Zero-Assignment Constraint for Graph Matching with Outliers

Fudong Wang[1], Nan Xue[1], Jin-Gang Yu[2], Gui-Song Xia[1*]

[1]Wuhan University, China

{fudong-wang, xuenan, guisong.xia}@whu.edu.cn

[2]South China University of Technology, China

jingangyu@scut.edu.cn

Abstract

Graph matching (GM), as a longstanding problem in computer vision and pattern recognition, still suffers from numerous cluttered outliers in practical applications. To address this issue, we present the zero-assignment constraint (ZAC) for approaching the graph matching problem in the presence of outliers. The underlying idea is to suppress the matchings of outliers by assigning zero-valued vectors to the potential outliers in the obtained optimal correspondence matrix. We provide elaborate theoretical analysis to the problem, i.e., GM with ZAC, and figure out that the GM problem with and without outliers are intrinsically different, which enables us to put forward a sufficient condition to construct valid and reasonable objective function. Consequently, we design an efficient outlier-robust algorithm to significantly reduce the incorrect or redundant matchings caused by numerous outliers. Extensive experiments demonstrate that our method can achieve the state-of-the-art performance in terms of accuracy and efficiency, especially in the presence of numerous outliers.

1. Introduction

In many real applications of computer vision and pattern recognition, the feature sets of interest represented as graphs are usually cluttered with numerous outliers [3, 42, 38, 30], which often reduce the accuracy of GM. Although recent works on GM [7, 11, 21, 22, 34, 44] can achieve satisfactory results for simple graphs that consist of only inliers or a few outliers, they still lack of ability to tolerate numerous outliers arising in complicated graphs. Empirically, the inliers in one graph are nodes that have highly-similar corresponding nodes in the other graph, while the outliers do not. Based on the empirical criterion, the aforementioned methods hope to match inliers to inliers correctly and force outliers to only match outliers. However, due to the com-
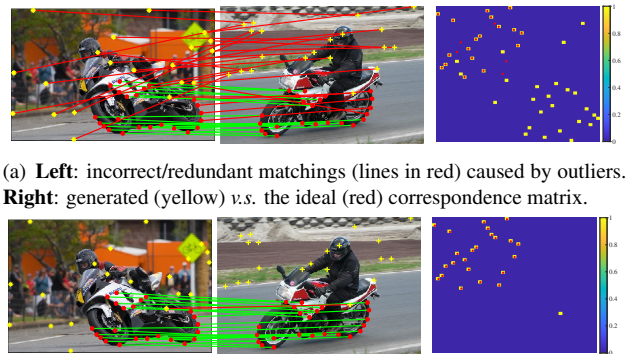
(a) **Left**: incorrect/redundant matchings (lines in red) caused by outliers. **Right**: generated (yellow) *v.s.* the ideal (red) correspondence matrix.

(b) **Left**: our graph matching result. **Right**: our correspondence matrix with zero-assignment constraint of outliers.

Figure 1: **ZAC** for graph matching in the presence of outliers. To suppress the undesired matchings of outliers in (a), we aim to assign the potential outliers with zero-valued vectors in our optimal correspondence matrix in (b), based on which we can both establish a theoretical foundation for graph matching with outliers and put forward an outlier identification approach that can significantly reduce incorrect or redundant matches caused by outliers in practice.

plicated mutual relationships between inliers and outliers, they usually result in incorrect matchings between inliers or redundant matchings between outliers (*e.g.*, Fig. 1 (a)).

In this paper, we are motivated to address this challenge by introducing the zero-assignment constraint for outliers: unlike the previous methods that hope to match outliers only to outliers, it's more reasonable to suppress the matchings of outliers. Equivalently, we try to assign each potential outlier with a zero-valued vector (*i.e.*, the zero-assignment constraint for outliers) in the solution of our objective function (*e.g.*, the correspondence matrix in Fig. 1 (b)).

To make our idea more reasonable and practical, we try our efforts in two aspects. First, based on the zero-assignment constraint, we establish the theoretical bases including the formulation of inliers and outliers and the quantitative distinguishability between them, and then find out a

---

*Corresponding author

sufficient condition such that the proposed objective function can only achieve its minimum at the ideal matching. Moreover, it also helps to demonstrate the intrinsic differences between GM with and without numerous outliers. Second, we propose an efficient GM algorithm consisting of fast optimization and explicit outlier identification. The optimization algorithm is modified based on the Frank-Wolfe method [18] combined with the k-cardinality linear assignment problem [10] and has low space and time complexity. And then, the zero-assignment vectors in the optimal solution of our objective function can be used to assign the nodes in two graphs with joint probabilities, which measure whether the nodes are inliers or outliers and help to identify and remove the potential outliers in practice.

Our main contributions are summarized as follows:

- We establish the theoretical foundations for GM problem with outliers based on the zero-assignment constraint and elaborate quantitative analyses of inliers and outliers, on which bases we can theoretically put forward a sufficient condition to guide us how to construct valid and reasonable objective function.

- We present an efficient GM algorithm with low space and time complexity by avoiding using the costly affinity matrix and designing fast optimization algorithm. Combined with our outlier identification approach, we can achieve state-of-the-art performance for complicated graphs cluttered with numerous outliers.

2. Related Work

Known to be NP-complete [12, 16, 20], the GM problem can only be solved in polynomial time with approximate solutions. Over the past decades, a myriad of literature have been extensively studied (see [9, 37] for surveys), we discuss the most related works in the following aspects.

Robustness to outliers. The dual decomposition approach [31] constructed a penalty potential in the objective function for unmatched features. The max-pooling-based method [8] was proposed to avoid the adverse effect of false matches of outliers. A domain adaptation-based outlier-removal strategy proposed in [34] aimed to remove outliers as a pre-processing step. However, they directly rely on empirical criterions of outliers and can not deal with complicated situations. In our work, we both explain theoretical analyses of outliers and present an efficient outlier identification approach, by which we can achieve much better matching accuracy in complicated applications.

Interpretability for graph matching. The probability-based works [41, 11] formulated GM from the maximum-likelihood estimation perspective. A random walk view [7] was introduced by simulating random walks with re-weighting jumps for GM. Some machine learning-based works [6, 25] went further to adjust attributes of graphs or improve the affinity matrix $\mathbf{K}$ (in Eq. (1)) based on priors learned from real data. A functional representation framework [34] was proposed to give geometric insights for both general and Euclidean GM. The pioneering works [39, 35] presented an end-to-end deep learning framework for GM. Our work aims to establish the mathematical foundation for GM with outliers and enhance its theoretical rationality.

Computational efficiency. Some existing works aimed to reduce the costly space complexity caused by $\mathbf{K}$ in Eq. (1). A typical work was the factorized graph matching [44], which factorized $\mathbf{K}$ as Kronecker product of several smaller matrices. However, it is highly time-consuming in practice due to the verbose iterations during optimization. Some methods like the graduated assignment method [13] and the integer-projected fixed point algorithm [24] proposed specific fast approximations while ended with unsatisfactory matching results. As comparison, our method has low space and time complexity and achieves better trade-off between time consumption and matching accuracy.

3. Graph matching with outliers

This section revisits the general formulation of GM and presents the theoretical foundation for GM with outliers.

3.1. General formulation of graph matching

Given two attributed graphs $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}, \mathcal{G}' = \{\mathcal{V}', \mathcal{E}'\}$, where $\mathcal{V} = \{V_i\}_{i=1}^m$ and $\mathcal{V}' = \{V_a'\}_{a=1}^n$ represent the node sets (assume $m \leq n$), $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ and $\mathcal{E}' \subseteq \mathcal{V}' \times \mathcal{V}'$ denote the edge sets. Generally, for each graph, $e.g.$, $\mathcal{G}$, the edges are represented by a (weighted) adjacency matrix $\mathcal{E} \in \mathbb{R}^{m \times m}$, where $\mathcal{E}_{ij} > 0$ if there is an edge (V_i, V_j), and $\mathcal{E}_{ij} = 0$ otherwise. In practice, graph $\mathcal{G}$ is usually associated with node attribute $\mathbf{v}_i \in \mathbb{R}^{d_v}$ of node V_i and edge attribute $\mathbf{A}_{ij} \in \mathbb{R}^{d_e}$ of edge $\mathcal{E}_{ij}$; the same to graph $\mathcal{G}'$.

Solving GM problem is to find an optimal binary correspondence $\mathbf{P} \in \{0, 1\}^{m \times n}$, where $\mathbf{P}_{ia} = 1$ when the nodes $V_i \in \mathcal{V}$ and $V_a' \in \mathcal{V}'$ are matched, and $\mathbf{P}_{ia} = 0$ otherwise. To find such an optimal correspondence, GM methods generally minimize or maximize an objective function that measures the mutual (dis-)similarity between graphs.

As a typical Quadratic Assignment Problem (QAP), GM formulated as Lawler's QAP [20, 23, 24, 7, 44] has been favored to maximize the sum of node and edge similarities

$$\max_{\mathbf{P} \in \mathcal{P}} \mathbf{P}_v^{\mathrm{T}} \mathbf{K} \mathbf{P}_v = \sum_{i,a} \mathbf{P}_{ia} \mathbf{K}_{ia;ia} + \sum_{(i,j),(a,b)} \mathbf{P}_{ia} \mathbf{K}_{ij;ab} \mathbf{P}_{jb}, \quad (1)$$

where $\mathbf{P}_v$ is the columnwise vectorized replica of $\mathbf{P}$. The affinity matrix $\mathbf{K} \in \mathbb{R}^{mn \times mn}$ has diagonal element $\mathbf{K}_{ia;ia}$ measuring the node affinity calculated with node attributes $(\mathbf{v}_i, \mathbf{v}_a')$ and non-diagonal element $\mathbf{K}_{ia;jb}$ measuring the edge affinity calculated with edge attributes $(\mathbf{A}_{ij}, \mathbf{B}_{ab})$.

Another famous formulation is Koopmans-Beckmann's QAP [16, 32, 1, 40], which maximizes a trace-form objec-

tive function measuring the node and edge similarities

$$\max_{\mathbf{P} \in \mathcal{P}} \text{tr}(\mathbf{U}^{\text{T}}\mathbf{P}) + \lambda \text{tr}(\mathcal{E}\mathbf{P}\mathcal{E}'\mathbf{P}^{\text{T}}), \qquad (2)$$

where $\{\mathbf{U}_{ia}\} \in \mathbb{R}^{m \times n}$ measures the node similarity between V_i and V_a' and $\lambda \geq 0$ is a weight.

Generally, GM methods impose the one-to-(at most)-one constraint, *i.e.*, the feasible field $\mathcal{P}$ can be defined as

$$\mathcal{P} \triangleq \left\{ \mathbf{P} \in \{0,1\}^{m \times n}; \mathbf{P}\mathbf{1} = \mathbf{1}, \mathbf{P}^{\text{T}}\mathbf{1} \leq \mathbf{1} \right\}, \qquad (3)$$

where $\mathbf{1}$ is a columnwise unit vector. In fact, Eq. (3) means that both inliers and outliers are equally treated to find their correspondences. Some methods like [6, 31] replace $\mathbf{P}\mathbf{1} = \mathbf{1}$ by $\mathbf{P}\mathbf{1} \leq \mathbf{1}$ to relax the one-to-(at most)-one constraint. However, they still lack of intrinsic theoretical analyses for the numerous outliers arising in both graphs.

3.2. Zero-assignment constraint for outliers

As stated previously in Sec. 1, we aim to only match inliers to inliers and suppress the matchings of outliers. To achieve our goal, we present the zero-assignment constraint for outliers in this section. Denoting the number of inliers in $\mathcal{G}'$ and $\mathcal{G}$ as k ($0 < k \leq m \leq n$), for better understanding, we first introduce some basic definitions in the following.

Definition 3.1. Denote $\mathscr{A} = \{1, 2, ..., m\}$ as the index set of nodes in graph $\mathcal{G}$. The index sets of inliers and outliers of $\mathcal{G}$ are respectively defined as,

$$\mathscr{A}_I \triangleq \{i \in \{1, 2, ..., m\}; \mathcal{V}_i \text{ is an inlier of } \mathcal{G}\}, \qquad (4)$$

$$\mathscr{A}_O \triangleq \{o \in \{1, 2, ..., m\}; \mathcal{V}_o \text{ is an outlier of } \mathcal{G}\}. \qquad (5)$$

The index sets $\mathscr{B} = \{1, 2, ..., n\}$, $\mathscr{B}_I$ and $\mathscr{B}_O$ are similarly defined for graph $\mathcal{G}'$. Obviously, we have $|\mathscr{A}_I| = |\mathscr{B}_I| = k$. The inliers and outliers sets are complementary and disjoint.

Proposition 1.

$$\mathscr{A}_I \cup \mathscr{A}_O = \mathscr{A}, \quad \mathscr{A}_I \cap \mathscr{A}_O = \varnothing, \qquad (6)$$

$$\mathscr{B}_I \cup \mathscr{B}_O = \mathscr{B}, \quad \mathscr{B}_I \cap \mathscr{B}_O = \varnothing. \qquad (7)$$

where $\varnothing$ denotes the empty set.

Next, we derive the zero-assignment constraint for outliers. Mathematically, the matching between $\mathcal{G}$ and $\mathcal{G}'$ consisting of inliers and outliers can be defined by a partial permutation τ and a partial permutation matrix $\mathbf{P}$ as follows.

Definition 3.2. The *partial permutation* τ between $\mathcal{G}$ and $\mathcal{G}'$ is defined as $\tau : \mathscr{A} \rightarrow \mathscr{B}$,

$$i \mapsto a = \tau(i) \in \mathscr{B}_I \text{ if } i \in \mathscr{A}_I; a = \varnothing \text{ if } i \in \mathscr{A}_O. \qquad (8)$$

And the inverse of τ can also be defined as $\tau^{-1} : \mathscr{B} \rightarrow \mathscr{A}$,

$$a \mapsto i = \tau^{-1}(a) \in \mathscr{A}_I \text{ if } a \in \mathscr{B}_I; i = \varnothing \text{ if } a \in \mathscr{B}_O. \qquad (9)$$

Given τ, the matching (or correspondence) between $\mathcal{G}$ and $\mathcal{G}'$ can be equivalently expressed by the *partial permutation matrix* $\mathbf{P} \in \{0,1\}^{m \times n}$ compatible with τ as

Definition 3.3. For $\mathbf{P} \in \{0,1\}^{m \times n}$ compatible with τ,

- *One-to-one constraint for inliers:* $\forall i \in \mathscr{A}_I$,

$$\mathbf{P}_{i,a=\tau(i)} = 1, \mathbf{P}_{i,a\neq\tau(i)} = 0, a \in \mathscr{B}_I. \qquad (10)$$

- *Zero-assignment constraint for outliers:*

$$\mathbf{P}_{i,:} \equiv \mathbf{0}^{\text{T}}, \forall i \in \mathscr{A}_O \text{ and } \mathbf{P}_{:,a} \equiv \mathbf{0}, \forall a \in \mathscr{B}_O. \qquad (11)$$

where $\mathbf{P}_{i,:}$ (or $\mathbf{P}_{:,a}$) is a row (or column) vector of $\mathbf{P}$, and $\mathbf{0}$ is a columnwise zero vector.

By this means, the feasible filed $\mathcal{P}_k$ can be redefined as

$$\left\{\mathbf{P} \in \{0,1\}^{m \times n}; \mathbf{P}\mathbf{1} \leq \mathbf{1}, \mathbf{P}^{\text{T}}\mathbf{1} \leq \mathbf{1}, \mathbf{1}^{\text{T}}\mathbf{P}\mathbf{1} = k\right\}. \qquad (12)$$

The explicit equation constraint $\mathbf{1}^{\text{T}}\mathbf{P}\mathbf{1} = k$ will be used to both present a proof for the rationality of our proposed objective function in Sec. 3.4 and design an efficient optimization algorithm in Sec. 4.1.

3.3. Consistency and distinguishability

Empirically, the GM methods assume that the unary and pairwise attributes of inlier $i \in \mathscr{A}_I$ and edge $(i, j) \in \mathscr{A}_I \times \mathscr{A}_I$ are consistent with those of the ideal matchings $a \in \mathscr{B}_I$ and $(a, b) \in \mathscr{B}_I \times \mathscr{B}_I$, while the outliers are on the contrary. Based on this empirical criterion, we furthermore elaborate a quantitative consistency of inliers and distinguishability between inliers and outiers, on which bases the rationality of our objective function can be guaranteed.

Denote $\{\mathbf{D}_{ia}\}_{ia}$ as the dissimilarity between nodes $V_i \in \mathcal{V}$ and $V_a' \in \mathcal{V}'$, $\{\mathbf{A}_{ij}\}_{ij}$ and $\{\mathbf{B}_{ab}\}_{ab}$ are the edge attributes of edges $(V_i, V_j) \in \mathcal{E}$ and $(V_a', V_b') \in \mathcal{E}'$. Meanwhile, denote $\{\tau^*, \mathbf{P}^* \in \mathcal{P}_k\}$ as the ideal matching between $\mathcal{G}$ and $\mathcal{G}'$. Consequently, beyond the empirical criterion, we can induce the consistency of inliers and distinguishability between inliers and outliers by $\{\tau^*, \mathbf{P}^* \in \mathcal{P}_k\}$ as follows.

Proposition 2. *Consistency between inliers.*

- *Unary consistency:* $\forall i \in \mathscr{A}_I, \forall a \in \mathscr{B}_I$,

$$\mathbf{D}_{ia'} = \min\{\mathbf{D}_{ia}, a \in \mathscr{B}\} \Leftrightarrow a' = \tau^*(i), \qquad (13)$$

$$\mathbf{D}_{i'a} = \min\{\mathbf{D}_{ia}, i \in \mathscr{A}\} \Leftrightarrow i' = \tau^{*-1}(a). \qquad (14)$$

- *Pairwise consistency:* $\forall i, j \in \mathscr{A}_I, \forall a, b \in \mathscr{B}_I$,

$$||\mathbf{A}_{ij} - \mathbf{B}_{a'b'}|| = \min\{||\mathbf{A}_{ij} - \mathbf{B}_{ab}||, a, b \in \mathscr{B}\}$$
$$\Leftrightarrow a' = \tau^*(i), b' = \tau^*(j), \qquad (15)$$

$$||\mathbf{B}_{ab} - \mathbf{A}_{i'j'}|| = \min\{||\mathbf{B}_{ab} - \mathbf{A}_{ij}||, i, j \in \mathscr{A}\}$$
$$\Leftrightarrow i' = \tau^{*-1}(a), j' = \tau^{*-1}(b). \qquad (16)$$

Proposition 3. *Distinguishability between inliers and outliers.*

- *Unary distinguishability:* $\forall (i,a) \in \mathscr{A}_O \times \mathscr{B}$ or $\mathscr{A} \times \mathscr{B}_O$,

$$\mathbf{D}_{ia} \geq \max\{\mathbf{D}_{i'\tau^*(i')}, i' \in \mathcal{A}_I\}. \quad (17)$$

- *Pairwise distinguishability:* $\forall (i,a), (j,b) \in \mathscr{A}_O \times \mathscr{B}$ or $\mathscr{A} \times \mathscr{B}_O$,

$$||\mathbf{A}_{ij} - \mathbf{B}_{ab}|| \geq \max\{||\mathbf{A}_{i'j'} - \mathbf{B}_{\tau^*(i')\tau^*(j')}|| , i',j' \in \mathcal{A}_I\}, \quad (18)$$

$$||\mathbf{B}_{ab} - \mathbf{A}_{ij}|| \geq \max\{||\mathbf{B}_{a'b'} - \mathbf{A}_{\tau^{*-1}(a')\tau^{*-1}(b')}|| , a',b' \in \mathcal{B}_I\}. \quad (19)$$

where $|| \cdot ||$ is an Euclidean norm.

By this means, we present a quantitative mathematical criteria of the local characteristics and mutual relationships of inliers and outliers, which is more concise and clear than empirical criteria. More importantly, the propositions above inspires us how to construct a reasonable objective function and find out a sufficient condition for proving the rationality.

3.4. Objective function with sufficient condition

A reasonable objective function $F(\mathbf{P})$ should satisfy two main properties: (1) preserve the unary and pairwise consistencies between the matched nodes (or edges) of two graphs and (2) achieve its optimum only at the ideal matching $\mathbf{P}^*$. Overall, our objective function is defined as

$$\min_{\mathbf{P} \in \mathcal{P}_k} F(\mathbf{P}) = \lambda_1 F_u(\mathbf{P}) + \lambda_2 F_p(\mathbf{P}), \quad (20)$$

where $F_u(\mathbf{P})$ and $F_p(\mathbf{P})$ are the unary and pairwise potentials. Precisely, we set $F_u(\mathbf{P}) = \sum_{ia} \mathbf{D}_{ia} \mathbf{P}_{ia}$ and

$$F_p(\mathbf{P}) \triangleq F_{p_1}(\mathbf{P}) + F_{p_2}(\mathbf{P}) \quad (21)$$

$$\triangleq \sum_{ij} \mathcal{E}_{ij} ||\mathbf{A}_{ij} - \sum_{a,b} \mathbf{P}_{ia} \mathbf{B}_{ab} \mathbf{P}_{jb}||^2$$

$$+ \sum_{ab} \mathcal{E}'_{ab} ||\mathbf{B}_{ab} - \sum_{i,j} \mathbf{P}_{ia} \mathbf{A}_{ij} \mathbf{P}_{jb}||^2 \quad (22)$$

$$\triangleq ||\mathbf{A} - \mathbf{P}\mathbf{B}\mathbf{P}^T||^2_{\mathcal{E}} + ||\mathbf{B} - \mathbf{P}^T\mathbf{A}\mathbf{P}||^2_{\mathcal{E}'}. \quad (23)$$

The property (1) is guaranteed since the minimization of $F(\mathbf{P})$ tends to find the minimizer $\hat{\mathbf{P}}$ that matches the nodes and edges in $\mathcal{G}$ (or $\mathcal{G}'$) to the mostly-consistent nodes and edges in $\mathcal{G}'$ (or $\mathcal{G}$). Next, we should make sure that it also satisfies the property (2). However, due to the cluttered outliers arising in both graphs, it may not hold for any arbitrarily given weighted adjacency matrices $\mathcal{E}, \mathcal{E}'$ or edge attributes $\mathbf{A}, \mathbf{B}$. Furthermore, we put forward a sufficient condition to support it.

Proposition 4. *Sufficient condition for objective function. Assume that the weighted adjacency matrices $\mathcal{E}, \mathcal{E}'$ and edge attributes $\mathbf{A}, \mathbf{B}$ satisfy that*

$$\mathcal{E}_{i \in \mathscr{A}_I, j \in \mathscr{A}_I} \geq \mathcal{E}_{i \in \mathscr{A}, j \in \mathscr{A}_O}, \mathcal{E}_{i \in \mathscr{A}_O, j \in \mathscr{A}}, \quad (24)$$

$$||\mathbf{A}_{i \in \mathscr{A}_I, j \in \mathscr{A}_I}|| \geq ||\mathbf{A}_{i \in \mathscr{A}, j \in \mathscr{A}_O}||, ||\mathbf{A}_{i \in \mathscr{A}_O, j \in \mathscr{A}}||, \quad (25)$$

and the same to $\mathcal{E}'$ and $\mathbf{B}$. Then, it's sufficient to prove that

$$\forall \mathbf{P} \in \mathcal{P}_k, \ F(\mathbf{P}) \geq F(\mathbf{P}^*), \quad (26)$$

the equation holds if and only if $\mathbf{P} = \mathbf{P}^*$.

Proof. Due to the over-length of the entire proof, we give the details in our supplementary materials, which also demonstrate the intrinsic differences between GM on simple graphs and on complicated graphs. □

Note that, the Eq. (24) and (25) tell us how to calculate proper $\{\mathcal{E}_{ij}\}$, $\{\mathbf{A}_{ij}\}$ (or $\{\mathcal{E}'_{ab}\}$, $\{\mathbf{B}_{ab}\}$): we should compute $\mathcal{E}_{ij}$ and $\mathbf{A}_{ij}$ to measure the similarities between the two end-nodes in edge (i, j) such that edges linked by two inliers have higher similarities than the edges linked by inlier-outlier or outlier-outlier. It will be followed and validated in the experiments section Sec. 5.

4. Outlier-robust graph matching algorithm

In this section, we propose an efficient algorithm to solve Eq. (20) and then design an outlier identification approach.

4.1. Optimization algorithm

Our optimization algorithm is based on the Frank-Wolfe method [18, 19], which is widely used for convex or non-convex optimization and achieve at least sub-linear convergence rate. Since it is a continuous line-search-based method, we should relax the discrete $\mathcal{P}_k$ into the continuous $\hat{\mathcal{P}}_k$ by relaxing $\mathbf{P}_{ia} \in \{0, 1\}$ into $\mathbf{P}_{ia} \in [0, 1]$. Given $F(\mathbf{P})$ is differentiable and $\hat{\mathcal{P}}_k$ is convex, the Frank-Wolfe method iterates the following steps till it converges:

$$\tilde{\mathbf{P}}^{(t+1)} \in \underset{\mathbf{P} \in \hat{\mathcal{P}}_k}{\operatorname{argmin}} \triangleq \langle \nabla F(\mathbf{P}^{(t)}), \mathbf{P} \rangle, \quad (27)$$

$$\mathbf{P}^{(t+1)} = \mathbf{P}^{(t)} + \alpha^{(t)}(\tilde{\mathbf{P}}^{(t+1)} - \mathbf{P}^{(t)}), \quad (28)$$

where $\nabla F(\mathbf{P}^{(t)})$ is the gradient of $F(\mathbf{P})$ at $\mathbf{P}^{(t)}$ and $\alpha^{(t)}$ is the step size obtained by exact or inexact line search [14].

Gradient computation. The gradient $\nabla F(\mathbf{P})$ can be efficiently calculated by matrix operations as follows,

$$\mathbf{W}_1 \triangleq 4||\mathbf{P}\mathbf{B}\mathbf{P}^T - \mathbf{A}|| \otimes \operatorname{sign}(\mathbf{P}\mathbf{B}\mathbf{P}^T - \mathbf{A}) \otimes \mathcal{E}, \quad (29)$$

$$\mathbf{W}_2 \triangleq 4||\mathbf{P}^T\mathbf{A}\mathbf{P} - \mathbf{B}|| \otimes \operatorname{sign}(\mathbf{P}^T\mathbf{A}\mathbf{P} - \mathbf{B}) \otimes \mathcal{E}', \quad (30)$$

$$\nabla F(\mathbf{P}) = \lambda_1 \mathbf{D} + \lambda_2 [\mathbf{W}_1 \mathbf{P}\mathbf{B}^T + \mathbf{A}\mathbf{P}\mathbf{W}_2^T], \quad (31)$$

where $\otimes$ is the pointwise multiplication and $\operatorname{sign}(\cdot)$ is the sign function.

The k-cardinality LAP. Eq. (27) plays a key role of the optimization. It is a linear programming (LP) problem that can be solved by LP algorithms like interior point method [29]. However, such methods have costly time complexity $O(m^3n^3/\ln(mn))$ [2]. Fortunately, one can prove that $\tilde{\mathbf{P}}^{(t+1)}$ is an extreme point [4] of $\hat{\mathcal{P}}_k$, thus, $\tilde{\mathbf{P}}^{(t+1)} \in \mathcal{P}_k$. Therefore, Eq. (27) boils down to a k-cardinality linear assignment problem (kLAP) [10]. We can adopt the approach [33] by which the kLAP is transformed into a standard LAP that can be efficiently solved by the Hungarian [17] or LAPJV [15] algorithm with much less time complexity $O(n^3)$.

Regularization. Someone may doubt that the explicit equation constraint $\mathbf{1}^T\mathbf{P}\mathbf{1} = k$ in the feasible filed is too strong. We can replace it with an implicit regularization term $(\mathbf{1}^T\mathbf{P}\mathbf{1} - k)^2$ and obtain a new objective function as

$$\min_{\mathbf{P}} F_r(\mathbf{P}, k) = F(\mathbf{P}) + \lambda_0(\mathbf{1}^T\mathbf{P}\mathbf{1} - k)^2. \qquad (32)$$

We set $\lambda_0 = 1$ is this paper. To solve Eq. (32), we can adopt the alternating optimization strategy: alternatively find the minimizer $\hat{\mathbf{P}}$ of Eq. (32) by Frank-Wolfe method with fixed k and then update $k = \mathbf{1}^T\hat{\mathbf{P}}\mathbf{1}$. Note that, in this case, Eq. (27) is solved by LP algorithms (interior point method in this paper) rather than the kLAP solvers since the constraint $\mathbf{1}^T\mathbf{P}\mathbf{1} = k$ dose not hold during solving Eq. (27).

Computational complexity. Since we do not use the affinity matrix $\mathbf{K}$, the space complexity is only $O(n^2)$. In optimization, each iteration takes time complexity $O(n^3)$ to solve the k-LAP or $O(m^3n^3/\ln(mn))$ to solve the LP, and $O(m^2n + mn^2)$ to compute the values and gradients of objective function. We are advised to adopt the kLAP-based approach based on the experimental analyses in Sec. 5.

4.2. Outlier identification and removal

After minimizing $F(\mathbf{P})$ or $F_r(\mathbf{P}, k)$, we obtain an optimal correspondence matrix $\hat{\mathbf{P}}$ that has two advantages beneficial to outlier identification: (1) $\hat{\mathbf{P}}$ optimally preserves the structural alignments between the two matched graphs. (2) The nearly zero-valued vectors $\hat{\mathbf{P}}_{i,:} \approx \mathbf{0}^T$ or $\hat{\mathbf{P}}_{:,a} \approx \mathbf{0}$ indicate that the node $V_i \in \mathcal{G}$ or $V_a' \in \mathcal{G}'$ can be identified as outliers, as an example shown in Fig. 2 (a).

An outlier removal approach is proposed based on this outlier identification criterion. Given $\hat{\mathbf{P}}$, we first calculate two vectors as $\hat{\mathbf{P}}\mathbf{1} = \{\hat{\mathbf{P}}_{i,:}\mathbf{1}\}_{i=1}^m$ and $\mathbf{1}^T\hat{\mathbf{P}} = \{\mathbf{1}^T\hat{\mathbf{P}}_{:,a}\}_{a=1}^n$, whose components with smaller values are more likely to be outliers. Then, $\hat{\mathbf{P}}\mathbf{1}$ and $\mathbf{1}^T\hat{\mathbf{P}}$ form the 2-dimensional coordinates of coupled nodes $\{(V_i, V_a')\}_{i,a}$ in the joint probability space, where the inliers and outliers can be significantly separated and clustered (*e.g.*, by k-means) into two classes, see an example in Fig. 2 (b). Assume that m', n' nodes of the two graphs are clustered as inliers by the clustering step, if $m' < k$ or $n' < k$, we pick out $k - m'$ or $k - n'$
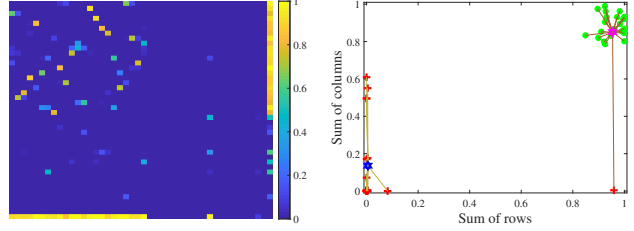

Figure 2: An example of the outlier identification and removal w.r.t. Fig. 1 (b). **Left**: the last row and column show the sums of column and row vectors of the optimal correspondence matrix $\hat{\mathbf{P}}$. **Right**: the inliers (green dots) and outliers (red signs) can be significantly separated and clustered into two classes. Note that, there is one outlier in each graph clustered as inlier due to its high similarity with the other inliers (see the matching result in Fig. 1 (b)).

nodes left with higher component values and put them back into inliers. If $m' > k$ or $n' > k$, the nodes with component values less than 0.5 will also be chosen as outliers. We iteratively execute this outlier removal procedure and then refine the inliers of two graphs till the enumerations of inliers keep unchanged. At last, the optimal solution solved w.r.t the refined graphs is our final matching result.

5. Experimental analysis

In this section, we evaluate and compare our methods (denoted as ZAC w.r.t. Eq. (20) and ZACR w.r.t Eq. (32)) with state-of-the-art graph matching methods including GA [13], RRWM [7], MPM [8], FGMD [44], BPFG [36] and FRGM [34] on widely used complicated datasets in terms of matching accuracy and time consumption. The codes of the compared methods are downloaded from the author's websites. Our code is available at https://github.com/wangfudong/ZAC_GM. For better evaluation of graph matching in the presence of outliers, we compute the commonly used indicators called *recall* $= \frac{\#\{correct\ matching\}}{\#\{groundtruth\ matching\}}$, *precision* $= \frac{\#\{correct\ matching\}}{\#\{total\ matching\}}$ and *F-measure*$= 2\frac{recall \cdot precision}{recall + precision}$.

5.1. Results on PASCAL dataset

We first conducted experiments on graphs in PASCAL dataset [25], which consists of 30 and 20 pairs of car and motorbike images (*e.g.*, Fig. 1), respectively. Each pair contains both inliers with known correspondence and randomly marked dozens of outliers. To generate graphs with outliers, we randomly selected 0, 4, ..., 20 outliers to both graphs, respectively. To generate the edges, our methods and FRGM applied complete graphs, while the others connected edges by Delaunay Triangulation, on which they achieved better performance than on complete graphs.

Similar with [44, 34], we set $\mathbf{K}_{ia;ia} = \exp(-d(\mathbf{v}_i -$
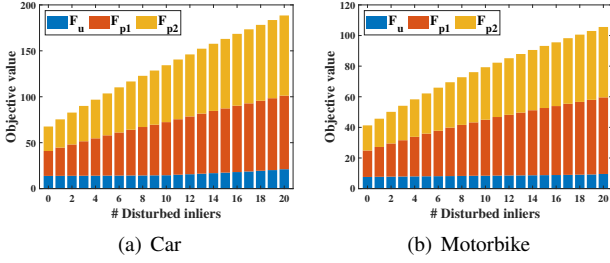
Figure 3: Statistical verification of the minimum values of our objective function Eq. (20).
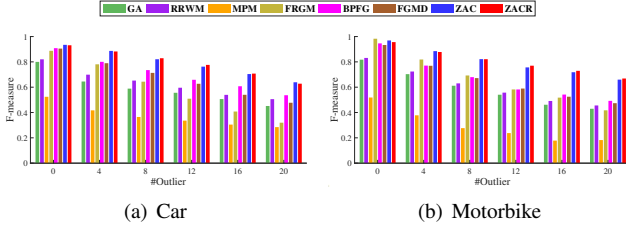

Figure 4: Average F-measure (%) w.r.t. number of outliers.

$\mathbf{v}'_a)$), and $\mathbf{K}_{ia;jb} = \exp(-\frac{1}{2}(|\mathbf{E}_{ij} - \mathbf{E}'_{ab}| + |\Theta_{ij} - \Theta'_{ab}|))$, where $\mathbf{v}_i, \mathbf{v}'_a$ were shape context [3], $d(\mathbf{v}_i - \mathbf{v}'_a)$ was the cost computed as χ^2 test statistic [3], $\mathbf{E}_{ij}, \mathbf{E}'_{ab}$ were distance matrices between nodes, $\Theta_{ij}, \Theta'_{ab}$ were the angles between the edges and the horizontal line. For our methods, we calculated $\mathbf{D}_{ia} = d(\mathbf{v}_i - \mathbf{v}'_a)$ to measure the node dissimilarity. For the weighted adjacency matrices $\mathcal{E}, \mathcal{E}'$ and edge attributes $\mathbf{A}_{ij}, \mathbf{B}_{ab}$, in order to honor the proposition 4, we set $\mathcal{E} = 1 \oslash \mathbf{E}, \mathcal{E}' = 1 \oslash \mathbf{E}'$ and $\mathbf{A} = \exp(-\mathbf{E}^2/\sigma_1^2), \mathbf{B} = \exp(-\mathbf{E}'^2/\sigma_2^2)$ with σ_1, σ_2 were the standard deviations of $\mathbf{E}, \mathbf{E}'$. The weights in Eq. (20) were $\lambda_1 = \lambda_2 = 1$.

First, we presented a statistical verification for proposition 4. For each graph pair with outliers, we randomly disturbed the ideal correspondences between inliers by forcing $0, 1, ..., 20$ inliers to be incorrectly matched. Then, we applied our optimization algorithm to minimize the objective function Eq. (20) under the mismatching constraints. We reported the series of obtained minimum values of objective function in Fig. 3. It shows that, with increasing number of disturbed matchings of inliers, the minimum values of objective function become higher. Only with no mismatchings (*i.e.*, the ideal ground-truth $\mathbf{P}^*$), the objective function achieves the lower limit of the series of minimum values. Namely, the proposition 4 can be guaranteed with our settings and optimization algorithm in practical cases.

Next, we compared all the methods in terms of matching accuracy and time consumption. For overall comparisons, we set a series of numbers $k = \lfloor ratio \cdot \min\{m, n\} \rfloor$ ($ratio = 0.3, 0.35, ..., 1$ such that $k \geq 5$ since $m, n \in [15, 75]$) in feasible fields $\hat{\mathcal{P}}_k$ for our method. And then, we also ran the compared methods with their soft-assignment
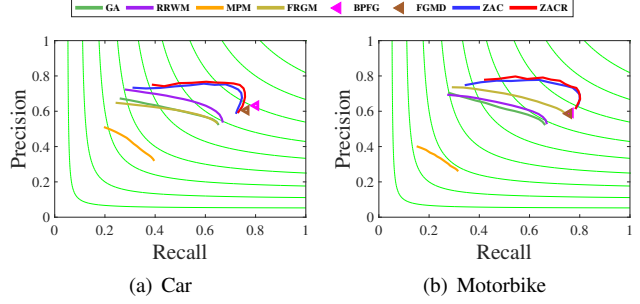

Figure 5: The average recall (%) and precision (%) w.r.t varying $ratio = 0.3, 0.35, ..., 1$ on PASCAL dataset.

#Outliers Methods	0	4	8	12	16	20
GA [13]	0.31	0.80	1.21	1.74	2.29	2.78
RRWM [7]	0.04	0.07	0.12	0.18	0.24	0.31
MPM [8]	0.35	0.61	0.94	1.40	2.06	3.05
FRGM [34]	0.44	0.61	0.78	0.96	1.14	1.36
BPFG [36]	1.07	23.84	37.79	61.04	83.41	122.59
FGMD [44]	0.68	10.01	12.67	15.44	19.47	24.21
ZAC	0.18	0.25	0.32	0.39	0.47	0.56
ZACR	0.53	0.75	0.89	1.05	1.20	1.36

Table 1: Average running time (s) w.r.t. number of outliers.

matrix and evaluated their matching accuracy with the top k matchings. Note that, since the methods FGMD [44] and BPFG [36] only obtain binary correspondences, we can only compute their matching accuracy with top $k = 1 \cdot \min\{m, n\}$ matchings.

Fig. 4 shows the highest average F-measure of all methods w.r.t the numbers of outliers. We can see that our methods ZAC and ZACR are more robust to outliers. Particularly, as shown in Fig. 5, with a wide range of $ratio$, our methods achieve much higher precision, which means that the proposed outlier identification and removal approach can efficiently reduce incorrect or redundant matchings. Tab. 1 reports the average time consumption, our methods take acceptable and intermediate time. Since the regularization term in Eq. (32) is more flexible than the equation constraint $\mathbf{1}^T \mathbf{P} \mathbf{1} = k$, ZACR has a little higher accuracy than ZAC. However, as mentioned in Sec. 4.1, since ZAC solves kLAP while ZACR uses LP solver, ZAC runs much faster than ZACR. Overall, ZAC achieves better trade-off between matching accuracy and time consumption than ZACR.

5.2. Results on VGG dataset

As the example shown in Fig. 1, graph pairs in PASCAL dataset are generated with similar shapes. Thus, the experiments above evaluate the performance of all the methods in terms of shape consistency. Furthermore, we conducted experiments on more practical dataset to evaluate all the GM methods with more complicated graphs under varying geo-
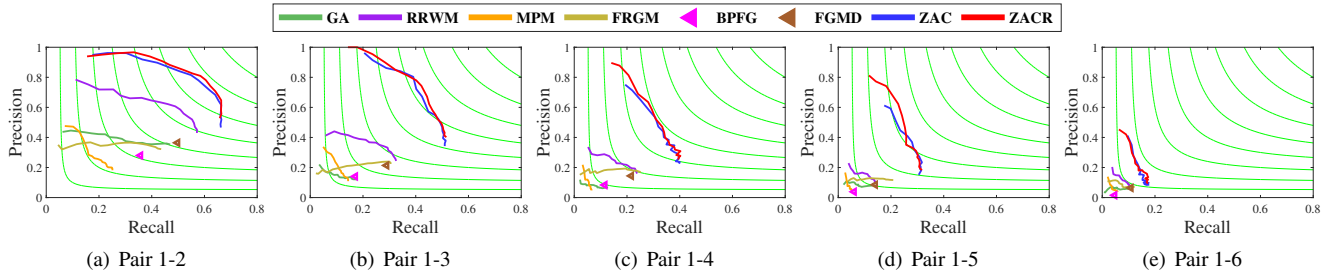
(a) Pair 1-2 (b) Pair 1-3 (c) Pair 1-4 (d) Pair 1-5 (e) Pair 1-6

Figure 6: Average recall (%) and precision (%) of all the methods with $ratio = 0.1, 0.15, ..., 1$. From pair 1-2 to 1-6, the graph pairs become more challenging for graph matching.

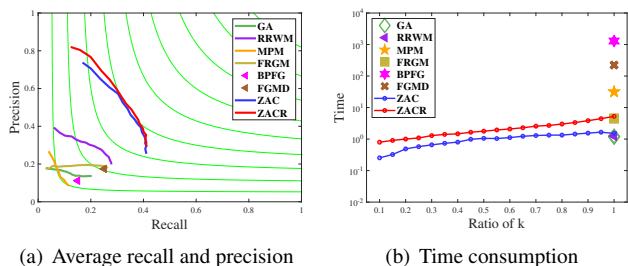(a) Average recall and precision (b) Time consumption

Figure 7: Average recall (%), precision (%) and time consumption (s) of all the methods on VGG dataset.

metric or physical factors.

We adopted the widely used VGG dataset[1] that consists of 8 groups of images (with sizes near 1000×1000) and each group has 6 images with varying blurring, viewpoint, rotation, light, zoom and JPEG compression (see examples in supplementary material). For each group, there exist 5 affine matrices $\mathbf{H}_{1s} \in \mathbb{R}^{3 \times 3}$, $s = 2, ..., 6$ that represent the ground-truth affine transformation from image 1 to images 2–6, respectively. We first formed graph pairs $\mathcal{G}, \mathcal{G}'$ between image 1 and images 2–6 in each group. Then, we utilized feature detector SIFT [26] to generate nodes of graphs. Note that, since the compared methods FGMD, BPFG and MPM were highly time consuming with large-scale complete graphs, we adjusted the threshold of SIFT such that the numbers of output features were around 100 and neglected repeated features. We computed the settings as the same as in PASCAL dataset except that $\{\mathbf{v}, \mathbf{v}_a'\}$ were SIFT features and set $\lambda_1 = 10, \lambda_2 = 1$.

An output matching result was evaluated as follows: for each node $V_i \in \mathcal{G}$ matched with $V_{a_i}' \in \mathcal{G}'$, we calculated its correct correspondence $V_{\sigma_i}' = \mathbf{H}_{1s} V_i$ using the ground-truth affine matrix $\mathbf{H}_{1s}$. Then, if the distance $||V_{a_i}' - V_{\sigma_i}'||$ was less than 10 pixels, the matching between V_i and V_{a_i}' was accepted as a correct matching. We set k with varying $ratio = 0.1, 0.15, ..., 1$ for the evaluation of recall and precision. Moreover, we also evaluated our time consumption w.r.t the varying $ratio$, since the number of nodes in

refined graphs obtained by our outlier identification and removal approach will be influenced by $ratio$. Note that, the time consumption of the other methods will not be affected by $ratio$ since they match all the nodes in graphs.

As shown in Fig. 6, under the varying geometric or physical conditions, our methods ZAC and ZACR can achieve much higher recall and precision. Fig. 7 (a) shows the overall average matching accuracy and time consumption w.r.t. varying $ratio$, our method ZAC and ZACR have much better matching accuracy within much less time consumption, even though on complicated graphs with numerous outliers and varying geometric or physical factors in practice.

5.3. Deformable graph matching

Deformable graph matching (DGM) [5, 43, 44, 34] is an important subproblem of GM, which focuses on incorporating rigid or non-rigid deformations between graphs. The main idea is to estimate both the correspondence $\mathbf{P}$ and deformation parameters τ by minimizing the sum of residuals

$$\min_{\mathbf{P}, \tau} J(\mathbf{P}, \tau) = \sum_{i,a} \mathbf{P}_{ia} ||V_a' - \tau(V_i)||^2 + \lambda_r \Upsilon(\tau), \quad (33)$$

where $V_i, V_a' \in \mathbb{R}^d$ are the nodes in $\mathcal{V}, \mathcal{V}'$, and $\Upsilon(\cdot)$ is a regularization term. Generally, the rigid or non-rigid deformation is parameterized as $\tau(\mathcal{V}) = s\mathcal{V}\mathbf{R} + t$ or $\tau(\mathcal{V}) = \mathcal{V} + \mathbf{W}\mathbf{G}$. See [44, 34] for more comprehensive reviews.
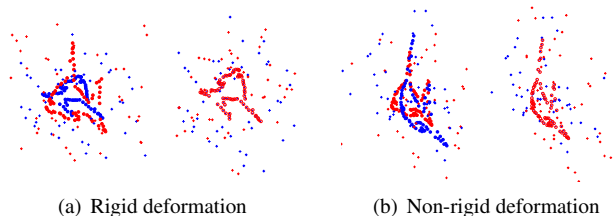

(a) Rigid deformation (b) Non-rigid deformation

Figure 8: Examples of deformable graph matching results of our method ZAC on the graphs under geometric deformations, to which the noises and outliers are also added.

Finding correct correspondence $\mathbf{P}$ plays the central role for solving Eq. (33). Once $\mathbf{P}$ is well-estimated, the geometric parameter τ can be solved with closed form [44, 34].
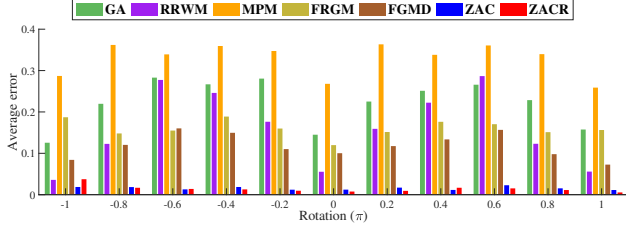
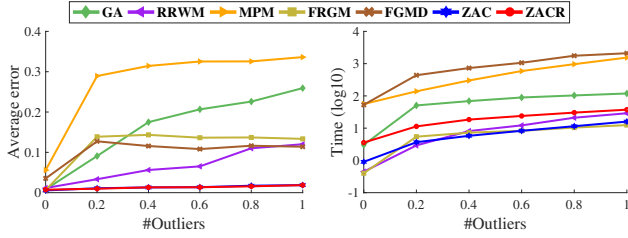Figure 9: Average errors w.r.t. varying rotations.

Figure 10: Average errors and time consumptions (s) w.r.t. varying outliers.

In this section, we applied all the GM methods with $k = 0.5 \times \min\{m, n\}$ to find $\mathbf{P}$ and then computed τ. We iteratively executed this procedure till it converges. All the GM methods applied the same settings used in Sec. 5.2 except that the node attributes $\mathbf{v}_i, \mathbf{v}'_a$ were shape context here.

We adopted the widely used 2D shape templates in [28, 27, 44, 34] for evaluation and comparison. We uniformly sampled 50% points of the shape template as inliers of $\mathcal{G}$ and $\mathcal{G}'$. And slight noises with uniform distribution $U(0, 0.01)$ were also added to $\mathcal{G}$ and $\mathcal{G}'$.

In this section, we conducted two series of experiments on graphs with varying deformations and outliers. First, we measured the robustness of each algorithm to rotations in rigid deformation. We rotated $\mathcal{G}'$ with varying degrees in $[-\pi, \pi]$ and then randomly added 10–50 outliers with Gaussian distribution $N(0, 0.5)$ to both $\mathcal{G}$ and $\mathcal{G}'$. Second, we evaluated the robustness to outliers. For the rigid deformation, we randomly rotated $\mathcal{G}'$ with degrees in $[-0.1\pi, 0.1\pi]$. For the non-rigid deformation, we deformed $\mathcal{G}'$ following the settings in [44, 34] by weight matrices $\mathbf{W}$ with Gaussian distribution $N(0, 0.5)$. And then, we incrementally added $\{0\%, 20\%, ..., 100\%\} \cdot \#inliers$ numbers of outliers with Gaussian distribution $N(0, 0.5)$ to $\mathcal{G}$ and $\mathcal{G}'$. See examples in Fig. 8. For all GM methods, we adopted the rotation-invariant shape context advised by [27, 34] for rigid deformations. For evaluation, we computed the average error between the transformed inliers $\{\tau(V_i)\}$ and their ground-truth matching point $\{V'_{\delta_i}\}$, i.e., $\frac{1}{\#inliers} \sum_i ||\tau(V_i) - V'_{\delta_i}||$.

Fig. 9 shows the average errors of all the methods w.r.t. varying rotations. We can see that our methods ZAC and ZACR can nearly perfectly recover and match all the graphs across all the rotations. Fig. 10 reports the average errors

and time consumptions on graphs with rigid and nonrigid deformations. Our methods have much less average error than all the other methods and runs faster than most of them. Note that, we did not compare BPFG due to its extremely unacceptable time consumption (more than 5 hours to match only one pair of graphs) in this experiment.

For more comprehensive evaluation, we also compared with two efficient point registration algorithms GLS [27] and CPD [28], which can address the deformable graph matching problem from the perspective of point registration. The comparison results are shown in Tab. 2, our methods ZAC and ZACR achieve comparable results with GLS and CPD for graphs without outliers, and have less average errors for complicated graphs with numerous outliers.

#Outliers Methods	0%	20%	40%	60%	80%	100%	
GLS [27]	**0.005**	0.006	0.011	0.021	0.030	0.044	Rigid
CPD [28]	**0.005**	0.052	0.085	0.112	0.124	0.135	
ZAC	**0.005**	**0.005**	**0.005**	**0.005**	**0.006**	**0.006**	
ZACR	0.009	**0.005**	0.006	**0.005**	**0.006**	**0.006**	
GLS [27]	0.010	**0.014**	0.043	0.050	0.064	0.092	Non-rigid
CPD [28]	**0.006**	0.015	0.042	0.065	0.069	0.083	
ZAC	**0.006**	0.016	**0.021**	0.022	0.028	**0.032**	
ZACR	**0.006**	**0.014**	**0.021**	0.021	**0.025**	**0.032**	

Table 2: Average errors w.r.t. varying outliers on graphs with rigid and non-rigid deformations.

6. Conclusion

This paper presents the zero-assignment constraint to address the problem of graph matching in the presence of outliers. Beyond the empirical criterion, we propose both theoretical foundations and quantitative analyses for this problem, on which bases we are inspired to construct reasonable objective function and find out the sufficient condition for its rationality. Moreover, we propose an efficient algorithm consisting of fast optimization and outlier identification, which ensures us to handle complicated graphs with numerous cluttered outliers in practice and achieve state-of-the-art performance in terms of accuracy and efficiency. In future work, we will go further to consolidate the theoretical foundation of graph matching problem with outliers by extending the zero-assignment constraint to the famous QAP formulations of graph matching in Eq. (1) and Eq. (2).

Acknowledgement

References

[1] HA Al-Mohamad and Salih o. Duffuaa. A linear programming approach for the weighted graph matching problem. *IEEE TPAMI*, 15(5):522–525, 1993.

[2] Kurt M. Anstreicher. Linear programming in O(n^3/ln nL) operations. *SIAM Journal on Optimization*, 9(4):803–812, 1999.

[3] Serge Belongie, Jitendra Malik, and Jan Puzicha. Shape matching and object recognition using shape contexts. *IEEE TPAMI*, 24(4):509–522, 2002.

[4] Harold P. Benson. Concave minimization: Theory, applications and algorithms. 1995.

[5] Tibério S. Caetano, Terry Caelli, Dale Schuurmans, and Dante Barone. Graphical models and point pattern matching. *IEEE TPAMI*, 28(10):1646–1663, 2006.

[6] Tiberio S. Caetano, Li Cheng, Quoc V. Le, and Alex J. Smola. Learning graph matching. *IEEE TPAMI*, 31(6):1048–1058, 2009.

[7] Minsu Cho, Jungmin Lee, and Kyoung Mu Lee. Reweighted random walks for graph matching. In *ECCV*, 2010.

[8] Minsu Cho, Jian Sun, Olivier Duchenne, and Jean Ponce. Finding matches in a haystack: A max-pooling strategy for graph matching in the presence of outliers. In *CVPR*, 2014.

[9] Donatello Conte, Pasquale Foggia, Carlo Sansone, and Mario Vento. Thirty years of graph matching in pattern recognition. *Int'l J. Pattern Recognition and Artificial Intelligence*, 18(3):265–298, 2004.

[10] Mauro Dell'Amico and Silvano Martello. The k-cardinality assignment problem. *Discrete Applied Mathematics*, 76(1-3):103–121, 1997.

[11] Amir Egozi, Yosi Keller, and Hugo Guterman. A probabilistic approach to spectral graph matching. *IEEE TPAMI*, 35(1):18–27, 2013.

[12] Michael R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, 1979.

[13] Steven Gold and Anand Rangarajan. A graduated assignment algorithm for graph matching. *IEEE TPAMI*, 18(4):377–388, 1996.

[14] Allen A. Goldstein. On steepest descent. *SIAM J. Control and Optimization*, 3(1):147–151, 1965.

[15] Roy Jonker and A. Volgenant. A shortest augmenting path algorithm for dense and sparse linear assignment problems. *Computing*, 38(4):325–340, 1987.

[16] Tjalling C. Koopmans and Martin Beckmann. Assignment problems and the location of economic activities. *Econometrica: J. the Econometric Society*, pages 53–76, 1957.

[17] Harold W. 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*, pages 29–47. Springer, 2010.

[18] Simon Lacoste-Julien and Martin Jaggi. On the global linear convergence of frank-wolfe optimization variants. In *NeurIPS*, 2015.

[19] Jean Lafond, Hoi-To Wai, and Eric Moulines. Nonconvex optimization with frank-wolfe algorithm and its variants. In *NeurIPS 2016 Workshop on Nonconvex Optimization for Machine Learning: Theory and Practice*, 2016.

[20] Eugene L. Lawler. The quadratic assignment problem. *Management Science*, pages 586–599, 1963.

[21] D. Khuê Lê-Huu and Nikos Paragios. Alternating direction graph matching. In *CVPR*, 2017.

[22] Jungmin Lee, Minsu Cho, and Kyoung Mu Lee. Hyper-graph matching via reweighted random walks. In *CVPR*, 2011.

[23] Marius Leordeanu and Martial Hebert. A spectral technique for correspondence problems using pairwise constraints. In *ICCV*, 2005.

[24] Marius Leordeanu, Martial Hebert, and Rahul Sukthankar. An integer projected fixed point method for graph matching and map inference. In *NeurIPS*. 2009.

[25] Marius Leordeanu, Rahul Sukthankar, and Martial Hebert. Unsupervised learning for graph matching. *IJCV*, 96(1):28–45, 2012.

[26] David G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004.

[27] Jiayi Ma, Ji Zhao, and Alan L. Yuille. Non-rigid point set registration by preserving global and local structures. *IEEE TIP*, 25(1):53–64, 2016.

[28] Andriy Myronenko and Xubo Song. Point set registration: Coherent point drift. *IEEE TPAMI*, 32(12):2262–2275, 2010.

[29] Yurii Nesterov and Arkadii Nemirovskii. *Interior-Point Polynomial Algorithms in Convex Programming*. Society for Industrial and Applied Mathematics, 1994.

[30] Tianwei Shen, Siyu Zhu, Tian Fang, Runze Zhang, and Long Quan. Graph-based consistent matching for structure-from-motion. In *ECCV*, 2016.

[31] Lorenzo Torresani, Vladimir Kolmogorov, and Carsten Rother. A dual decomposition approach to feature correspondence. *IEEE TPAMI*, 35(2):259–271, 2013.

[32] Shinji Umeyama. An eigendecomposition approach to weighted graph matching problems. *IEEE TPAMI*, 10(5):695–703, 1988.

[33] A. Volgenant. Solving the k-cardinality assignment problem by transformation. *European Journal of Operational Research*, 157(2):322–331, 2004.

[34] Fu-Dong Wang, Gui-Song Xia, Nan Xue, Yipeng Zhang, and Marcello Pelillo. A functional representation for graph matching. *IEEE TPAMI*, early access, 2019.

[35] Runzhong Wang, Junchi Yan, and Xiaokang Yang. Learning combinatorial embedding networks for deep graph matching. In *ICCV*, 2019.

[36] Tao Wang, Haibin Ling, Congyan Lang, , and Songhe Feng. Graph matching with adaptive and branching path following. *IEEE TPAMI*, 40(12):2853–2867, 2018.

[37] Junchi Yan, Xu-Cheng Yin, Weiyao Lin, Cheng Deng, Hongyuan Zha, and Xiaokang Yang. A short survey of recent advances in graph matching. In *ICMR*, 2016.

[38] Bangpeng Yao and Fei-Fei Li. Action recognition with exemplar based 2.5D graph matching. In *ECCV*, 2012.

[39] Andrei Zanfir and Cristian Sminchisescu. Deep learning of graph matching. In *CVPR*, 2018.

[40] Mikhail Zaslavskiy, Francis Bach, and Jean-Philippe Vert. A path following algorithm for the graph matching problem. *IEEE TPAMI*, 31(12):2227–2242, 2009.

[41] Ron Zass and Amnon Shashua. Probabilistic graph and hypergraph matching. In *CVPR*, 2008.

[42] Yun Zeng, Chaohui Wang, Yang Wang, Xianfeng Gu, Dimitris Samaras, and Nikos Paragios. Dense non-rigid surface registration using high-order graph matching. In *CVPR*, 2010.

[43] Yefeng Zheng and David Doermann. Robust point matching for nonrigid shapes by preserving local neighborhood structures. *IEEE TPAMI*, 28(4):643–649, 2006.

[44] Feng Zhou and Fernando De la Torre. Factorized graph matching. *IEEE TPAMI*, 38(9):1774–1789, 2016.