Toward More Robust Graph Matching: Models and Algorithms

Junchi Yan
Department of Computer Science and Engineering
MoE Key Lab of Artificial Intelligence
Shanghai Jiao Tong University
yanjunchi@sjtu.edu.cn

Abstract

Graph matching refers to finding vertex correspondence among two or multiple graphs, being fundamental in many applications such as image registration, DNA alignment, and automatic software bug finding. In contrast to classic two-graph setting, recently matching multiple graphs emerge for their practical usefulness and methodological potential. Starting by a brief introduction for traditional two-graph matching, we walk through the recent development of multiple graph matching methods, including details for both models and algorithms. We show how learning can be inter-played with graph matching. The hope is to prompt the up-to-date advance to readers in a concrete way. Directions for future work are also discussed.

1 Introduction

Compared with vector like data, graph is a versatile representation for many real-world objects whereby relational structures are involved. One fundamental problem for processing graph data is graph matching (GM), which involves finding common vertex correspondences over a pair or more graphs. Departure from point matching that only considers unary (vertex) feature similarity, graph matching establishes the vertex correspondence by considering both unary and higher-order namely hyperedge (typically second-order i.e. edge) information. Due to the introduction of higher-order information, graph matching becomes generally NP-hard in contrast to point matching which can be solved with global optimum using the Hungarian method [Munkres, 1957] in polynomial time regarding with the number of vertex.

For its robustness, graph matching has been widely used in the analysis of image [Duchenne *et al.*, 2011b], graphics [Kim *et al.*, 2012], and genome [Zaslavskiy *et al.*, 2009] etc. Techniques have also been devised for cross social network matching [Cao *et al.*, 2018]. In all these domains, graph matching serves as a building block for downstream applications such as image recognition, graphics clustering, whereby the similarity based on aligned structure is utilized.

Over the decades, a large body of works has been developed on graph matching. Due to its NP-hardness in nature, most methods seek different approximation techniques to pursuit the trade-off between accuracy and efficiency. Traditionally, methods are mostly focused on pairwise graph matching i.e. each time only two graphs are involved for matching. Recently, an increasing line of methods emerge for matching multiple graphs jointly. In our analysis, it not only opens up new space for advanced learning of graph matching, there are also imperatives for developing new methods in this setting:

1) From the methodology perspective, the availability to multiple graphs sharing common or similar structure provides a way of facilitating the disambiguation for local matching by fusing global observations. This is especially helpful when there exists local noise on a few graphs rendering local matching among these graphs inherently ambiguous.

2) From the application perspective, in real world applications one is often given a set of graphs, for instance a collection of images, or a set of DNA. Hence it is unnecessary and even harmful to perform two-graph matching pair by pair, and joint matching becomes a natural choice.

As a long-standing problem over decades, graph matching has attracted a number of literature reviews [Bunke, 2000; Conte *et al.*, 2004; Foggia *et al.*, 2014; Vento, 2015; Yan *et al.*, 2016c]. These articles cover different subareas of graph matching, as well as techniques loosely related to graph matching such as graph kernel and embedding [Livi and Rizzi, 2013], graph isomorphism and graph-subgraph isomorphism [Santo *et al.*, 2003], graph edit distance [Gao *et al.*, 2010]. However, multi-graph matching approaches are hardly presented in the above surveys (partly due to only until recently is there an emerging line of work [Solé-Ribalta and Serratosa, 2011; Huang *et al.*, 2012; Pachauri *et al.*, 2013; Yan *et al.*, 2013; Solé-Ribalta and Serratosa, 2013; Yan *et al.*, 2014; Chen *et al.*, 2014; Shi *et al.*, 2016]). One exception is the very recent study [Yan *et al.*, 2016c] in which multi-graph matching is briefly discussed in Section 3.3 without mathematical details (taking up less than a half page). In contrast, this article is motivated for a more comprehensible and dedicated description, helping readers not only learn the basic background but also specific models and algorithms. Perhaps more importantly, via the detailed disclosure of related works, one can better understand the trend for future work.

In this article, we start with a brief introduction on traditional pairwise graph matching methods, including affinity function, relaxation techniques, and solvers, which lay the foundation for the advance of multi-graph matching. Then

we shift to the recent advance for multi-graph matching. Finally potential directions for this emerging area is discussed.

2 Pairwise Matching for Two Graphs

Two-graph matching can be modeled as the quadratic assignment problem (QAP), which is known NP-complete. In particular, as graph matching refers to a constrained optimization problem in discrete domain, the development of two-graph matching can be viewed by two orthogonal facets: i) how the optimization objective is modeled; ii) how the optimization procedure is devised. Specifically the first problem involves the modeling of affinity function which measures the similarity between vertices and edges. While the second refers to algorithmic solvers and many of the methods resort to different relaxation techniques to relax the combinatorial constraint to continuous one, typically followed by a post step converting the solution back to the discrete domain.

2.1 Objective function

Graph matching aims to establishing the vertex correspondence such that the aligned two graphs achieve a maximum affinity value (namely the objective function) which sums up all the vertex-to-vertex and edge-to-edge affinity values. When only first and second order edge information is considered, one can use an affinity matrix $\mathbf{K} \in \mathbb{R}^{n_1 n_2 \times n_1 n_2}$ to encode the affinity between graph $\mathcal{G}_1$ ($|\mathcal{G}_1| = n_1$) and $\mathcal{G}_2$ ($|\mathcal{G}_2| = n_2$), whereby the diagonal elements store the vertex-to-vertex similarity while off-diagonal carry the edge-to-edge similarity. By letting $\mathbf{X} \in \{0,1\}^{n_1 \times n_2}$ denote the binary correspondence matrix (namely assignment matrix), the quadratic assignment problem can be written by:

$$J(\mathbf{x}) = \mathbf{x}^\top \mathbf{K} \mathbf{x} \tag{1}$$
$$\mathbf{X}\mathbf{1}_{n_2} = \mathbf{1}_{n_1}, \mathbf{X}^\top \mathbf{1}_{n_1} \leq \mathbf{1}_{n_2}, \mathbf{X} \in \{0,1\}^{n_1 \times n_2}$$

where $\mathbf{x} = \mathrm{vec}(\mathbf{X})$ is the column-wise vectorized version of matrix $\mathbf{X}$. The constraints refer to the fact that each vertex in $\mathcal{G}_1$ shall find their one-to-one correspondence in $\mathcal{G}_2$ i.e. there is no outlier in $\mathcal{G}_1$. This QAP form is popular in literature [Leordeanu and Hebert, 2005; Leordeanu *et al.*, 2009; Cho *et al.*, 2010; Leordeanu *et al.*, 2012], being a more compact writing equivalent to [Gold and Rangarajan, 1996].

Beyond the second-order affinity with the form of matrix, higher-order similarity can be encoded by tensor representation, whereby three or more edges are grouped as a tuple and the tensor element stores the similarity between each pair of tuples. The widely used tensor based objective can be written by [Lee *et al.*, 2011] (constraints are omitted for briefly):

$$J(\mathbf{x}) = \mathbf{H} \otimes_1 \mathbf{x} \cdots \otimes_p \mathbf{x} \tag{2}$$

where $\mathbf{H}$ is the affinity tensor and p is the order of the affinity. Termed as hypergraph matching, various higher-order methods [Zass and Shashua, 2008; Chertok and Keller, 2010; Duchenne *et al.*, 2011a; Yan *et al.*, 2015c; Ngoc *et al.*, 2015] have been proposed for improved matching accuracy at the cost of increased time and space complexity. The common strategy is to transform the higher-order problem into the second-order case in an iterative fashion.

In fact, most graph matching work set the affinity matrix/tensor by a predefined parametric function. For instance, the Gaussian kernel is widely used e.g. in [Cho *et al.*, 2010]:

$$K_{ia,jb}(\sigma) = \exp\left((d_{ij} - d_{ab})^2 / -\sigma^2\right) \tag{3}$$

where d_{ij}, d_{ab} can be the Euclidean distances between two points normalized to $[0,1]$ by dividing the largest edge length. The diagonal elements can be measured by the distance between two vertices. In computer vision, typical embodiments include the distance between two feature descriptors such as SIFT and deep CNN features. Note here the parameter σ is a constant and given in advance.

From Eq. 3, one can find the affinity function is parameterized by the manually set parameter σ. Departure from the way of predefining affinity parameters, recent studies also aim to learn the parameters in either unsupervised or supervised way [Leordeanu *et al.*, 2012; Caetano *et al.*, 2009; Leordeanu *et al.*, 2011]. In particular, [Cho *et al.*, 2013] learns the graph structure model for matching.

2.2 Optimization algorithms

Continuous relaxation based method

To circumvent the inherent difficulty from the combinatorial nature, many relaxation techniques have been devised over the years. In this article, we list three popular categories:

i) spectral relaxations on the matching matrix [Leordeanu and Hebert, 2005; Cour *et al.*, 2006]. The matching constraint is loosened by $\|\mathbf{x}\|_2 = 1$ which can be solved efficiently;

ii) doubly-stochastic relaxation on the matching matrix [Gold and Rangarajan, 1996; Leordeanu and Hebert, 2005; Leordeanu *et al.*, 2009]: according to Eq. 1, the solution domain is relaxed to the continuous space:

$$\mathbf{X}\mathbf{1}_{n_2} = \mathbf{1}_{n_1}, \mathbf{X}^\top \mathbf{1}_{n_1} \leq \mathbf{1}_{n_2}, \mathbf{X} \in [0,1]^{n_1 \times n_2} \tag{4}$$

iii) semidefinite-programming (SDP) [Torr, 2003; Schellewald and Schnörr, 2005]. The relaxation model for optimization in) [Schellewald and Schnörr, 2005] can be written as:

$$\min_{\mathbf{Y}} \mathrm{Tr}(\mathbf{Q}\mathbf{Y}) \quad \text{s.t.} \quad \mathbf{Y} \succeq \mathbf{0}, \quad \mathrm{Tr}(\mathbf{A}_i \mathbf{Y}) = \mathbf{c}_i \tag{5}$$

where constraints are defined by s series of $\mathbf{A}_i$ and $\mathbf{c}_i$. There is off-the-shelf solver for such an SDP problem, while the derived high dimensional variable $\mathbf{Y} = \begin{pmatrix} 1 & \mathbf{x}^\top \\ \mathbf{x} & \mathbf{x}\mathbf{x}^\top \end{pmatrix} \in \mathbb{R}^{(n_1 n_2 + 1) \times (n_1 n_2 + 1)}$ causes scalability issue.

Discrete method

Different from the above methods working in the continuous domain, and involves a post binarization, several methods tend to directly compute the solution in discrete assignment space. The method Integer Projected Fixed Point (IPFP) [Leordeanu *et al.*, 2009] is devised with the hope that an optimal solution can be found along a (quasi) discrete course. Sampling based methods [Lee *et al.*, 2010; Suh *et al.*, 2012] directly generate discrete solutions via Monte Carlo Sampling. More recently, [Adamczewski *et al.*, 2015] devises a tailored Tabu search for graph matching.

Path-following paradigm
From the optimization perspective, the continuation method (i.e. path-following) is widely adopted. In [Gold and Rangarajan, 1996], a particular deterministic annealing procedure is performed in the continuous space. Similar path-following techniques are widely used in recent work [Zaslavskiy *et al.*, 2009; Liu *et al.*, 2012; Zhou and Torre, 2016].

3 Joint Matching for Multiple Graphs

Many existing multi-graph matching methods are built on the concept of consistency. In general it refers to the fact that the bijection correspondence between graph $\mathcal{G}_1$ and $\mathcal{G}_2$ shall be consistent with an derived one through an intermediate graph $\mathcal{G}_3$: namely $\mathbf{X}_{12} = \mathbf{X}_{13}\mathbf{X}_{32}$. It is also termed as cycle consistency in literature. Obviously consistency is the necessary condition for ideal ground truth bijections among multiple graphs. Hence the consistency violation extent can partially reflect the matching accuracy. As a matter of fact, affinity function may not always increase with the matching accuracy and even the global maximum of affinity score may not correspond to the ground truth matchings. This is due to: i) the ubiquitous existence of noise and ii) the inherent approximation nature for the parametric affinity function modeling to real-world graph-like data. In this regard, supplementary to affinity objective, consistency serves as an important additional estimation to matching accuracy.

We divide recent multi-graph matching methods into three groups. Methods in the first group iteratively transform the multi-graph matching problem into a pairwise matching task at each iteration. Hence off-the-shelf two-graph matching solvers can be readily reused. While the second group involves methods that take the initial (noisy) pairwise matching result as input, and try to recover a cycle-consistent solution by post-processing. The methods of last group take a clustering or low rank recovery perspective in feature space.

3.1 Iterative pairwise matching based methods

Methods in this category often explicitly define the consistency measurement. Here we first review two formal definitions appearing in [Yan *et al.*, 2015a; 2016a], while the similar ideas have been widely accepted in literature.

Definition 1. *For N graphs $\{\mathcal{G}_k\}_{k=1}^N$ and a set of pairwise matching solutions $\mathbb{X} = \{\mathbf{X}_{ij}\}_{i=1,j=i+1}^{N-1,N}$, the unary consistency $C_u(k,\mathbb{X}) \in (0,1]$ for graph $\mathcal{G}_k$ is defined by:*

$$C_u(k,\mathbb{X}) = 1 - \frac{\sum_{i=1}^{N-1}\sum_{j=i+1}^{N}\|\mathbf{X}_{ij} - \mathbf{X}_{ik}\mathbf{X}_{kj}\|_F/2}{nN(N-1)/2} \quad (6)$$

where $\|\|_F$ is the Frobenius norm.

Definition 2. *For N graphs $\{\mathcal{G}_k\}_{k=1}^N$ and a set of matching solutions $\mathbb{X}$, for any pair of graph $\mathcal{G}_i$ and $\mathcal{G}_j$, the pairwise consistency $C_p(\mathbf{X}_{ij},\mathbb{X}) \in (0,1]$ is defined by:*

$$C_p(\mathbf{X}_{ij},\mathbb{X}) = 1 - \frac{\sum_{k=1}^{N}\|\mathbf{X}_{ij} - \mathbf{X}_{ik}\mathbf{X}_{kj}\|_F/2}{nN} \quad (7)$$

Based on the above two definitions, we introduce two representative works which iteratively update pairwise matching. These notations will also be used in the rest of this paper.

We discuss two complementary methods from [Yan *et al.*, 2015a] and [Yan *et al.*, 2016a], which are state-of-the-arts in this line of work [Solé-Ribalta and Serratosa, 2011; 2013].

Example I: star-shape centralized methods
The first work is firstly proposed in [Yan *et al.*, 2013] which is further extended to [Yan *et al.*, 2015a]. For N graphs the multi-graph matching objective function is written by:

$$\mathbb{X}^* = \arg\max_{\mathbb{X}} \sum_{i,j=1,i\neq j}^{N} \mathbf{x}_{ij}^\top \mathbf{K}_{ij}\mathbf{x}_{ij} \quad (8)$$

$$\text{s. t.} \quad \mathbf{X}_{ij}\mathbf{1}_n = \mathbf{1}_n \quad \mathbf{1}_n^\top\mathbf{X}_{ij} = \mathbf{1}_n^\top \quad \mathbf{X}_{ij} = \mathbf{X}_{ji}^\top \in \{0,1\}^{n\times n}$$

Seeing the redundancy of exhaustively involving all pairwise matching solutions, one can introduce a series of basis solutions $\{\mathbf{X}_{rk}\}_{k=1,k\neq r}^N$ induced by a pre-selected reference graph $\mathcal{G}_r$. Then the other pairwise matchings can be derived by $\mathbf{X}_{ij} = \mathbf{X}_{ir}\mathbf{X}_{rj}$ and its vectorized form can be written by

$$\mathbf{x}_{uf} = \mathbf{F}\mathbf{x}_{ur}, \quad \mathbf{F} = \mathbf{X}_{fr} \otimes \mathbf{I} \quad (9)$$

The authors in [Yan *et al.*, 2013] devise an alternating optimization procedure and at each iteration $\mathbf{x}_{ur}$ is to be solved by fixing the other basis solutions:

$$J(\mathbf{x}_{ur}) = \mathbf{x}_{ur}^\top\mathbf{K}_{ur}\mathbf{x}_{ur} + \sum_{f=1,f\neq r,u}^{N} \mathbf{x}_{uf}^\top\mathbf{K}_{uf}\mathbf{x}_{uf} \quad (10)$$

According to Eq. 9, it can also be written in QAP form:

$$J(\mathbf{x}_{ur}) = \mathbf{x}_{ur}^\top\left(\mathbf{K}_{ur} + \sum_{f=1,f\neq r,u}^{N} \mathbf{F}_{fr}^\top\mathbf{K}_{uf}\mathbf{F}_{fr}\right)\mathbf{x}_{ur} \quad (11)$$

Off-the-shelf two-graph matching solvers are mostly based on the QAP form, and can be applied directly. As a result, given the reference graph $\mathcal{G}_r$, the algorithm alternatively chooses the index graph $\mathcal{G}_u$ for updating $\mathbf{x}_{ur}$ until the iteration converges or exceeds to the maximum round.

Seeing the reference graph and alternating optimization order are both randomly set in [Yan *et al.*, 2013], the extended work [Yan *et al.*, 2015a] aims to adaptively find the appropriate reference graph $\mathcal{G}_r$ and alternating order. Specifically, $\mathcal{G}_r$ is found by maximizing the unary consistency $C_u(k,\mathbb{X})$ defined in Eq. 6 while the alternating updating order is according to the pairwise consistency score $C_p(\mathbf{X}_{ij},\mathbb{X})$ defined in Eq. 7: more inconsistent matchings shall have higher priority for updating to avoid error accumulation.

Such a star-shape centralized framework inherently suffers from fragility as the information flow are all through the central reference graph which can become the bottleneck. The underlying assumption is that $\mathcal{G}_r$ shall be similar or easier to match with other graphs, which can not hold in general.

Example II: distributed composition methods
To mitigate the above issue, a distributed multi-graph matching framework is proposed in [Yan *et al.*, 2014]. Similar to the star-shape centralized framework, a matching will be updated iteratively that involves an intermediate graph. The difference

is that there is no centralized reference graph, and the intermediate graph is distributed among every graph in the set. Moreover, the consistency constraint in Eq. 9 is relaxed by adding a consistency regularizer in the objective along with affinity score, to allow for more flexible search of matchings.

In the extended article [Yan *et al.*, 2016a] the method iteratively updates the pairwise matching by maximizing a weighted objective including both affinity and consistency:

$$k^* = \arg\max_{k=1}^{N}(1-\lambda)J(\mathbf{X}_{ik}\mathbf{X}_{kj}) + \lambda C_p(\mathbf{X}_{ik}\mathbf{X}_{kj}, \mathbb{X}) \quad (12)$$

where $J(\mathbf{X}) = \text{vec}(\mathbf{X})^T\mathbf{K}\text{vec}(\mathbf{X})$ is the affinity function for pairwise matching. This compositional approach encourages the estimation of matchings between dissimilar graphs by composition of matchings along a path of similar graphs.

Efficient variants can be devised. For instance by replacing the pairwise consistency term with the unary one, one can obtain (see Eq. 9 in [Yan *et al.*, 2016a]):

$$k^* = \arg\max_{k=1}^{N}(1-\lambda)J(\mathbf{X}_{ik}\mathbf{X}_{kj}) + \lambda C_u(k, \mathbb{X}) \quad (13)$$

The computational advantage is all $\{C_u(k, \mathbb{X}^{(t-1)})\}_{k=1}^N$ need be pre-computed at each iteration only once.

Note that the above approach does not involve any particular QAP solver, instead the pairwise matching is computed by a composition based trial to generate a candidate solution. This character enables a simple while effective outlier-resistant approach as also presented in [Yan *et al.*, 2016a].

We first introduce the two definitions about node-wise consistency and affinity score as defined in [Yan *et al.*, 2016a].

Definition 3. *Given $\{\mathcal{G}_k\}_{k=1}^N$ and $\mathbb{X}$, for node $\{\mathcal{N}_{u^k}\}_{u^k=1}^n$ in graph $\mathcal{G}_k$, its consistency w.r.t. $\mathbb{X}$ is defined by $C_n(u^k, \mathbb{X}) = 1 - \frac{\sum_{i=1}^{N-1}\sum_{j=i+1}^{N}\|\mathbf{Y}(u^k,:)\|_F/2}{N(N-1)/2} \in (0,1]$ where $\mathbf{Y} = \mathbf{X}_{kj} - \mathbf{X}_{ki}\mathbf{X}_{ij}$ and $\boldsymbol{Y}(u^k,:)$ is the u^kth row of matrix $\mathbf{Y}$.*

Definition 4. *Given $\{\mathcal{G}_k\}_{k=1}^N$, $\mathbb{X}$, $\mathbb{K}$, for node $\{\mathcal{N}_{u^k}\}_{u^k=1}^n$ in $\mathcal{G}_k$, its affinity w.r.t. $\mathbb{X}$ and $\mathbb{K}$ is defined by $S_n(u^k, \mathbb{X}, \mathbb{K}) = \sum_{i=1,\neq k}^{N} vec(\mathbf{X}_{ki}^{u^k})^T\mathbf{K}_{ki}vec(\mathbf{X}_{ki})$, where $\mathbf{X}^{u^k}$ denotes the matrix $\mathbf{X}$ with zeros except for the u^k-th rows as is.*

Assume the number of common inliers $\overline{n} \leq n$ is given, then the bijection solution $\mathbf{X} \in \{0,1\}^{n\times n}$ can be processed to zeroing out those outlier related rows/columns, resulting a new assignment matrix containing $\overline{n}$ one elements which is denoted by $\psi(\mathbf{X})$. To identify the $\overline{n}$ unknown outliers, one can either use the node-wise consistency or node-wise affinity as defined above to zeroing out suspect outliers: the top $\overline{n}$ node with highest node-wise consistency (affinity) score is maintained by the notation $\psi_c(\mathbf{X}, \mathbb{X}, \overline{n})$ ($\psi_a(\mathbf{X}, \mathbb{X}, \mathbb{K}, \overline{n})$).

Then the affinity score is rewritten as:

$$J^{\psi_a}(\mathbf{X}) = \text{vec}\left(\psi_a(\mathbf{X}, \mathbb{X}, \overline{n})\right)^T\mathbf{K}\text{vec}(\psi_a(\mathbf{X}, \mathbb{X}, \overline{n})) \quad (14)$$

The consistency terms are accordingly modified as follows:

$$C_u^{\psi_c}(k, \mathbb{X}, \overline{n}) = 1 - \frac{\sum_{i=1,j=i+1}^{N-1,N}\|\psi_c(\mathbf{X}_{ij} - \mathbf{X}_{ik}\mathbf{X}_{kj}, \mathbb{X}, \overline{n})\|_F}{\overline{n}N(N-1)} \quad (15)$$

$$C_p^{\psi_c}(\mathbf{X}_{ij}, \mathbb{X}, \overline{n}) = 1 - \frac{\sum_{k=1}^{N}\|\psi_c(\mathbf{X}_{ij} - \mathbf{X}_{ik}\mathbf{X}_{kj}, \mathbb{X}, \overline{n})\|_F}{2\overline{n}N} \quad (16)$$

While the above work does not explore how to estimate $\overline{n}$.

Remarks for possible improvement

From the above two methods, we make an observation that the first star-like centralized method [Yan *et al.*, 2015a] only adopts a single graph as the intermediate one for information flow, while the other method [Yan *et al.*, 2016a] goes to another extreme allowing any graph to be an intermediate bridge. When the graphs form multiple fine-grained clusters, one reference graph is incapable to capture and diffuse the global information effectively. Also, reiterating every graph for evaluating Eq. 12 is computationally intensive. One possible strategy in the between is a method for devising multiple reference graphs to improve the capacity for information flow. On the other hand, distributed graph matching may be performed locally in each cluster to reduce the computing cost. There also leaves open questions for how to select multiple reference graph and graph clusters for future work. Moreover how to effectively sample cycles remains an open question.

3.2 Global consistency recovery based methods

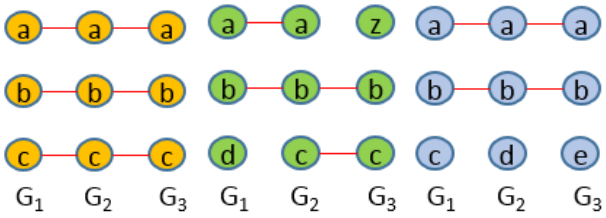
Another body of work aim to recover a globally consistent pairwise matching set from putative pairwise matchings. Spectral techniques [Kim *et al.*, 2012; Pachauri *et al.*, 2013; Huang and Guibas, 2013] are developed to extract the consistent matches by the spectrum (top eigenvectors) of the matrix composed of all putative matches. The underlying rationale is that the problem can be formulated as quadratic integer programming which can be relaxed into a generalized Rayleigh problem [Pachauri *et al.*, 2013]. In the seminal work [Huang and Guibas, 2013], the authors show theoretical conditions for exact recovery. They note that if the pairwise matchings are cycle-consistent then the bulk matrix storing all matchings is low-rank and positive semidefinite. This leads to a convex relaxation method for estimating cycle-consistent matchings by finding the nearest positive semidefinite matrix to the input matrix stacking by all initial matchings. Improvement is made in [Chen *et al.*, 2014] by assuming the underlying rank of variable matrix can be estimated reliably. Then two extensions are made: i) partially matching to allow for when different groups of common inliers are shared among different graph subsets, ii) robust recovery from a small fraction of observation given large portion of hidden or erroneous matches.

We start with some notations and concepts, and then state-of-the-art methods are discussed. In [Huang and Guibas, 2013; Chen *et al.*, 2014], a virtual universe with n^+ entities is introduced and each entity can be observed by at least one graph. Hence each pairwise matching can be written by $\mathbf{X}_{ij} = \mathbf{P}_i\mathbf{P}_j^\top$ for $\mathbf{P}_i \in \{0,1\}^{n_i\times n^+}$ which can be interpreted as the mapping from $\mathcal{G}_i$ to nodes in universe ($n_i \leq n^+$).

In line with the literature, we define a bulk matrix $\mathbf{X} \in \{0,1\}^{M\times M}$ where $M = \sum_{i=1}^N n_i$ is the total number of node copies in all N graphs, and stack $\mathbf{P}_i$ into $\mathbf{P} \in \{0,1\}^{M\times n^+}$:

$$\mathbf{X} = \begin{pmatrix} \mathbf{X}_{11} & \mathbf{X}_{12} & \cdots & \mathbf{X}_{1N} \\ \mathbf{X}_{21} & \mathbf{X}_{22} & \cdots & \mathbf{X}_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{X}_{N1} & \vdots & \vdots & \mathbf{X}_{NN} \end{pmatrix}, \quad \mathbf{P} = \begin{pmatrix} \mathbf{P}_1 \\ \mathbf{P}_2 \\ \vdots \\ \mathbf{P}_N \end{pmatrix} \quad (17)$$

Here the matrix in each entry $\mathbf{X}_{ij} \in \{0,1\}^{n_i\times n_j}$ is the pairwise match for recovery. Also an input bulk matrix $\overline{\mathbf{X}}$ can be



(a) bijection ($n = 3$) (b) partial ($n^+ = 4$) (c) common ($\overline{n} = 2$)

Figure 1: The three correspondence recovery scenarios.

defined whose submatrix entry $\overline{\mathbf{X}}_{ij}$ is obtained by pairwise matching. It has been well shown [Huang and Guibas, 2013; Pachauri *et al.*, 2013] that cycle-consistency holds if and only if the factorization holds: $\mathbf{X} = \mathbf{P}\mathbf{P}^T$.

This also equivalently leads to the property of both positive semidefinite and low-rank (recall n^+ is the universe size):

$$\mathbf{X} \succeq 0, \quad \text{rank}(\mathbf{X}) = n^+ \tag{18}$$

This formula gives a compact and solid condition for cycle-consistency. By using universe, it also allows for partial matching which is practically useful [Chen *et al.*, 2014].

There are many methods based on the above insights and they can be grouped by three main scenarios: i) bijection, ii) partial matching; iii) common inlier elicitation. The difference of the problem setting is illustrated in Fig. 1. Note we do not consider one/many-to-many matching case here.

Scenario I: Recovery under bijection

Several methods are focused on the setting that all matchings between graph pairs are bijection. Given the initial (noisy) pairwise matching input, agreement maximization (i.e. discrepancy minimization) between the initial (noisy) input match $\overline{\mathbf{X}}$ and the optimization variable $\mathbf{X}$ for recovery:

$$\max_{\mathbf{X} \succeq \mathbf{0}, \mathbf{X}_{ij} \in \mathbb{P}} < \overline{\mathbf{X}}, \mathbf{X} > \tag{19}$$

where $< \cdot, \cdot >$ is the matrix inner product and $\mathbb{P}$ is the permutation matrix space. Directly solving this combinatorial problem is intractable and relaxation methods are devised.

The spectral clustering method [Pachauri *et al.*, 2013] relaxes $\mathbf{X}$ into a rank-n symmetric matrices whose non-zero eigenvalues are N. To solve this generalized Rayleigh problem, the N leading eigenvectors $\mathbf{v}_l$ of $\overline{\mathbf{X}}$ (assume $N \leq n$) is computed such that the continuous relaxed solution is $\mathbf{V} = N \sum_{l=1}^{n} \mathbf{v}_l \mathbf{v}_l^\top$. Post-processing is performed to convert $\mathbf{V}$ to the discrete solution $\mathbf{X}$ e.g. using the Hungarian method. In a more tight convex relaxation method based on Augmented Lagrangian Method (ALM) [Huang and Guibas, 2013], the positive semidefinite constraint is preserved, and theoretical condition for exact recovery is derived.

Scenario II: Recovery under partial matching

The MatchLift method [Chen *et al.*, 2014] is one of the first studying the partial matching problem. For problem Eq. 19, to better explore the prior knowledge about the universe size n^+, they further lift the positive semidefinite constraint for $\mathbf{X}$ to the following form (they remove the doubly stochastic constraints for efficiency and empirical effectiveness):

$$\begin{bmatrix} k & \mathbf{1}^\top \\ \mathbf{1} & \mathbf{X} \end{bmatrix} = \begin{bmatrix} \mathbf{1}^\top \\ \mathbf{P} \end{bmatrix} \begin{bmatrix} \mathbf{1} & \mathbf{P}^\top \end{bmatrix} \succeq \mathbf{0} \tag{20}$$

Then they present an alternating direction methods of multipliers (ADMM) to solve the resulting problem.

MatchALS [Zhou *et al.*, 2015] devises the following objective with additional sparsity and nuclear norm regularization and relaxes $\mathbf{X}$ into continuous domain $[0, 1]$:

$$\min_{\mathbf{X}} \underbrace{-\sum_{i,j} < \mathbf{S}_{ij}, \mathbf{X}_{ij} >}_{\text{node affinity i.e. } - < \mathbf{S}, \mathbf{X} >} + \underbrace{\alpha < \mathbf{1}, \mathbf{X} >}_{\text{sparsity}} + \underbrace{\lambda \|\mathbf{X}\|_*}_{\text{rank}} \tag{21}$$
$$\mathbf{X}_{ii} = \mathbf{I}_{n_i}, \quad \mathbf{X}_{ij} = \mathbf{X}_{ij}, \quad \mathbf{0} \leq \mathbf{X} \leq \mathbf{1}$$

Double-stochastic constraint for each $\mathbf{P}_{ij}$ is also added.

Note that for optimization tractability, $\mathbf{S}_{ij} \in \mathbb{R}^{n_i \times n_j}$ is the first-order node-to-node affinity and no edge information is used, and $\mathbf{S} \in \mathbb{R}^{M \times M}$ is the bulk version for $\mathbf{S}_{ij}$. Moreover the positive semidefinite constraint on $\mathbf{X}_{ij}$ is discarded in [Zhou *et al.*, 2015] partly due to the use of rank constraint. Readers are referred to [Zhou *et al.*, 2015] for details of a fast alternating minimization algorithm for the above model.

Scenario III: Recovery under common inlier elicitation

A drawback of the above methods [Zhou *et al.*, 2015; Chen *et al.*, 2014] is the use of an excessively large n^+ to accommodate a flexible universe incurring high algorithmic complexity. One natural idea similar to [Yan *et al.*, 2016a] (recall Eq. 14, 15, 16 in Sec. 3.1) is to extract $\overline{n} < n^+$ common inliers among graphsA. In this regard, [Wang *et al.*, 2018] devises the following objective where $\mathbf{P} \in \{0, 1\}^{M \times \overline{n}}$ and the affinity term used in Eq. 21 is omitted for efficiency as here we solve factorized $\mathbf{P}$ rather than $\mathbf{X}$ (see Eq. 17) directly:

$$\min_{\mathbf{P} \in \mathbb{P}_i^{n_i \times \overline{n}}, \text{rank}(\mathbf{Z} \leq \overline{n})} \underbrace{\frac{1}{4} \|\overline{\mathbf{X}} - \mathbf{P}\mathbf{P}^\top\|_F^2}_{\text{discrepancy to input } \overline{\mathbf{X}}} + \underbrace{\frac{\lambda}{2} \|\mathbf{C}_i \mathbf{P}_i - \mathbf{Z}_i\|_F^2}_{\text{orthographic projection}} \tag{22}$$

where $\mathbb{P}$ denotes the partial permutation matrix space as:

$$\mathbf{0} \leq \mathbf{P}_i \mathbf{1} \leq \mathbf{1}, \quad \mathbf{P}_i^\top \mathbf{1} = \mathbf{1}$$

It holds as the $\overline{n}$ common inliers is observed in each graph.

Note the second term is the orthographic projection constraint that involves a constant matrix $\mathbf{C}_i \in \mathbb{R}^{2 \times n_i}$ related to the so-called measurement matrix in structure from motion [Tomasi and Kanade, 1992] while $\mathbf{Z}_i \in \mathbb{R}^{2 \times \overline{n}}$ is the $2i - 1$ and $2i$-th row of the auxiliary variable $\mathbf{Z} \in \mathbb{R}^{2n \times \overline{n}}$. A scalable block coordinate descent technique is developed in [Wang *et al.*, 2018] to solve this problem.

In fact, methods in Sec. 3.1 involve non-convex combinatorial optimization which may rely on heuristic strategy e.g. the compositional technique. For the convex-relaxation methods in Sec. 3.2, [Huang and Guibas, 2013; Chen *et al.*, 2014] still suffer scalability issue as SDP constraint is directly incorporated. Moreover, most methods [Huang and Guibas, 2013; Chen *et al.*, 2014; Wang *et al.*, 2018] surprisingly never explore the affinity information during optimization (see Eq. 19, 23). Finally for all these methods, the size of the universe n^+ or the common inliner set size $\overline{n}$ need to be explicitly provided, rendering them less applicable in real problems.

Scenario IV: Distributed multi-graph matching

The main bottleneck for methods in Sec. 3.2 is the adoption of the bulk matrix $\mathbf{X}$. One natural idea is to divide the whole matrix into overlapping submatrix, and accordingly the graph set is clustered into multiple overlapping groups. The authors in [Hu *et al.*, 2018] give a theoretical study on the connection between the cycle-consistency on overlapped clusters $\mathcal{V}_i$ and that on the whole graph set $\cup_i \mathcal{V}_i$. Furthermore, they devise the following objective which can be viewed as a distributed version of Eq. 21 in [Zhou *et al.*, 2015] by summing up the objectives in all overlapping clusters $\mathcal{V}_i$:

$$\min_{\mathbf{X}} \sum_i (< \alpha \mathbf{1}_{\mathcal{V}_i} - \mathbf{S}_{\mathcal{V}_i}, \mathbf{X}_{\mathcal{V}_i} > + \lambda \|\mathbf{X}_{\mathcal{V}_i}\|_*) \qquad (23)$$
$$\text{s. t.} \quad \mathbf{X}_{\mathcal{V}_i^{i\cap j}} = \mathbf{X}_{\mathcal{V}_j^{i\cap j}}, \quad \forall i, j$$

Here $\mathcal{V}_i^{i\cap j}$ denotes the overlapping graphs of graph cluster $\mathcal{V}_i$ and $\mathcal{V}_j$ in $\mathcal{V}_i$ and similar for $\mathcal{V}_j^{i\cap j}$. This constraint ensures the local consistency in each $\mathbf{X}_{\mathcal{V}_i}$ can lead to the global consistency for the whole bulk matrix $\mathbf{X}$. The additional constraints in Eq. 21 are also used while literally omitted here for space saving. For how a set of valid clusters $\mathcal{V}_j$ based on a given set of graphs and initial pairwise matchings, [Hu *et al.*, 2018] also presents a greedy construction method.

Note that in another concurrent work [Leonardos *et al.*, 2017], the authors propose a decentralized version of the centralized spectral method [Pachauri *et al.*, 2013] for multi-graph matching. However, akin to [Pachauri *et al.*, 2013], their method can only apply to the bijection case, while the model in Eq. 23 is applicable to the more general partial matching problem (in line with the model in Eq. 21).

3.3 Rank-1/Clustering based methods

Rank-1 methods These methods shuffle the feature vectors of graph nodes, such that the aligned vectors shall be identical to each other and the resulting stacking matrix shall be rank one. The low-rank sparsity decomposition model is often used [Zeng *et al.*, 2012; Yan *et al.*, 2015b]. However, in their models, a permutation matrix is required restricted to the bijection assumption. Moreover, the resulting optimization problem is sensitive to initial point. Hence clustering methods are developed to increase the flexibility and robustness whereby the rank-1 condition need no strictly be satisfied.

Clustering methods In [Yan *et al.*, 2016b], a feature space clustering view is adopted to the multi-graph/image matching problem, with an iterative constrained clustering algorithm alternating between a matching step and the computing of mean features. Along this perspective, a density-based clustering method called QuickMatch is proposed in [Tron *et al.*, 2017], whereby QuickShift [Vedaldi and Soatto, 2008] is used for finding clusters from modes by non-parametric estimate of the density distribution. One advantage is that neither the universe size nor the common inlier number need to be specified. Moreover it naturally allows for partial matching.

Finally it is worth noting that different from methods in Sec. 3.1, none of the methods in Sec. 3.2 and Sec. 3.3 explicitly explores the second-order or higher-order affinity information (e.g. $\mathbf{K}_{ij}$ in Eq. 8) for iterative optimization, though such an affinity could have been used to generate the initial pairwise matching i.e. $\overline{\mathbf{X}}$ for subsequent recovery, or could be indirectly encoded by the node-wise unary similarity matrix $\mathbf{S}_{ij}$ as used in Eq. 21. We believe this is the main reason for the superiority in accuracy by methods [Yan *et al.*, 2015a; 2016a] compared with [Chen *et al.*, 2014; Pachauri *et al.*, 2013] as verified by a third-party evaluation [Shi *et al.*, 2016].

4 Outlook and Concluding Remarks

Based on the review of recent advance from two-graph to multi-graph matching, we identify promising active directions for ongoing efforts on multi-graph matching.

Incremental multiple graph matching

The above multi-graph matching methods however mostly consider the offline setting i.e. all graphs are available for joint matching in one-shot. In applications graphs often arrive over time hence online incremental multi-graph matching is worth further study. One recent effort in this direction refers to [Chakraborty *et al.*, 2016], however only first-order node-wise similarity is considered, and the method is only applied on small-scale data. An exception is a recent work [Yu *et al.*, 2018a], which handles the online case with the composition technique proposed in [Yan *et al.*, 2016a].

Learning by/for graph matching

There has been efforts [Cho *et al.*, 2013] on learning geometrical affinity function's parameters supervised by the manually labeled correspondence. There are two possibilities for further exploration. The first is using cycle-consistency as the supervision signal for training the affinity function instead of the manual correspondence labels. On the other hand, deep features e.g. based on convolutional neural networks can be learned instead of traditional handcrafted feature descriptors. More recently, the seminal work present a deep network based pipeline for graph matching [Zanfir and Sminchisescu, 2018], which shows promising results by learning the feature and affinity function with high-capacity networks.

Repeated object discovery by matching

For co-detecting or co-segmenting repeated objects in a images, the capability for automatically discovering repeated objects by matching is desirable. For similar objects distributed in multiple images, proposal flow [Ham *et al.*, 2016] is proposed. [Yu *et al.*, 2018b] explores a more challenging scenario for automatically discovering two repeated objects in one image, while how to solve the case for multiple object remains open. Another relevant technique is progressive matching and graph structure updating [Cho and Lee, 2012], whereby the graph structure and affinity is not fixed but can be dynamically adjusted in the matching procedure.

Concluding remarks

This paper is a retrospective review on the course of multi-graph matching, which is relatively a new area compared with the classic setting of two-graph matching. By incorporating more graph data, it opens the room for new theory, models, and algorithms to push the frontier of graph matching.

References

[Adamczewski *et al.*, 2015] K. Adamczewski, Y. Suh, and K. Lee. Discrete tabu search for graph matching. In *ICCV*, 2015.

[Bunke, 2000] H. Bunke. Graph matching: theoretical foundations, algorithms, and applications. In *Vision Interface*, 2000.

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

[Cao *et al.*, 2018] X. Cao, W. Zhang, and Y. Yu. A bootstrapping framework with interactive information modeling for network alignment. *IEEE Access*, 2018.

[Chakraborty *et al.*, 2016] Anirban Chakraborty, Abir Das, and Amit K Roy-Chowdhury. Network consistent data association. *IEEE transactions on pattern analysis and machine intelligence*, 38(9):1859–1871, 2016.

[Chen *et al.*, 2014] Y. Chen, L. Guibas, and Q. Huang. Near-optimal joint object matching via convex relaxation. In *ICML*, 2014.

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

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

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

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

[Conte *et al.*, 2004] D. Conte, P. Foggia, C. Sansone, and M. Vento. Thirty years of graph matching in pattern recognition. *IJPRAI*, 2004.

[Cour *et al.*, 2006] T. Cour, P. Srinivasan, and J. Shi. Balanced graph matching. In *NIPS*, 2006.

[Duchenne *et al.*, 2011a] O. Duchenne, F. Bach, I. Kweon, and J. Ponce. A tensor-based algor ithm for high-order graph matching. *PAMI*, 2011.

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

[Foggia *et al.*, 2014] P. Foggia, G. Percannella, and M. Vento. Graph matching and learning in pattern recognition in the last 10 years. *IJPRAI*, 33(1), 2014.

[Gao *et al.*, 2010] X. Gao, B. Xiao, D. Tao, and X. Li. A survey of graph edit distance. *Pattern Analysis and Applications*, 2010.

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

[Ham *et al.*, 2016] B. Ham, M. Cho, C. Schmid, and J. Ponce. Proposal flow. In *CVPR*, 2016.

[Hu *et al.*, 2018] N. Hu, Q. Huang, B. Thibert, and L. Guibas. Distributable consistent multi-graph matching. *CVPR*, 2018.

[Huang and Guibas, 2013] Q. Huang and L. Guibas. Consistent shape maps via semidefinite programming. In *SGP*, 2013.

[Huang *et al.*, 2012] 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.

[Kim *et al.*, 2012] V. G. Kim, W. Li, N. J. Mitra, S. DiVerdi, and T. Funkhouser. Exploring collections of 3d models using fuzzy correspondences. In *SIGGRAPH*, 2012.

[Lee *et al.*, 2010] J. Lee, M. Cho, and K.M. Lee. A graph matching algorithm using data-driven markov chain monte carlo sampling. In *ICPR*, 2010.

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

[Leonardos *et al.*, 2017] S. Leonardos, X. Zhou, and K. Daniilidis. Distributed consistent data association. *ICRA*, 2017.

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

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

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

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

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

[Livi and Rizzi, 2013] L. Livi and A. Rizzi. The graph matching problem. *Pattern Analysis and Applications*, 16(3):253–283, 2013.

[Munkres, 1957] J. Munkres. Algorithms for the assignment and transportation problems. *Journal of the Society for Industrial & Applied Mathematics*, 5(1), 1957.

[Ngoc *et al.*, 2015] Q. Ngoc, A. Gautier, and M. Hein. A flexible tensor block coo rd inate ascent scheme for hypergraph matching. In *CVPR*, 2015.

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

[Santo *et al.*, 2003] M. De Santo, P. Foggia, C. Sansone, and M. Vento. A large database of graphs and its use for benchmarking graph isomorphism algorithms. *Pattern Recognition Letters*, 24(8):1067–1079, 2003.

[Schellewald and Schnörr, 2005] C. Schellewald and C. Schnörr. Probabilistic subgraph matching based on convex relaxation. In *EMMCVPR*, 2005.

[Shi *et al.*, 2016] X. Shi, H. Ling, W. Hu, J. Xing, and Y. Zhang. Tensor power iteration for multi-graph matching. In *CVPR*, 2016.

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

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

[Suh *et al.*, 2012] Y. Suh, M. Cho, and K. M. Lee. Graph matching via sequential monte carlo. In *ECCV*, 2012.

[Tomasi and Kanade, 1992] C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: a factorization method. *International Journal of Computer Vision*, 9(2):137–154, 1992.

[Torr, 2003] P. H. S. Torr. Solving markov random fields using semidefinite programmin. In *AISTATS*, 2003.

[Tron *et al.*, 2017] R. Tron, X. Zhou, C. Esteves, and K. Daniilidis. Fast multi-image matching via density-based clustering. In *CVPR*, pages 4057–4066, 2017.

[Vedaldi and Soatto, 2008] A. Vedaldi and S. Soatto. Quick shift and kernel methods for mode seeking. In *ECCV*, 2008.

[Vento, 2015] M. Vento. A long trip in the charming world of graphs for pattern recognition. *Pattern Recognition*, 48(2):291–301, 2015.

[Wang *et al.*, 2018] Q. Wang, X. Zhou, and K. Daniilidis. Multi-image semantic matching by mining consistent features. In *CVPR*, 2018.

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

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

[Yan *et al.*, 2015a] J. Yan, J. Wang, H. Zha, X. Yang, and S. Chu. Consistency-driven alternating optimization for multigraph matching: A unified approach. *IEEE Transactions on Image Processing*, 24(3):994–1009, 2015.

[Yan *et al.*, 2015b] J. Yan, H. Xu, H. Zha, X. Yang, H. Liu, and S. Chu. A matrix decomposition perspective to multiple graph matching. In *ICCV*, 2015.

[Yan *et al.*, 2015c] J. Yan, C. Zhang, H. Zha, W. Liu, X. Yang, and S. Chu. Discrete hyper-graph matching. In *CVPR*, 2015.

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

[Yan *et al.*, 2016b] J. Yan, Z. Ren, H. Zha, and S. Chu. A constrained clustering based approach for matching a collection of feature sets. In *ICPR*, pages 3832–3837, 2016.

[Yan *et al.*, 2016c] J. Yan, X. Yin, W. Lin, C. Deng, H. Zha, and X. Yang. A short survey of recent advances in graph matching. In *ICMR*, 2016.

[Yu *et al.*, 2018a] T. Yu, J. Yan, W. Liu, and B. Li. Incremental multi-graph matching via diversity and randomness based graph clustering. In *ECCV*, 2018.

[Yu *et al.*, 2018b] T. Yu, J. Yan, J. Zhao, and B. Li. Joint cuts and matching of partitions in one graph. In *CVPR*, 2018.

[Zanfir and Sminchisescu, 2018] A. Zanfir and C. Sminchisescu. Deep learning of graph matching. In *CVPR*, 2018.

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

[Zass and Shashua, 2008] R. Zass and A. Shashua. Probabilistic graph and hypergraph matching. In *CVPR*, 2008.

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

[Zhou and Torre, 2016] F. Zhou and F. Torre. Factorized graph matching. *IEEE PAMI*, 2016.

[Zhou *et al.*, 2015] X. Zhou, M. Zhu, and K. Daniilidis. Multi-image matching via fast alternating minimization. In *ICCV*, 2015.