRGM-D on geometrically deformed graphs, we do not compare the GM algorithms used in Sec. 7.1–Sec. 7.3 because those methods can neither be directly used for this task nor handle graphs with significant geometric deformations. All comparisons are conducted on both synthetic and real-world datasets that are commonly used to evaluate GM or point registration algorithms. We obtained the codes of the compared methods from the author’s websites and implemented all the experiments on a desktop with a 3.5GHz Intel Xeon CPU E3-1240 and 16 GB memory.

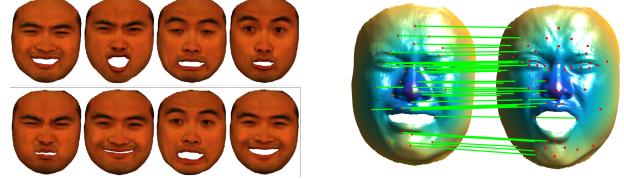


Fig. 6. Left: Some instances of 3D faces. Right: an example of matching unequal-sized graph pairs with sizes (40, 50).

7.1 Results on 3D face

This section evaluated our algorithm FRGM-G with experiments conducted on graphs in low-dimensional manifold, i.e., 3D face. We both (1) evaluated FRGM-G w.r.t. varying graph settings and (2) compared FRGM-G with other state-of-the-art GM methods.

The experiments were performed on 383 continuous frames of 3D faces [62] with gradually changing expressions. We selected 38 frames whose expressions were more dissimilar to each other, and each frame was marked with 50 landmarks as the ground-truth. Some examples are shown in Fig. 6. For each pair of faces, we construct two graphs \mathcal{G}_1 and \mathcal{G}_2 with node attributes $\{\mathbf{v}_i^{(1)}\}_{i=1}^m$ and $\{\mathbf{v}_j^{(2)}\}_{j=1}^n$ consisting of the HKS [49] feature descriptors. The edge attribute matrices \mathbf{E}_1 and \mathbf{E}_2 were computed as the geodesic distance between graph nodes on the faces. For the implementation of FRGM-G, the unary term measuring node dissimilarity is computed as $\mathbf{U}_{ij} \triangleq \|\mathbf{v}_i^{(1)} - \mathbf{v}_j^{(2)}\|$. We updated the raw matrices \mathbf{E}_1 and \mathbf{E}_2 into $\hat{\mathbf{E}}_1 = \exp(-\mathbf{E}_1^2/\sigma^2)$ and $\hat{\mathbf{E}}_2 = \exp(-\mathbf{E}_2^2/\sigma^2)$ to honor the inner product illustrated in Proposition 1. Then, we used $\hat{\mathbf{E}}_1$ and $\hat{\mathbf{E}}_2$ to compute the functionals, i.e., inner products $\mathbb{F}_{\mathcal{V}_1}(\cdot, \cdot)$ and $\mathbb{F}_{\mathcal{V}_2}(\cdot, \cdot)$ defined in Eqn. (8). We computed \mathbf{D} with Eqn. (10) and set $\alpha_1 = 0.99, \alpha_2 = 0.5$.

We evaluated our FRGM-G in two aspects. First, we conducted experiments on graphs with varying $\sigma = 0.05, 0.1, \dots, 1$, whose results are reported in Fig. 7 (a). It shows that, with $\sigma \approx [0.3, 0.5]$, FRGM-G usually achieves higher accuracy. This is consistent with the analysis of eigenvalues in Fig. 2. Second, to evaluate FRGM-G on graphs with varying edge densities, we used k -nn graphs, i.e., each node was connected by the k -nearest neighborhood nodes to generate adjacency matrices \mathcal{E}_1 and \mathcal{E}_2 . The edge density was determined by k , which was set to 10%, 20%, ..., 100% of the number of nodes (m, n). The results (with $\sigma = 0.5$) are shown in Fig. 7 (b). In this experiment, FRGM-G achieves higher accuracy with more edges because the geometric structures such as inner product or metric will be more complete with more edges.

We then compared our FRGM-G (with $\sigma = 0.5$) and the other GM algorithms with complete graphs of sizes (m, n) . For the compared methods, the node affinity was computed as $\mathbf{K}_{ij;ij} = \exp(-\|\mathbf{v}_i^{(1)} - \mathbf{v}_j^{(2)}\|)$, and the edge affinity was computed as $\mathbf{K}_{i_1 j_1; i_2 j_2} = \exp(-(\mathbf{E}_{1_{i_1 i_2}} - \mathbf{E}_{2_{j_1 j_2}})^2/500)$, as used similarly in [19]. The comparison results are shown in Tab. 2. Because the

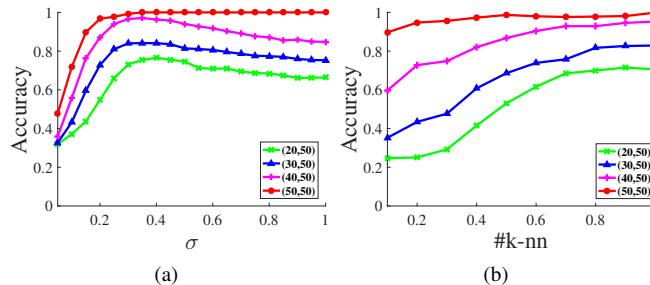


Fig. 7. Left: evaluation results on parameter $\sigma \in (0, 1]$. Right: results on k -nn connected graph pairs with varying edge densities.

TABLE 2

Comparison results of average accuracy (%) on the 3D face dataset.

(m,n)	(25,50)	(30,50)	(35,50)	(40,50)	(45,50)	(50,50)
GA [35]	12.00	15.41	21.39	29.26	39.94	50.54
PM [28]	6.70	7.03	11.27	16.01	22.16	33.30
SM [11]	11.24	11.62	16.76	23.18	37.96	54.59
SMAC [30]	26.49	33.69	45.87	58.11	68.41	82.43
IPFP-S [23]	11.35	7.12	5.95	10.07	7.63	69.19
RRWM [12]	19.14	28.56	44.71	55.47	65.59	90.05
FGM-D [19]	46.81	59.91	75.06	84.66	92.01	99.78
MPM [44]	8.32	11.17	16.76	29.39	40.66	52.76
DTA [38]	37.25	41.80	49.36	53.64	57.53	61.41
FRGM-G	76.14	79.88	89.16	91.58	98.04	100.00

geometric structures defined on function spaces of graphs are more efficient for representing the distinguishing feature of graphs, our proposed FRGM-G achieves much higher average accuracy in both equal-sized and unequal-sized cases.

7.2 Results on synthetic data

In this section, we performed a comparative evaluation of both FRGM-G and FRGM-E on synthesized graphs following [12], [19], [20]. The synthetic nodes of \mathcal{G}_1 and \mathcal{G}_2 were generated as follows: for graph \mathcal{G}_1 , n_{in} inlier points were randomly generated on \mathbb{R}^2 with Gaussian distribution $\mathcal{N}(0, 1)$. Graph \mathcal{G}_2 with noise was generated by adding Gaussian noise $\mathcal{N}(0, \sigma^2)$ to each $V_i^{(1)}$ to evaluate the robustness to noise. Graph \mathcal{G}_2 with outliers was generated by adding n_{out} additional points on \mathbb{R}^2 with a Gaussian distribution $\mathcal{N}(0, 1)$ to evaluate the robustness to outliers.

For the compared methods, we computed the node affinity as $\mathbf{K}_{ij;ij} = \exp(-\|\mathbf{v}_i^{(1)} - \mathbf{v}_j^{(2)}\|)$ with $\mathbf{v}_i^{(1)}, \mathbf{v}_j^{(2)}$ computed by shape context [3], and we computed the edge affinity as $\mathbf{K}_{i_1j_1;i_2j_2} = \exp(-(\|V_{i_1}^{(1)} - V_{i_2}^{(1)}\| - \|V_{j_1}^{(2)} - V_{j_2}^{(2)}\|)^2/0.15)$ as used in [19]. For FRGM-G, we computed the unary term \mathbf{U} by shape context and set $\alpha_1 = 0.99, \alpha_2 = 0.5$. For FRGM-E, we set $\lambda_1 = 0.99, \lambda_2 = 0.5$. To compute the matrix $\mathbf{S} \in \mathbb{R}^{m \times m}$ that indicated the adjacent nodes in $\{\mathcal{T}(V_i^{(1)})\}_{i=1}^m$, we performed a Delaunay triangulation on \mathcal{V}_1 to connect the edges. Then these edges were divided into two parts using k-means by considering the edge length. Edges with longer lengths were then abandoned.

Average accuracy. Since the compared methods have a considerable space complexity of $O(m^2 n^2)$ when the graphs are fully connected, they can hardly handle complete graphs with more than a hundred nodes. Therefore, for fairness, we performed the experiment on two types of graphs: smaller graphs that were fully connected and larger graphs that were connected by Delaunay triangulation. We first applied all methods to complete graphs of size $n_{in} = 20$ with either the noise level σ varying from 0 to 0.5 (by intervals of 0.05) or number of outliers n_{out} varying from 0 to 20 (by intervals of 2). Then, we enlarged the size of the graphs to $n_{in} = 100$ and connected the edges by Delaunay triangulation. Similarly, noise and outliers were added.

As shown in Fig. 8 (a) and (b), under the complete graph setting, our algorithms FRGM-G and FRGM-E achieve higher average

accuracy than the other algorithms in the case with noise and achieve competitive results in the case with outliers, respectively. As shown in Fig. 8 (c) and (d), with larger graphs connected by Delaunay triangulation, both FRGM-G and FRGM-E outperform all the other methods. Moreover, we can observe that all algorithms achieve higher accuracy on complete graphs than graphs connected by Delaunay triangulation.

Running time. To compare the time consumptions of all methods, we tested all methods on graphs with inliers varying as $n_{in} = 10, 20, \dots, 100$ and noise $\sigma = 0.2$. Considering the effect of the number of edges on time consumption, we used both complete and Delaunay-triangulation-connected graphs.

As shown in Fig. 9, under the same conditions in which graphs are either complete or connected through Delaunay triangulation, our algorithms FRGM-G and FRGM-E achieve higher average accuracy within an intermediate running time. For all methods, matching complete graphs consumes more time than Delaunay-triangulation-connected graphs. Compared with GA, SM, PM, SMAC, and IPFP-S, which run faster, our method can achieve higher average accuracy. The methods RRWM, FGM and MPM can achieve competitive accuracy when the graphs are fully connected. However, the time consumptions of these methods rapidly increase and become larger than that of ours method, and they will take an unacceptable amount time to match complete graphs that have more than a hundred nodes.

Large-scale graph matching. To test the efficiency of FRGM-G and FRGM-E on large-scale graphs, we used more challenging settings for evaluation. We set the number of inliers as $n_{in} = 100, 300, 500, 1000$ with Gaussian noise or outliers. The number of outliers was set to 20%, 40%, ..., 100% of the number of inliers.

Tab. 3 reports the results of FRGM-G and FRGM-E on complete graphs with hundreds and thousands of nodes. Both FRGM-G and FRGM-E are very robust to outliers and less robust to strong noise with larger graphs. FRGM-E is much faster than FRGM-G because FRGM-E searches for the optimal solution upon a simpler Euclidean space \mathbb{R}^d , while FRGM-G searches for the optimal solution upon a more complex function space $\mathcal{F}(\mathcal{V}_2, \mathbb{R})$. Since the compared methods need to store affinity matrices with a size of approximately $n_{in}^2(n_{in} + n_{out})^2$, applying these methods to large-scale graphs with hundreds or thousands of nodes is infeasible.

7.3 Results on real-world image datasets

We also performed comparative evaluations on real-world datasets, including the CMU House and Hotel sequences, the PASCAL Cars and Motorbikes pairs [41], which are commonly used to evaluate GM algorithms. All methods are applied to complete graphs. The CMU House and Hotel sequences consist of 111 and 101 frames of a synthetic house and hotel, respectively. Each image contains 30 points that are manually marked with known correspondence. In this experiment, we matched all the image pairs separated by 10, 20, ..., 90 frames. The unequal-sized cases are set as 20-vs-30 and 25-vs-30. For the compared methods, we computed the node affinity $\mathbf{K}_{ij;ij} = \exp(-\|\mathbf{v}_i^{(1)} - \mathbf{v}_j^{(2)}\|)$ with shape context, and we computed the edge affinity as $\mathbf{K}_{i_1j_1;i_2j_2} = \exp(-(\|V_{i_1}^{(1)} - V_{i_2}^{(1)}\| - \|V_{j_1}^{(2)} - V_{j_2}^{(2)}\|)^2/2500)$ used in [19].

Average accuracy. As shown in Fig. 11, for the house sequence, our algorithms FRGM-G and FRGM-E achieve higher accuracy in both equal-sized and unequal-sized cases. For the hotel sequence, FRGM-G outperforms all the other methods.

The PASCAL dataset consists of 30 pairs of car images and 20 pairs of motorbike images. Each pair contains both inliers with known correspondence and randomly selected outliers. In the unequal-sized case, we added 5, 10, 15, and 20 outliers to \mathcal{G}_2 . For the compared methods, we computed the same node affinity used on the CMU dataset, and we computed the edge affinity with the edge length and edge angle, which were used in [19].

Outlier removal effectiveness. The outliers occurring in graphs often seriously affect the matching accuracy. We tested our proposed outlier-removal strategy on PASCAL dataset. We first applied it to

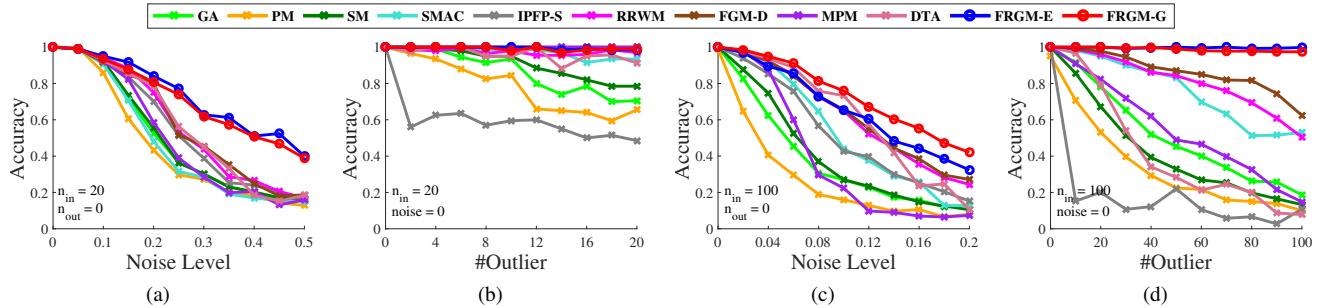


Fig. 8. Comparisons of the robustness to noise and outliers. For complete graphs, the accuracies with respect to the noise and number of outliers are shown in (a) and (b), respectively. The results for graphs connected by Delaunay triangulation are shown in (c) and (d). FRGM-G and FRGM-E outperform all the others for graphs with noise and outliers.

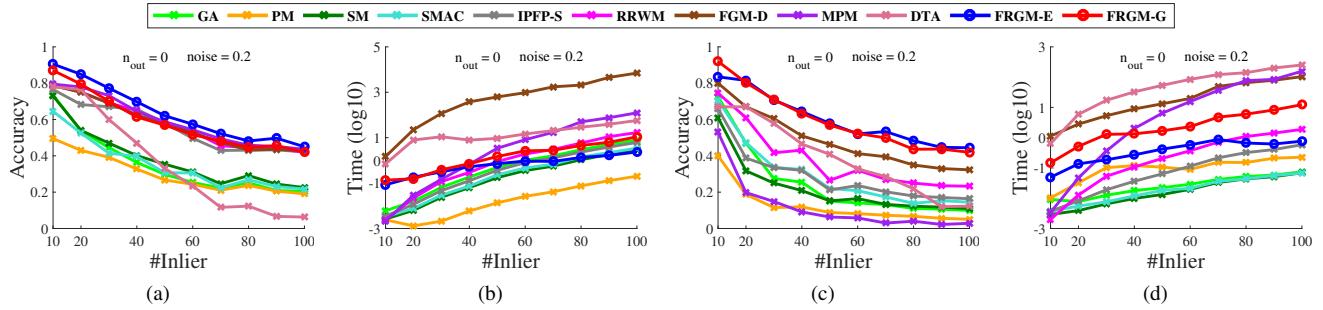


Fig. 9. Comparisons of running time and average accuracy. The graphs in (a) and (b) are complete, and the graphs in (c) and (d) are connected through Delaunay triangulation. FRGM-G and FRGM-E outperform all the others in terms of matching accuracy with modest running time.

TABLE 3
Average accuracy and running time of FRGM-G (left) v.s. FRGM-E (right) on synthetic data with varying inliers, noise and outliers.

#Inlier	Noise (σ)	0.02	0.04	0.06	0.08	0.10
100	time (s)	0.07 / 0.11	0.11 / 0.32	0.14 / 0.54	0.24 / 0.73	0.25 / 0.88
	acc. (%)	98.38 / 98.86	94.28 / 95.70	88.74 / 90.70	81.10 / 85.20	72.16 / 75.76
300	time (s)	2.07 / 4.43	3.78 / 6.80	7.88 / 7.06	15.20 / 7.29	19.18 / 7.34
	acc. (%)	95.33 / 96.34	85.08 / 87.66	71.32 / 76.11	60.17 / 63.73	48.60 / 50.79
500	time (s)	18.10 / 27.18	34.43 / 27.83	86.30 / 28.94	161.14 / 29.61	214.29 / 30.33
	acc. (%)	93.14 / 94.35	77.62 / 80.27	60.32 / 63.80	46.74 / 49.70	37.44 / 39.14
1000	time (s)	120.85 / 124.66	233.14 / 179.87	426.40 / 184.53	711.78 / 187.67	1810.42 / 191.76
	acc. (%)	88.16 / 89.43	63.29 / 66.34	43.79 / 45.23	32.20 / 33.47	24.23 / 25.27
#Inlier	#Outlier	0.2	0.4	0.6	0.8	1.0
100	time (s)	0.06 / 0.03	0.08 / 0.04	0.10 / 0.07	0.15 / 0.14	0.22 / 0.17
	acc. (%)	99.86 / 99.98	99.74 / 99.88	99.22 / 99.85	98.77 / 99.76	98.08 / 99.68
300	time (s)	1.26 / 0.18	2.04 / 0.34	5.53 / 0.74	7.87 / 1.17	10.33 / 2.02
	acc. (%)	99.90 / 99.98	99.66 / 99.92	99.33 / 99.93	98.66 / 99.83	98.19 / 99.83
500	time (s)	27.72 / 0.97	48.34 / 2.01	74.65 / 3.43	136.82 / 5.89	222.28 / 10.91
	acc. (%)	99.94 / 99.95	99.86 / 99.95	99.44 / 99.89	98.18 / 99.87	97.54 / 99.96
1000	time (s)	181.63 / 4.48	337.41 / 10.99	461.22 / 30.94	502.51 / 54.95	626.03 / 77.72
	acc. (%)	99.92 / 99.99	99.68 / 99.97	99.16 / 99.97	97.95 / 99.97	97.05 / 99.96

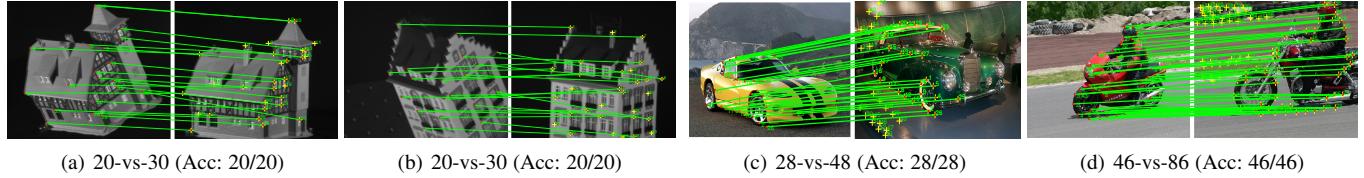


Fig. 10. Examples of matching unequal-sized graphs using FFRGM-G (in (a) and (b)) and FRGM-E (in (c) and (d)). The red dots are inliers in \mathcal{G}_1 , and the yellow plus signs are inliers with outliers in \mathcal{G}_2 . The lines in green are correct matches.

graph pairs with outliers as a preprocessing step, and then we executed all the algorithms on the preprocessed graphs. As shown in Tab. 4, the average accuracy of all the methods is greatly improvement, and almost all the methods improve their performance by more than 10%. Moreover, as shown in Fig. 12, FRGM-G achieves the highest average accuracy, and FRGM-E achieves a competitive result.

7.4 Results on geometrically deformed graphs

In this section, we evaluated our algorithm FRGM-D for matching graphs with geometric deformations. We chose 5 templates: Olympic logo (113 nodes), whale (150 nodes), Chinese character (105 nodes), tropical fish (91 nodes) and UCF fish (98 nodes), which have been widely used by registration methods [51], [52], [63]. Fig. 13 shows

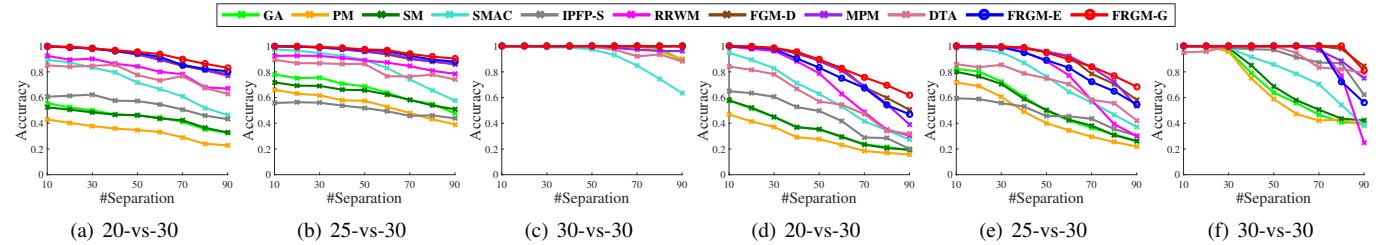


Fig. 11. Comparison of average accuracy on the *house* ((a)-(c)) and *hotel* ((d)-(f)) sequences in both equal-sized and unequal-sized cases. FRGM-G achieves higher average accuracies than the other methods.

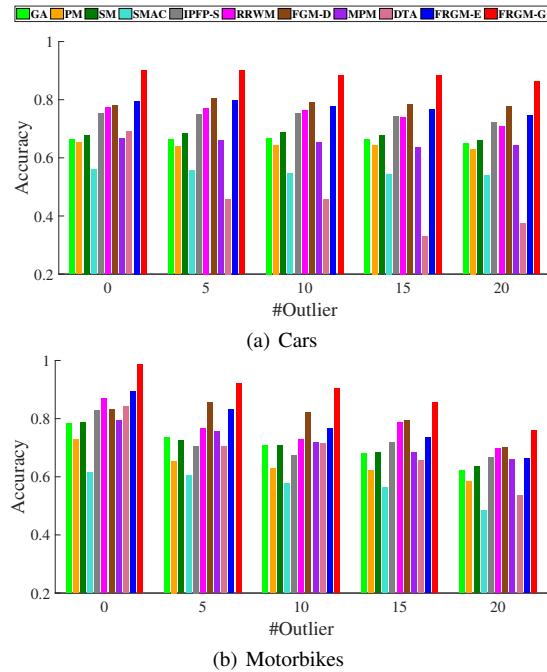


Fig. 12. Comparison on car and motorbike image pairs with outliers. FRGM-G achieves the highest average accuracy.

TABLE 4

Effectiveness of outlier-removal strategy. It improves the average matching accuracy by more than 10% for almost all the methods.

Method	Out. Re.	Cars	Motorbikes
GA [35]	w/o	34.50	45.97
	w/	66.14	70.60
PM [28]	w/o	37.04	43.56
	w/	64.21	64.26
SM [11]	w/o	38.04	47.13
	w/	67.72	70.75
SMAC [30]	w/o	38.53	43.84
	w/	54.91	56.89
IPFP-S [23]	w/o	38.53	43.84
	w/	74.36	71.84
RRWM [12]	w/o	53.84	65.64
	w/	75.09	76.92
FGM-D [19]	w/o	49.05	67.31
	w/	78.72	80.01
MPM [44]	w/o	58.02	65.73
	w/	65.11	72.17
DTA [38]	w/o	46.24	69.03
	w/	66.33	73.99
FRGM-E	w/o	32.74	43.19
	w/	77.67	77.74
FRGM-G	w/o	55.02	68.19
	w/	88.59	88.51

some results obtained by FRGM-D, in which the graphs are disturbed by geometric deformations, noises, missing points and outliers.

Robustness to deformations. For each template denoted as graph \mathcal{G}_1 , the graph \mathcal{G}_2 was generated by adding geometric deformations to

\mathcal{G}_1 . To evaluate the robustness to rotation and scaling, we rotated the template \mathcal{G}_1 by varying degrees in $[-\pi, \pi]$ and scaled \mathcal{G}_1 with varying scaling factors in $[0.1, 1]$ and $[2, 20]$. In addition, the graph \mathcal{G}_2 was also disturbed with noise with distribution $\mathcal{N}(0, 0.02)$ and outliers $n_{out} = 100$ with distribution $\mathcal{N}(0, 0.25)$. To evaluate the robustness to nonrigid deformations, we deformed \mathcal{G}_1 by weight matrix \mathbf{W} with distribution $\mathcal{N}(0, \sigma^2)$ with varying $\sigma \in [0, 0.5]$ and abandoned the extremely deformed \mathcal{G}_2 . Then, some slight noises with distribution $\mathcal{N}(0, 0.02)$ were also added. For our algorithm FRGM-D, in the alternation that estimated the correspondence \mathbf{P} , the unary term \mathbf{U} was computed using the rotation-invariant shape context that was also used in some other works [53], [54]. Moreover, we solved FRGM-D by the proposed AFW method.

For all methods, we computed the average error between each point $\tau(V_i^{(1)})$ and the corresponding point $V_{\delta_i}^{(2)}$, i.e., $\frac{1}{m} \sum_i \|\tau(V_i^{(1)}) - V_{\delta_i}^{(2)}\|$. Moreover, to evaluate the parameterization \mathcal{T} (i.e., binary correspondence \mathbf{P}) obtained by FRGM-D, we also reported the average error between $\mathcal{T}(V_i^{(1)})$ and the correspondence $V_{\delta_i}^{(2)}$, i.e., $\frac{1}{m} \sum_i \|\mathcal{T}(V_i^{(1)}) - V_{\delta_i}^{(2)}\|$. We denoted these two types of average errors as FRGM-D1 and FRGM-D2 for our algorithm. As shown in Fig. 14 (a) and (b), our algorithm is more robust to rotation and scaling factor. As shown in Fig. 14 (c), FRGM-D1 is competitive with GMM and CPD, and FRGM-D2 has less average errors due to the well-estimated \mathbf{P} .

Robustness to noise and outliers. In this experiment, each template \mathcal{G}_1 was first deformed with similarity and nonrigid deformations to obtain \mathcal{G}_2 . Then, \mathcal{G}_2 was disturbed by noises $\mathcal{N}(0, \sigma^2)$ with $\sigma \in [0, 0.05]$ and ratios of outliers varying in $[0, 1]$. In addition, we also randomly neglected inliers in \mathcal{G}_1 with missing point ratios in $[-0.5, 0]$.

As shown in Fig. 15 (a) and (b), under similarity deformation, both FRGM-D1 and FRGM-D2 have less average errors for graphs with missing inliers, outliers and noises. As shown in Fig. 15 (c) and (d), under nonrigid deformation, FRGM-D2 has less average error. FRGM-D1 is competitive with GMM and CPD in the cases with noise and results in higher average error when there are too many missing points or outliers.

Running time of FW and AFW. Finally, we evaluated the average execution time on the graphs with rigid or nonrigid geometric deformations when the algorithm FRGM-D was solved by FW or AFW, respectively. As shown in Tab. 5, the AFW-based implementation is nearly 10 times faster.

TABLE 5
The execution times for FW and AFW implementations.

Template \ Method	Temp-1	Temp-2	Temp-3	Temp-4	Temp-5
FW	35.3s	69.4s	25.8s	24.9s	30.0s
AFW	4.4s	7.8s	3.8s	3.3s	3.6s

8 CONCLUSION AND DISCUSSION

In this paper, we introduce a functional representation for the GM problem. The main idea is to represent both the graphs and node-to-

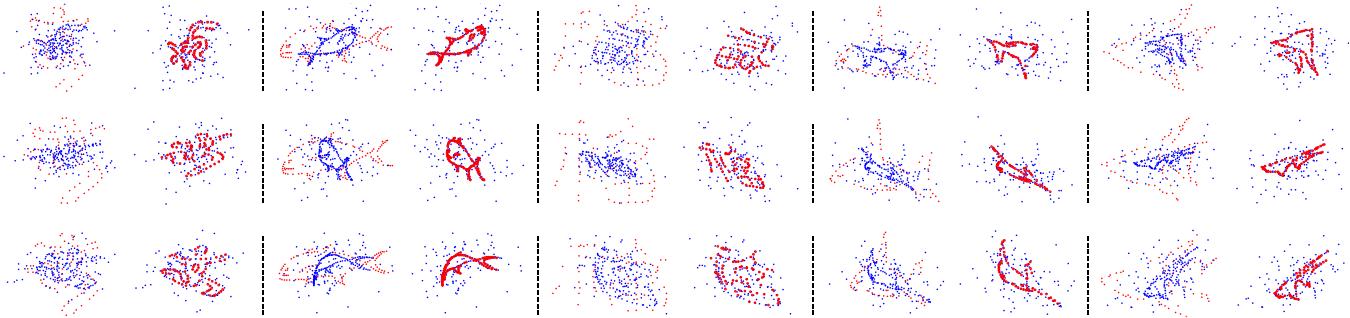


Fig. 13. Examples of matching graphs with geometric deformations by our algorithm FRGM-D. From left to right: the Olympic logo, whale, Chinese character, tropical fish and UCF fish. From top to bottom: the graphs \mathcal{G}_1 (red dots) are deformed with similarity (the first row), affine (the second row) and nonrigid (the third row) transformations. Graphs \mathcal{G}_2 are disturbed by geometric deformations, noises, missing points and outliers.

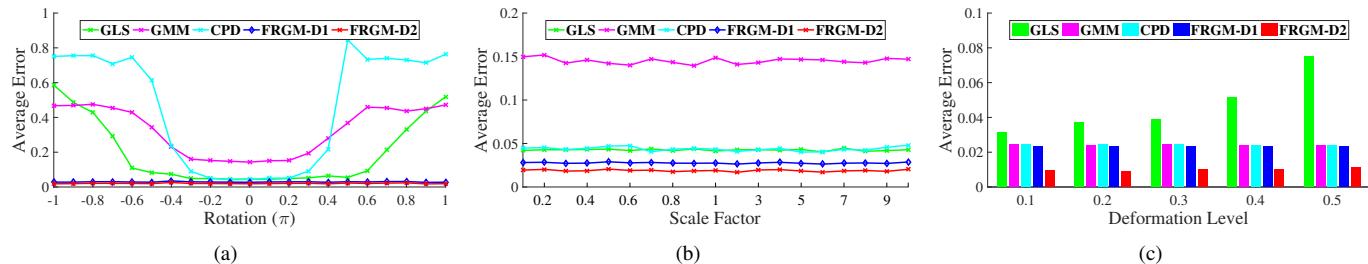


Fig. 14. Comparisons on rotation (a), scaling factor (b) and deformation level (c). Both FRGM-D1 and FRGM-D2 have less average errors than the other algorithms.

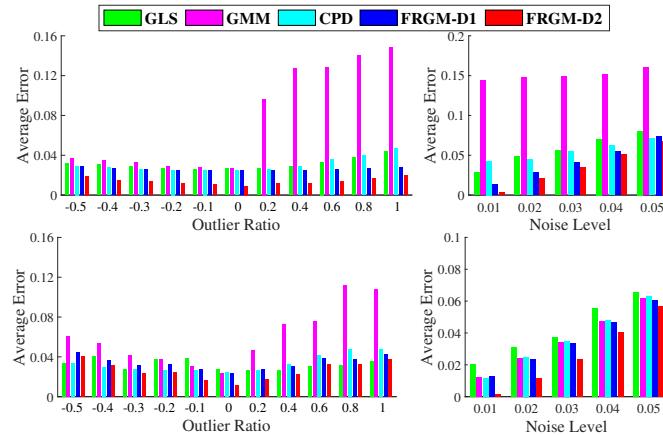


Fig. 15. Comparisons on outliers and noises for similarity deformation (the first row) and nonrigid deformation (the second row).

node correspondence by linear function spaces and linear functional representation map, and then represent the pairwise information of graphs by geometric-aware functionals defined on the function spaces. It follows with three main contributions of functional representation. First, the representation provides geometric insights for the general GM, by which we can construct more appropriate objective functions and algorithms. Second, the linear representation map is a new parameterization approach for the Euclidean GM and helps to handle both conventional and geometrically deformed graphs. Third, the representation of graph attributes can be used as a replacement for the costly affinity matrix and reduce the space complexity. Finally, both efficient algorithms and optimization strategy have been proposed to solve the proposed GM algorithms with better performance.

Beyond the scope explored in this paper, there are some other problems that may benefit from our work. There are three inspirations.

- Considering the basis functions and function space of a graph, some flexible choices are available for different real applications. For example, for graphs on 3D surfaces, we can define the eigenfunctions of the Laplace-Beltrami operator

as basis functions; for graphs embedded in hidden spaces, we can first embed graphs into some specific vector spaces and then construct the basis functions and function space aforementioned in Sec. 3.

- For the hypergraph or multigraph matching problem, which results in much higher computational complexity, the functional representation may be used to reduce the space complexity by representing the higher-order edge attributes and the matching configuration among the product space of function spaces.
- The proposed approximated Frank-Wolfe method can be used to improve the other GM algorithms or some problems that need to solve objective functions upon the feasible field $\tilde{\mathcal{P}}$ by some modifications.

For the limitations of FRGM, one should note that the current version of FRGM-G can only handle undirected graphs due to the construction of the inner product or metric, which requires symmetric edge attributes. In future work, we will address this issue by extending the inner product or metric with more general functionals, such as bilinear functional on the product space of function spaces of graphs.

ACKNOWLEDGEMENT

This work was supported by the National Natural Science Foundation of China under Grant 61771350 and Grant 61842102. Nan Xue is also supported by China Scholarship Council.

REFERENCES

- [1] J. Yan, X. Yin, W. Lin, C. Deng, H. Zha, and X. Yang, "A short survey of recent advances in graph matching," in *ICMR*, 2016.
- [2] D. Conte, P. Foggia, C. Sansone, and M. Vento, "Thirty years of graph matching in pattern recognition," *Int'l J. Pattern Recognition and Artificial Intelligence*, vol. 18, no. 3, pp. 265–298, 2004.
- [3] S. Belongie, J. Malik, and J. Puzicha, "Shape matching and object recognition using shape contexts," *IEEE TPAMI*, vol. 24, no. 4, pp. 509–522, 2002.
- [4] V. Garro and A. Giachetti, "Scale space graph representation and kernel matching for non rigid and textured 3d shape retrieval," *IEEE TPAMI*, vol. 38, no. 6, pp. 1258–1271, 2016.

- [5] O. Duchenne, A. Joulin, and J. Ponce, "A graph-matching kernel for object categorization," in *ICCV*, 2011.
- [6] W. Lin, Y. Shen, J. Yan, M. Xu, J. Wu, J. Wang, and K. Lu, "Learning correspondence structures for person re-identification," *IEEE TIP*, vol. 26, no. 5, pp. 2438–2453, 2017.
- [7] H. Xiao, W. Lin, B. Sheng, K. Lu, J. Yan, J. Wang, E. Ding, Y. Zhang, and H. Xiong, "Group re-identification: Leveraging and integrating multi-grain information," in *ACM Multimedia*, 2018, pp. 192–200.
- [8] B. Yao and F. Li, "Action recognition with exemplar based 2.5D graph matching," in *ECCV*, 2012.
- [9] T. Shen, S. Zhu, T. Fang, R. Zhang, and L. Quan, "Graph-based consistent matching for structure-from-motion," in *ECCV*, 2016.
- [10] D. Lowe, "Distinctive image features from scale-invariant keypoints," *IJCV*, vol. 60, no. 2, pp. 91–110, 2004.
- [11] M. Leordeanu and M. Hebert, "A spectral technique for correspondence problems using pairwise constraints," in *ICCV*, 2005.
- [12] M. Cho, J. Lee, and K. Lee, "Reweighted random walks for graph matching," in *ECCV*, 2010.
- [13] J. Lee, M. Cho, and K. Lee, "Hyper-graph matching via reweighted random walks," in *CVPR*, 2011.
- [14] J. Yan, C. Zhang, H. Zha, W. Liu, X. Yang, and S. Chu, "Discrete hypergraph matching," in *CVPR*, 2015.
- [15] E. Loiola, N. de Abreu, P. Netto, P. Hahn, and T. Querido, "A survey for the quadratic assignment problem," *European J. Operational Research*, vol. 176, no. 2, pp. 657–690, 2007.
- [16] E. Lawler, "The quadratic assignment problem," *Management Science*, pp. 586–599, 1963.
- [17] T. Koopmans and M. Beckmann, "Assignment problems and the location of economic activities," *Econometrica: J. the Econometric Society*, pp. 53–76, 1957.
- [18] M. Garey and D. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, 1979.
- [19] F. Zhou and F. D. la Torre, "Factorized graph matching," *IEEE TPAMI*, vol. 38, no. 9, pp. 1774–1789, 2016.
- [20] B. Jiang, J. Tang, C. Ding, and B. Luo, "Binary constraint preserving graph matching," in *CVPR*, 2017.
- [21] D. Lê-Huu and N. Paragios, "Alternating direction graph matching," in *CVPR*, 2017.
- [22] F. Wang, N. Xue, Y. Zhang, X. Bai, and G.-S. Xia, "Adaptively transforming graph matching," in *ECCV*, 2018.
- [23] M. Leordeanu, M. Hebert, and R. Sukthankar, "An integer projected fixed point method for graph matching and map inference," in *NIPS*, 2009.
- [24] M. Zaslavskiy, F. Bach, and J. Vert, "A path following algorithm for the graph matching problem," *IEEE TPAMI*, vol. 31, no. 12, pp. 2227–2242, 2009.
- [25] H. Kuhn, "The hungarian method for the assignment problem," in *50 Years of Integer Programming 1958–2008 - From the Early Years to the State-of-the-Art*. Springer, 2010, pp. 29–47.
- [26] J. Ullmann, "An algorithm for subgraph isomorphism," *J. ACM*, vol. 23, no. 1, pp. 31–42, 1976.
- [27] L. Cordella, P. Foggia, C. Sansone, and M. Vento, "A (sub)graph isomorphism algorithm for matching large graphs," *IEEE TPAMI*, vol. 26, no. 10, pp. 1367–1372, 2004.
- [28] R. Zass and A. Shashua, "Probabilistic graph and hypergraph matching," in *CVPR*, 2008.
- [29] Y. Lu, K. Huang, and C.-L. Liu, "A fast projected fixed-point algorithm for large graph matching," *Pattern Recognition*, vol. 60, pp. 971–982, 2016.
- [30] T. Cour, P. Srinivasan, and J. Shi, "Balanced graph matching," in *NIPS*, 2006.
- [31] P. Torr, "Solving markov random fields using semi definite programming," in *Int'l Conf. Artificial Intelligence and Statistics*, 2003.
- [32] C. Schellewald and C. Schnörr, "Probabilistic subgraph matching based on convex relaxation," in *Energy Minimization Methods in Computer Vision and Pattern Recognition*, 2005.
- [33] Z. Liu and H. Qiao, "GNCCP —graduated nonconvexity and concavity procedure," *IEEE TPAMI*, vol. 36, no. 6, pp. 1258–1267, 2014.
- [34] Z. Liu, H. Qiao, X. Yang, and S. Hoi, "Graph matching by simplified convex-concave relaxation procedure," *IJCV*, vol. 109, no. 3, pp. 169–186, 2014.
- [35] S. Gold and A. Rangarajan, "A graduated assignment algorithm for graph matching," *IEEE TPAMI*, vol. 18, no. 4, pp. 377–388, 1996.
- [36] Y. Tian, J. Yan, H. Zhang, Y. Zhang, X. Yang, and H. Zha, "On the convergence of graph matching: Graduated assignment revisited," in *ECCV*, 2012.
- [37] L. Torresani, V. Kolmogorov, and C. Rother, "A dual decomposition approach to feature correspondence," *IEEE TPAMI*, vol. 35, no. 2, pp. 259–271, 2013.
- [38] K. Adamczewski, Y. Suh, and K. M. Lee, "Discrete tabu search for graph matching," in *ICCV*, 2015.
- [39] A. Egozi, Y. Keller, and H. Guterman, "A probabilistic approach to spectral graph matching," *IEEE TPAMI*, vol. 35, no. 1, pp. 18–27, 2013.
- [40] T. Caetano, J. McAuley, L. Cheng, Q. Le, and A. Smola, "Learning graph matching," *IEEE TPAMI*, vol. 31, no. 6, pp. 1048–1058, 2009.
- [41] M. Leordeanu, R. Sukthankar, and M. Hebert, "Unsupervised learning for graph matching," *IJCV*, vol. 96, no. 1, pp. 28–45, 2012.
- [42] A. Zanfir and C. Sminchisescu, "Deep learning of graph matching," in *CVPR*, 2018.
- [43] F. Zhou and F. D. la Torre, "Deformable graph matching," in *IEEE CVPR*, 2013.
- [44] M. Cho, J. Sun, O. Duchenne, and J. Ponce, "Finding matches in a haystack: A max-pooling strategy for graph matching in the presence of outliers," in *CVPR*, 2014.
- [45] M. Cuturi, "Sinkhorn distances: Lightspeed computation of optimal transport," in *NIPS*, 2013, pp. 2292–2300.
- [46] W. Rudin, *Functional Analysis*. McGraw-Hill, 2006.
- [47] M. Ovsjanikov, M. Ben-Chen, J. Solomon, A. Butscher, and L. Guibas, "Functional maps: a flexible representation of maps between shapes," *ACM TOG.*, vol. 31, no. 4, pp. 30:1–30:11, 2012.
- [48] E. Rodolà, L. Cosmo, M. Bronstein, A. Torsello, and D. Cremers, "Partial functional correspondence," *Comput. Graph. Forum*, vol. 36, no. 1, pp. 222–236, 2017.
- [49] J. Sun, M. Ovsjanikov, and L. Guibas, "A concise and provably informative multi-scale signature based on heat diffusion," *Comput. Graph. Forum*, vol. 28, no. 5, pp. 1383–1392, 2009.
- [50] N. Courty, R. Flamary, D. Tuia, and A. Rakotomamonjy, "Optimal transport for domain adaptation," *IEEE TPAMI*, vol. 39, no. 9, pp. 1853–1865, 2017.
- [51] A. Myronenko and X. Song, "Point set registration: Coherent point drift," *IEEE TPAMI*, vol. 32, no. 12, pp. 2262–2275, 2010.
- [52] B. Jian and B. Vemuri, "Robust point set registration using gaussian mixture models," *IEEE TPAMI*, vol. 33, no. 8, pp. 1633–1645, 2011.
- [53] J. Ma, J. Zhao, and A. Yuille, "Non-rigid point set registration by preserving global and local structures," *IEEE TIP*, vol. 25, no. 1, pp. 53–64, 2016.
- [54] Y. Zheng and D. Doermann, "Robust point matching for nonrigid shapes by preserving local neighborhood structures," *IEEE TPAMI*, vol. 28, no. 4, pp. 643–649, 2006.
- [55] T. Caetano, T. Caelli, D. Schuurmans, and D. Barone, "Graphical models and point pattern matching," *IEEE TPAMI*, vol. 28, no. 10, pp. 1646–1663, 2006.
- [56] S. Lacoste-Julien and M. Jaggi, "On the global linear convergence of frank-wolfe optimization variants," in *NIPS*, 2015.
- [57] J. Lafond, H. Wai, and E. Moulines, "Non-convex optimization with frank-wolfe algorithm and its variants," in *NIPS 2016 Workshop on Nonconvex Optimization for Machine Learning: Theory and Practice*, 2016.
- [58] A. Goldstein, "On steepest descent," *SIAM J. Control and Optimization*, vol. 3, no. 1, pp. 147–151, 1965.
- [59] R. Jonker and A. Volgenant, "A shortest augmenting path algorithm for dense and sparse linear assignment problems," *Computing*, vol. 38, no. 4, pp. 325–340, 1987.
- [60] L. D. M. P. Bredies, K., "A generalized conditional gradient method and its connection to an iterative shrinkage method," *Comput. Optim. Appl.*, vol. 42, no. 2, pp. 173–193, 2009.
- [61] R. Cominetti and J. S. Martín, "Asymptotic analysis of the exponential penalty trajectory in linear programming," *Math. Program.*, vol. 67, pp. 169–187, 1994.
- [62] L. Zhang, N. Snavely, B. Curless, and S. Seitz, "Spacetime faces: high resolution capture for modeling and animation," *ACM TOG*, vol. 23, no. 3, pp. 548–558, 2004.
- [63] T. Chen, B. Vemuri, A. Rangarajan, and S. Eisenschenk, "Group-wise point-set registration using a novel cdf-based havrda-charvát divergence," *IJCV*, vol. 86, no. 1, pp. 111–124, 2010.



Fu-Dong Wang received a B.Sc. degree in Mathematics from Wuhan University, Wuhan, China, in 2016. He is currently pursuing his Ph.D. degree in the State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing (LIESMARS) at Wuhan University. His research interests include graph matching, optimization theory and 3D computer vision.



Nan Xue received a B.Sc. degree in Computational Mathematics from Wuhan University, Wuhan, China, in 2014. He is currently working toward a Ph.D. degree in the State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing (LIESMARS), Wuhan University. His research interests include low-level computer vision and structure-from-motion. He is a student member of IEEE.



Yipeng Zhang currently is a Research Assistant at the School of Computer Science, Wuhan University. His major research interest is in the Computer Aided Design, Parallel Computing and Artificial Neural Network. He joins Wuhan University after he acquires the master degree from overseas.



Gui-Song Xia (M'10-SM'15) received his Ph.D. degree in image processing and computer vision from CNRS LTCI, Télécom ParisTech, Paris, France, in 2011. From 2011 to 2012, he has been a Post-Doctoral Researcher with the Centre de Recherche en Mathématiques de la Décision, CNRS, Paris-Dauphine University, Paris, for one and a half years. He is currently working as a full professor in computer vision and photogrammetry at Wuhan University. He has also been working as Visiting Scholar at DMA, École Normale Supérieure (ENS-Paris) for two months in 2018. His current research interests include mathematical modeling of images and videos, structure from motion, perceptual grouping, and remote sensing imaging. He serves on the Editorial Boards of the journals Pattern Recognition, Signal Processing: Image Communications, and EURASIP Journal on Image & Video Processing.



Marcello Pelillo (SM'04-F'13) is full professor of computer science with University of Venice, Italy, where he leads the Computer Vision and Pattern Recognition group. He held visiting research positions with Yale University, McGill University, the University of Vienna, York University, the University College London. He is general chair for ICCV 2017 and has served as program chair for several conferences (EMMCVPR, SIMBAD, etc.). He serves (has served) on the Editorial Boards of the journals the IEEE Transactions on Pattern Analysis and Machine Intelligence, Pattern Recognition, Frontiers in Computer Image Analysis, and serves on the Advisory Board of the International Journal of Machine Learning and Cybernetics. He is a fellow of the IEEE and the IAPR.