Highlights

Factorized Multi-Graph Matching

Liangliang Zhu, Xinwen Zhu, Xiurui Geng

- The equivalence between the two kinds of multi-graph matching methods is proved.
- A multi-graph matching method that can avoid cycle-consistency problem is proposed.
- The proposed method is of low computational complexity.
- An efficient updating algorithm is proposed to solve the optimization problem.

Factorized Multi-Graph Matching

Liangliang Zhu[a,b,c], Xinwen Zhu[a,b,c], Xiurui Geng[a,b,c,*]

[a] *Aerospace Information Research Institute, Chinese Academy of Sciences, Beijing, 100864, China*
[b] *School of Electronic, Electrical and Communication Engineering, University of Chinese Academy of Sciences, Beijing, 100049, China*
[c] *Key Laboratory of Technology in Geo-Spatial Information Processing and Application System, Chinese Academy of Sciences, Beijing, 100864, China*

Abstract

In recent years, multi-graph matching has become a popular yet challenging task in graph theory. There exist two major problems in multi-graph matching, i.e., the cycle-consistency problem, and the high time and space complexity problem. On one hand, the pairwise-based multi-graph matching methods are of low time and space complexity, but in order to keep the cycle-consistency of the matching results, they need additional constraints. Besides, the accuracy of the pairwise-based multi-graph matching is highly dependent on the selected optimization algorithms. On the other hand, the tensor-based multi-graph matching methods can avoid the cycle-consistency problem, while their time and space complexity is extremely high. In this paper, we found the equivalence between the pairwise-based and the tensor-based multi-graph matching methods under some specific circumstances. Based on this finding, we proposed a new multi-graph matching method, which not only avoids the cycle-consistency problem, but also reduces the complexity. In addition, we further improved the proposed method by introducing a lossless factorization of the affinity matrix in the multi-graph matching methods. Synthetic and real data experiments demonstrate the superiority of our method.

Keywords: Graph matching, Multi-graph matching, Tensor, Factorization

*Corresponding author
Email address: genxr@sina.com.cn (Xiurui Geng)

1. Introduction

Graph matching, which aims to find the correspondences between two given graphs, is a fundamental yet challenging problem in computer vision [1], pattern recognition [2], data mining [3], and so on [4, 5, 6]. In the last few decades, extensive research has been done on the graph matching problem: The graph matching models have evolved from the simplest linear assignment problem [7] to the quadratic assignment problem [8] (the mostly used), and to high-order matching problem [9, 10, 11]. The optimal solution of the linear assignment problem can be obtained by the Hungarian method [12] in polynomial time. However, the quadratic assignment problem and high-order matching problem are known to be NP-hard. Therefore, many algorithms [13, 14, 15] are proposed to efficiently obtain an approximation solution for them. In addition, a few studies [16, 17] have also been done to overcome the high time and space complexity of the graph matching problem. Recently, researchers have moved their focus to the multi-graph matching (MGM) problem, which is designed to simultaneously establish the correspondences between more than two graphs.

In MGM, a major task is to ensure that the solution is cycle-consistent. For example, if node a corresponds to node b, and b corresponds to node c, then node a must correspond to node c. For example, in Fig. 1a, the matching result is cycle-consistent, on the contrary, the cycle-consistency is not satisfied in the matching result in Fig. 1b. It's obvious that when employing a pairwise matching method to MGM, the independently calculated pairwise matching results could be cycle-inconsistent. Therefore, many MGM methods were proposed to solve this problem, and most of them can be divided into two categories [18]: one-shot methods and iterative methods. The one-shot methods [19, 20, 21] try to directly extract a cycle-consistent solution from the matching results obtained by other pairwise matching methods. Since no reinforcement is performed to increase the objective function of MGM, these methods are usually sensitive to their inputs. The iterative methods [22, 23, 24] aim to maximize the objective function of MGM, and account for the cycle-consistency at the same time. In [22, 23], the cycle-consistency is strictly obeyed over the whole solution updating procedure. Meanwhile, in [24], the importance of cycle-consistency

gradually increases during the iteration, which is more flexible.

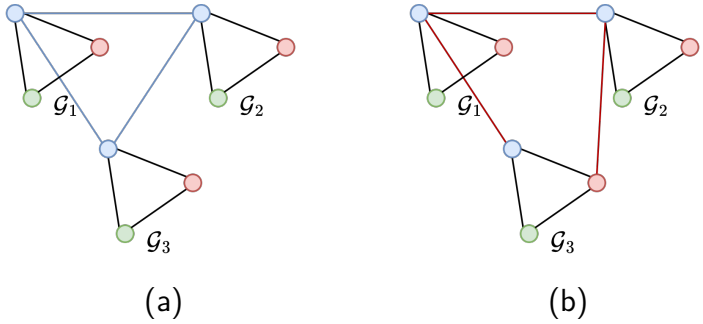


Figure 1: An illustration of the cycle-consistency in MGM (a) a cycle-consistent example (b) a cycle-inconsistent example

Notice that the aforementioned methods are all based on the pairwise matching techniques, thus, maintaining the cycle-consistency of the matching results becomes an inevitable task for them. Recently, in [25], the authors introduced a tensor-like formulation for the objective function of MGM, in which the cycle-consistency problem is naturally avoided. Unfortunately, the proposed formulation is of extremely high time and space complexity. Therefore, during implementation, the authors only apply the tensor-like formulation for the first-order (node-to-node similarity) part of the objective function in MGM. Besides, a simplified pairwise-based MGM model is used to formulate the second-order (edge-to-edge similarity) part, so that the time and space complexity of the whole algorithm is acceptable in practice.

Although the pairwise-based MGM and the tensor-based MGM seem different in formulation, it is surprising that the former can be attributed to a special case of the latter, which will be discussed in detail in the following sections. This finding indicates that the cycle-consistency problem could be avoided in the pairwise-based MGM methods. In addition, by utilizing the sparsity of the affinity matrix in the model [17], we hope to further reduce the complexity of the proposed method. Moreover, due to the integer constraints in MGM, there exist many local maximums in the objective function. Therefore, the stochastic gradient descent method is adopted in the presented approximation algorithm, which is expected to alleviate this problem.

2. Background

In this section, we first provide a brief introduction of the pairwise matching problem, then the formulations of the pairwise-based and tensor-based MGM are presented. For convenience, the definitions of the used notations in this paper are introduced in the following.

Generally, a graph is denoted by a set $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, where $\mathcal{V}$ represents a set of nodes and $\mathcal{E}$ denotes edges. In addition, the number of nodes in $\mathcal{G}$ is $n_v = |\mathcal{V}|$, and the number of edges is $n_e = |\mathcal{E}|$. Moreover, in this paper, a graph $\mathcal{G}$ is also associated with 4 incidence matrices: $\mathbf{H}^{11} \in \{0,1\}^{n_v \times n_v}, \mathbf{H}^{12} \in \{0,1\}^{n_v \times n_e}, \mathbf{H}^{21} \in \{0,1\}^{n_e \times n_v}, \mathbf{H}^{22} \in \{0,1\}^{n_e \times n_e}$. The incidence matrix $\mathbf{H}^{ab}$ is used to indicate the connections between the a^{th}-order features and the b^{th}-order features. For instance, if the i^{th} second-order feature (the i^{th} edge) is connected to the j^{th} first-order feature (the j^{th} node), then $\mathbf{H}^{21}_{ij}$ is set to 1. In addition, for the k^{th} graph $\mathcal{G}^{[k]}$ from a set of multiple graphs, its numbers of nodes and edges are denoted $n_v^{[k]}$ and $n_e^{[k]}$, respectively. The incidence matrices of $\mathcal{G}^{[k]}$ are given in the form of $\mathbf{H}^{[k]^{ab}}$.

2.1. Pairwise Matching

Given two graphs $\mathcal{G}^{[1]}$ and $\mathcal{G}^{[2]}$ (assume that $n_v^{[1]} \leq n_v^{[2]}$), then the pairwise matching model can be formulated into the well-known quadratic assignment problem, where the objective is to find an indicator matrix $\mathbf{X}$ that maximizes the following quadratic score function.

$$\begin{aligned} \hat{\mathbf{X}} &= \arg\max_{\mathbf{X}} \sum_{i_1 i_2, j_1 j_2} \mathbf{K}^{[12]}_{i_1 i_2, j_1 j_2} \mathbf{X}_{i_1 i_2} \mathbf{X}_{j_1 j_2} \\ &= \arg\max_{\mathbf{X}} \ vec(\mathbf{X})^{\mathrm{T}} \mathbf{K}^{[12]} vec(\mathbf{X}) \\ &\text{s.t. } \mathbf{X}\mathbf{1}^{[2]} = \mathbf{1}^{[1]}, \mathbf{X}^{\mathrm{T}}\mathbf{1}^{[1]} \leq \mathbf{1}^{[2]}, \mathbf{X} \in \{0,1\}^{n_v^{[1]} \times n_v^{[2]}}, \end{aligned} \tag{1}$$

where $\mathbf{1}^{[i]}$ is an all-ones vector with the size of $n_v^{[i]} \times 1$, and $\mathbf{K}^{[12]} \in \mathbb{R}^{n_v^{[1]} n_v^{[2]} \times n_v^{[1]} n_v^{[2]}}$ is the affinity matrix. The diagonal element $\mathbf{K}^{[12]}_{i_1 i_2, i_1 i_2}$ measures the similarity between i_1^{th} node in $\mathcal{G}^{[1]}$ and i_2^{th} node in $\mathcal{G}^{[2]}$. And the non-diagonal element $\mathbf{K}^{[12]}_{i_1 i_2, j_1 j_2}$ measures the similarity between the edge formed by the i_1^{th}, j_1^{th} nodes in $\mathcal{G}^{[1]}$ and that formed by the i_2^{th}, j_2^{th} nodes in $\mathcal{G}^{[2]}$.

Due to the fact that in many real applications, the edges of a graph are sparse, thus, many elements in $\mathbf{K}^{[12]}$ are zeros. In addition, the nonzero elements in $\mathbf{K}^{[12]}$ are contained in the node-to-node similarity matrix $\mathbf{S} \in \mathbb{R}^{n_v^{[1]} \times n_v^{[2]}}$ and the edge-to-edge similarity matrix $\mathbf{T} \in \mathbb{R}^{n_e^{[1]} \times n_e^{[2]}}$. In [17], the authors noticed this phenomenon and proposed the following factorization of the affinity matrix:

$$\mathbf{K}^{[12]} = (\mathbf{V}^{[2]} \otimes \mathbf{V}^{[1]}) diag(\mathbf{D}) (\mathbf{V}^{[2]} \otimes \mathbf{V}^{[1]})^{\mathrm{T}}, \tag{2}$$

where $\otimes$ represents the Kronecker product, and

$$\mathbf{V}^{[1]} = \left[\mathbf{H}^{[1]^{21}}, \mathbf{H}^{[1]^{11}}\right], \mathbf{V}^{[2]} = \left[\mathbf{H}^{[2]^{21}}, \mathbf{H}^{[2]^{11}}\right],$$
$$\mathbf{D} = \begin{bmatrix} \mathbf{T} & -\mathbf{T}\mathbf{H}^{[2]^{21}} \\ -\mathbf{H}^{[1]^{12}}\mathbf{T} & \mathbf{H}^{[1]^{12}}\mathbf{T}\mathbf{H}^{[2]^{21}} + \mathbf{S} \end{bmatrix}.$$

It should be noted that in (2), $diag(\cdot)$ vectorizes the given matrix first, and then constructs a diagonal matrix from the generated vector. The proposed factorization can avoid computing the costly affinity matrix without loss of information, thus, it has attracted extensive attention in recent years. Moreover, if the graphs are undirected, the affinity matrix $\mathbf{K}$ is then supersymmetric, i.e., $\mathbf{K}^{[12]}_{i_1 i_2, j_1 j_2} = \mathbf{K}^{[12]}_{j_1 i_2, i_1 j_2} = \mathbf{K}^{[12]}_{i_1 j_2, j_1 i_2} = \mathbf{K}^{[12]}_{j_1 j_2, i_1 i_2}$. This indicates that there are $n_v^{[1]} n_v^{[2]} + 4 n_e^{[1]} n_e^{[2]}$ nonzero elements in $\mathbf{K}^{[12]}$. But when using the factorization in (2), we only need to store $(n_v^{[1]} + n_e^{[1]})(n_v^{[2]} + n_e^{[2]})$ elements, i.e., the matrix $\mathbf{D}$. Consequently, the computational complexity of subsequent calculations can be reduced.

2.2. Multi-graph Matching

Given m graphs, the objective function of most MGM methods is formulated as follows

$$\begin{aligned} \hat{\mathbf{X}}^{[pq]} = & \arg\max_{\mathbf{X}^{[\mathbf{pq}]}} \sum_{p,q,p \neq q} vec(\mathbf{X}^{[pq]})^{\mathrm{T}} \mathbf{K}^{[pq]} vec(\mathbf{X}^{[pq]}) \\ \text{s.t. } & \mathbf{X}^{[pq]} \mathbf{1}^{[q]} = \mathbf{1}^{[p]}, (\mathbf{X}^{[pq]})^{\mathrm{T}} \mathbf{1}^{[p]} \leq \mathbf{1}^{[q]}, \\ & \mathbf{X}^{[pq]} \in \{0,1\}^{n_v^{[p]} \times n_v^{[q]}}, \mathbf{X}^{[pr]} \mathbf{X}^{[rq]} = \mathbf{X}^{[pq]}, \end{aligned} \tag{3}$$

where the notations with a superscript of $^{[pq]}$ are the corresponding variables for $\mathcal{G}^{[p]}$ and $\mathcal{G}^{[q]}$. For example, $\mathbf{K}^{[pq]} \in \mathbb{R}^{n_v^{[p]} n_v^{[q]} \times n_v^{[p]} n_v^{[q]}}$ is the affinity matrix between $\mathcal{G}^{[p]}$ and $\mathcal{G}^{[q]}$. It can be

found that (3) is equivalent to maximizing the sum of all objective functions of the pairwise graph matching problems. Besides, $\mathbf{X}^{[pr]}\mathbf{X}^{[rq]} = \mathbf{X}^{[pq]}$ ensures the cycle-consistency of the matching results.

The iterative methods consider the cycle-consistency constraint during the whole variable updating procedure. For example, in [22, 23], the authors wrote the objective function of MGM into the following form

$$\begin{aligned}&\sum_{p,q,p\neq q} vec(\mathbf{X}^{[pr]}\mathbf{X}^{[rq]})^{\mathrm{T}}\mathbf{K}^{[pq]}vec(\mathbf{X}^{[pr]}\mathbf{X}^{[rq]})\\ =&\sum_{p} vec(\mathbf{X}^{[pr]})^{\mathrm{T}}(\sum_{q,q\neq p}(\mathbf{X}^{[qr]}\otimes\mathbf{I})^{\mathrm{T}}\mathbf{K}^{[pq]}(\mathbf{X}^{[qr]}\otimes\mathbf{I}))vec(\mathbf{X}^{[pr]}),\end{aligned} \tag{4}$$

where $\mathbf{I}$ is an identity matrix. Then, for a specific $\mathbf{X}^{[pr]}$, the solution can be obtained by solving a pairwise matching problem with the affinity matrix of $\mathbf{K}'^{[pr]} = \sum_{q,q\neq p}(\mathbf{X}^{[qr]} \otimes \mathbf{I})^{\mathrm{T}}\mathbf{K}^{[pq]}(\mathbf{X}^{[qr]} \otimes \mathbf{I})$. Notice that when any element in $\{\mathbf{X}^{[qr]}|q \neq r\}$ changes during iteration, the affinity matrix $\mathbf{K}'^{[pr]}$ needs to be regenerated to ensure the cycle-consistency. Besides, in [23], the authors also introduced the factorization of the affinity matrix into their work, so that the proposed method can be solved in a path-following way. However, due to the fact that the proposed method needs to continuously generate the affinity matrix $\mathbf{K}'^{[pr]}$ during iteration, the computational complexity of the factorized version of their method is not reduced.

Meanwhile, the one-shot methods optimize every pairwise graph matching problem independently, and obtain a matrix $\mathbf{W}$ that collects all pairwise results, i.e.,

$$\mathbf{W} = \begin{bmatrix}\mathbf{X}^{[11]} & \cdots & \mathbf{X}^{[1m]}\\ \vdots & \ddots & \vdots\\ \mathbf{X}^{[m1]} & \cdots & \mathbf{X}^{[mm]}\end{bmatrix}. \tag{5}$$

Notice that if the cycle-consistency is satisfied, the matrix $\mathbf{W}$ can be decomposed as $\mathbf{W} = \mathbf{U}^{\mathrm{T}}\mathbf{U}$, where $\mathbf{U} = [\mathbf{X}^{[11]}, \cdots, \mathbf{X}^{[1m]}]$. Most one-shot methods utilize the low-rank property of $\mathbf{W}$, and try to reconstruct a cycle-consistent $\mathbf{W}$ from the results of other pairwise matching methods. It can be found that, after obtaining $\mathbf{W}$, the affinity matrices are no longer involved in further calculations. Thus, the one-shot methods are usually of low computational

complexity. However, in the one-shot methods, no reinforcement is performed to further increase the objective function of MGM. Therefore, when the same pairwise matching method is used, the performance of the one-shot methods is generally worse than that of the iterative methods.

Recently, a tensor form of the MGM problem is proposed, where the cycle-consistent problem is naturally avoided. Let $\mathcal{X} \in \{0,1\}^{n_v^{[1]} \times n_v^{[2]} \times \cdots \times n_v^{[m]}}$ be the indicator tensor, and $\mathcal{X}_{i_1:i_m} = 1$ indicates that the i_1^{th} node in $\mathcal{G}^{[1]}$, $\ldots$, and the i_m^{th} node in $\mathcal{G}^{[m]}$ are matched. Notice that the subscript $_{i_1:i_m}$ is the short for $_{i_1,i_2,\cdots,i_m}$, and $\mathcal{X}_{i_1:i_m}$ is the element with the index of $i_1, i_2, \cdots, i_m$ in the tensor $\mathcal{X}$. Then, the tensor-based MGM can be formulated as follows (without loss of generality, assume that the first graph has the smallest number of nodes).

$$
\begin{aligned}
\hat{\mathcal{X}} &= \arg\max_{\mathcal{X}} \sum_{i_1:i_m} \mathcal{S}_{i_1:i_m} \mathcal{X}_{i_1:i_m} + \sum_{i_1:i_m, j_1:j_m} \mathbf{K}_{i_1:i_m, j_1:j_m} \mathcal{X}_{i_1:i_m} \mathcal{X}_{j_1:j_m} \\
&= \arg\max_{\mathcal{X}} \ vec(\mathcal{S})^{\mathrm{T}} vec(\mathcal{X}) + vec(\mathcal{X})^{\mathrm{T}} \mathbf{K} vec(\mathcal{X}) \\
&\text{s.t. } \mathcal{X} \in \{0,1\}^{n_v^{[1]} \times n_v^{[2]} \times \cdots \times n_v^{[m]}}, \\
&\quad \mathcal{X} \times_2 \mathbf{1}^{[2]} \cdots \times_m \mathbf{1}^{[m]} = \mathbf{1}^{[1]}, \\
&\quad \mathcal{X} \times_1 \mathbf{1}^{[1]} \cdots \times_m \mathbf{1}^{[m]} \leq \mathbf{1}^{[2]}, \\
&\quad \cdots, \\
&\quad \mathcal{X} \times_1 \mathbf{1}^{[1]} \cdots \times_{m-1} \mathbf{1}^{[m-1]} \leq \mathbf{1}^{[m]},
\end{aligned}
\tag{6}
$$

where $\mathcal{S} \in \mathbf{R}^{n_v^{[1]} \times \cdots \times n_v^{[m]}}$ is the node similarity tensor, and $\mathbf{K} \in \mathbf{R}^{(n_v^{[1]} \cdots n_v^{[m]}) \times (n_v^{[1]} \cdots n_v^{[m]})}$ is the affinity matrix in MGM. In (6), $\times_k$ represents the k-mode product of a tensor $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_K}$ with a matrix $\mathbf{B} \in \mathbb{R}^{I_k \times J}$, which is defined as follows in this paper

$$
(\mathcal{A} \times_k \mathbf{B})_{i_1 \cdots j \cdots i_N} = \sum_{i_k=1}^{I_k} \mathcal{A}_{i_1 \cdots i_k \cdots i_K} \mathbf{B}_{i_k j}. \tag{7}
$$

In addition, if $\mathbf{K}_{i_1:i_m, i_1:i_m}$ (the diagonal elements) equals $\mathcal{S}_{i_1:i_m}$, the two parts of the objective function can be integrated into one piece, i.e., $vec(\mathcal{X})^{\mathrm{T}} \mathbf{K} vec(\mathcal{X})$.

Though there is no cycle-consistent constraint in the tensor-based MGM, the time and space complexity of it geometrically increases with the number of graphs. Thus, the authors

[25] adopted a simplified version in practice, where a tensor $\mathcal{C} \in \mathbb{R}^{(n_v^{[1]} n_v^{[2]}) \times \cdots \times (n_v^{[m-1]} n_v^{[m]})}$ is constructed so that

$$vec(\mathcal{S})^{\mathrm{T}} vec(\mathcal{X}) = \mathcal{C} \times_1 vec(\mathbf{X}^{[12]}) \cdots \times_{m-1} vec(\mathbf{X}^{[(m-1)m]}). \tag{8}$$

Then, the simplified objective function can be written as follows

$$\begin{aligned} \hat{\mathbf{X}}^{[p(p+1)]} = & \underset{\mathbf{X}^{[\mathbf{p}(\mathbf{p+1})]}}{\arg\max} \mathcal{C} \times_1 vec(\mathbf{X}^{[12]}) \cdots \times_{m-1} vec(\mathbf{X}^{[(m-1)m]}) \\ & + \sum_{p=1}^{m-1} vec(\mathbf{X}^{[p(p+1)]})^{\mathrm{T}} \mathbf{K}^{[p(p+1)]} vec(\mathbf{X}^{[p(p+1)]}). \end{aligned} \tag{9}$$

Nonetheless, the computational complexity of the proposed algorithm still cannot be ignored. In the section of complexity analysis in [25], the authors mentioned that it could cost 10 minutes for their method to match 12 graphs all with 10 nodes. Meanwhile, the existing pairwise-based MGM methods usually cost only seconds to accomplish the same task.

3. Method

In this section, we first discuss the equivalence between the pairwise-based MGM and the tensor-based MGM. Based on the equivalence, we proposed a new MGM method, where the cycle-consistency problem is avoided. To further reduce the complexity, we introduce the factorization of the affinity matrix into the proposed method. In addition, an approximation algorithm based on the stochastic gradient descent is presented in this section to solve the optimization problem in (9).

3.1. Pairwise-based and Tensor-based MGM

Assume that for all m graphs to be matched, there exists a virtual graph $\mathcal{G}^{[0]}$, which has $n_v^{[0]} = \min(n_v^{[1]}, \cdots, n_v^{[m]})$ nodes, and the indicator matrix between $\mathcal{G}^{[0]}$ and $\mathcal{G}^{[i]}$ is $\mathbf{X}^{[0i]} \in \{0,1\}^{n_v^{[0]} \times n_v^{[i]}}$. Similarly, $\mathbf{X}^{[i0]} = \mathbf{X}^{[0i]^{\mathrm{T}}}$ is the indicator matrix between $\mathcal{G}^{[i]}$ and $\mathcal{G}^{[0]}$. Then, we have the following theorem for the overall indicator tensor $\mathcal{X}$ (the detailed derivation can be seen in the appendix).

Theorem 1. *Let* $\mathbf{X}^{[0i]}$ *be the indicator matrix between the virtual graph* $\mathcal{G}^{[0]}$ *and the* i^{th} *graph* $\mathcal{G}^{[i]}$*, then the indicator tensor* $\mathcal{X}$ *for all* m *graphs can be factorized as follows*

$$\mathcal{X} = \mathcal{I} \times_1 \mathbf{X}^{[01]} \times_2 \mathbf{X}^{[02]} \cdots \times_m \mathbf{X}^{[0m]}, \tag{10}$$

where $\mathcal{I}$ *is a diagonal tensor with the size of* $n_v^{[0]} \times n_v^{[0]} \times \cdots \times n_v^{[0]}$*, and the diagonal elements are all ones.*

Based on Theorem 1, when the affinity matrix $\mathbf{K}$ is specially constructed, the objective function of the tensor-based MGM can be written into a pairwise form. The details are given in Theorem 2, and the derivation can be seen in the appendix.

Theorem 2. *When the affinity matrix* $\mathbf{K}$ *in the tensor-based MGM is specially constructed as follows*

$$\mathbf{K}_{i_1:i_m,j_1:j_m} = \sum_{p,q,p\neq q} \mathbf{K}^{[pq]}_{i_p i_q, j_p j_q},$$

where $\mathbf{K}^{[pq]}$ *is the affinity matrix between* $\mathcal{G}^{[p]}$ *and* $\mathcal{G}^{[q]}$*. Then, the objective function in the tensor-based MGM can be written as*

$$vec(\mathcal{X})^{\mathrm{T}} \mathbf{K} vec(\mathcal{X}) = \sum_{p,q,p\neq q} vec(\mathbf{X}^{[p0]}\mathbf{X}^{[0q]})^{\mathrm{T}} \mathbf{K}^{[pq]} vec(\mathbf{X}^{[p0]}\mathbf{X}^{[0q]}). \tag{11}$$

It can be found that if the virtual graph is selected as $\mathcal{G}^{[r]}$, (11) is exactly the same as (4), which is the objective function of the pairwise-based MGM methods. Therefore, the pairwise-based MGM is actually a special case of the tensor-based MGM.

Furthermore, when the similarity between a set of features is given by the sum of all possible pairwise similarities, the special construction of K in Theorem 2 is always satisfied. Such a similarity measure strategy is natural and reasonable, and on this condition, the equivalence presented in Theorem 2 is universal. For example, in Fig. 2, the whole similarity for the three green labeled edges is given by the following equation

$$\mathbf{K}_{111,222} = \mathbf{K}^{[12]}_{11,22} + \mathbf{K}^{[21]}_{11,22} + \mathbf{K}^{[13]}_{11,22} + \mathbf{K}^{[31]}_{11,22} + \mathbf{K}^{[23]}_{11,22} + \mathbf{K}^{[32]}_{11,22}.$$

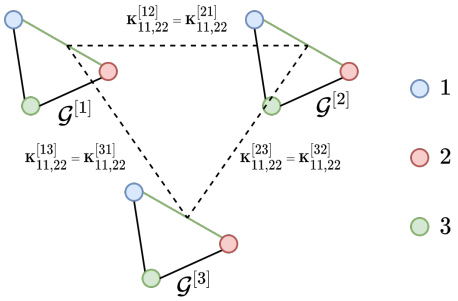


Figure 2: Illustration of the special construction of the affinity matrix $\mathbf{K}$ in Theorem 2

3.2. Factorized MGM

From (10), it can be found that the correspondences between all graphs to be matched can be completely determined by m indicator matrices $\{\mathbf{X}^{[0i]}|i = 1, \cdots, m\}$. Therefore, instead of obtaining a cycle-consistent matching result for all graphs from the pairwise matching results, we directly calculate the m indicator matrices $\{\mathbf{X}^{[0i]}|i = 1, \cdots, m\}$ between the virtual graph and the graphs to be matched. Moreover, a cycle-consistent matching result can be obtained by setting $\mathbf{X}^{[pq]} = \mathbf{X}^{[p0]}\mathbf{X}^{[0q]}$ for all p, q. Because there is no cycle-consistency constraint between $\mathbf{X}^{[01]}, \cdots, \mathbf{X}^{[0m]}$, the cycle-consistency problem is avoided in the proposed method.

Inspired by (11), the virtual graph is selected as one of the graphs to be matched, and the indicator matrix $\mathbf{X}^{[p0]}$ between $\mathcal{G}^{[p]}$ and the virtual graph can be obtained by solving the following two-graph matching problem:

$$\hat{\mathbf{X}}^{[p0]} = \arg\max_{\mathbf{X}^{[p0]}} \ vec(\mathbf{X}^{[p0]})^{\mathrm{T}}\mathbf{K}'^{[p0]}vec(\mathbf{X}^{[p0]}), \tag{12}$$

where $\mathbf{K}'^{[p0]} = \sum_{q,q\neq p}(\mathbf{X}^{[q0]} \otimes \mathbf{I})^{\mathrm{T}}\mathbf{K}^{[pq]}(\mathbf{X}^{[q0]} \otimes \mathbf{I})$. Because $\mathbf{X}^{[q0]}$ is a permutation matrix, the computational complexity of generating $\mathbf{K}'^{[pr]}$ can be reduced to $O(mn_v^3)$[23] (without loss of generality, we assume that all graphs have n_v nodes and n_e edges). However, it is still unacceptable when there are numerous nodes.

We notice that in most pairwise matching methods, only the gradients of the objective function to the variables are required. Therefore, we prefer to directly deduce the gradients

for $\mathbf{X}^{[p0]}$, which is

$$\nabla \mathbf{X}^{[p0]} = 2 \sum_{q,q\neq p} mat(\mathbf{K}^{[pq]} vec(\mathbf{X}^{[p0]}\mathbf{X}^{[0q]}))\mathbf{X}^{[q0]}. \tag{13}$$

In (13), $mat(\cdot)$ is used to reshape a vector into a matrix, and the derivation of the gradient is presented in the appendix. By directly calculating the gradients, the procedure of generating $\mathbf{K}'^{[p0]}$ can be skipped. Moreover, the calculation of $\mathbf{K}^{[pq]} vec(\mathbf{X}^{[p0]}\mathbf{X}^{[0q]})$ can be accelerated by introducing the lossless factorization of the affinity matrix in [17]. The details can be seen in Theorem 3, and the corresponding derivation is given in the appendix.

Theorem 3. *Let* $\mathbf{U} = [\mathbf{X}^{[01]}, \cdots, \mathbf{X}^{[0m]}]$ *be a matrix that contains all solutions, then the objective function of the MGM problem can be written as*

$$J(\mathbf{U}) = \sum_{p,q,p\neq q} vec(\mathbf{D}^{[pq]})^{\mathrm{T}} vec(\mathbf{V}^{[p]^{\mathrm{T}}}\mathbf{X}^{[p0]}\mathbf{X}^{[0q]}\mathbf{V}^{[q]})^{\circ 2}, \tag{14}$$

where

$$\mathbf{D}^{[pq]} = \begin{bmatrix} \mathbf{T}^{[pq]} & -\mathbf{T}^{[pq]}\mathbf{H}^{[q]^{21}} \\ -\mathbf{H}^{[p]^{12}}\mathbf{T}^{[pq]} & \mathbf{H}^{[p]^{12}}\mathbf{T}^{[pq]}\mathbf{H}^{[q]^{21}} + \mathbf{S}^{[pq]} \end{bmatrix},$$

and $^{\circ 2}$ *represents the element-wise power of 2. In this case, the gradient of* $J(\mathbf{U})$ *to* $\mathbf{X}^{[p0]}$ *is*

$$\nabla \mathbf{X}^{[p0]} = 2 \sum_{q,q\neq p} \mathbf{V}^{[p]}(\mathbf{D}^{[pq]} \circ (\mathbf{V}^{[p]^{\mathrm{T}}}\mathbf{X}^{[p0]}\mathbf{X}^{[0q]}\mathbf{V}^{[q]}))\mathbf{V}^{[q]^{\mathrm{T}}}\mathbf{X}^{[q0]},$$

where $\circ$ *represents the element-wise multiplication.*

In the following, we present an approximation algorithm that can efficiently obtain the solutions for the proposed factorized MGM model.

3.3. Optimization of Factorized MGM

It can be found that any graph in $\{\mathcal{G}^{[1]}, \cdots, \mathcal{G}^{[m]}\}$ could be selected as the virtual graph $\mathcal{G}^{[0]}$, and for each different virtual graph, the solution could be different. In order to find the best choice of the virtual graph, we follow the procedure in [23], where the pairwise matching results are required.

In this paper, the famous Reweighted Random Walks for Graph Matching (RRWM) [26] is adopted as the pairwise matching method. The pseudocode of RRWM is given in

Algorithm 1. When two graphs $\mathcal{G}^{[p]}$ and $\mathcal{G}^{[q]}$ are to be matched, the corresponding gradient $\nabla \mathbf{X}^{[pq]}$ in Algorithm 1 has the following expression (the detailed derivations of (15) is given in the appendix)

$$\nabla \mathbf{X}^{[pq]} = 2\mathbf{V}^{[p]}(\mathbf{D}^{[pq]} \circ (\mathbf{V}^{[p]^{\mathrm{T}}} \mathbf{X}^{[pq]} \mathbf{V}^{[q]}))\mathbf{V}^{[q]^{\mathrm{T}}}. \tag{15}$$

Algorithm 1 The workflow of RRWM

Input: The parameters α, β ($\alpha = 0.2, \beta = 30$ by default), and the objective function.

Output: The pair-wise matching result $\mathbf{X}$.

Initialize $\mathbf{X}$ as an all-ones matrix.
repeat
  Calculate the gradient of $\mathbf{X}$, which is $\nabla \mathbf{X}$.
  Reweight: $\mathbf{Y} = \exp(\beta \nabla \mathbf{X} / \max \nabla \mathbf{X})$.
  repeat
    Row normalization: $\mathbf{Y}_{ab} = \mathbf{Y}_{ab} / \sum_b \mathbf{Y}_{ab}$.
    Column normalize: $\mathbf{Y}_{ab} = \mathbf{Y}_{ab} / \sum_a \mathbf{Y}_{ab}$.
  until $\mathbf{Y}$ converges
  Update $\mathbf{X}$ by $\mathbf{X} = \alpha \nabla \mathbf{X} + (1 - \alpha)\mathbf{Y}$.
until $\mathbf{X}$ converges
Discretize $\mathbf{X}$.

After the pairwise matching results $\mathbf{W}$ is obtained, let $\mathbf{U}^{[r]}$ be $[\mathbf{X}^{[r1]}, \mathbf{X}^{[r2]}, \cdots, \mathbf{X}^{[rm]}]$. Then the virtual graph $\mathcal{G}^{[0]}$ is set as $\mathcal{G}^{[r]}$ that has the minimum value of $||\mathbf{W} - \mathbf{U}^{[r]^{\mathrm{T}}} \mathbf{U}^{[r]}||_F$, i.e., the one corresponding to the best low-order approximation, where $||\cdot||_F$ represents the Frobenius norm. And the corresponding matrix $\mathbf{U}^{[r]}$ is selected as the initial solution of $\mathbf{U} = [\mathbf{X}^{[01]}, \cdots, \mathbf{X}^{[0m]}]$ for the next step.

Notice that $\mathbf{X}^{[r1]}, \cdots, \mathbf{X}^{[rm]}$ are all permutation matrices, thus, the computational complexity of calculating $\mathbf{X}^{[pr]}\mathbf{X}^{[rq]}$ can be reduced to $O(n_v)$. Thus, the computational complexity of generating $\mathbf{U}^{[r]^{\mathrm{T}}}\mathbf{U}^{[r]}$ is $O(m^2 n_v)$, and that of determining the best virtual graph is $O(m^3 n_v)$.

Algorithm 2 The workflow of further optimization

Input: The m graphs to be matched, the index of the virtual graph r, and corresponding matching result $\mathbf{U}$.

Output: The pairwise matching result $\mathbf{W}$.

Backup the solution: $\mathbf{U}' = \mathbf{U}$.
repeat
  for p = 1:m and $p \neq r$ **do**
    Use RRWM to update $\mathbf{X}^{[p0]}$ according to the global gradient $\nabla\mathbf{X}^{[p0]}$.
    if $\mathbf{U} \neq \mathbf{U}'$ and $J(\mathbf{U}) < J(\mathbf{U}')$ **then**
      Restore previous result: $\mathbf{U} = \mathbf{U}'$.
    else
      Backup the solution: $\mathbf{U}' = \mathbf{U}$.
    end if
  end for
until $\mathbf{U}$ converges {Global updating.}
repeat
  for p = 1:m and $p \neq r$ **do**
    for q = 1:m and $q \neq p$ **do**
      Use RRWM to update $\mathbf{X}^{[p0]}$ according to the local gradient $\nabla\mathbf{X}_q^{[p0]}$.
      if $\mathbf{U} \neq \mathbf{U}'$ and $J(\mathbf{U}) < J(\mathbf{U}')$ **then**
        Restore previous result: $\mathbf{U} = \mathbf{U}'$.
      else
        Backup the solution: $\mathbf{U}' = \mathbf{U}$.
      end if
    end for
  end for
until $\mathbf{U}$ converges {Local updating.}
Calculate $\mathbf{W}$ by $\mathbf{W} = \mathbf{U}^{\mathrm{T}}\mathbf{U}$.

In conventional iterative MGM methods, to further optimize the solutions, $\mathbf{X}^{[p0]}$ is updated according to the global gradient $\nabla\mathbf{X}^{[p0]}$. However, we find that such a strategy could quickly get stuck in a local maximum due to the integer constraints. Therefore, inspired by the stochastic gradient descent method, we additionally adopt a local updating strategy, which is given in the following.

For a specific pair of p and q, we first obtain a solution of $\mathbf{X}^{[p0]}$ by setting the following formulation as the objective function in Algorithm 1:

$$\begin{aligned}\hat{\mathbf{X}}^{[p0]} &= \arg\max_{\mathbf{X}^{[p0]}} vec(\mathbf{X}^{[p0]}\mathbf{X}^{[0q]})^{\mathrm{T}}\mathbf{K}^{[pq]}vec(\mathbf{X}^{[p0]}\mathbf{X}^{[0q]}) \\ &= \arg\max_{\mathbf{X}^{[p0]}} vec(\mathbf{D}^{[pq]})^{\mathrm{T}}vec(\mathbf{V}^{[p]^{\mathrm{T}}}\mathbf{X}^{[p0]}\mathbf{X}^{[0q]}\mathbf{V}^{[q]})^{\circ 2}.\end{aligned} \tag{16}$$

Notice that the objective function in (16) is actually a part of the global objective function $J(\mathbf{U})$, and the corresponding local gradient $\nabla\mathbf{X}_q^{[p0]}$ can be simply given by

$$\nabla\mathbf{X}_q^{[p0]} = 2\mathbf{V}^{[p]}(\mathbf{D}^{[pq]} \circ (\mathbf{V}^{[p]^{\mathrm{T}}}\mathbf{X}^{[p0]}\mathbf{X}^{[0q]}\mathbf{V}^{[q]}))\mathbf{V}^{[q]^{\mathrm{T}}}\mathbf{X}^{[q0]}.$$

If the overall objective function $J(\mathbf{U})$ increases after updating $\mathbf{X}^{[p0]}$ in $\mathbf{U}$, then the updated $\mathbf{U}$ is saved, otherwise, the previous $\mathbf{U}$ will continue to be used. The algorithm will continuously traverse all possible p and q, until $\mathbf{U}$ converges. The complete pseudocode of our method can be seen in Algorithm 2.

In the proposed method, not only the cycle-consistency problem is avoided, but also the costly affinity matrix. Besides, in each iteration, the most time-consuming part is to calculate the gradient $\nabla\mathbf{X}^{[pq]}$ or $\nabla\mathbf{X}_q^{[p0]}$. Since $\mathbf{V}^{[p]}, \mathbf{V}^{[q]}$ are binary matrices, and $\mathbf{X}^{[p0]}, \mathbf{X}^{[q0]}$ are permutation matrices, the computational complexity of calculating the gradients can be greatly reduced, which is $O((n_v + n_e)^2)$. Therefore, the whole computational complexity of our algorithm is $O(m^3 n_v + km^2(n_v + n_e)^2)$, where k is a variable that relates to the number of iterations.

4. Experiments

In this paper, four experiments are carried out, including one synthetic dataset and three real datasets. Our method, namely Factorized Multi-Graph Matching (FMGM), is compared

with five other MGM methods: Three one-shot methods, including solving the multi-way matching problem by permutation synchronization (MatchSync) [19], near-optimal joint object matching via convex relaxation (MatchLift) [20], and multi-image matching via fast alternating minimization (MatchALS) [21]; Two iterative methods, including joint optimization for consistent MGM (JOMGM) [22], and consistency-driven alternating optimization for MGM (CDMGM) [23]. Notice that the factorized version of CDMGM is of higher computational complexity, but the performance is not significant improved, and sometimes could be worse. Thus, the non-factorized version of CDMGM is used for comparison. In addition, a simple method that directly uses the pairwise matching method to find the correspondences between the first graph and the left graphs is also evaluated as a reference, which is denoted Origin.

In JOMGM, the pairwise matching method Integer Projected Fixed Point (IPFP) [27] is recommended, while in CDMGM and our method FMGM, the pairwise matching method RRWM [26] is used. Therefore, in the Origin method and three one-shot methods, RRWM is selected as the pairwise matching solver for fair comparisons. Besides, the codes of the three one-shot methods are from [28]. The left methods are coded by us and the default parameters recommended in the papers are used. In addition, the Hungarian method [12] is used to discretize the solution for all methods.

Notice that in all four experiments, the nodes of the graphs are associated with 2D locations. Therefore, all graphs are constructed by generating the sparse delaunay triangulation among the nodes. By following the previous works, the similarity between two nodes or edges are given by the Gaussian kernel $\exp(-\|\boldsymbol{f}_i - \boldsymbol{f}_j\|/\sigma)$, where $\boldsymbol{f}_i$ and $\boldsymbol{f}_j$ are the corresponding feature vectors of two nodes or edges, and σ is the normalization factor. The matching accuracy, which is defined as the number of correctly matched inliers divided by the total number of inliers, is used as the evaluation metric.

4.1. Synthetic Dataset

In this experiment, we randomly generate 10 points on a 2D plane for each trail. To test the anti-noise performance of all methods, different zero-mean Gaussian noise is added to

the coordinates of the nodes to generate numerous similar graphs. In addition to noise, the influence of outliers is also evaluated. Furthermore, for each setting, 200 sets of graphs are generated, and the mean accuracy of the matching results is used to evaluate all methods. In the anti-noise experiment, the standard deviation of the Gaussian noise ranges from 0 to 0.2. In the anti-outlier experiment, the number of outliers ranges from 0 to 5. For each pair of graphs, the node-to-node similarity matrix is set to $\mathbf{0}$. The edge-to-edge similarity matrix is generated by using the length of the edges as the feature, and σ is set as $0.1\max(\|\boldsymbol{f}_i - \boldsymbol{f}_j\|)$. Moreover, in both experiments, all methods are implemented to match different number of graphs. The results of the experiments can be seen in Figs. 3 and 4.

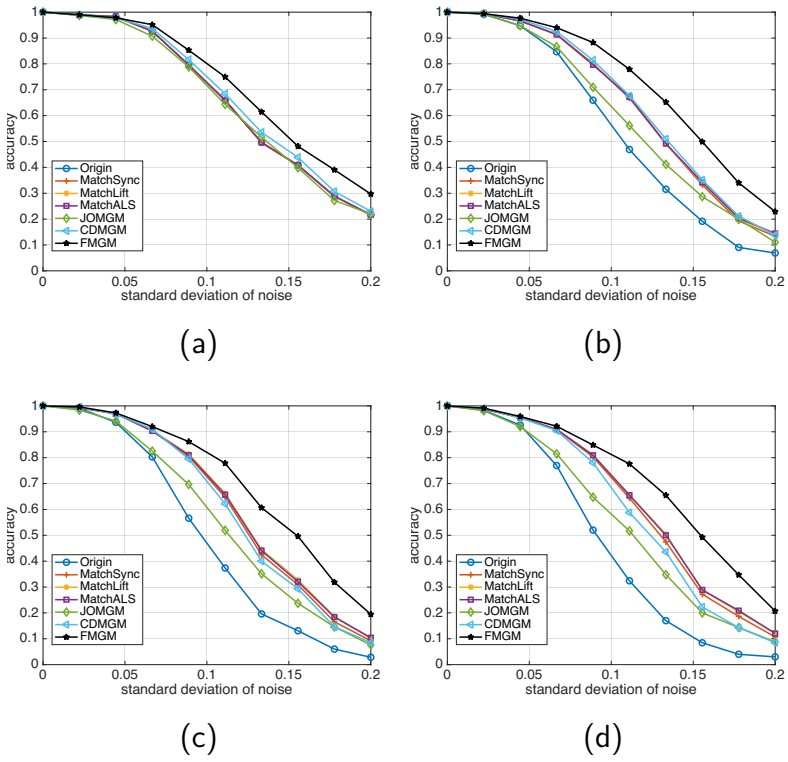


Figure 3: The results of anti-noise experiment (a) 3 graphs (b) 6 graphs (c) 9 graphs (d) 12 graphs

It can be found from Figs. 3 and 4 that as there are more graphs required to be matched, the matching accuracy of all methods generally decrease. This is because a correct matching result requires all pairwise matching results to be correct. Hence, as there are more graphs, achieving a correct matching result becomes more difficult. It can be seen in Figs. 3a and 4a

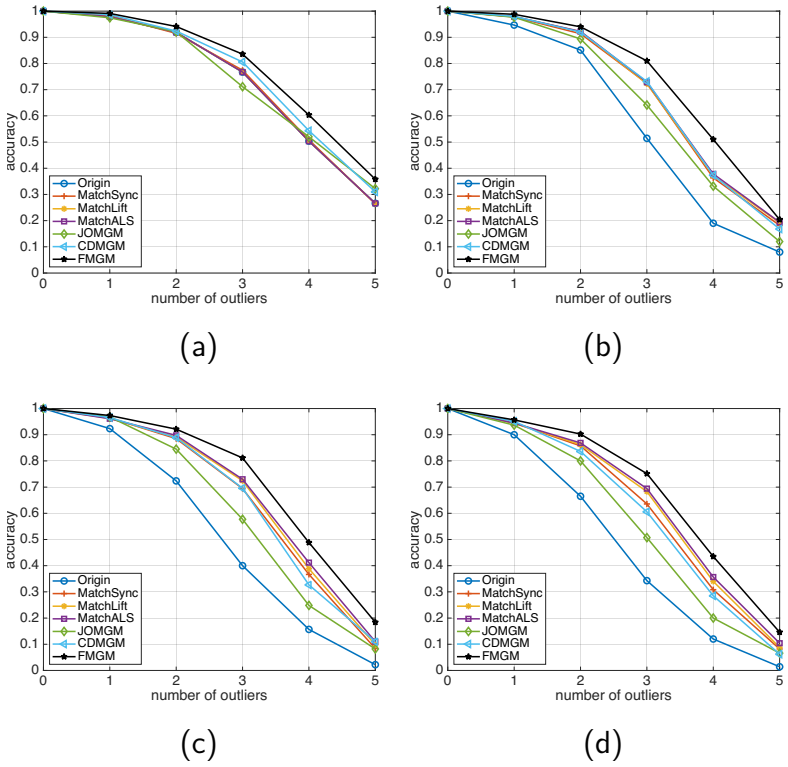


Figure 4: The results of anti-outlier experiment (a) 3 graphs (b) 6 graphs (c) 9 graphs (d) 12 graphs

that, when there are 3 graphs to be matched, the performance of all MGM methods is close to that of Origin. However, as the number of graphs increases, the accuracy of them gradually becomes higher than that of Origin, which indicates the superiority of considering all possible pairwise matching results. The accuracy curves of the three one-shot methods in Figs. 3 and 4 are close to each other in different settings, which is because they share the same input. In general, under varying parameters, our method FMGM achieves the best accuracy over all other methods in both experiments, especially when the number of graphs is large.

4.2. CMU House/Hotel Dataset

The CMU House/Hotel dataset contains a sequence of image of the same object taken from different viewpoints. More specifically, the CMU House dataset contains 111 frames of a toy house with 30 labeled landmark points in each frame, and the CMU Hotel dataset contains 101 frames of a toy hotel with 30 labeled landmark points in each frame. Some

example images of the CMU House/Hotel dataset is shown in Fig. 5, from which, it can be seen that the larger the sequence gap, the more different the angle of the viewpoints.

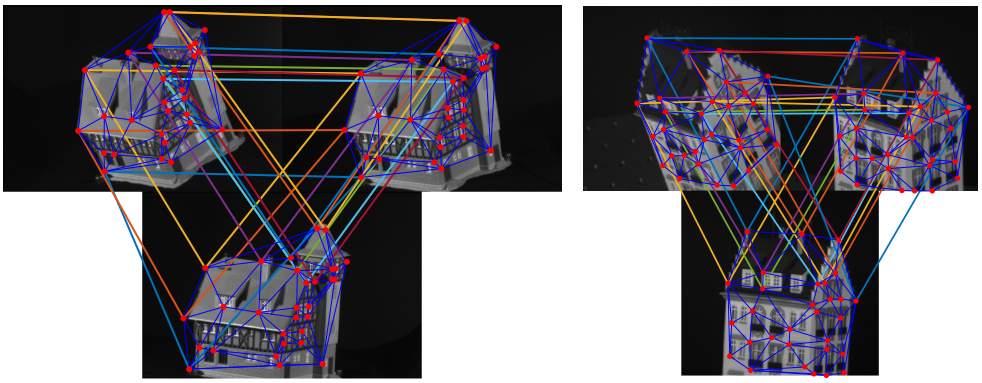


Figure 5: Some frames of the CMU House/Hotel Dataset (for better visualization, only the correspondences of the first 10 nodes are plotted) (a) the 1, 41, 81 frames of the house dataset (b) the 1, 41, 81 frames of the hotel dataset

Similar to the synthetic experiment, we only rely on the length of the edges, and the same σ is used. For each trail, we randomly select m graphs from the CMU house or hotel dataset, and 10 labeled landmark points are also randomly selected for each graph. The number of graphs m ranges from 4 to 12, and for each m, 2000 implementations are carried out to obtain a statistical result, which can be seen in Fig. 6. Besides, we also evaluate the performance of all methods when there exist outliers, the results are shown in Fig. 7, where 4 graphs with sequence gap ranges from 1 to 20 are required to be matched.

It can be seen from Fig. 6 that, since the inputs of the three one-shot methods are the same, they are similar in performance. Because using a different pairwise solver, the performance of JOMGM is worse than other MGM methods. Meanwhile, it can be found from Fig. 6 that when there is no noise or outlier, the matching performance of all MGM methods is sensitive to the number of graphs. The iterative methods CDMGM and FMGM continue increasing the objective function during optimization, thus, they outperform all other methods in accuracy. Moreover, because of the effective optimization strategy used in FMGM, our method achieves a higher accuracy over all other methods. In Fig. 7, due to

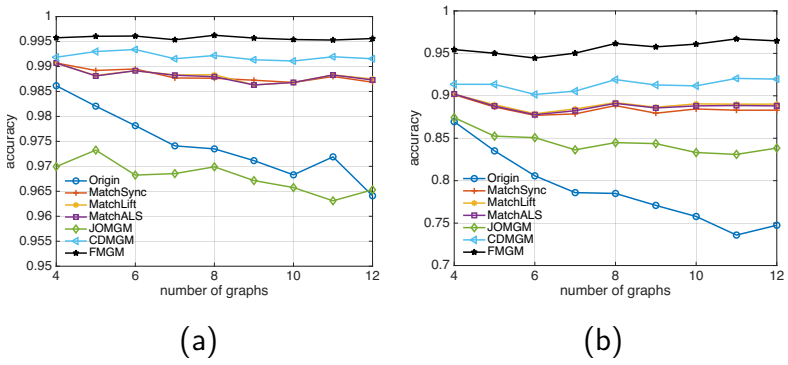


Figure 6: The results of the CMU house/hotel dataset (no outlier) (a) CMU house (b) CMU hotel

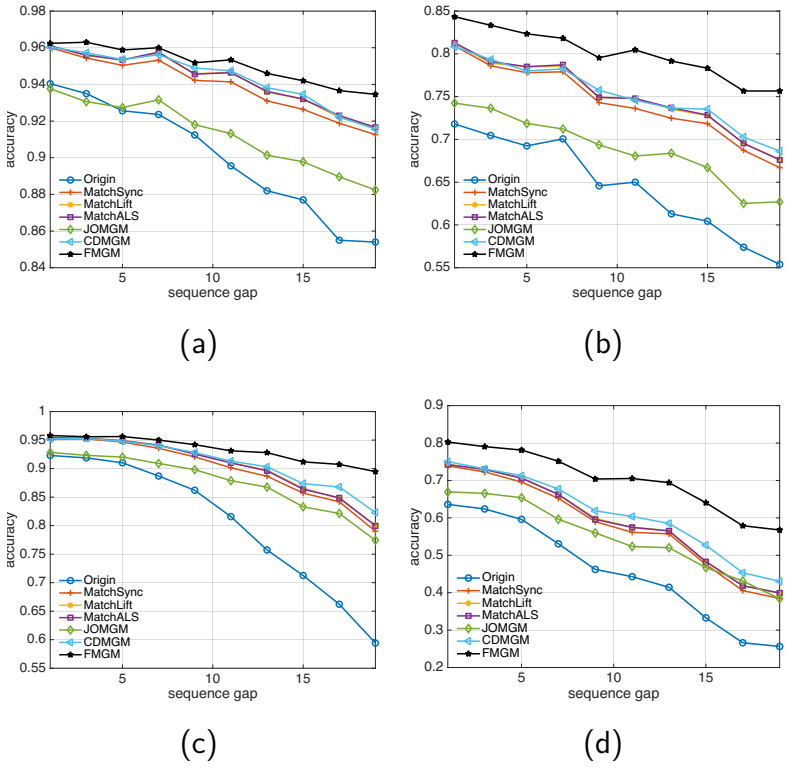


Figure 7: The results of the CMU house/hotel dataset (with outlier) (a) CMU house with 2 outliers (b) CMU house with 5 outliers (c) CMU hotel with 2 outliers (d) CMU hotel with 5 outliers

the existence of outliers, the performance of all methods is degraded, especially when the sequence gap is large. However, the matching accuracy of FMGM is still higher than other methods in all settings. This further demonstrates the superiority of our method.

4.3. Willow Object Dataset

In this section, the Willow Object dataset [29], which contains a total of 256 images from 5 categories, is used to evaluate the performance of all methods. There are at least 40 images in each category of the Willow Object dataset, and each image is annotated with the same 10 distinctive category specific keypoints. Some images of the Willow Object dataset are shown in Fig. 8.

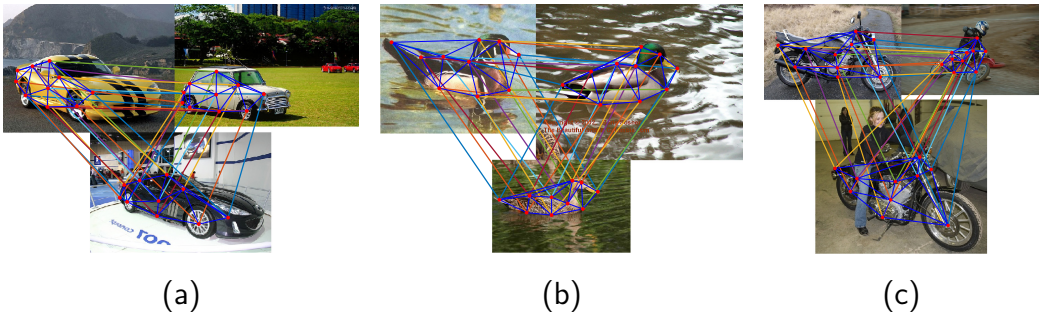


Figure 8: Some examples of the Willow dataset (a) three images of the car category (b) three images of the duck category (c) three images of the motorbike category

We first carry out an experiment on a subset of the Willow Object dataset, which is denoted by PF-Willow dataset in [30]. The PF-Willow dataset splits the Willow Object dataset into 10 subclasses, including car(G), (S), (M), duck(S), motorbike(G), (S), (M), and winebottle(woC), (wC), (M). Notice that (G), (S), (M) represent general, side and mixed viewpoints, respectively, and (C) denotes the background clutter. In each subclass, there are 10 images, and all methods are applied to match them all at a time.

Due to the fact that the object appearance in each class varies greatly, in addition to the edges, we also adopt the node features extracted from a pretrained convolutional neural network as the literature [28] did. The pretrained convolutional neural network [31] extracts a 640×1 feature vector for every node, which is used to generate the node-to-node similarity matrix of each pair of graphs. As for the edge, a vector $[l, \theta]^{\mathrm{T}}$ is taken as its feature, where l is the normalized length of the edge and θ is the angle between the edge and the horizontal line. Besides, in the node-to-node similarity matrix and the edge-to-edge similarity matrix, σ is set as $0.25 \max(\|\boldsymbol{f}_i - \boldsymbol{f}_j\|)$ to map the distance to similarity. The matching result is listed

Table 1: The matching accuracy of the PF-Willow dataset

	Origin	MatchSync	MatchLift	MatchALS	JOMGM	CDMGM	FMGM
car(G)	0.6	0.8	0.8	0.8	1.0	1.0	1.0
car(S)	0.2	0.8	0.8	0.8	1.0	1.0	1.0
car(M)	0.1	0.5	0.7	0.7	1.0	1.0	1.0
duck(S)	1.0	1.0	1.0	1.0	1.0	1.0	1.0
motorbike(G)	0.8	1.0	1.0	1.0	1.0	1.0	1.0
motorbike(S)	1.0	1.0	1.0	1.0	0.8	1.0	1.0
motorbike(M)	0.8	1.0	1.0	1.0	0.8	1.0	1.0
winebottle(M)	0.6	0.6	0.6	0.6	0.6	0.6	1.0
winebottle(wC)	0.2	0.8	0.8	0.8	0.2	0.6	1.0
winebottle(woC)	0.6	0.8	0.8	0.8	0.8	0.8	0.8
mean	0.59	0.83	0.85	0.85	0.82	0.90	0.98

in Table 1, from which, it can be found that FMGM obtains the highest matching accuracy for all subclasses, and the mean accuracy is also obviously higher than other methods.

In addition, we also compare the performance of all methods on the whole Willow Object dataset. In this part, we randomly select m images from the same category, and then match them all at a time. The number of images m ranges from 3 to 15, and for each m, we repeat the experiment for 500 times to obtain a statistical result. The average matching accuracy of all methods is shown in Fig. 9.

From Fig. 9c, it can be found that all methods are able to achieve a perfect matching result for the face category. In Figs. 9a and 9b, it can be seen that, for the car and duck categories, the matching accuracy of all methods generally decreases with the increase of number of graphs. However, for the motorbike and winebottle categories, CDMGM and FMGM could obtain a better matching result when the number of graphs increases. This is because the appearances of the motorbike and winebottle categories are more stable than those of the car and duck categories. In this case, when more images are required to be matched, more information can be utilized to improve the matching accuracy. From the mean accuracy curves shown in Fig. 9f, it can be found that FMGM always obtains the best matching results under different numbers of graphs. This further demonstrates the

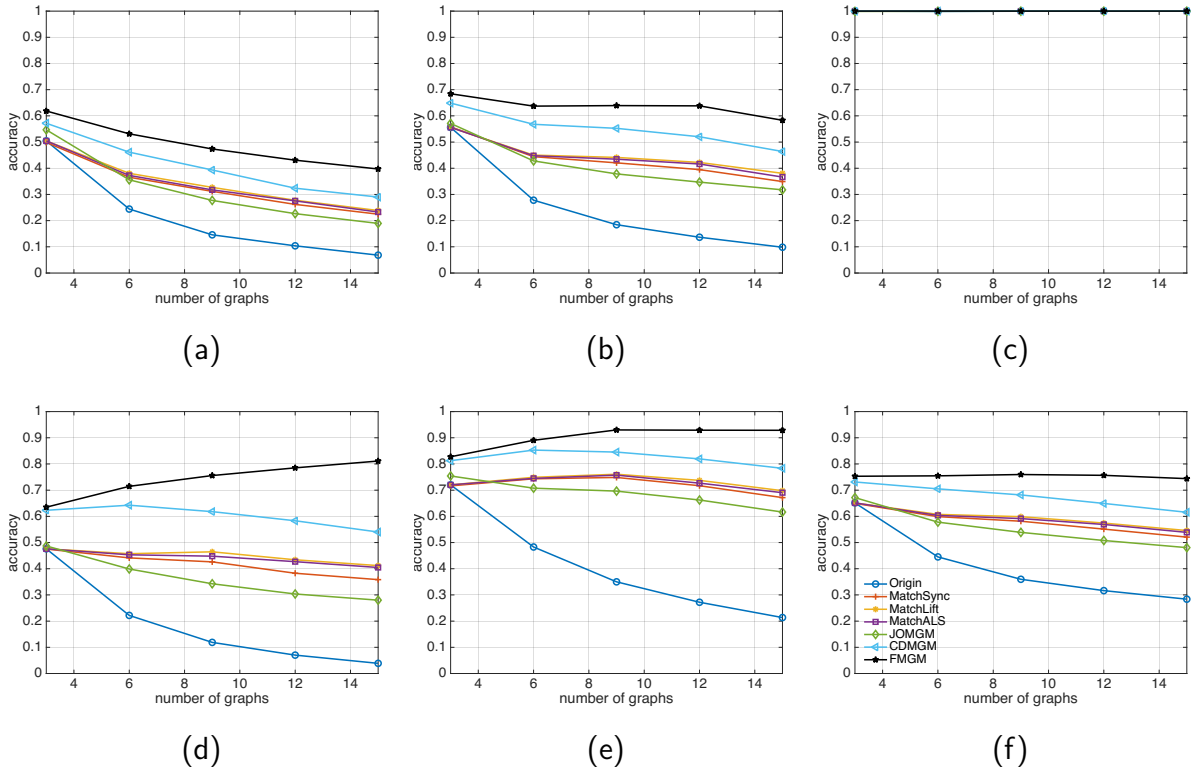


Figure 9: The matching accuracy of all methods on the Willow Object dataset (a) car (b) duck (c) face (d) motorbike (e) winebottle (f) mean accuracy of all objects

effectiveness of the proposed approximation algorithm in FMGM.

4.4. Pascal Visual Object Classes Dataset

In this section, the famous Pascal Visual Object Classes (VOC) dataset [32] is used to evaluate the performance of the proposed method. The Pascal VOC dataset contains 10,103 images, which are divided into 20 object categories, including plane, bicycle, bird, boat, bottle, bus, car, cat, chair, cow, table, dog, horse, motorbike, person, plant, sheep, sofa, train, and tvmonitor. For the objects in the images, the number of annotated keypoints ranges from 2 to 20. Therefore, for convenience, we only select the images with more than 4 annotated keypoints for each category. In addition, for each trail, we randomly select four annotated keypoints to match. The experimental setup is the same as that of the Willow Object dataset experiment. The average matching results of 500 runs are listed in Tables 2 and 3, where the number of graphs required to be matched are 3 and 6, respectively.

Table 2: The matching accuracy of the Pascal VOC dataset (3 graphs)

	Origin	MatchSync	MatchLift	MatchALS	JOMGM	CDMGM	FMGM
plane	0.2655	0.258	0.2655	0.2655	**0.3095**	0.291	0.304
bicycle	0.156	0.1495	0.156	0.156	**0.172**	0.1635	0.1675
bird	0.152	0.1555	0.152	0.152	**0.192**	0.1665	0.1725
boat	0.184	0.184	0.184	0.184	0.188	0.1845	**0.2085**
bottle	0.1235	0.11	0.1235	0.1235	0.1225	**0.124**	0.1235
bus	0.59	0.563	0.59	0.59	0.6515	0.671	**0.6785**
car	0.308	0.307	0.308	0.308	0.3365	0.318	**0.337**
cat	0.732	0.727	0.732	0.732	0.7855	0.7605	**0.788**
chair	0.3205	0.3115	0.3205	0.3205	0.373	0.381	**0.4005**
cow	0.782	0.78	0.782	0.782	0.855	0.828	**0.8565**
table	0.241	0.2235	0.241	0.241	0.287	0.279	**0.3105**
dog	0.7275	0.7335	0.7275	0.7275	0.7875	0.7535	**0.791**
horse	0.7725	0.7765	0.7725	0.7725	0.8185	0.812	**0.8325**
motorbike	0.2145	0.2075	0.2145	0.2145	0.238	0.2365	**0.248**
person	0.415	0.3935	0.415	0.415	0.4905	0.5155	**0.5335**
plant	0.4955	0.5005	0.4955	0.4955	**0.564**	0.5215	0.549
sheep	0.6585	0.6725	0.6585	0.6585	0.7475	0.7115	**0.7495**
sofa	0.453	0.445	0.453	0.453	0.5465	0.5675	**0.5775**
train	0.61	0.5707	0.61	0.61	**0.656**	0.636	**0.656**
tvmonitor	0.4685	0.461	0.4685	0.4685	0.547	0.5535	**0.5645**
mean	0.4335	0.4265	0.4335	0.4335	0.4834	0.4737	**0.4924**

It can be found that the matching accuracy of the Origin method in Table 3 is much lower than that in Table 2. The reason is that the Origin method does not consider all pairwise matching results. Hence, when there are more graphs, the error would accumulate. Though the performance of the MGM methods is better than that of Origin, it can be seen that the improvement of the MGM methods is limited for those classes with matching accuracy lower than 0.5. In addition, as can be seen in Tables 2 and 3, FMGM achieves the highest matching accuracy for most classes, and is more efficient than other methods when the number of graphs is 6. This indicates the superiority of the proposed method in

Table 3: The matching accuracy of the Pascal VOC dataset (6 graphs)

	Origin	MatchSync	MatchLift	MatchALS	JOMGM	CDMGM	FMGM
plane	0.0665	0.1115	0.122	0.1255	0.141	0.127	**0.155**
bicycle	0.016	0.025	0.026	0.026	0.0315	0.029	**0.034**
bird	0.017	0.021	0.0225	0.022	0.0375	0.0335	**0.043**
boat	0.013	0.021	0.0225	0.021	**0.0265**	0.0265	0.0215
bottle	0.005	0.0025	**0.007**	0.0055	0.0035	0.0035	0.004
bus	0.3065	0.4815	0.4775	0.472	0.5315	0.6	**0.625**
car	0.1015	0.1895	0.188	0.199	0.2005	0.1945	**0.2175**
cat	0.5735	0.691	0.6895	0.6885	0.6905	**0.703**	0.7015
chair	0.106	0.2015	0.217	0.2175	0.254	0.2675	**0.341**
cow	0.6025	0.8435	0.8545	0.845	0.844	0.8725	**0.879**
table	0.036	0.0685	0.0705	0.072	0.095	0.1025	**0.1675**
dog	0.493	0.704	0.711	0.7075	0.731	0.734	**0.76**
horse	0.518	0.7065	0.7145	0.718	0.7495	0.76	**0.796**
motorbike	0.053	0.085	0.0935	0.095	0.1145	0.1015	**0.1225**
person	0.1615	0.3285	0.3395	0.3345	0.3905	0.477	**0.565**
plant	0.2665	0.3925	0.4045	0.3985	**0.4175**	0.4005	0.416
sheep	0.4385	0.6175	0.6245	0.6215	0.6855	0.6555	**0.695**
sofa	0.164	0.3415	0.3435	0.336	0.399	0.488	**0.5665**
train	0.4033	0.4987	0.496	0.4973	0.5193	**0.5213**	0.516
tvmonitor	0.198	0.309	0.31	0.3095	0.3425	0.379	**0.4505**
mean	0.227	0.332	0.3367	0.3356	0.3602	0.3738	**0.4038**

matching multiple graphs.

4.5. Computational Complexity Analysis

Another major advantage of FMGM is its low computational complexity, which is demonstrated in Fig. 10 and Table 4. In Fig. 10a, the number of nodes is set to 20, and the number of graphs ranges from 4 to 20. And in Fig. 10b, the number of graphs is set to 3, and the number of nodes ranges from 10 to 50. All methods are implemented in MATLAB 2022a on an i5-1155G7 CPU and 16GB RAM computer for 20 times.

In MatchSync, after obtaining the pairwise result $\mathbf{W}$, the algorithm only needs to cal-

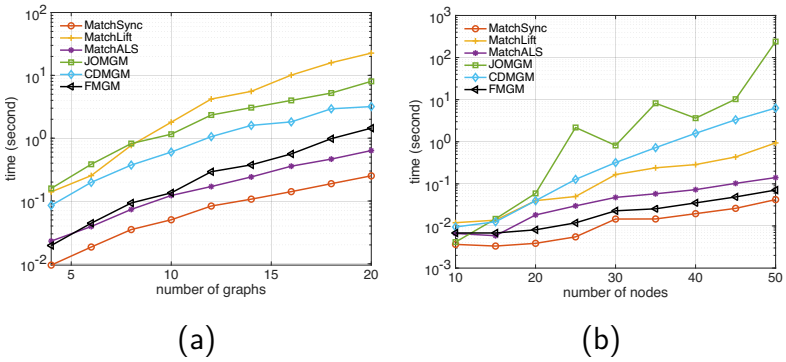


Figure 10: The computational time of the MGM methods (a) time vs number of graphs (b) time vs number of nodes

Table 4: The complexity comparison of the MGM methods

method	time complexity
MatchSync [23]	$O(m^2n_v^3) + \tau_{pair}$
MatchLift [21]	$O(km^3n_v^3) + \tau_{pair}$
MatchALS [21]	$O(km^2n_v^3) + \tau_{pair}$
JOMGM [22]	$O(km^2n_v^3 + km(n_v^2 + n_e^2))$
CDMGM [23]	$O(m^3n_v + km^2n_v^3 + km(n_v^2 + n_e^2)) + \tau_{pair}$
Tensor-based [25]	$O(km^2n_v^m + kmn_v^6)$
FMGM	$O(m^3n_v + km^2(n_v + n_e)^2) + \tau_{pair}$

τ_{pair} is the computational complexity of calculating the pairwise matching results, which is $O(m^2(n_v^2 + n_e^2))$ when using RRWM

k represents the number of iterations, which varies for different methods

culate the singular value decomposition of the matrix $\mathbf{W}$. Therefore, the computational complexity of MatchSync is the lowest comparing with other MGM methods. Because MatchLift is based on the costly semidefinite programming, its computational time is close to that of the two iterative methods JOMGM and CDMGM as can be seen in Fig. 10. Though the computational complexity of JOMGM is lower than that of CDMGM (Table 4), but its computational time is generally higher than that of CDMGM (Fig. 10). After some investigation, we find the reason is that JOMGM usually takes much more iterations to obtain a converged solution.

From Table 4, it can be found that comparing with the tensor-based MGM methods,

whose computational complexity grows geometrically with the size of graphs, FMGM is much more efficient. Due to the procedure of determining the best virtual graph, whose computational complexity is $O(m^3 n_v)$, the computational time of FMGM increases faster than that of MatchALS (Fig. 10a) and gradually becomes close to that of CDMGM. Meanwhile, in Fig. 10b, it can be seen that, when matching graphs with numerous nodes, FMGM costs less time than MatchALS does. This is because of the proposed efficient gradient calculation formula, which is insensitive to the number of nodes. In summary, the computational complexity of FMGM is close to that of the one-shot methods, and is lower than that of the iterative methods. Therefore, considering the superiority of FMGM in matching accuracy over other methods, it is recommended in most cases, especially when there are plenty of nodes in the graphs.

5. Conclusion

In this paper, we demonstrate that the pairwise-based MGM is a special case of the tensor-based MGM. Based on this finding, a new MGM method is proposed, where not only the complexity is reduced, but also the cycle-consistency problem is avoided. Besides, the factorization of the affinity matrix is introduced into our work to further reduce the complexity. In order to alleviate the local maximum problem in MGM, we also present an approximation algorithm based on the stochastic gradient descent method. Both the synthetic and real data experiments demonstrate the efficiency and accuracy of our method. However, due to the updating strategy used in the approximation algorithm, FMGM could cost more time to find a better solution when the number of graphs is large. Besides, when there exists a complex relationship between the graphs to be matched, the performance of FMGM may be unsatisfactory. How to overcome the mentioned two limitations of FMGM will be our future research topics.

Appendix A. Detail Proofs of the Theorems

Theorem 1. *Let $\mathbf{X}^{[0i]}$ be the indicator matrix between the virtual graph $\mathcal{G}^{[0]}$ and the i^{th} graph $\mathcal{G}^{[i]}$, then the indicator tensor $\mathcal{X}$ for all m graphs can be factorized as follows*

$$\mathcal{X} = \mathcal{I} \times_1 \mathbf{X}^{[01]} \times_2 \mathbf{X}^{[02]} \cdots \times_m \mathbf{X}^{[0m]}, \tag{10}$$

where $\mathcal{I}$ is a diagonal tensor with the size of $n_v^{[0]} \times n_v^{[0]} \times \cdots \times n_v^{[0]}$, and the diagonal elements are all ones.

Proof. In the matrix $\mathbf{X}^{[0n]}$, the (i_0, i_n) element is 1, if the i_0^{th} node in $\mathcal{G}^{[0]}$ and the i_n^{th} node in $\mathcal{G}^{[n]}$ is matched. Similarly, if i_1^{th} node in $\mathcal{G}^{[1]}$, ..., and i_m^{th} node in $\mathcal{G}^{[m]}$ are matched, $\mathcal{X}_{i_1:i_m}$ equals 1.

Assume that i_0^{th} node in $\mathcal{G}^{[0]}$, i_1^{th} node in $\mathcal{G}^{[1]}$, ..., and i_m^{th} node in $\mathcal{G}^{[m]}$ are matched. Let $\mathcal{X}' = \mathcal{I} \times_1 \mathbf{X}^{[01]} \times_2 \mathbf{X}^{[02]} \cdots \times_m \mathbf{X}^{[0m]}$, then according to the definition of the k-mode product, it can be calculated by

$$\mathcal{X}'_{j_1:j_m} = \sum_{l_1:l_m} \mathcal{I}_{l_1:l_m} \Pi_{k=1}^{m} \mathbf{X}^{[0k]}_{l_k j_k}. \tag{A.1}$$

Notice that only when $l_1 = \cdots = l_m$, $\mathcal{I}_{l_1:l_m}$ equals 1,and $\mathbf{X}^{[0k]}_{l_k j_k}$ equals 1, only when $l_k = i_0, j_k = i_k$. Thus, $\mathcal{I}_{l_1:l_m} \Pi_{k=1}^{m} \mathbf{X}^{[0k]}_{l_k j_k}$ equals 1, only when $i_0 = l_1 = \cdots = l_m$, and $j_1 = i_1, \cdots, j_m = i_m$. This indicates that $\mathcal{X}'_{j_1:j_m}$ equals 1, only when $j_1 = i_1, \cdots, j_m = i_m$, i.e., $\mathcal{X} = \mathcal{X}'$. □

Theorem 2. *When the affinity matrix $\mathbf{K}$ in the tensor-based MGM is specially constructed as follows*

$$\mathbf{K}_{i_1:i_m, j_1:j_m} = \sum_{p,q,p\neq q} \mathbf{K}^{[pq]}_{i_p i_q, j_p j_q},$$

where $\mathbf{K}^{[pq]}$ is the affinity matrix between $\mathcal{G}^{[p]}$ and $\mathcal{G}^{[q]}$. Then, the objective function in the tensor-based MGM can be written as

$$vec(\mathcal{X})^{\mathrm{T}} \mathbf{K} vec(\mathcal{X}) = \sum_{p,q,p\neq q} vec(\mathbf{X}^{[p0]} \mathbf{X}^{[0q]})^{\mathrm{T}} \mathbf{K}^{[pq]} vec(\mathbf{X}^{[p0]} \mathbf{X}^{[0q]}). \tag{11}$$

Proof. Let $\hat{\mathbf{K}}^{[pq]}$ be a matrix, which has the following expression

$$\hat{\mathbf{K}}^{[pq]}_{i_1:i_m,j_1:j_m} = \mathbf{K}^{[pq]}_{i_p i_q, j_p j_q}. \tag{A.2}$$

Then the affinity matrix $\mathbf{K}$ can be written as

$$\mathbf{K} = \sum_{p,q,p\neq q} \hat{\mathbf{K}}^{[pq]}. \tag{A.3}$$

Notice that the matrix $\hat{\mathbf{K}}^{[pq]}$ can be written as follows

$$\begin{aligned} &\hat{\mathbf{K}}^{[pq]}_{i_1:i_m,j_1:j_m} \\ =&\sum_{k_p k_q, l_p l_q} \mathbf{K}^{[pq]}_{k_p k_q, l_p l_q} (\mathbf{1}^{[m]}_{i_m} \cdots \mathbf{I}^{[q]}_{k_q i_q} \cdots \mathbf{I}^{[p]}_{k_p i_p} \cdots \mathbf{1}^{[1]}_{i_1})(\mathbf{1}^{[m]}_{j_m} \cdots \mathbf{I}^{[q]}_{l_q j_q} \cdots \mathbf{I}^{[p]}_{l_p j_p} \cdots \mathbf{1}^{[1]}_{j_1}), \end{aligned} \tag{A.4}$$

where $\mathbf{1}^{[i]}$ represents an all-ones $n_v^{[i]} \times 1$ vector, and $\mathbf{I}^{[p]}$ represents an identity matrix with the size of $n_v^{[p]} \times n_v^{[p]}$. Because $(\mathbf{1}^{[m]}_{i_m} \cdots \mathbf{I}^{[q]}_{k_q i_q} \cdots \mathbf{I}^{[p]}_{k_p i_p} \cdots \mathbf{1}^{[1]}_{i_1})$ equals the element with the index of $(i_1 : i_m, k_p k_q)$ in $(\mathbf{1}^{[m]} \cdots \otimes \mathbf{I}^{[q]} \cdots \otimes \mathbf{I}^{[p]} \cdots \otimes \mathbf{1}^{[1]})$, $\hat{\mathbf{K}}^{[pq]}$ can be further written as

$$\begin{aligned} \hat{\mathbf{K}}^{[pq]}_{i_1:i_m,j_1:j_m} =& \sum_{k_p k_q, l_p l_q} \mathbf{K}^{[pq]}_{k_p k_q, l_p l_q} (\mathbf{1}^{[m]} \cdots \otimes \mathbf{I}^{[q]} \cdots \otimes \mathbf{I}^{[p]} \cdots \otimes \mathbf{1}^{[1]})_{i_1:i_m, k_p k_q} \\ &(\mathbf{1}^{[m]} \cdots \otimes \mathbf{I}^{[q]} \cdots \otimes \mathbf{I}^{[p]} \cdots \otimes \mathbf{1}^{[1]})_{j_1:j_m, l_p l_q} \\ =&(\mathbf{1}^{[m]} \cdots \otimes \mathbf{I}^{[q]} \cdots \otimes \mathbf{I}^{[p]} \cdots \otimes \mathbf{1}^{[1]}) \mathbf{K}^{[pq]} (\mathbf{1}^{[m]} \cdots \otimes \mathbf{I}^{[q]} \cdots \otimes \mathbf{I}^{[p]} \cdots \otimes \mathbf{1}^{[1]})^{\mathrm{T}}. \end{aligned} \tag{A.5}$$

Notice that

$$\begin{aligned} &(\mathbf{1}^{[m]} \cdots \otimes \mathbf{I}^{[q]} \cdots \otimes \mathbf{I}^{[p]} \cdots \otimes \mathbf{1}^{[1]})^{\mathrm{T}} vec(\mathcal{X}) \\ =&vec(\mathcal{X} \times_1 \mathbf{1}^{[1]} \cdots \times_p \mathbf{I}^{[p]} \cdots \times_q \mathbf{I}^{[q]} \cdots \times_m \mathbf{1}^{[m]}) \\ =&vec(\mathcal{I} \times_1 \mathbf{X}^{[01]} \mathbf{1}^{[1]} \cdots \times_p \mathbf{X}^{[0p]} \mathbf{I}^{[p]} \cdots \times_q \mathbf{X}^{[0q]} \mathbf{I}^{[q]} \cdots \times_m \mathbf{X}^{[0m]} \mathbf{1}^{[m]}) \\ =&vec(\mathcal{I} \times_1 \mathbf{1}^{[0]} \cdots \times_p \mathbf{X}^{[0p]} \cdots \times_q \mathbf{X}^{[0q]} \cdots \times_m \mathbf{1}^{[0]}) \\ =&vec(\mathbf{X}^{[p0]} \mathbf{X}^{[0q]}), \end{aligned} \tag{A.6}$$

therefore, we have

$$\begin{aligned} &vec(\mathcal{X})^{\mathrm{T}} \mathbf{K} vec(\mathcal{X}) \\ =& \sum_{p,q,p\neq q} vec(\mathcal{X})^{\mathrm{T}} \hat{\mathbf{K}}^{[pq]} vec(\mathcal{X}) \\ =& \sum_{p,q,p\neq q} vec(\mathbf{X}^{[p0]} \mathbf{X}^{[0q]})^{\mathrm{T}} \mathbf{K}^{[pq]} vec(\mathbf{X}^{[p0]} \mathbf{X}^{[0q]}). \end{aligned} \tag{A.7}$$

□

Theorem 3. *Let* $\mathbf{U} = [\mathbf{X}^{[01]}, \cdots, \mathbf{X}^{[0m]}]$ *be a matrix that contains all solutions, then the objective function of the MGM problem can be written as*

$$J(\mathbf{U}) = \sum_{p,q,p\neq q} vec(\mathbf{D}^{[pq]})^{\mathrm{T}} vec(\mathbf{V}^{[p]^{\mathrm{T}}} \mathbf{X}^{[p0]} \mathbf{X}^{[0q]} \mathbf{V}^{[q]})^{\circ 2}, \tag{14}$$

where

$$\mathbf{D}^{[pq]} = \begin{bmatrix} \mathbf{T}^{[pq]} & -\mathbf{T}^{[pq]} \mathbf{H}^{[q]^{21}} \\ -\mathbf{H}^{[p]^{12}} \mathbf{T}^{[pq]} & \mathbf{H}^{[p]^{12}} \mathbf{T}^{[pq]} \mathbf{H}^{[q]^{21}} + \mathbf{S}^{[pq]} \end{bmatrix},$$

and $^{\circ 2}$ *represents the element-wise power of 2. In this case, the gradient of* $J(\mathbf{U})$ *to* $\mathbf{X}^{[p0]}$ *is*

$$\nabla \mathbf{X}^{[p0]} = 2 \sum_{q,q\neq p} \mathbf{V}^{[p]} (\mathbf{D}^{[pq]} \circ (\mathbf{V}^{[p]^{\mathrm{T}}} \mathbf{X}^{[p0]} \mathbf{X}^{[0q]} \mathbf{V}^{[q]})) \mathbf{V}^{[q]^{\mathrm{T}}} \mathbf{X}^{[q0]},$$

where $\circ$ *represents the element-wise multiplication.*

Proof. According to the factorization of the second-order affinity matrix, it can be indicated that

$$\mathbf{K}^{[pq]} = (\mathbf{V}^{[q]} \otimes \mathbf{V}^{[p]}) diag(\mathbf{D}^{[pq]}) (\mathbf{V}^{[q]} \otimes \mathbf{V}^{[p]})^{\mathrm{T}}, \tag{A.8}$$

where

$$\mathbf{D}^{[pq]} = \begin{bmatrix} \mathbf{T}^{[pq]} & -\mathbf{T}^{[pq]} \mathbf{H}^{[q]^{21}} \\ -\mathbf{H}^{[p]^{12}} \mathbf{T}^{[pq]} & \mathbf{H}^{[p]^{12}} \mathbf{T}^{[pq]} \mathbf{H}^{[q]^{21}} + \mathbf{S}^{[pq]} \end{bmatrix}.$$

Therefore, the factorization of the objective function is

$$\begin{aligned} & vec(\mathbf{X})^{\mathrm{T}} \mathbf{K} vec(\mathbf{X}) \\ =& \sum_{p\neq q} vec(\mathbf{X}^{[p0]} \mathbf{X}^{[0q]})^{\mathrm{T}} \mathbf{K}^{[pq]} vec(\mathbf{X}^{[p0]} \mathbf{X}^{[0q]}) \\ =& \sum_{p\neq q} vec(\mathbf{X}^{[p0]} \mathbf{X}^{[0q]})^{\mathrm{T}} (\mathbf{V}^{[q]} \otimes \mathbf{V}^{[p]}) diag(\mathbf{D}^{[pq]}) (\mathbf{V}^{[q]} \otimes \mathbf{V}^{[p]})^{\mathrm{T}} vec(\mathbf{X}^{[p0]} \mathbf{X}^{[0q]}) \\ =& \sum_{p\neq q} vec(\mathbf{V}^{[q]^{\mathrm{T}}} \mathbf{X}^{[p0]} \mathbf{X}^{[0q]} \mathbf{V}^{[q]})^{\mathrm{T}} diag(\mathbf{D}^{[pq]}) vec(\mathbf{V}^{[q]^{\mathrm{T}}} \mathbf{X}^{[p0]} \mathbf{X}^{[0q]} \mathbf{V}^{[q]}) \\ =& \sum_{p,q,p\neq q} vec(\mathbf{D}^{[pq]})^{\mathrm{T}} vec(\mathbf{V}^{[q]^{\mathrm{T}}} \mathbf{X}^{[p0]} \mathbf{X}^{[0q]} \mathbf{V}^{[q]})^{\circ 2}. \end{aligned} \tag{A.9}$$

In addition, the gradient is

$$\begin{aligned}
\nabla \mathbf{X}^{[p0]} &= 2 \sum_{q,q\neq p} mat(\mathbf{K}^{[pq]} vec(\mathbf{X}^{[p0]}\mathbf{X}^{[0q]}))\mathbf{X}^{[q0]} \\
&=2 \sum_{q,q\neq p} mat((\mathbf{V}^{[q]} \otimes \mathbf{V}^{[p]})^{\mathrm{T}} diag(\mathbf{D}^{[pq]}) vec(\mathbf{V}^{[q]^{\mathrm{T}}}\mathbf{X}^{[p0]}\mathbf{X}^{[0q]}\mathbf{V}^{[q]}))\mathbf{X}^{[q0]} \\
&=2 \sum_{q,q\neq p} mat((\mathbf{V}^{[q]} \otimes \mathbf{V}^{[p]})^{\mathrm{T}} vec(\mathbf{D}^{[pq]} \circ (\mathbf{V}^{[p]^{\mathrm{T}}}\mathbf{X}^{[p0]}\mathbf{X}^{[0q]}\mathbf{V}^{[q]}))\mathbf{X}^{[q0]} \\
&=2 \sum_{q,q\neq p} mat(vec(\mathbf{V}^{[p]}(\mathbf{D}^{[pq]} \circ (\mathbf{V}^{[p]^{\mathrm{T}}}\mathbf{X}^{[p0]}\mathbf{X}^{[0q]}\mathbf{V}^{[q]}))\mathbf{V}^{[q]^{\mathrm{T}}}))\mathbf{X}^{[q0]} \\
&=2 \sum_{q,q\neq p} \mathbf{V}^{[p]}(\mathbf{D}^{[pq]} \circ (\mathbf{V}^{[p]^{\mathrm{T}}}\mathbf{X}^{[p0]}\mathbf{X}^{[0q]}\mathbf{V}^{[q]}))\mathbf{V}^{[q]^{\mathrm{T}}}\mathbf{X}^{[q0]}.
\end{aligned} \tag{A.10}$$

□

Appendix B. Derivations of the Gradients

The derivation of the gradient $\nabla \mathbf{X}^{[p0]}$: The objective function is

$$\begin{aligned}
&\sum_{p,q,p\neq q} vec(\mathbf{X}^{[p0]}\mathbf{X}^{[0q]})^{\mathrm{T}}\mathbf{K}^{[pq]} vec(\mathbf{X}^{[p0]}\mathbf{X}^{[0q]}) \\
=&\sum_{p} vec(\mathbf{X}^{[p0]})^{\mathrm{T}}(\sum_{q,q\neq p} (\mathbf{X}^{[q0]} \otimes \mathbf{I})^{\mathrm{T}}\mathbf{K}^{[pq]}(\mathbf{X}^{[q0]} \otimes \mathbf{I}))vec(\mathbf{X}^{[p0]}).
\end{aligned} \tag{B.1}$$

Thus, the gradient $\nabla \mathbf{X}^{[p0]}$ can be deduced as follows

$$\begin{aligned}
\nabla \mathbf{X}^{[p0]} =&2(\sum_{q,q\neq p} (\mathbf{X}^{[q0]} \otimes \mathbf{I})^{\mathrm{T}}\mathbf{K}^{[pq]}(\mathbf{X}^{[q0]} \otimes \mathbf{I}))vec(\mathbf{X}^{[p0]}) \\
=&2 \sum_{q,q\neq p} (\mathbf{X}^{[q0]} \otimes \mathbf{I})^{\mathrm{T}}\mathbf{K}^{[pq]} vec(\mathbf{X}^{[p0]}\mathbf{X}^{[0q]}) \\
=&2 \sum_{q,q\neq p} \mathbf{I} mat(\mathbf{K}^{[pq]} vec(\mathbf{X}^{[p0]}\mathbf{X}^{[0q]}))\mathbf{X}^{[q0]} \\
=&2 \sum_{q,q\neq p} mat(\mathbf{K}^{[pq]} vec(\mathbf{X}^{[p0]}\mathbf{X}^{[0q]}))\mathbf{X}^{[q0]}.
\end{aligned} \tag{B.2}$$

The derivation of the gradient $\nabla \mathbf{X}^{[pq]}$: The objective function is

$$vec(\mathbf{X}^{[pq]})^{\mathrm{T}}\mathbf{K}^{[pq]} vec(\mathbf{X}^{[pq]}). \tag{B.3}$$

It can be simply indicated that the gradient is

$$\nabla \mathbf{X}^{[pq]} = 2mat(\mathbf{K}^{[pq]} vec(\mathbf{X}^{[pq]})). \tag{B.4}$$

By substituting the factorization of the affinity matrix, the gradient can be further written as

$$\begin{aligned}
\nabla \mathbf{X}^{[pq]} =& 2mat((\mathbf{V}^{[q]} \otimes \mathbf{V}^{[p]}) diag(\mathbf{D}^{[pq]})(\mathbf{V}^{[q]} \otimes \mathbf{V}^{[p]})^{\mathrm{T}} vec(\mathbf{X}^{[pq]})) \\
=& 2mat((\mathbf{V}^{[q]} \otimes \mathbf{V}^{[p]}) diag(\mathbf{D}^{[pq]}) vec({\mathbf{V}^{[p]}}^{\mathrm{T}} \mathbf{X}^{[pq]} \mathbf{V}^{[q]})) \\
=& 2mat((\mathbf{V}^{[q]} \otimes \mathbf{V}^{[p]}) vec(\mathbf{D}^{[pq]} \circ ({\mathbf{V}^{[p]}}^{\mathrm{T}} \mathbf{X}^{[pq]} \mathbf{V}^{[q]}))) \\
=& 2\mathbf{V}^{[p]}(\mathbf{D}^{[pq]} \circ ({\mathbf{V}^{[p]}}^{\mathrm{T}} \mathbf{X}^{[pq]} \mathbf{V}^{[q]})) {\mathbf{V}^{[q]}}^{\mathrm{T}}.
\end{aligned} \tag{B.5}$$

References

[1] J. Yan, X.-C. Yin, W. Lin, C. Deng, H. Zha, X. Yang, A short survey of recent advances in graph matching, in: Proceedings of the 2016 ACM on International Conference on Multimedia Retrieval, ICMR '16, Association for Computing Machinery, New York, NY, USA, 2016, p. 167–174. doi:10.1145/2911996.2912035.
URL https://doi.org/10.1145/2911996.2912035

[2] D. HIDOVIĆ, M. PELILLO, Metrics for attributed graphs based on the maximal similarity common subgraph, International Journal of Pattern Recognition and Artificial Intelligence 18 (03) (2004) 299–313. arXiv:https://doi.org/10.1142/S0218001404003216, doi:10.1142/S0218001404003216.
URL https://doi.org/10.1142/S0218001404003216

[3] C. J. Romanowski, R. Nagi, M. Sudit, Data mining in an engineering design environment: Or applications from graph matching, Computers & Operations Research 33 (11) (2006) 3150–3160, part Special Issue: Operations Research and Data Mining. doi:https://doi.org/10.1016/j.cor.2005.01.025.
URL https://www.sciencedirect.com/science/article/pii/S0305054805000262

[4] J. Hays, M. Leordeanu, A. A. Efros, Y. Liu, Discovering texture regularity as a higher-order correspondence problem, in: Proceedings of the 9th European Conference on Computer Vision - Volume Part II, ECCV'06, Springer-Verlag, Berlin, Heidelberg, 2006, p. 522–535. doi:10.1007/11744047_40.

[5] M. Chertok, Y. Keller, Spectral symmetry analysis, IEEE Transactions on Pattern Analysis and Machine Intelligence 32 (07) (2010) 1227–1238. doi:10.1109/TPAMI.2009.121.

[6] U. Gaur, Y. Zhu, B. Song, A. Roy-Chowdhury, A "string of feature graphs" model for recognition of complex activities in natural videos, in: 2011 International Conference on Computer Vision, 2011, pp. 2595–2602. doi:10.1109/ICCV.2011.6126548.

[7] S. Martello, P. Toth, Linear assignment problems, in: S. Martello, G. Laporte, M. Minoux, C. Ribeiro (Eds.), Surveys in Combinatorial Optimization, Vol. 132 of North-Holland Mathematics Studies, North-Holland, 1987, pp. 259–282. doi:https://doi.org/10.1016/S0304-0208(08)73238-9.
URL https://www.sciencedirect.com/science/article/pii/S0304020808732389

[8] P. Pardalos, F. Rendl, H. Wolkowicz, The Quadratic Assignment Problem: A Survey and Recent Developments, DIMACS Series in Discrete Mathematics and Theoretical Computer Science, 1994, pp. 1–42. doi:10.1090/dimacs/016/01.

[9] O. Duchenne, F. Bach, I.-S. Kweon, J. Ponce, A tensor-based algorithm for high-order graph matching, IEEE Transactions on Pattern Analysis and Machine Intelligence 33 (12) (2011) 2383–2395. doi:10.1109/TPAMI.2011.110.

[10] Y. Wu, M. Gong, W. Ma, S. Wang, High-order graph matching based on ant colony optimization, Neurocomputing 328 (2019) 97–104, chinese Conference on Computer Vision 2017. doi:https://doi.org/10.1016/j.neucom.2018.02.104.
URL https://www.sciencedirect.com/science/article/pii/S0925231218309561

[11] Anita, A. Yadav, Discrete artificial electric field algorithm for high-order graph matching, Applied Soft Computing 92 (2020) 106260. doi:https://doi.org/10.1016/j.asoc.2020.106260.
URL https://www.sciencedirect.com/science/article/pii/S1568494620302003

[12] H. W. Kuhn, The Hungarian Method for the Assignment Problem, Naval Research Logistics Quarterly 2 (1–2) (1955) 83–97. doi:10.1002/nav.3800020109.

[13] B. J. van Wyk, M. A. van Wyk, Kronecker product graph matching., ACM Transactions on Asian and Low-Resource Language Information Processing 36 (2003) 2019–2030.

[14] M. Zaslavskiy, F. R. Bach, J.-P. Vert, A path following algorithm for the graph matching problem., IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI) 31 (2009) 2227–2242.

[15] B. Jiang, J. Tang, C. H. Q. Ding, B. Luo, Nonnegative orthogonal graph matching., in: AAAI Conference on Artificial Intelligence (AAAI), 2017, pp. 4089–4095.

[16] T. C. Koopmans, M. Beckmann, Assignment problems and the location of economic activities, Econometrica 25 (1) (1957) 53–76.
URL http://www.jstor.org/stable/1907742

[17] F. Zhou, F. De la Torre, Factorized graph matching, IEEE Transactions on Pattern Analysis and Machine Intelligence 38 (9) (2016) 1774–1789. doi:10.1109/TPAMI.2015.2501802.

[18] J. Yan, X.-C. Yin, W. Lin, C. Deng, H. Zha, X. Yang, A short survey of recent advances in graph matching, in: Proceedings of the 2016 ACM on International Conference on Multimedia Retrieval, ACM, 2016.

[19] D. Pachauri, R. Kondor, V. Singh, Solving the multi-way matching problem by permutation synchro-

nization., in: Conference on Neural Information Processing Systems (NeurIPS), 2013, pp. 1860–1868.

[20] Y. Chen, L. J. Guibas, Q.-X. Huang, Near-optimal joint object matching via convex relaxation., in: International Conference on Machine Learning (ICML), 2014, pp. 100–108.

[21] X. Zhou, M. Zhu, K. Daniilidis, Multi-image matching via fast alternating minimization., in: IEEE International Conference on Computer Vision (ICCV), IEEE, 2015, pp. 4032–4040.

[22] J. Yan, Y. Tian, H. Zha, X. Yang, Y. Zhang, S. M. Chu, Joint optimization for consistent multiple graph matching., in: IEEE International Conference on Computer Vision (ICCV), IEEE, 2013, pp. 1649–1656.

[23] J. Yan, J. Wang, H. Zha, X. Yang, S. Chu, Consistency-driven alternating optimization for multigraph matching: A unified approach, IEEE Transactions on Image Processing 24 (3) (2015) 994–1009. doi:10.1109/TIP.2014.2387386.

[24] J. Yan, Y. Li, W. Liu, H. Zha, X. Yang, S. M. Chu, Graduated consistency-regularized optimization for multi-graph matching, in: D. Fleet, T. Pajdla, B. Schiele, T. Tuytelaars (Eds.), Computer Vision – ECCV 2014, Springer International Publishing, Cham, 2014, pp. 407–422.

[25] X. Shi, H. Ling, W. Hu, J. Xing, Y. Zhang, Tensor power iteration for multi-graph matching., in: Computer Vision and Pattern Recognition (CVPR), IEEE, 2016, pp. 5062–5070.

[26] M. Cho, J. Lee, K. M. Lee, Reweighted random walks for graph matching., in: European Conference on Computer Vision (ECCV), 2010, pp. 492–505.

[27] M. Leordeanu, M. Hebert, R. Sukthankar, An integer projected fixed point method for graph matching and map inference, in: Y. Bengio, D. Schuurmans, J. Lafferty, C. Williams, A. Culotta (Eds.), Advances in Neural Information Processing Systems, Vol. 22, Curran Associates, Inc., 2009.

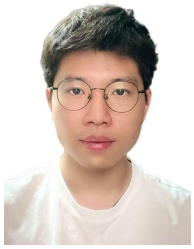
[28] Q. Wang, X. Zhou, K. Daniilidis, Multi-image semantic matching by mining consistent features., in: Computer Vision and Pattern Recognition (CVPR), 2018, pp. 685–694.

[29] M. Cho, K. Alahari, J. Ponce, Learning graphs to match, in: 2013 IEEE International Conference on Computer Vision, 2013, pp. 25–32. doi:10.1109/ICCV.2013.11.

[30] B. Ham, M. Cho, C. Schmid, J. Ponce, Proposal flow: Semantic correspondences from object proposals, in: arXiv:1703.07144, 2017.

[31] A. Krizhevsky, I. Sutskever, G. E. Hinton, Imagenet classification with deep convolutional neural networks, in: F. Pereira, C. Burges, L. Bottou, K. Weinberger (Eds.), Advances in Neural Information Processing Systems, Vol. 25, Curran Associates, Inc., 2012.

[32] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, A. Zisserman, The pascal visual object classes challenge: A retrospective, International Journal of Computer Vision 111 (1) (2015) 98–136.



Liangliang Zhu received the B.S. degree from the Beijing Institute of Technology, Beijing, China, in 2018. He is pursuing the Ph.D. degree with the Key Laboratory of Technology in Geo-spatial Information Process and Application Systems, Aerospace Information Research Institute, Chinese Academy of Sciences, Beijing. His research interests include image registration, and machine learning.

Xinwen Zhu received the B.S. degree from Tianjin University, China, in 2020. She is pursuing the Ph.D. degree with the Key Laboratory of Technology in Geo-spatial Information Process and Application Systems, Aerospace Information Research Institute, Chinese Academy of Sciences, Beijing. Her research interests include hyperspectral image processing, and machine learning.

Xiurui Geng received the B.S. and M.S. degrees in applied mathematics from Beihang University, Beijing, China, in 1999 and 2002, respectively, and the Ph.D. degree in hyperspectral remote sensing from the Institute of Remote Sensing Applications, Chinese Academy of Sciences, Beijing, in 2005. He is a Professor with the Key Laboratory of Technology in Geo-spatial Information Process and Application Systems, Aerospace Information Research Institute, Chinese Academy of Sciences. His research interests include hyperspectral image processing, machine learning, and the basic theory of matrices.