A Matrix Decomposition Perspective to Multiple Graph Matching

Junchi Yan[1,2], Hongteng Xu[3,4], Hongyuan Zha[3,5], Xiaokang Yang[1], Huanxi Liu[1]*, Stephen Chu[2]
[1]Shanghai Jiao Tong University, [2]IBM Research – China
[3]College of Computing, [4]School of Electronic Engineering, Georgia Institute of Technology
[5]Software Engineering Institute, East China Normal University
{yanjunchi,xkyang,lhxsjtu}@sjtu.edu.cn hxu42@gatech.edu zha@cc.gatech.edu schu@us.ibm.com

Abstract

Graph matching has a wide spectrum of applications and in its general form is known NP-hard. In many vision tasks, one realistic problem arises for finding the global node mappings across a batch of corrupted weighted graphs. This paper is an attempt to connect graph matching, especially multi-graph matching to the matrix decomposition model and its relevant convex optimization algorithms. Our method aims to extract the common inliers and their synchronized permutations from disordered weighted graphs in the presence of deformation and outliers. Under the proposed framework, several variants can be derived in the hope of accommodating to specific types of noises. Experimental results empirically show that the proposed paradigm exhibits interesting behaviors and often performs competitively with the state-of-the-arts.

1. Introduction

Graph matching (GM) is one of the essential problems in computer science and mathematics. It has received wide attentions over the last decades [13, 19], and lies at the heart of many important computer vision problems and applications as diverse as object recognition [15], flow analysis [40, 36], object tracking [53], person re-identification [45], and image labeling [63] among others, which require correspondence matching between feature sets. Different from the point based matching methods such as RANSAC [18] and Iterative Closet Point (ICP) [64], GM methods incorporate both unary *node-to-node*, as well as second-order *edge-to-edge* similarity as structural information. By encoding the geometric information in the graph representation and matching process, GM methods can usually produce more satisfactory node correspondence results than other methods. Due to its NP-hard nature, current GM methods either involve finding approximate solutions [65, 10, 29] or obtaining the global optimum in polynomial time for particular types of graphs [17, 39, 1].

Most GM methods focus on establishing correspondences between two feature sets [30, 21, 10, 5, 16]. In general, the pairwise GM problem can be formulated as a quadratic assignment problem (QAP), which can be divided into two categories [65]: i) the Koopmans-Beckmann's QAP [25] written by $\text{tr}(\mathbf{X}^\top\mathbf{F}_i\mathbf{X}\mathbf{F}_j) + \text{tr}(\mathbf{K}_p^\top\mathbf{X})$ where $\mathbf{X}$ refers to the assignment matrix between two graphs. $\mathbf{F}_i$, $\mathbf{F}_j$ are the weighted adjacency matrices and $\mathbf{K}_p$ is the node-to-node similarity matrix; ii) the more general Lawler's QAP [26] by $\text{vec}(\mathbf{X})^\top\mathbf{K}\text{vec}(\mathbf{X})$ where $\mathbf{K}$ is the second-order affinity matrix. Note the Koopmans-Beckmann's QAP can always be represented as a special case of the Lawler's by setting $\mathbf{K} = \mathbf{F}_j \otimes \mathbf{F}_i$ and a large portion of extant GM works including both two-graph matching [21, 10, 49, 65] and multi-graph matching [56, 57, 55] adopt the Lawler's formula perhaps due to its generality and compactness.

Nevertheless, in many applications it is often required to find the global matchings across a batch of graphs. With the advance of imaging and scanning technologies, there is an increasing need for multi-graph matching [52]. Moreover, graphic shape analysis and search often require to model objects by multi-view assembly [20] that also lends itself to the problem of graph clustering, classification and indexing. A recent work [37] applies multi-graph matching to the problem of multi-sources topic alignment.

2. Related Work

Pairwise graph matching Pairwise graph matching that involves matching two graphs one time has been extensively studied in literature. Most pairwise graph matching methods [22, 28, 48, 29, 10, 49, 16] are based on the general Lawler's QAP and relax the solution domain from the permutation (assignment) matrix space to the (sub) doubly-stochastic matrix space. Meanwhile there is another thread [50, 43, 44, 2, 4, 61, 38] that explores a more special case: the Koopmans-Beckmann's QAP. Note the weighted adjacency matrix $\mathbf{F}_1, \mathbf{F}_2$ cannot be recovered from the pairwise

*Corresponding author.

affinity matrix $\mathbf{K}$ in the Lawler's problem, as it directly encodes the pairwise affinity between edges and nodes.

We also briefly mention several works in parallel with the present work: i) learning the affinity over graphs [5, 31, 30, 9]; ii) encoding higher-order affinity terms [31, 8, 14, 27, 59]; iii) progressive matching [11] or a pipeline for integrating feature detection and matching [12].

Multi-graph matching Multi-graph matching (MGM) is less studied compared with two-graph matching. Very recently, there is an emerging trend for devising advanced MGM (or specifically registration) methods [46, 23, 41, 56, 47, 55, 7, 58], because directly borrowing on-the-shelf two-graph solvers does not fully utilize the global affinity and matching consistency information. These various MGM methods are mostly motivated to better address or capture the following aspects of the problem:

i) *Matching cycle-consistency.* One drawback of employing a two-graph matching solver for multi-graph is that the so-called cycle-consistency cannot be automatically fulfilled. Consider a simple and concrete example, given graphs $\mathcal{G}_i, \mathcal{G}_j, \mathcal{G}_a, \mathcal{G}_b$ of equal size in the absence of outliers, the four pairwise matchings $\mathbf{X}_{ij}$, $\mathbf{X}_{ia}$, $\mathbf{X}_{aj}$, $\mathbf{X}_{ib}$ computed by two-graph matching independently may lead to cycle-inconsistency *i.e.* $\mathbf{X}_{ia}\mathbf{X}_{aj} \neq \mathbf{X}_{ib}\mathbf{X}_{bj}$.

Multi-graph matching models are devised to capture consistency either iteratively or in a one-shot fashion: a) *Iterative methods* [47, 56, 57, 55] try to maximize the affinity and in the mean time account for consistency. [56, 57] enforce the cycle-consistency by strictly enforcing the equality constraints $\mathbf{X}_{ij} = \mathbf{X}_{ib}\mathbf{X}_{bj}$ during the whole iterative variable updating procedure. This mechanism works in a 'hard' expectation-maximization fashion. Thus the performance can be sensitive to the initial point and variable rotating order and inherently vulnerable to error accumulation. Similar mechanism is devised by [47] which extends the Graduated Assignment algorithm for two-graph matching [21] to multi-graph one. A more flexible and robust mechanism is proposed in [55] and the journal extension [54], where the consistency is gradually infused for optimization over iterations. b) In contrast, *one-shot methods* are devised to achieve overall consistency in one shot. By employing spectral techniques, [41, 7, 66] all enforce consistency as a post-processing step whose inputs are the putative pairwise matchings $\{\mathbf{X}\}_{i,j=1}^{N}$ calculated by a two-graph matching solver. Since no reinforcement is performed to boost the overall affinity as done by the above iterative methods, these two methods are found dependent on the input matchings by a two-graph matching solver.

ii) *Affinity across graphs.* In [47, 56, 57, 55], the objective for multi-graph matching is explicitly written by adding up pairwise affinity terms $\{\text{vec}(\mathbf{X})^{\top}\mathbf{K}\text{vec}(\mathbf{X})\}_{i,j=1}^{N}$ under the Lawler's form, and the pairwise matchings $\{\mathbf{X}\}_{i,j=1}^{N}$ are updated iteratively which strictly (or gradually) satisfy the cycle-consistency constraints. While for the aforementioned two non-iterative methods [41, 7], affinity information is only considered locally for a pair of graphs at hand. In their post-process step where consistency is enforced, affinity cues are totally dropped off.

Based on the above discussion, we make two important observations: i) cycle-inconsistency is fundamentally caused by the use of pairwise matchings $\{\mathbf{X}\}_{i,j=1}^{N}$ – a distributed and redundant representation for multi-graph matching; ii) for affinity modeling, existing multi-graph matching methods are mostly based on the pairwise affinity representation, regardless of Lawler's or Koopmans-Beckmann's QAP. Though the pairwise affinity matrix $\mathbf{K}$ in Lawler's QAP is general and flexible for two-graph matching, while employing this *pairwise* affinity metric to the N-graphs problem is unnatural. In fact, the affinity matrix becomes an N-order tensor if one refers to an N-graph affinity as a counterpart to the pairwise affinity matrix. However, this extension is not efficient nor compact, and will cause additional mathematical difficulties and computational cost due to the involvement of tensors.

Motivated by seeking a fresh and compact framework for addressing the above two shortcomings, this paper explicitly uses the *per-graph* weighted adjacency matrices $\{\mathbf{F}_i\}_{i=1}^{N}$ instead of the *pairwise* affinity matrices $\{\mathbf{K}\}_{i,j=1}^{N}$ since the latter we believe is not a natural nor effective affinity metrics for multi-graph matching. For a similar reason, we abandon the widely used pairwise matching representation $\{\mathbf{X}\}_{i,j=1}^{N}$ and turn to a compact set of N permutations $\{\mathbf{X}\}_{i=1}^{N}$ to an virtual common reference. However, such a reformulation can be challenging for optimization due to the strong multi-modality of the energy landscape. Yet focusing on $\{\mathbf{F}_i\}_{i=1}^{N}$ and $\{\mathbf{X}\}_{i=1}^{N}$ naturally allows us to take a matrix decomposition perspective towards the multi-graph matching problem, in the hope of dismissing some noises to alleviate the optimization difficulty. Another good news is that matrix decomposition is known robust to noise and well studied by sound theory *e.g.* Robust PCA [6], efficient algorithms [33, 34] and successful applications [42, 60, 62] for vision problems – but still relatively new to graph matching.

Novelty We summarize the fresh ideas of our approach:

i) *Compact and global representation* Our model dispenses with pairwise terms for both affinity $\{\mathbf{K}\}_{i,j=1}^{N}$ and matchings $\{\mathbf{X}\}_{i,j=1}^{N}$ which dominate extant MGM methods. As a result, our model is inherently free from matching cycle-inconsistency, and can also capture more global affinity clues beyond two graphs at one time;

ii) *Matrix decomposition perspective* Our formulation connects the MGM problem to the intensively studied matrix decomposition model. Different matrix decomposition variants can be readily borrowed which correspond to choosing a solver tailored to a certain type of noise, *e.g.* ℓ_F and ℓ_1 norm for Gaussian and Laplacian noise respectively.

iii) *Explicit use of the number of common inliers* Our model has a parameter for the number of common inliers. Thus the knowledge of common liner number can be explicitly leveraged. In many cases, this number is known by given a template consisting of common inliers, or can be estimated from graphs (this paper also tentatively provides an inlier number estimation method). In contrast, extant MGM methods have few mechanisms to utilize this information, though obtaining accurate n_c is often challenging.

3. Proposed formulation and algorithm

3.1. Problem formulation

Let n_c as the number of common inliers across N graphs $\{\mathcal{G}_i\}_{i=1}^N$, and n_i as the number of total nodes in graph $\mathcal{G}_i$ ($n_i \geq n_c$), where the number of outliers is $n_i - n_c$ varying by each graph. Denote $\mathbf{F}_i^c \in \mathbb{R}^{n_c \times n_c}$ as the weighted adjacency matrix formed by n_c inliers in graph $\mathcal{G}_i$, which is augmented by random column permutation with outliers, to the augmented version of $\mathbf{F}_i \in \mathbb{R}^{n_i \times n_i}$. The goal is to identify the n_c common inliers, and establish the one-to-one node correspondences among them. Also, we use $\mathbf{X}$ to denote the node-to-node assignment matrix between two graphs. We further define the assignment matrix domain: $\mathbb{X}_i$ that matches the i^{th} graph to a virtual reference graph that only includes n_c common inliers:

$$\mathbb{X}_i = \{\mathbf{X}_i \in \{0,1\}^{n_i \times n_c}, \quad \mathbf{1}_{n_i}^\top \mathbf{X}_i = \mathbf{1}_{n_c}^\top \quad \mathbf{X}_i \mathbf{1}_{n_c} \leq \mathbf{1}_{n_i}\} \quad (1)$$

where $\mathbf{1}_{n_c} \in \mathbb{R}^{n_c \times 1}$ (*resp.* $\mathbf{1}_{n_i} \in \mathbb{R}^{n_i \times 1}$) denotes the column vector for all its entries being 1.

For GM, one considers not only the unary affinity, but also the second-order edge-wise measurements. In line with the Koopmans-Beckmann's QAP that explicitly makes use of the weighted adjacency matrix, the two-graph matching problem for $\mathcal{G}_1$, $\mathcal{G}_2$ can be written by $\|\mathbf{X}^\top \mathbf{F}_1 \mathbf{X} - \mathbf{F}_2\|$ [50]. For MGM, by implicitly setting a virtual reference graph, we can vectorize $\mathbf{X}_i^\top \mathbf{F}_i \mathbf{X}_i \in \mathbb{R}^{n_c \times n_c}$ for each graph $\mathcal{G}_i$ and stack them in one matrix: $\mathbf{D} \in \mathbb{R}^{n_c^2 \times N}$.

$$\mathbf{D} = [\mathrm{vec}(\mathbf{X}_1^\top \mathbf{F}_1 \mathbf{X}_1), \ldots, \mathrm{vec}(\mathbf{X}_N^\top \mathbf{F}_N \mathbf{X}_N)]$$

where each variable $\{\mathbf{X}_i\}_{i=1}^N$ reorders the inliers to enforce the columns similar to each other in $\mathbf{D}$. Hence $\mathbf{D}$ is supposed to be rank-deficient, ideally rank($\mathbf{D}$)=1 when exact graph matching (all graphs are the same except their nodes are permuted) is supposed. Or its rank would be smaller than the number of graphs as in practice the weighted adjacency matrices are often coupled with noises, which violates the low-rank assumption. Moreover, the noises are often sparse and randomly imposed. This observation leads to the low-rank sparsity decomposition formulation: By replacing rank($\cdot$) and ℓ_0 norm by their convex surrogates, *i.e.* nuclear norm $\|\cdot\|_*$ and ℓ_1-norm respectively [6], we reach a widely adopted formulation in computer vision [60]

$$\min_{\{\mathbf{X}_i\}_{i=1}^N \in \mathbb{X}_i, \mathbf{L}, \mathbf{S}} \|\mathbf{L}\|_* + \lambda \|\mathbf{S}\|_1 \quad s.t. \quad \mathbf{D} = \mathbf{L} + \mathbf{S} \qquad (2)$$

Note the above formula is a regularized consensus problem, which is popularly tackled by Alternative Direction Method of Multiplier (ADMM) [3] in literature such as [33, 62, 24, 32]. For instance, the previous work [62, 24] employ the ADMM style solver to solve a similar regularized consensus problem. However, the graph matching problem we address in this paper involves the *second-order* weighted adjacency matrix, beyond the unary point-wise feature vector such as SIFT descriptor or point coordinate information. This leads to a more complicated optimization problem, since the objective function involves quadratic term $\mathbf{X}^\top \mathbf{F} \mathbf{X}$ in $\mathbf{D}$, instead of linear terms as considered in [62, 24] for first-order *point* set alignment.

In the following, we will present the formulation and solvers in the presence of outliers and not, followed by a brief discussion on possible extensions to cope with different types of noises. Finally, a cycle-consistency based approach for estimating the number of common inliers n_c will be presented, which can serve as a general preprocessing baseline for our method when the parameter n_c is unknown.

3.2. Approach overview and algorithm details

Overview The key idea for solving problem (2) by Lagrangian multipliers to take care of i) the decomposition residue from the raw matrix $\mathbf{D}$ and its additive components $\mathbf{L}$, $\mathbf{S}$ as well as the node equality correspondence constraint as shown in Eq.1, which is encoded by $\mathbf{W}$ in this paper; ii) the residue between the raw matching variable $\{\mathbf{X}_i\}_{i=1}^N$ and the relaxed continuous variable $\{\bar{\mathbf{X}}_i\}_{i=1}^N$. A coordinate descent fashion variable updating procedure is applied to reach a solution (perhaps a local optimum), with respect to variable $\{\mathbf{X}_i\}_{i=1}^N$, $\{\bar{\mathbf{X}}_i\}_{i=1}^N$, the decompositions $\mathbf{L}$, $\mathbf{S}$, and the Lagrangian multipliers $\mathbf{Y}$ and $\mathbf{Z}$ as depicted in the algorithmic chart of Alg.1. More details are as follows.

We start with the general case in the presence of outliers. The relaxed form (2) is still not directly trackable due to the optimization with respect to the N number of $\mathbf{X}_i$ is highly nonlinear and fundamentally a combinatorial problem. We introduce the following notations for $\mathbf{W}(\mathbf{L}, \mathbf{S}, \{\mathbf{X}_i\}_{i=1}^N) \in \mathbb{R}^{(n_c^2+n_c) \times N}$, $\mathbf{G} \in \mathbb{R}^{n_c \times N}$, $\mathbf{M}_i \in \mathbb{R}^{(n_i+n_i n_c) \times n_i n_c}$ and $\mathbf{h}_i \in \mathbb{R}^{(n_i+n_i n_c) \times 1}$ to facilitate the later exposition:

$$\mathbf{G} = [(\mathbf{I}_{n_c} \otimes \mathbf{1}_{n_1}^\top)\mathrm{vec}(\mathbf{X}_1), \ldots, (\mathbf{I}_{n_c} \otimes \mathbf{1}_{n_N}^\top)\mathrm{vec}(\mathbf{X}_N)],$$

$$\mathbf{W}(\mathbf{L}, \mathbf{S}, \{\mathbf{X}_i\}_{i=1}^N) = \begin{bmatrix} \mathbf{D} \\ \mathbf{G} \end{bmatrix} - \begin{bmatrix} \mathbf{L} + \mathbf{S} \\ \mathbf{1}_{n_c} \mathbf{1}_N^\top \end{bmatrix},$$

$$\mathbf{M}_i = \begin{bmatrix} \mathbf{1}_{n_c}^\top \otimes \mathbf{I}_{n_i} \\ -\mathbf{I}_{n_c n_i} \end{bmatrix},$$

$$\mathbf{h}_i = \begin{bmatrix} \mathbf{1}_{n_i} \\ \mathbf{0}_{n_c n_i} \end{bmatrix}$$

Here $\mathbf{I}_{n_c n_i}$ is the square identity matrix with size $n_c n_i$, and $\mathbf{0}_{n_c n_i} \in \mathbb{R}^{n_c n_i \times 1}$ is a vector with all zero entries. By explicitly writing out $\mathbf{X}_i \in \mathbb{X}_i$, and relaxing $\mathbf{X}_i$ to the continuous domain within [0,1] as widely adopted by graph matching methods [22, 49, 65] *etc.*, we reach:

$$\min_{\mathbf{L},\mathbf{S},\{\mathbf{X}_i\}_{i=1}^N} \|\mathbf{L}\|_* + \lambda\|\mathbf{S}\|_1 \tag{3}$$
$$s.t. \quad \mathbf{W}(\mathbf{L},\mathbf{S},\{\mathbf{X}_i\}_{i=1}^N) = \mathbf{0}, \quad \{\mathbf{M}_i\text{vec}(\mathbf{X}_i) \leq \mathbf{h}_i\}_{i=1}^N$$

The equality constraint $\mathbf{W}$=$\mathbf{0}$ refers to the decomposition of low-rank and sparsity components from the raw matrix (the upper part), as well as the one-to-one matching for each inliers (the lower part). The inequality $\mathbf{M}_i\text{vec}(\mathbf{X}_i) \leq \mathbf{h}_i$ relates to the fact for each node (inlier/outlier) in the graph, there is at most one correspondence from another graph.

To simplify the mathematical difficulty in handling the quadratic terms $\mathbf{X}_i^\top\mathbf{F}_i\mathbf{X}_i$ in $\mathbf{W}$, we introduce the new variable $\overline{\mathbf{X}}_i$ by replacing $\mathbf{X}_i^\top\mathbf{F}_i\mathbf{X}_i$ with $\overline{\mathbf{X}}_i^\top\mathbf{F}_i\mathbf{X}_i$ for $\overline{\mathbf{X}}_i = \mathbf{X}_i$. We define $\overline{\mathbf{D}} \in \mathbb{R}^{n_c^2\times N}$ as the relaxed version of $\mathbf{D}$:

$$\overline{\mathbf{D}} = [\text{vec}(\overline{\mathbf{X}}_1^\top\mathbf{F}_1\mathbf{X}_1), \ldots, \text{vec}(\overline{\mathbf{X}}_N^\top\mathbf{F}_N\mathbf{X}_N)]$$

Accordingly, we use $\overline{\mathbf{D}}$ to replace $\mathbf{D}$ in $\mathbf{W}$, which is term as $\overline{\mathbf{W}}(\mathbf{L},\mathbf{S},\{\mathbf{X}_i,\overline{\mathbf{X}}_i\}_{i=1}^N)$. It induces the new expression:

$$\min_{\mathbf{L},\mathbf{S},\{\mathbf{X}_i,\overline{\mathbf{X}}_i\}_{i=1}^N} \|\mathbf{L}\|_* + \lambda\|\mathbf{S}\|_1 \tag{4}$$
$$s.t. \quad \overline{\mathbf{W}}(\mathbf{L},\mathbf{S},\{\mathbf{X}_i,\overline{\mathbf{X}}_i\}_{i=1}^N) = 0,$$
$$\{\mathbf{X}_i = \overline{\mathbf{X}}_i, \mathbf{M}_i\text{vec}(\mathbf{X}_i) \leq \mathbf{h}_i\}_{i=1}^N$$

We use the augmented Lagrangian multiplier method to solve this problem. By introducing the new variable $\mathbf{Y} \in \mathbb{R}^{(n_c^2+n_c)\times N}$ and $\{\mathbf{Z}_i \in \mathbb{R}^{n_i\times n_c}\}_{i=1}^N$, the augmented Lagrangian function of problem (4) is written as:

$$\begin{aligned}&\mathcal{L}(\mathbf{L},\mathbf{S},\mathbf{Y},\{\mathbf{X}_i,\overline{\mathbf{X}}_i,\mathbf{Z}_i\}_{i=1}^N) \\ =&\|\mathbf{L}\|_* + \lambda\|\mathbf{S}\|_1 + <\mathbf{Y},\overline{\mathbf{W}}> + \frac{\mu_1}{2}\|\overline{\mathbf{W}}\|_F^2 \\ &+\sum_{i=1}^N <\mathbf{Z}_i,\mathbf{X}_i-\overline{\mathbf{X}}_i> + \sum_{i=1}^N \frac{\mu_2}{2}\|\mathbf{X}_i-\overline{\mathbf{X}}_i\|_F^2\end{aligned} \tag{5}$$

μ_1, μ_2 are positive scalars which follow the updating rule $\mu_1^{t+1} = \rho_1\mu_1^t$ ($\mu_2^{t+1} = \rho_2\mu_2^t$) for ρ_1>1(ρ_2>1) over iteration t, and $\mathbf{Y}$, $\{\mathbf{Z}_i\}_{i=1}^N$ are the Lagrange multipliers. $\|\cdot\|_F$ denotes the Frobenius norm *i.e.* $\|\mathbf{A}\|_F$=$\sqrt{\sum_{i,j}a_{ij}^2}$, and $< \cdot,\cdot >$ denotes the matrix inner product such that $<\mathbf{A},\mathbf{B}>=\text{Trace}(\mathbf{A}^\top\mathbf{B})$ for the input matrix $\mathbf{A}$ and $\mathbf{B}$.

Now we solve the above augmented Lagrangian function in an alternating optimization manner, over iteration t.

Update L, E: The problem of updating $\mathbf{L}$, $\mathbf{S}$ are both convex programs, which can be written in the form of proximal operators associated with a nuclear or ℓ_1 norm respectively. Moreover, when updating $\mathbf{L}$, $\mathbf{S}$, only the first n_c^2 rows of $\mathbf{Y}$ are involved. Thus we use $\overline{\mathbf{Y}} \in \mathbb{R}^{n_c^2\times N}$ to denote these rows. Then the partial minimization is as follows and can be solved in a closed form by the technique used in [33].

$$\begin{aligned}(\mathbf{L}^{t+1},\mathbf{S}^{t+1}) =& \arg\min_{\mathbf{L},\mathbf{S}} \|\mathbf{L}\|_* + \lambda\|\mathbf{S}\|_1 \\ &+ <\overline{\mathbf{Y}}^t, \mathbf{D}^t - \mathbf{L} - \mathbf{S}> + \frac{\mu_1}{2}\|\mathbf{D}^t-\mathbf{L}-\mathbf{S}\|_F^2\end{aligned} \tag{6}$$

Update Y: $\mathbf{Y}$ can be updated efficiently by:

$$\mathbf{Y}^{t+1} = \mathbf{Y}^t + \mu_1\overline{\mathbf{W}}(\mathbf{L}^t,\mathbf{S}^t,\{\mathbf{X}_i^t,\overline{\mathbf{X}}_i^t\}_{i=1}^N) \tag{7}$$

Update $\mathbf{X}_1,\ldots,\mathbf{X}_N$: By fixing other variables, the partial objective function *w.r.t.* $\{\mathbf{X}_i\}_{i=1}^N$ can be written as:

$$\begin{aligned}\min_{\{\mathbf{X}_i\}_{i=1}^N} &\left\langle \mathbf{Y}^{t+1}, \begin{bmatrix}\mathbf{D}^t\\ \mathbf{G}^t\end{bmatrix} - \begin{bmatrix}\mathbf{L}^{t+1}+\mathbf{S}^{t+1}\\ \mathbf{1}_{n_c}\mathbf{1}_N^\top\end{bmatrix}\right\rangle \\ &+\frac{\mu_1}{2}\left\|\begin{bmatrix}\mathbf{D}^t\\ \mathbf{G}^t\end{bmatrix} - \begin{bmatrix}\mathbf{L}^{t+1}+\mathbf{S}^{t+1}\\ \mathbf{1}_{n_c}\mathbf{1}_N^\top\end{bmatrix}\right\|_F^2 \\ &+\sum_{i=1}^N <\mathbf{Z}_i^t,\mathbf{X}_i-\overline{\mathbf{X}}_i^t> + \sum_{i=1}^N\frac{\mu_2}{2}\|\mathbf{X}_i-\overline{\mathbf{X}}_i^t\|_F^2\end{aligned} \tag{8}$$

This can be equivalently written as:

$$\begin{aligned}\min_{\{\mathbf{X}_i\}_{i=1}^N} &\frac{\mu_1}{2}\left\|\begin{bmatrix}\mathbf{L}^{t+1}+\mathbf{S}^{t+1}\\ \mathbf{1}_{n_c}\mathbf{1}_N^\top\end{bmatrix} + \frac{\mathbf{Y}^{t+1}}{\mu_1} - \begin{bmatrix}\mathbf{D}^t\\ \mathbf{G}^t\end{bmatrix}\right\|_F^2 \\ &+\sum_{i=1}^N\frac{\mu_2}{2}\left\|\overline{\mathbf{X}}_i^t + \frac{1}{\mu_2}\mathbf{Z}_i^t - \mathbf{X}_i\right\|_F^2\end{aligned} \tag{9}$$

Note that (9) can be decoupled into N independent subproblems where the i^{th} subproblem corresponds to solving the following optimization problem *w.r.t.* $\mathbf{X}_i$, where we relax $\mathbf{X}_i$ to the continuous domain from its constrained discrete domain $\mathbb{X}_i$ as defined in formula (1).

$$\begin{aligned}\min_{\mathbf{X}_i} \quad &\text{vec}(\mathbf{X}_i)^\top\left(\frac{\mu_1}{2}\mathbf{B}_i^{t\top}\mathbf{B}_i^t + \frac{\mu_2}{2}\mathbf{I}_{n_in_c}\right)\text{vec}(\mathbf{X}_i) \\ &-\left(\mu_1\mathbf{A}_i^{t+1\top}\mathbf{B}_i^t + \mu_2\mathbf{C}_i^{t\top}\right)\text{vec}(\mathbf{X}_i) \\ &s.t. \quad \mathbf{M}_i\text{vec}(\mathbf{X}_i) \leq \mathbf{h}_i\end{aligned} \tag{10}$$

$\{\mathbf{A}_i^{t+1}$, $\mathbf{B}_i^t$, $\mathbf{b}_i^t$, $\mathbf{C}_i^t\}_{i=1}^N$ are defined as follows, where $\mathbf{e}_i$ is the binary column vector. Its i^{th} element is 1 and others 0.

$$\mathbf{A}_i^{t+1} = \left(\begin{bmatrix}\mathbf{L}^{t+1}+\mathbf{S}^{t+1}\\ \mathbf{1}_{n_c}\mathbf{1}_N^\top\end{bmatrix} + \frac{1}{\mu_1}\mathbf{Y}^{t+1}\right)\mathbf{e}_i$$
$$\mathbf{B}_i^t = \mathbf{I}_{n_c}\otimes\mathbf{b}_i^t, \quad \mathbf{b}_i^t = \begin{bmatrix}\overline{\mathbf{X}}_i^{t\top}\mathbf{F}_i\\ \mathbf{1}_{n_i}^\top\end{bmatrix}$$
$$\mathbf{C}_i^t = \text{vec}\left(\overline{\mathbf{X}}_i^t + \frac{1}{\mu_2}\mathbf{Z}_i^t\right)$$

Update $\overline{\mathbf{X}}_1,\ldots,\overline{\mathbf{X}}_N$: Similarly, $\overline{\mathbf{X}}_i$ is solved by:

$$\begin{aligned}\min_{\overline{\mathbf{X}}_i} \quad &\text{vec}(\overline{\mathbf{X}}_i^\top)^\top\left(\frac{\mu_1}{2}\overline{\mathbf{B}}_i^{t+1\top}\overline{\mathbf{B}}_i^{t+1} + \frac{\mu_2}{2}\mathbf{I}_{n_in_c}\right)\text{vec}(\overline{\mathbf{X}}_i^\top) \\ &-\left(\mu_1\overline{\mathbf{A}}_i^{t+1\top}\overline{\mathbf{B}}_i^{t+1} + \mu_2\overline{\mathbf{C}}_i^{t\top}\right)\text{vec}(\overline{\mathbf{X}}_i^\top) \\ &s.t. \quad \mathbf{M}_i\text{vec}(\overline{\mathbf{X}}_i) \leq \mathbf{h}_i\end{aligned} \tag{11}$$

where $\{\overline{\mathbf{A}}_i^{t+1}$, $\overline{\mathbf{B}}_i^{t+1}$, $\overline{\mathbf{b}}_i^{t+1}$, $\overline{\mathbf{C}}_i^t\}_{i=1}^N$ are defined by:

$$\overline{\mathbf{A}}_i^{t+1} = \left(\mathbf{L}^{t+1}+\mathbf{S}^{t+1}+\frac{1}{\mu_1}\overline{\mathbf{Y}}^{t+1}\right)\mathbf{e}_i,$$
$$\overline{\mathbf{B}}_i^{t+1} = \overline{\mathbf{b}}_i^{t+1}\otimes\mathbf{I}_{n_c}, \quad \overline{\mathbf{b}}_i^{t+1} = (\mathbf{F}_i\mathbf{X}_i^{t+1})^\top,$$
$$\overline{\mathbf{C}}_i^t = \text{vec}\left(\mathbf{X}_i^{t+1\top} - \frac{1}{\mu_2}\mathbf{Z}_i^{t\top}\right)$$

Algorithm 1 Graph Matching via Matrix Decomposition

```
Input $\{\mathbf{F}_i\}_{i=1}^N$, initial $\mathbf{P}_i^0$, $\mu_1$, $\mu_2$, $\lambda$, $T$;
for $t = 1 : T$ do
    Update $\mathbf{L}, \mathbf{S}$ and $\mathbf{Y}$ by Eq.6 and Eq.7 respectively;
    Update $\{\mathbf{X}_i\}_{i=1}^N$, $\{\bar{\mathbf{X}}_i\}_{i=1}^N$, $\{\mathbf{Z}_i\}_{i=1}^N$ by Eq.8, 11, 12;
    Update $\mathbf{D}$ and $\mathbf{G}$ by Eq.13 and Eq.14 respectively;
    Update $\mu_1 = \mu_1\rho, \mu_2 = \mu_2\rho$;
end for
Obtain the binarized $\{\mathbf{P}_i\}_{i=1}^N$ by the Hungarian method.
```

Update $\mathbf{Z}_1, \ldots, \mathbf{Z}_N$: $\{\mathbf{Z}_i\}_{i=1}^N$ can be updated by:

$$\mathbf{Z}_i^{t+1} = \mathbf{Z}_i^t + \mu_2(\bar{\mathbf{X}}_i^{t+1} - \mathbf{X}_i^{t+1}) \tag{12}$$

Update D and **G**: They are updated by their definitions:

$$\mathbf{D}^{t+1} = [\text{vec}(\bar{\mathbf{X}}_1^{t+1\top}\mathbf{F}_1\mathbf{X}_1^{t+1}), \ldots, \text{vec}(\bar{\mathbf{X}}_N^{t+1\top}\mathbf{F}_N\mathbf{X}_N^{t+1})] \tag{13}$$

$$\mathbf{G}^{t+1} = [(\mathbf{I}_{n_c} \otimes \mathbf{1}_{n_1}^\top)\text{vec}(\mathbf{X}_1^{t+1}), \ldots, (\mathbf{I}_{n_c} \otimes \mathbf{1}_{n_N}^\top)\text{vec}(\mathbf{X}_N^{t+1})] \tag{14}$$

In line with many graph matching works [10, 49, 56], as a standard postprocessing, we apply Hungarian method on $\{\mathbf{X}_i\}_{i=1}^N$ to obtain binarized $\{\mathbf{P}_i\}_{i=1}^N$ when iteration ends.

Simplification for the case $n_i = n_c$ When no outlier exists for a given graph, the *partial* permutation matrix $\{\mathbf{X}_i, \bar{\mathbf{X}}_i\}_{i=1}^N$ becomes a *full* squared permutation matrix. For Eq.10, the following equation always holds:

$$\text{vec}(\mathbf{X}_i)^\top\mathbf{B}_i^\top\mathbf{B}_i\text{vec}(\mathbf{X}_i) = \|\mathbf{B}_i\text{vec}(\mathbf{X}_i)\|_2^2 = \|\text{vec}(\mathbf{b}_i\mathbf{X}_i)\|_2^2 = \|\text{vec}(\mathbf{b}_i)\|_2^2$$

Thus the quadratic term in Eq.10 can be dropped off. Similarly, for Eq.11 we have:

$$\text{vec}(\bar{\mathbf{X}}_i^\top)^\top\bar{\mathbf{B}}_i^\top\bar{\mathbf{B}}_i\text{vec}(\bar{\mathbf{X}}_i^\top) = \|\bar{\mathbf{B}}_i\text{vec}(\bar{\mathbf{X}}_i^\top)\|_2^2 = \|\text{vec}(\bar{\mathbf{b}}_i)\|_2^2.$$

Thus updating $\mathbf{X}$ is reduced as follows, and similar for $\bar{\mathbf{X}}$:

$$\min_{\mathbf{X}_i|\mathbf{M}_i\text{vec}(\mathbf{X}_i)\leq\mathbf{h}_i} -\left(\mu_1\mathbf{A}_i^\top\mathbf{B}_i + \mu_2\mathbf{C}_i^\top\right)\text{vec}(\mathbf{X}_i)$$

The algorithm is outlined in Alg.1. Note the optimal solution $\{\mathbf{P}_i\}_{i=1}^N$ that can derive the same pairwise mappings $\{\mathbf{P}_{ij}{=}\mathbf{P}_i\mathbf{P}_j^\top\}_{i,j=1}^N$ is *not unique*, as the overall formulation is non-convex. This is analogous to clustering where the label index value by a clustering algorithm is not critical.

3.3. Extensions and alternative variants

The above algorithm can be extended in several aspects benefiting from the established connection from graph matching to matrix decomposition, given the fact that the latter has been studied intensively in recent years. As it is impossible to exhaustively list all variants to Alg.1 in this paper, here we try to organize these variants into three aspects and leave a comprehensive evaluation under different settings to future work.

Unary features modeling It is easy to combine the first-order node-wise feature by augmenting **D**:

$$\mathbf{D} = \begin{bmatrix} \text{vec}(\mathbf{X}_1^\top\mathbf{F}_1\mathbf{X}_1), \ldots, \text{vec}(\mathbf{X}_N^\top\mathbf{F}_N\mathbf{X}_N) \\ \text{vec}(\mathbf{f}_1\mathbf{X}_1), \ldots, \text{vec}(\mathbf{f}_N\mathbf{X}_N) \end{bmatrix} \tag{15}$$

where $\{\mathbf{f}_i\} \in \mathbb{R}^{d\times n_i}$ is the node feature descriptor of dimension d. This augmentation does not increase the order of the problem *w.r.t.* **X** thus can be solved similarly.

Vectorized edge weight modeling In a more general setting, the weight of each edge in graphs is encoded by a vector instead of scalar. This is useful in computer vision problems since it allows the length and orientation of an edge to be both considered. The variant can be illustrated by:

$$\mathbf{D} = \begin{bmatrix} \text{vec}(\mathbf{X}_1^\top\mathbf{F}_1^l\mathbf{X}_1), \ldots, \text{vec}(\mathbf{X}_N^\top\mathbf{F}_N^l\mathbf{X}_N) \\ \text{vec}(\mathbf{X}_1^\top\mathbf{F}_1^o\mathbf{X}_1), \ldots, \text{vec}(\mathbf{X}_N^\top\mathbf{F}_N^o\mathbf{X}_N) \end{bmatrix} \tag{16}$$

Note the weight is distributed to $\{\mathbf{F}_i^l\}_{i=1}^N$ and $\{\mathbf{F}_i^o\}_{i=1}^N$.

Noise modeling In current model described above, we use ℓ_1 norm to model the noise in Eq.2. This term can be replaced by other norms *e.g.* Frobenius norm, which is regarded effective to Gaussian noises, and $\ell_{2,1}$ norm to sample-specific noises [34], or the re-weighting of these norms by keeping the constrain $\mathbf{D} = \mathbf{L} + \mathbf{S}$:

$$\min_{\mathbf{L},\mathbf{S}} \|\mathbf{L}\|_* + \lambda_{21}\|\mathbf{S}\|_{2,1} + \lambda_F\|\mathbf{S}\|_F + \lambda_1\|\mathbf{S}\|_1$$

There are relevant solvers *e.g.* [35] to these problems.

3.4. Estimating number of inliers n_c

Readers may concern that there is a parameter n_c for forming the decomposition matrix **D** which is not needed by other MGM methods [47, 56, 55, 41]. In fact, these MGM methods have no mechanism to explicitly explore and leverage n_c but always maximize the affinity and consistency for *all points without distinction*. In this spirit, a trivial setting is letting $n_c = n_i$ which denotes our model treats all points equally in line with the above peer methods.

The good news is that this parameter is an interface for utilizing n_c when it is known or can be estimated by a certain means: On one hand, n_c can be easily obtained by manual in some practical situations, *e.g.* counting the number of inlier landmarks on a template. In fact, many two-graph matching methods assume one of the two graphs contains only inliers and try to find their correspondences from the other graph with outliers, and this assumption does not hinder their success in many applications. On the other hand, given a batch of graphs with outliers, it is more desirable to estimate n_c automatically. Though n_c estimation might be considered as a chicken-egg problem in two-graph matching, we believe more graphs help reduce the ambiguity. As an initial endeavor, we present a heuristic approach outlined in Alg.2 which is problem independent. Moreover, this paper does not claim its effectiveness, but it serves as one replaceable component for a practical MGM pipeline.

First, we perform independent pairwise graph matching via on-the-shelf pairwise GM solvers *e.g.* RRWM [10] to obtain a matching configuration $\mathbb{X} = \{\mathbf{X}_{ij}\}_{i,j=1}^N$ between

Algorithm 2 Heuristic consistency-driven n_c estimation

```
Input Putative $\mathbb{X}$ obtained by a pairwise matching solver;
Compute $\mathbf{c}_i = \sum_{j,k=1}^{N} rsum(\frac{\mathbf{X}_{ik}-\mathbf{X}_{ij}\mathbf{X}_{jk}}{2})$.
Reorder $\mathbf{c}_i$ in descending order $\mathbf{c}_i^d$;
Compute the overall $\mathbf{c}^d = \sum_{i=1}^{N} \mathbf{c}_i^d/n$;
Set $c_n = \arg\min_i(abs(\mathbf{c}^d - 1/n))$;
```

each pair of graphs $\mathcal{G}_i$ and $\mathcal{G}_j$[1]. By assuming a reasonable overall matching accuracy of $\mathbb{X}$, we expect the inliers shall have better mapping consistency over different pairwise matching paths. We concretize this idea as follows:

For each $\mathcal{G}_i$, its nodes are mapped to the correspondences in $\mathcal{G}_j$ by $\mathbf{X}_{ij}$, and further $\mathcal{G}_k$ by $\mathbf{X}_{jk}$. We measure the node consistency for each node in $\mathcal{G}_i$ by $\mathbf{c}_i = \sum_{j,k=1}^{N} rsum(\frac{\mathbf{X}_{ik}-\mathbf{X}_{ij}\mathbf{X}_{jk}}{2})$ where $\mathbf{u}{=}rsum(\mathbf{A})$ denotes the summation of the absolute values of elements in each row of $\mathbf{A} \in \mathbb{R}^{n\times n}$ *i.e.* $\{u_i = \sum_{k=1}^{n} |a_{ik}|\}_{i=1}^{n}$. Then we order the element in $\mathbf{c}_i$ by descending order which is termed as $\mathbf{c}_i^d$. Repeating this step for all $\{\mathcal{G}_i\}_{i=1}^{N}$ we obtain a set of $\{\mathbf{c}_i^d\}_{i=1}^{N}$ which are further averaged by $\mathbf{c}^d = \sum_{i=1}^{N} \mathbf{c}_i^d/n$. $\mathbf{c}^d$ is used to measure the node-wise consistency. It also has an intuitive interpretation as the mean probability of obtaining consistent mappings by all possible matching paths $\{\mathcal{G}_i \to \mathcal{G}_j \to \mathcal{G}_k\}_{i,j,k=1}^{N}$ for all nodes in each graph.

On the other hand, when the node mapping over all inliers is all (or mostly) correct — the assumption of Alg.2, then the outliers in one graph will only (or mostly) correspond to outliers in others. Let $Prob(u^k{=}u^{k'})$ denote the probability for outlier u^k and $u^{k'}$ are the same, where u^k and $u^{k'}$ are the nodes in $\mathcal{G}_k$ corresponding to the node u^i in $\mathcal{G}_i$. The former is via the mapping by $\mathbf{X}_{ik}$ while the latter is via the chaining $\mathbf{X}_{ij}\mathbf{X}_{jk}$. Given no prior knowledge about the distribution of outlier mappings, we assume within outliers, it is a uniform prior $Prob(u^k{=}u^{k'}){=}\frac{1}{n}$. Then we find the element in $\mathbf{c}^d$ which is closest to $Prob(u^k{=}u^{k'})$ *i.e.* $i^*{=}\arg\min_i(abs(\mathbf{c}^d - \frac{1}{n}))$, and set $n_c = i^*$.

3.5. Retrospection on peer methods

First, our batch-wise formulation allows for integrating *all* graph information simultaneously during optimization. In contrast, the state-of-the-art methods [56, 57] use a base graph $\mathcal{G}_b$ which only encodes *a linear portion* of the total $\frac{N(N-1)}{2}$ number of pairwise affinities: $\{\mathbf{K}_{uf}\}_{f=1,\neq b,u}^{N}$ per iteration: $\max_{\mathbf{X}_{ub}} \text{vec}(\mathbf{X}_{ub})\mathbf{K}_{ub}\text{vec}(\mathbf{X}_{ub}) + \sum_{f=1,\neq b,u}^{N} \text{vec}(\mathbf{X}_{uf})\mathbf{K}_{uf}\text{vec}(\mathbf{X}_{uf})$. A similar mechanism is adopted in [47] where a virtual reference graph is created to proceed the optimization. Although all pairwise matching results obtained from independent pairwise matching solvers are considered simultaneously in [41, 7, 55], while [41, 7] are designed to capture the matching cycle-consistency but ignore the graph attribute or pairwise affinity information in their model and [55] approximately boosts the pairwise matching accuracy via a local intermediate graph by $\max_u \text{vec}(\mathbf{X}_{iu}\mathbf{X}_{uj})^\top \mathbf{K}_{ij}\text{vec}(\mathbf{X}_{iu}\mathbf{X}_{uj})$.

Second, our method is built on Robust PCA [6] which is supposed to offer some robustness by decoupling the noise from the weighted adjacency stacking matrix $\mathbf{D}$. In contrast, existing MGM methods [47, 56, 55] directly perform optimization on the corrupted objective function without separating the noises, which bear some limitation especially when the graph attributes are largely corrupted. This is because the objective function may deviate from the semantic matching accuracy and an effective algorithm aiming at achieving a higher objective score may adversely generate less accurate solutions. Moreover, our formulation can accommodate unequal size of graphs as long as they contain the same number of common inliers. Comparatively, the peer multi-graph methods [47, 56, 55, 41] and most existing two-graph methods [49, 65, 10, 29] use dummy nodes to make the considered graphs of equal size, and augment $\mathbf{X}$ to a square permutation matrix. The dummy nodes may bring about stability issues dependent on the problem and further increase the size of the graphs. Our model also can automatically and explicitly identify outliers and this advantage comes from an explicit usage of knowing the exact or estimated number of common inliers.

4. Experiments and Discussion

The experiments are performed on both synthetic and real-image data. The synthetic test is controlled by quantitatively varying the disturbance of deformation, outlier and edge density. The real-image datasets are tested with varying viewing angles, scales, shapes, and spurious outliers. The matching accuracy over all graphs, is calculated by averaging all pairwise matching accuracy $\frac{\sum_{i=1}^{N-1}\sum_{j=i+1}^{N} \text{Acc}_{ij}}{N(N-1)/2}$. Each Acc_{ij} computes the matches between the correspondence matrix $\mathbf{X}_{ij}^{alg}$ given by the ground truth $\mathbf{X}_{ij}^{tru}$: $\text{Acc}_{ij} = \frac{\text{tr}(\mathbf{X}_{ij}^{alg}\mathbf{X}_{ij}^{tru})}{\text{tr}(\mathbf{1}_{n2\times n1}\mathbf{X}_{ij}^{tru})}$. Note we only calculate the accuracy for common inliers and ignore the meaningless matchings over outliers. The above methodologies follow a standard protocol widely adopted by many related works such as [10, 49, 65].

4.1. Dataset description and settings

Several unified protocols are adopted in all tests: i) RRWM [10] is adopted as the standard pairwise solver to generate initial pairwise matchings which is a must for the comparing MGM methods [56, 41, 55, 7]. Its input is the pairwise affinity matrix $\{\mathbf{K}_{ij}\}_{i,j=1}^{N}$; ii) all edge attributes in the graph are fully sampled for building the stacking adjacency matrix $\mathbf{D}$ and pairwise affinity $\mathbf{K}$; iii) in order

[1] For discussion convenience, we assume the number of outliers are equal which can usually be easily realized in real applications. If this does not hold, we add additional random outliers to satisfy this condition. Thus each $\mathbf{P}_{ij}$ is a square permutation matrix without discrimination for outliers.

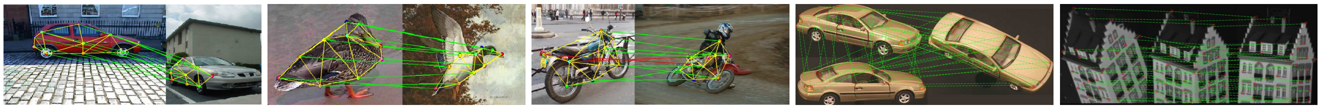


Figure 1: Left to right: car, duck, motorbike of Willow-ObjectClass; volvoC70 of Pose sequence; hotel of CMU sequence.

to differentiate graph matching from feature point matching, following a common setting [10, 49, 56, 55], only the second-order edge attributes are used thus the diagonal of **K** are all set zero; iv) given the edge attributes d_{ij} in $\mathcal{G}_1$ and d_{ab} in $\mathcal{G}_2$, the corresponding elements in $\mathbf{K}_{12}$ is set by $K_{12}(ij,ab)$=exp($\frac{(d_{ij}-d_{ab})^2}{-\sigma^2}$) ($\sigma^2$=0.15 throughout this paper), refer to [10, 49, 65] for more details on building **K**.

Synthetic data The synthetic test follows the widely used protocol [56, 21, 10, 49, 65, 48]. For each trial, a reference graph of n_{in} nodes is created by assigning a random weight to its edge, uniformly sampled from the interval $[0, 1]$. Then the 'perturbed' graphs are created by adding a Gaussian deformation to the edge weight d_{ij}^r, as sampled from $N(0, \varepsilon)$: $d_{ij}^p = d_{ij}^r$+$N(0, \varepsilon)$ where the superscript 'p' and 'r' denotes for 'perturb' and 'reference' respectively.

CMU-POSE Sequence This data contains three sequences. The first two are from the CMU house (30 landmarks, 101 frames), hotel (30 landmarks, 111 frames) sequences (http://vasc.ri.cmu.edu//idb/html/motion/) which are commonly used in [56, 10, 49, 5, 65, 9]. The third sequence is sampled from the sedan (VolvoC70) sequence (19 landmarks, 225 frames) which is viewed from various angles covering a range of 70 degrees from the POSE dataset [51]. For each test, an image sequence is sampled as spaced evenly by three frames. Specifically, we select n_{in}=10 landmarks out of all n_{ant} annotated points (n_{ant}=30 for the two CMU sequences, n_{ant}=19 for Pose sedan sequence), and randomly choose n_{out}=4 nodes from the rest $n_{ant}-n_{in}$ nodes as outliers. In line with [56, 55, 57], edge attributes are set by the Euclidian distance between two key points in the images, which are further normalized by the longest distance between two points in the considered image.

WILLOW-ObjectClass This dataset [9] is constructed by images from Caltech-256 and PASCAL VOC2007. We use three categories Duck (50 images), Motorbike and Car (both 40 images) where 10 features are manually labeled on the object for each image. No outlier is added for testing. Since this dataset is more ambiguous due to large deformation, we add edge orientation information in addition with the length used in the CMU-POSE data. The edge orientation is computed by the angle between the edge and the horizontal line in line with [65]. Thus the affinity matrix is $\mathbf{K} = \frac{1}{2}\mathbf{K}^o + \frac{1}{2}\mathbf{K}^l$. For our method, this is done by using Eq.16 and *cos* value of the orientation as adopted in [27, 14] to deal with the discontinuity brought by the radian measurement. One may further improve it by adding *sin* while we find *cos* is almost enough in our tests.

4.2. Results and discussion

The result on the above datasets are depicted in Fig.2. We evaluate Alg.1 and its variant as discussed in Sec.3.3 by replacing the ℓ_1 norm with ℓ_F in Eq.2, which is termed as Alg.1^2 by a superscript '2' in the legends for ℓ_F norm. Here we assume the number of common inliers n_c is known.

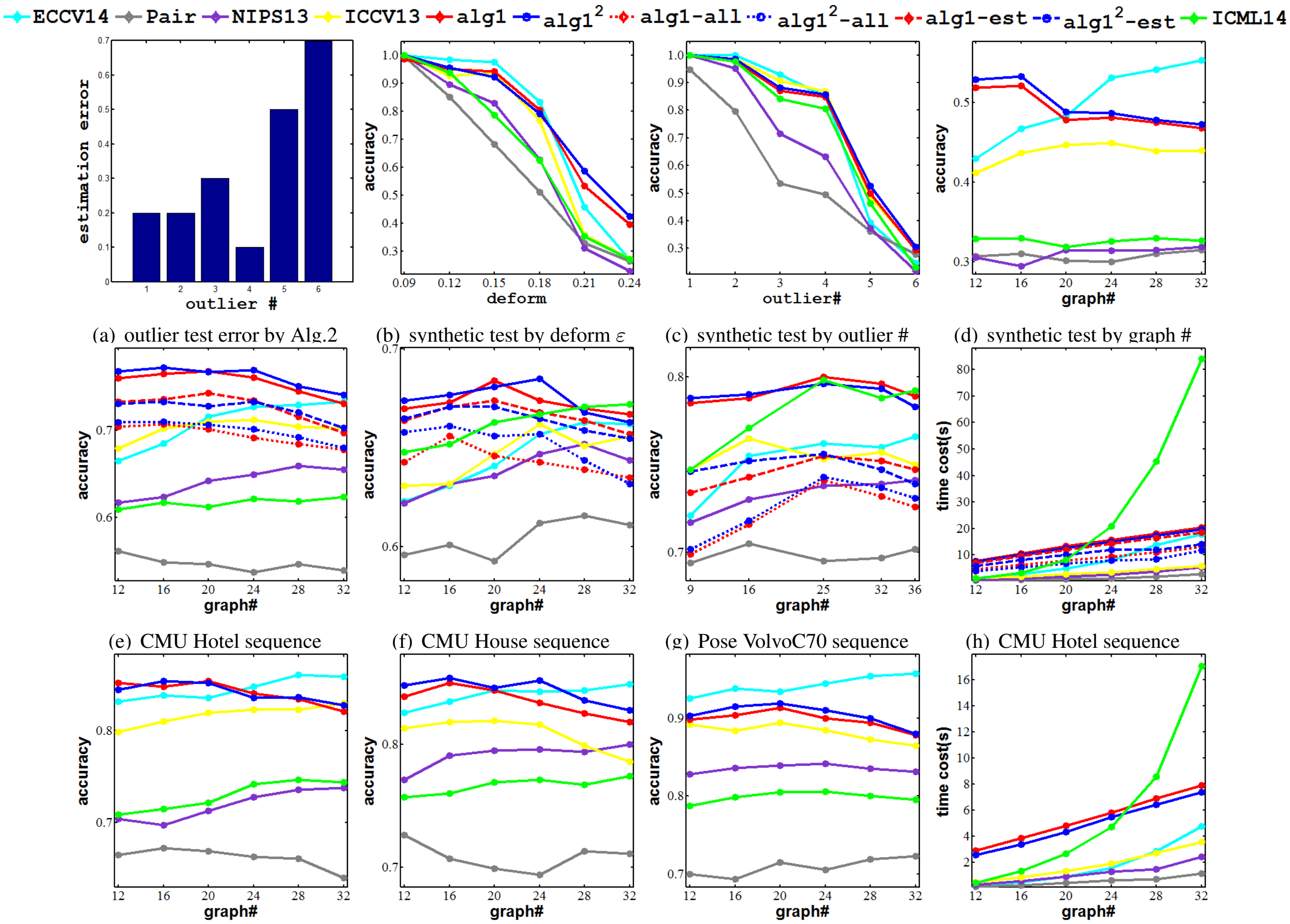
In case the input parameter n_c is unknown, we derive two variants by i) setting $n_c = n_i$ for all graphs thus all graphs are of equal size. The equal size setting is the same with other MGM methods [47, 56, 57, 55, 41] and can be achieved by adding dummy nodes or random outliers. The methods are termed as Alg.1-all and Alg.1^2-all for two norms where the post-fix 'all' denotes *all* points are considered in matrix **D**; ii) setting $n_c = n_{est}$ where n_{est} is the estimation by Alg.2. The methods are termed by Alg.1-est and Alg.1^2-est, where 'est' stands for 'estimated'. We set $\lambda = 1, \rho = 1.1, \mu_1^0 = \mu_2^0 = \rho/\|\mathbf{F}_1\|_2, T = 5$ in all tests.

Our methods are initialized trivially by performing two-graph matching using RRWM [10] between $\mathcal{G}_1$ (contains most points $n_1 \geq \{n_i\}_{i=2}^N$) and the rest graphs $\{\mathbf{P}_i^0 = \mathbf{P}_{i1}\}_{i=1}^N$. In case n_c is given either by ground truth or by estimation and therefore $n_c < n_i$, we simply keep the first n_c rows of each $\{\mathbf{P}_{i1}\}_{i=1}^N$ as the initial $\{\mathbf{P}_i^0\}_{i=1}^N$, which contains one and only one nonzero element in each of its rows.

Very recent MGM methods [55, 7, 56, 41] are evaluated whose parameters are mostly set as in the original papers. First we use Fig.2(a) to evaluate the efficacy of Alg.2 for inlier number estimation where the error denotes the averaged $|n_c - n_c^*|$ over the outlier test in Fig.2(c). We summarize several behaviors of Alg.1 and its variants:

i) **Robustness to deformation but not outliers** Our methods are more robust in the presence of large deformation without outliers as evidenced by Fig.2(b). And the ℓ_F norm version (in blue) seems even better which can be explained by the fact that the widely adopted (also this paper) deformation noises is Gaussian. We assume the robustness comes from the built-in matrix decomposition model dismissing large noises. However, when more outliers are added, the performance gain almost fades away as evidenced in Fig.2(c). We conjecture this is because the built-in RPCA model may treat the outliers as inliers corrupted by large deformation noises, as its tolerance to arbitrary deformation may adversely have the side-effect of losing discrimination to outliers and activate them in $\mathbf{X}_i$.

ii) **Scalablity to N but sacrificing the accuracy** The overhead of our method is linear to N as shown in Fig.2(h) and Fig.2(l). The main cost is solving Eq.8 and Eq.11, at the expense of $O(n^6)$ for the worst case when the interior point



(a) outlier test error by Alg.2 (b) synthetic test by deform ϵ (c) synthetic test by outlier # (d) synthetic test by graph #

(e) CMU Hotel sequence (f) CMU House sequence (g) Pose VolvoC70 sequence (h) CMU Hotel sequence

(i) Willow-ObjectClass car (j) Willow-ObjectClass duck (k) Willow-ObjectClass bike (l) Willow-ObjectClass bike

Figure 2: Results on benchmarks. We set σ^2 as .15, 0.1 and .05 for synthetic, Willow-ObjectClass and CMU-POSE test. In Fig.2(b), $N = 20, n_{in} = 10, n_{out} = 0$; In Fig.2(c), $N = 20, n_{in} = 10, \epsilon = .06$; In Fig.2(d), $n_{in} = 10, n_{out} = 0, \epsilon = 0.2$.

method is performed. The overhead is $O(Nn^6)$ since each $\mathbf{X}_i$ is computed separately – see complexity in Table 1 for comparison. Though showing superior performance when $N < 20$ perhaps due to the use of matrix $\mathbf{D}$ other than distributed pairwise term $\mathbf{K}_{ij}$, yet the accuracy decreases as N grows especially evidenced by Fig.2(g) and Fig.2(i) when $N > 20$. We conjecture this is due to more variables are involved bringing optimization challenges. Similar problem happens on the solver [56] which adopts a coordinate ascent optimization based on a set of base variables. The peer solvers [55, 56] in theory is not linearly scalable to N as their complexity is $O(N^3)$ or $O(N^2)$ as shown in Table 1.

iii) **Knowing n_c improves the performance** As evidenced in Fig.2(e), 2(f), 2(g), Alg.1, Alg.1^2 exactly knowing n_c (in solid curve) perform best. The two approximate methods (Alg.1-est, Alg.1^2-est in dashed curve) are in the between. Note even trivially letting $n_c = n_i$ as the setting all graphs are of equal size in which most extant approaches are based on, Alg.1-all, Alg.1^2-all (dotted curve) still perform competitively, as shown in the the second row of Fig.2.

Table 1: Complexity comparison: τ_{raw} is the overhead for initial pairwise matching, and n is the number of nodes.

method	time complexity	iteration round
ECCV14 [55]	$O(N^3n^2) + N^2\tau_{\text{raw}}$	few
ICML14 [7]	$O(n^3N^3)$	moderate
ICCV13 [56, 57]	$O(N^2n^4 + N^2\tau_{\text{raw}})$	few
NIPS13 [41]	$O(N^2n^3 + N^2\tau_{\text{raw}})$	one-shot
Our method	$O(Nn^6)$	moderate

5. Conclusion and Future Work

This paper is an endeavor of taking a matrix decomposition perspective to the multi-graph matching problem, which allows for reusing algorithms originally tailored to matrix decomposition. Our model differs from peer methods by discarding the pairwise affinity terms and pairwise matchings representation. Observed interesting behaviors show the potential of this framework.

Acknowledgement The work is supported by NSF DMS-1317424, NSFC (61527804, 61129001, 61221001), STCSM (15JC1401700, 14XD1402100, 13511504501) and the 111 Program (B07022).

References

[1] A. V. Aho, M. Ganapathi, and S. W. Tjiang. Code generation using tree matching and dynamic programming. *TOPLAS*, 1989.

[2] H. A. Almohamad and S. O. Duffuaa. A linear programming approach for the weighted graph matching problem. *TPAMI*, 1993.

[3] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Ecjstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 2010.

[4] T. Caelli and S. Kosinov. An eigenspace projection clustering method for inexact graph matching. *TPAMI*, 2004.

[5] T. Caetano, J. McAuley, L. Cheng, Q. Le, and A. J. Smola. Learning graph matching. *TPAMI*, 31(6):1048–1058, 2009.

[6] E. Candes, X. Li, Y. Ma, and J. Wright. Robust principal component analysis? *Journal of the ACM*, 2011.

[7] Y. Chen, G. Leonidas, and Q. Huang. Matching partially similar objects via matrix completion. In *ICML*, 2014.

[8] M. Chertok and Y. Keller. Efficient high order matching. *TPAMI*, 2010.

[9] M. Cho, K. Alahari, and J. Ponce. Learning graphs to match. In *ICCV*, 2013.

[10] M. Cho, J. Lee, and K. M. Lee. Reweighted random walks for graph matching. In *ECCV*, 2010.

[11] M. Cho and K. M. Lee. Progressive graph matching: Making a move of graphs via probabilistic voting. In *CVPR*, 2012.

[12] T. Collins, P. Mesejo, and A. Bartoli. An analysis of errors in graph-based keypoint matching and proposed solutions. In *ECCV*, 2014.

[13] D. Conte, P. Foggia, C. Sansone, and M. Vento. Thirty years of graph matching in pattern recognition. *IJPRAI*, 18(03):265–298, 2004.

[14] O. Duchenne, F. Bach, I. Kweon, and J. Ponce. A tensor-based algorithm for high-order graph matching. In *CVPR*, 2009.

[15] O. Duchenne, A. Joulin, and J. Ponce. A graph-matching kernel for object categorization. In *ICCV*, 2011.

[16] A. Egozi, Y. Keller, and H. Guterman. A probabilistic approach to spectral graph matching. *TPAMI*, pages 18–27, 2013.

[17] D. Eppstein. Subgraph isomorphism in planar graphs and related problems. In *SODA*, 1995.

[18] M. A. Fischler and R. C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 1981.

[19] P. Foggia, G. Percannella, and M. Vento. Graph matching and learning in pattern recognition in the last 10 years. *IJPRAI*, 2014.

[20] T. Funkhouser, M. Kazhdan, P. Shilane, P. Min, W. Kiefer, A. Tal, and D. Dobkin. Modeling by example. *TOG*, 2004.

[21] S. Gold and A. Rangarajan. A graduated assignment algorithm for graph matching. *TPAMI*, 1996.

[22] S. Gold and A. Rangarajan. Softmax to softassign: neural network algorithms for combinatorial opti-mization. *J. Artif. Neural Netw*, 1996.

[23] Q. Huang, G. Zhang, L. Gao, S. Hu, A. Butscher, and L. Guibas. An optimization approach for extracting and encoding consistent maps in a shape collection. *ACM Transactions on Graphics (TOG)*, 2012.

[24] K. Jia, T. H. Chan, Z. Zeng, G. Wang, T. Zhang, and Y. Ma. Roml: a robust feature correspondence approach for matching objects in a set of images. *arXiv:1403.7877*, 2014.

[25] T. C. Koopmans and M. Beckmann. Assignment problems and the location of economic activities. *Econometrica*, pages 53–76, 1957.

[26] E. Lawler. The quadratic assignment problem. *Management Science*, 1963.

[27] J. Lee, M. Cho, and K. M. Lee. Hyper-graph matching via reweighted randomwalks. In *CVPR*, 2011.

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

[29] M. Leordeanu, M. Hebert, and R. Sukthankar. An integer projected fixed point method for graph matching and map inference. In *NIPS*, 2009.

[30] M. Leordeanu, R. Sukthankar, and M. Hebert. Unsupervised learning for graph matching. *Int. J. Comput. Vis.*, pages 28–45, 2012.

[31] M. Leordeanu, A. Zanfir, and C. Sminchisescu. Semi-supervised learning and optimization for hypergraph matching. In *ICCV*, 2011.

[32] Y. Li, Z. Y. Yan, J., and J. Yang. Optimum subspace learning and error correction for tensors. In *ECCV*, 2010.

[33] Z. Lin, M. Chen, and Y. Ma. The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices. *arXiv preprint arXiv:1009.5055*, 2010.

[34] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yong, and Y. Ma. Robust recovery of subspace structures by low-rank representation. *TPAMI*, 2013.

[35] J. Liu, S. Ji, and J. Ye. Slep: Sparse learning with efficient projections. *Arizona State University*, 6, 2009.

[36] L. Liu, W. Lin, and Y. Zhong. Traffic flow matching with clique and triplet cues. In *IEEE International Workshop on Multimedia Signal Processing*, 2015.

[37] S. Liu, X. Wang, J. Chen, and et al. Topicpanorama: a full picture of relevant topics. *VCG*, 2014.

[38] Z. Liu, H. Qiao, and L. Xu. An extended path following algorithm for graph-matching problem. *TPAMI*, pages 1451–1456, 2012.

[39] E. M. Luks. Isomorphism of graphs of bounded valence can be tested in polynomial time. *Journal of Computer and System Sciences*, 25(1), 1982.

[40] Y. Mi, L. Liu, W. Lin, and W. Wangg. Extracting recurrent motion flows from crowded scene videos: A coherent motion-based approach. In *Intl. Conf. Multimedia Big Data workshop*, 2015.

[41] D. Pachauri, R. Kondor, and S. Vikas. Solving the multi-way matching problem by permutation synchronization. In *NIPS*, 2013.

[42] G. A. Peng, Y.G. Rasl: Robust alignment by sparse and low-rank decomposition for linearly correlated images. In *CVPR*, 2010.

[43] G. Scott and H. Longuet-Higgins. An algorithm for associating the features of two images. *Biological Sciences*, pages 21–26, 1991.

[44] L. S. Shapiro and M. Brady. Feature-based correspondence: an eigenvector approach. *Image Vision Comput.*, pages 283–288, 1992.

[45] Y. Shen, W. Lin, J. Yan, M. Xu, J. Wu, and J. Wang. Person re-identification with correspondence structure learning. In *ICCV*, 2015.

[46] A. Solé-Ribalta and F. Serratosa. Models and algorithms for computing the common labelling of a set of attributed graphs. *CVIU*, 2011.

[47] A. Sole-Ribalta and F. Serratosa. Graduated assignment algorithm for multiple graph matching based on a common labeling. *IJPRAI*, 2013.

[48] P. S. T. Cour and J. Shi. Balanced graph matching. In *NIPS*, 2006.

[49] Y. Tian, J. Yan, H. Zhang, Y. Zhang, X. Yang, and H. Zha. On the convergence of graph matching: Graduated assignment revisited. In *ECCV*, 2012.

[50] S. Umeyama. An eigendecomposition approach to weighted graph matching problems. *TPAMI*, 1988.

[51] F. Viksten, P. Forssén, B. Johansson, and A. Moe. Comparison of local image descriptors for full 6 degree-of-freedom pose estimation. In *ICRA*, 2009.

[52] M. L. Williams, R. C. Wilson, and E. Hancock. Multiple graph matching with bayesian inference. *Pattern Recognition Letters*, pages 1275–1281, 1997.

[53] J. Xiao, H. Cheng, H. Sawhney, and F. Han. Vehicle detection and tracking in wide field-of-view aerial video. In *CVPR*, 2010.

[54] J. Yan, M. Cho, H. Zha, X. Yang, and S. Chu. Multi-graph matching via affinity optimization with graduated consistency regularization. *TPAMI*, 2016.

[55] J. Yan, Y. Li, W. Liu, H. Zha, X. Yang, and S. Chu. Graduated consistency-regularized optimization for multi-graph matching. In *ECCV*, 2014.

[56] J. Yan, Y. Tian, H. Zha, X. Yang, Y. Zhang, and S. Chu. Joint optimization for consistent multiple graph matching. In *ICCV*, 2013.

[57] J. Yan, J. Wang, H. Zha, and X. Yang. Consistency-driven alternating optimization for multigraph matching: A unified approach. *TIP*, 2015.

[58] J. Yan, J. Wang, H. Zha, X. Yang, and S. M. Chu. Multi-view point registration via alternating optimization. In *AAAI*, 2015.

[59] J. Yan, C. Zhang, H. Zha, W. Liu, X. Yang, and S. M. Chu. Discrete hypergraph matching. In *CVPR*, 2015.

[60] J. Yan, M. Zhu, H. Liu, and Y. Liu. Visual saliency detection via sparsity pursuit. *Signal Processing Letters, IEEE*, 2010.

[61] M. Zaslavskiy, F. R. Bach, and J.-P. Vert. A path following algorithm for the graph matching problem. *TPAMI*, 2009.

[62] Z. Zeng, T. H. Chan, K. Jia, and D. Xu. Finding correspondence from multiple images via sparse and low-rank decomposition. In *ECCV*, 2012.

[63] Z. Zeng, S. Xiao, K. Jia, T. Chan, S. Gao, D. Xu, and Y. Ma. Learning by associating ambiguously labeled images. In *CVPR*, 2013.

[64] Z. Zhang. Iterative point matching for registration of free-form curves and surfaces. *IJCV*, 1994.

[65] F. Zhou and F. D. Torre. Factorized graph matching. In *CVPR*, 2012.

[66] X. Zhou and M. Z. amd Kostas Daniilidis. Multi-image matching via fast alternating minimization. In *arXiv:1505.04845v1*, 2015.