Joint optimization for consistent multiple graph matching

Junchi Yan*
Shanghai Jiao Tong University
IBM Research - China
yanjunchi@sjtu.edu.cn

Yu Tian
Shanghai Jiao Tong University
Shanghai, China, 200240
yutian@sjtu.edu.cn

Hongyuan Zha
Georgia Institute of Technology
Atlanta, Georgia, 30032
zha@cc.gatech.edu

Xiaokang Yang
Shanghai Jiao Tong University
Shanghai, China, 200240
xkyang@sjtu.edu.cn

Ya Zhang
Shanghai Jiao Tong University
Shanghai, China, 200240
ya_zhang@sjtu.edu.cn

Stephen M. Chu
IBM T.J. Waston Research Center
Yorktown Heights, NY USA, 10598
schu@us.ibm.com

Abstract

The problem of graph matching in general is NP-hard and approaches have been proposed for its suboptimal solution, most focusing on finding the one-to-one node mapping between two graphs. A more general and challenging problem arises when one aims to find consistent mappings across a number of graphs more than two. Conventional graph pair matching methods often result in mapping inconsistency since the mapping between two graphs can either be determined by pair mapping or by an additional anchor graph. To address this issue, a novel formulation is derived which is maximized via alternating optimization. Our method enjoys several advantages: 1) the mappings are jointly optimized rather than sequentially performed by applying pair matching, allowing the global affinity information across graphs can be propagated and explored; 2) the number of concerned variables to optimize is in linear with the number of graphs, being superior to local pair matching resulting in $O(n^2)$ variables; 3) the mapping consistency constraints are analytically satisfied during optimization; and 4) off-the-shelf graph pair matching solvers can be reused under the proposed framework in an 'out-of-the-box' fashion. Competitive results on both the synthesized data and the real data are reported, by varying the level of deformation, outliers and edge densities.

*Corresponding author. The work is supported by NSF IIS-1116886, NSF IIS-1049694, NSFC 61129001/F010403 and the 111 Project (B07022).

1. Introduction

Graph matching is an essential problem in theoretical computer science due to its powerful characteristics of abstraction. The problem of graph matching is to establish a consistent mapping between the nodes of two or more graphs. It is related to various research areas such as computer vision, pattern recognition, and machine learning. It has attracted considerable research interests [4] for decades yet remains challenging, due to its NP-hard property.

Graph matching is typically and mostly considered under the two-graph scenario, and has been explored in a variety of tasks such as feature tracking, image retrieval, object recognition, and shape matching [9]. For instance, object matching can be regarded as finding consistent correspondences between two sets of features, by maximizing the fitness regarding the unary and edge similarities in two graphs. In this setting, the problem can be formulated as graph matching that aims to find a point-to-point mapping that preserves as much as possible the relationships between nodes. Many approaches have been proposed, ranging from the early work [7] to the recent work [3, 21] etc.

However, in many of other scenarios, given a set of graphs, which all represent equivalent or related structures, it is required to find global consistent correspondences for the *all* graphs. Relevant applications could be found in [24], where representations obtained from Infrared, Optical, Cartographic and SAR images must be combined, as well as in [15] where a prototype has to be synthesized from noisy data representing the same object. To address the multi-

ple graph matching problem, a simple strategy could be applying the state-of-the-art two-graph matching algorithms on each graph pair individually and independently: for instance, given graph G_a, G_b and G_c, firstly one can apply the pair matching solver e.g. [3, 21] to find the node mapping $\mathbf{X}_{ab}$, $\mathbf{X}_{bc}$ respectively, and then obtain the mapping $\mathbf{X}_{ac}$ induced by the intermediate graph G_b. One obvious weakness is the affinity measurement between G_a and G_c is ignored, which might be more accurate than the defined affinity between G_a-G_b and G_b-G_c especially in case G_b is heavily corrupted. On the other hand, applying pair match solver between individual G_a and G_c is likely to generate inconsistent or redundant matching solutions compared with the mapping induced by $\mathbf{X}_{ab}$, $\mathbf{X}_{bc}$, especially given large number of nodes or significant corruption.

Aiming at addressing these challenges, we make the following main contributions in theory and algorithm:

1) We derive the original objective formulation into a new one where the redundant mapping variables are compactly represented by the set of basis mapping variables. Moreover, we show the number of the basis variables scales linearly with the number of graphs. This formulation mathematically ensures the mapping constraints and allows for joint pair mappings optimization, mixing the local and the global affinity information. Its robustness against noise and local distortion are indicated in the experiments;

2) Using the derived formulation, we propose an alternating optimization method which allows reusing any of the existing pair matching solvers. The convergence property is analyzed theoretically and observed experimentally.

2. Related work

A myriad of methods have been proposed for graph matching. Most approaches address the graph matching problem in the context of finding correspondence between a pair of graphs, for which an incomplete list is as follows: [14] formulate graph matching as a constrained integer quadratic programming (IQP) problem with a concave optimization scheme. [7] proposed the well known Graduated Assignment algorithm (GAGM) by relaxing the integer constraint. [23] designed the Successive Projection Graph Matching (SPGM), and recently [10] proposed the Integer Projected Fixed Point (IPFP) method with climbing and convergence properties that optimizes the IQP in the discrete domain. Another line of optimization techniques adopts the technique of spectral matching to approximately solve the matching task: [9] proposed an efficient Spectral Matching (SM) method to the IQP based on spectral relaxation, which computes the leading eigenvector of symmetric nonnegative affinity matrix. [20] extended SM to Spectral Matching with Affine Constraint (SMAC) by introducing affine constraints into the spectral decomposition that encodes the one-to-one matching constraints. Recently, [13] showed point matching capable of handling significant affine or similarity transformation can be formulated as an IQP problem which also appeared in graph matching

Beyond the second order edge affinity, recent work explore the graph structure by hyper-edges, e.g. triple-node tuples [5,8,12,26]. Another research thread in parallel is focusing on learning the affinity matrix (or tensor in the hyper graph case) for graph matching [2, 6, 11] etc. While this paper focuses on the optimization framework, rather than how come from the affinity information (learning or learning-free); or to what extend the affinity information is explored (edge or hyper-edge). Thus these work are in parallel and out of the scope of this paper.

One limitation of the aforementioned approaches is that the matching is conducted under a graph pair, rather than an arbitrary number of graphs *jointly* and *consistently*. The multiple graph matching problem is also termed as common labelling, especially in the context of graph cluster representation [18], and has many connections with how to measure the similarity for the attributed graphs, in terms of random graph synthesizing, representation, classification and clustering e.g. [1, 16, 17, 25] etc. Specifically, [25] synthesizes an ensemble of attributed graphs into the distribution of a random graph and further classify the query graph to a certain category. By using the second order information, [16] extend the first order random graphs (FORGs) [25] to the second-order random graphs (SORGs) for graph clustering and recognition. [1] propose an unsupervised learning method for graphs clustering and median graph building based on a set of graphs. These works are all based on graph pair matching as the similarity metric, and share the common weakness that the local matching error taken at initial stages might propagate and lead to bad global results.

Compared with the pair matching problem, the multiple graph matching has not been intensively studied by the community, and only a few address this problem using principled optimization. The proof-of-concept (as claimed in the paper by the authors) [24] provide a Bayesian view to extend the matching graph pairs to multiple ones, but no solver is proposed. Using the generalized softassign algorithm, [18] substitute the assignation matrices associated with each graph pair by an assignation hypercube. It generates always consistent isomorphisms, and alleviates the lacking of global knowledge in optimization. However simply extending to the N-dimension softassign suffers the prohibitive computational complexity due to directly manipulating the N-dimension stochastic hypercube during softassign iteration. A more recent state-of-the-art [19] is the extension and improvement of [18]: hypercube is first obtained by pair isomorphism, and then a clean-up step is performed to enforce the mapping consistency. This algorithm computes the hypercube via sequential local pair matching, resulting in more computational efficiency. How-

ever, none of the previous work has shown how the multiple graph matching formulation can be back transformed into a pair matching problem in its IQP formulation, both algorithmically and mathematically.

To our best knowledge, this is the first work on extending the IQP formulation for pair matching to multiple ones. Our novel formulation explores the global affinity and unveil the underlying connection with pair matching. And it is scalable to N-graph case.

3. Preliminaries for graph matching

3.1. Graph pair matching

First we briefly introduce the widely used formulation of pair graph matching. Concretely, given two graphs $G^L(V^L, E^L, A^L)$ and $G^R(V^R, E^R, A^R)$, where V denotes nodes, E, edges and A, attributes, there is an affinity matrix defined as $\mathbf{M}_{ia;jb}$ that measures the affinity with the candidate edge pair (v_i^L, v_j^L) vs. (v_a^R, v_b^R). And the diagonal term $\mathbf{M}_{ia;ia}$ describes the unary affinity of a node match (v_i^L, v_a^R). By introducing a permutation matrix $\mathbf{x} \in \{0,1\}^{n_L \times n_R}$ whereby $\mathbf{p}_{ia} = 1$ if node V_i^L matches node V_a^R (and $\mathbf{p}_{ia} = 0$ otherwise), it leads to the following constrained IQP formulation:

$$\mathbf{p}^* = \arg_{\mathbf{p}} \max(\mathbf{p}^T \mathbf{M} \mathbf{p}) \quad s.t. \quad \mathbf{A}\mathbf{p} = \mathbf{1} \quad \mathbf{p} \in \{0,1\} \quad (1)$$

here $\mathbf{x}$ is the vectorized permutation matrix, and **Ax=1** refers to the one-to-one node mapping constraint for two graphs.

3.2. Multiple graph matching

Given N graphs and the associated affinity matrix $\mathbf{M}_{ij}$, a natural extension for multi-graph matching is:

$$\mathbf{P}^* = \arg_{\mathbf{P}} \max \sum_{i,j=1,2,...,N;i>j} \lambda_{ij} \mathbf{p}_{ij}^T \mathbf{M}_{ij} \mathbf{p}_{ij} \quad (2)$$
$$s.t. \quad \mathbf{A}\mathbf{p}_{ij} = \mathbf{1} \quad \mathbf{p}_{ij} \in \{0,1\}$$
$$\forall i,j = 1,2,...,N; \quad i > j \quad |\mathbf{P}| = \frac{N(N-1)}{2}$$

where $\mathbf{P}$ is the vectorized pairwise permutation matrix set among multiple views: $\{\mathbf{p}_{ij}\}(\forall i,j = 1,2,...,N; i > j)$. And λ is the weight for each pair matching score.

The above formulation tells that optimizing $\mathbf{p}_{ij}$, $\mathbf{p}_{jk}$ and $\mathbf{x}_{ik}$ could not guarantee the consistency between $\mathbf{p}_{ik}$ and the mapping induced by the chain $\mathbf{p}_{ij}$, $\mathbf{p}_{jk}$ - see Fig. 1 for example. One alternative solution is to enforce some certain constraints like $f(\mathbf{p}_{ij}, \mathbf{p}_{ik}, \mathbf{p}_{jk}) = 0$ in addition to the extended formulation 2, and this calls for exploring the explicit mathematical connection. Even knowing the explicit constraints as will be shown in the rest of this paper, we still need new algorithms to solve this problem as it might be different from the conventional pair matching problem.

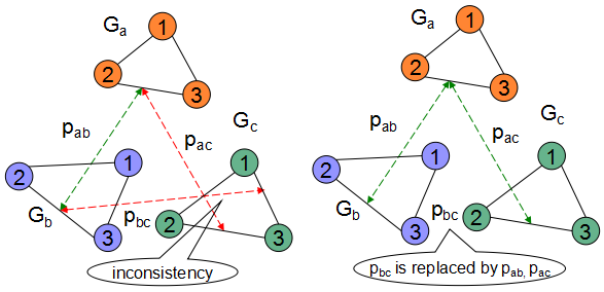


Figure 1: Graph a, b, c: (a) the edge mapping between graph b and c by the mapping chain through $\mathbf{p}_{ab}$ and $\mathbf{p}_{ac}$ is inconsistent to the direct mapping $\mathbf{p}_{bc}$, this suggests the redundancy of three mappings; (b) when the $\mathbf{p}_{bc}$ is subsumed by a formulation only containing $\mathbf{p}_{ab}$ and $\mathbf{p}_{ac}$, the redundancy and inconsistency naturally disappear.

4. Proposed framework and algorithm

Given a graph set with more than two graphs, we are interested in finding their node mapping among the graphs. Directly applying the existing pairwise matching methods will sequentially match the graphs by pair and a local matching mistake between two graphs may propagate along the matching chain to deteriorate the overall performance. Thus we are motivated to design a principled multiple graph matching framework in nature, and furthermore, to explore the possibility of reusing the resource of the off-the-shelf pairwise matching solvers.

4.1. Reduce the redundant $\mathbf{P}_{bc}$

In the following, we show how to find the representation of $\mathbf{P}_{bc}$ that satisfies consistency over multiple graphs. For clarity reason, the following mathematical derivations will specifically focus on the triple-view graph matching scenario, while it should be noted the formulation can be readily extended to any number of graphs - this will be discussed shortly after the illustration of triple case. Given three graphs G_a, G_b, G_c, let $\mathbf{p}_{ab}$, $\mathbf{p}_{ac}$, $\mathbf{p}_{bc}$ denote the matching correspondences mapping between view pair $\{G_a, G_b\}$, $\{G_a, G_c\}$ and $\{G_b, G_c\}$, respectively.

In particular, we are interested in how to formulate the triple-view graph matching problem in a unified optimization framework, and preferably, the pairwise matching techniques can be reused. We illustrate our idea using a concrete

example as shown in the following[1]:

$$\mathbf{p}_{ab} = \Big[0,1,0,0,\vdots,0,0,1,0,\vdots,1,0,0,0,\vdots,0,0,0,1\Big]^T \quad (3)$$

$$\mathbf{p}_{ac} = \Big[0,0,0,1,\vdots,0,0,1,0,\vdots,0,1,0,0,\vdots,1,0,0,0\Big]^T \quad (4)$$

where $\mathbf{p}_{ab}$ in Eq.3 indicates that point {1,2,3,4} in view a corresponds to point {2,3,1,4} in view b respectively. And $\mathbf{p}_{ac}$ in Eq.4 denotes point {1,2,3,4} in view a corresponds to {4,3,2,1} in view c respectively. In this way, the correspondences between view b and view c can be determined from the known mapping between pair ab and pair ac:

$$\mathbf{p}_{bc} = \Big[0,1,0,0,\vdots,0,0,0,1,\vdots,0,0,1,0,\vdots,1,0,0,0\Big]^T \quad (5)$$

While the above observation is obvious, it still lacks the strict mathematical derivation. In what follows, we show how to obtain $\mathbf{p}_{bc}$ when given $\mathbf{p}_{ab}$ and $\mathbf{p}_{ac}$, respectively. This is the key step to build our final formulation.

Still in the context of four-point graph matching, first define the matrix $\mathbf{Q}_{41}$ as follows:

$$\mathbf{Q}_{41} = \begin{bmatrix} 1,0,0,0,\vdots,0,0,0,0,\vdots,0,0,0,0,\vdots,0,0,0,0 \\ 0,1,0,0,\vdots,0,0,0,0,\vdots,0,0,0,0,\vdots,0,0,0,0 \\ 0,0,1,0,\vdots,0,0,0,0,\vdots,0,0,0,0,\vdots,0,0,0,0 \\ 0,0,0,1,\vdots,0,0,0,0,\vdots,0,0,0,0,\vdots,0,0,0,0 \end{bmatrix} \quad (6)$$

And we have: $\mathbf{Q}_{41}\mathbf{p}_{ab} = [0,1,0,0]^T$. Note that $\mathbf{Q}_{41}$ is used to get the first block of $\mathbf{p}_{ab}$. And for $\mathbf{Q}_{41}$, the subscript '1' denotes the position of the identity sub-matrix in Eq. 6. Similarly, we have: $\mathbf{Q}_{41}\mathbf{p}_{ac} = [0,0,0,1]^T$. As such, one can obtain:

$$(\mathbf{Q}_{41}\mathbf{x}_{ab}) \otimes (\mathbf{Q}_{41}\mathbf{x}_{ac}) = \Big[0,0,0,0,\vdots,0,0,0,1,\vdots,0,0,0,0,\vdots,0,0,0,0\Big]^T$$

where $\otimes$ is the Kronecker Product. Thus we have:

$$\mathbf{p}_{bc} = \sum_{i=1}^{4}(\mathbf{Q}_{4i}\mathbf{p}_{ab}) \otimes (\mathbf{Q}_{4i}\mathbf{p}_{ac}) \quad (7)$$

In general, for N node graph, we have:

$$\mathbf{p}_{bc} = \sum_{i=1}^{N}(\mathbf{Q}_{Ni}\mathbf{p}_{ab}) \otimes (\mathbf{Q}_{Ni}\mathbf{p}_{ac}) \quad (8)$$

We have shown $\mathbf{p}_{bc}$ can be analytically derived from $\mathbf{p}_{ab}$ and $\mathbf{p}_{ac}$. And triple-view matching objective can be written:

$$\max \mathbf{p}_{ab}^T\mathbf{M}_{ab}\mathbf{p}_{ab} + \mathbf{p}_{ac}^T\mathbf{M}_{ac}\mathbf{p}_{ac} + \mathbf{p}_{bc}^T\mathbf{M}_{bc}\mathbf{p}_{bc} \quad (9)$$

[1]When the number of nodes are different in two graphs, one can add dummy nodes or introduce slack variables as done in [7]. Thus we always assume equal number of nodes.

where $\mathbf{p}_{bc} = \sum_{i=1}^{N}(\mathbf{Q}_{Ni}\mathbf{p}_{ab}) \otimes (\mathbf{Q}_{Ni}\mathbf{p}_{ac})$. Without loss of generality, the weights of each term in (9) are omitted.

Then we address how to formulate $\mathbf{p}_{bc}^T\mathbf{M}_{bc}\mathbf{p}_{bc}$ using $\mathbf{p}_{ab}$ and $\mathbf{p}_{ac}$ while keeping its IQP formulation. Note the following equation (the so-called mixed-product property):

$$(\mathbf{Q}_{Ni}\mathbf{p}_{ab}) \otimes (\mathbf{Q}_{Ni}\mathbf{p}_{ac}) = (\mathbf{Q}_{Ni} \otimes \mathbf{Q}_{Ni})(\mathbf{p}_{ab} \otimes \mathbf{p}_{ac}) \quad (10)$$

Therefore $\mathbf{p}_{bc}^T\mathbf{M}_{bc}\mathbf{p}_{bc}$ can be written as:

$$\begin{aligned}
&[\sum_{i=1}^{N}(\mathbf{Q}_{Ni}\mathbf{p}_{ab}) \otimes (\mathbf{Q}_{Ni}\mathbf{p}_{ac})]^T\mathbf{M}_{bc}[\sum_{i=1}^{N}(\mathbf{Q}_{Ni}\mathbf{p}_{ab}) \otimes (\mathbf{Q}_{Ni}\mathbf{p}_{ac})] \\
=&\sum_{i=1}^{N}(\mathbf{p}_{ab} \otimes \mathbf{p}_{ac})^T(\mathbf{Q}_{Ni} \otimes \mathbf{Q}_{Ni})^T\mathbf{M}_{bc}\sum_{i=1}^{N}(\mathbf{Q}_{Ni} \otimes \mathbf{Q}_{Ni})(\mathbf{p}_{ab} \otimes \mathbf{p}_{ac}) \\
=&(\mathbf{p}_{ab} \otimes \mathbf{p}_{ac})^T[\sum_{i=1}^{N}(\mathbf{Q}_{Ni} \otimes \mathbf{Q}_{Ni})^T\mathbf{M}_{bc}\sum_{i=1}^{N}(\mathbf{Q}_{Ni} \otimes \mathbf{Q}_{Ni})](\mathbf{p}_{ab} \otimes \mathbf{p}_{ac}) \\
=&\Big[\mathbf{p}_{ab}^1\mathbf{p}_{ac}^T, \mathbf{p}_{ab}^2\mathbf{p}_{ac}^T, \ldots, \mathbf{p}_{ab}^{N^2}\mathbf{p}_{ac}^T\Big]\mathbf{M}_{bc}^Q\begin{bmatrix}\mathbf{p}_{ab}^1\mathbf{p}_{ac} \\ \mathbf{p}_{ab}^2\mathbf{p}_{ac} \\ \vdots \\ \mathbf{p}_{ab}^{N^2}\mathbf{p}_{ac}\end{bmatrix} \\
=&\sum_{i=1}^{N^2}\sum_{j=1}^{N^2}\mathbf{p}_{ab}^i\mathbf{p}_{ab}^j\mathbf{p}_{ac}^T\mathbf{M}_{bc}^Q(i,j)\mathbf{p}_{ac} \qquad (11)
\end{aligned}$$

where $\mathbf{p}_{ab}^i$ is the ith scalar element in $\mathbf{p}_{ab}$, while $\mathbf{M}_{bc}^Q(i,j)$ denotes the sub-matrix within row $(i-1)*N^2+1 : i*N^2$ and column $(j-1)*N^2+1 : j*N^2$. And we have:

$$\mathbf{M}_{bc}^Q = \sum_{i=1}^{N}(\mathbf{Q}_{Ni} \otimes \mathbf{Q}_{Ni})^T\mathbf{M}_{bc}\sum_{i=1}^{N}(\mathbf{Q}_{Ni} \otimes \mathbf{Q}_{Ni}) \quad (12)$$

Note that $\mathbf{M}_{bc}^Q$ is a constant during optimization, thus the mathematically difficult operator $\otimes$ actually disappears. Then we obtain the transformed new objective function:

$$\mathbf{p}_{ab}^T\mathbf{M}_{ab}\mathbf{p}_{ab} + \mathbf{p}_{ac}^T\mathbf{M}_{ac}x_{ac} + \sum_{i,j=1}^{N^2}\mathbf{p}_{ab}^i\mathbf{p}_{ab}^j\mathbf{p}_{ac}^T\mathbf{M}_{bc}^Q(i,j)\mathbf{p}_{ac} \quad (13)$$

Now we show how to replace the term including $\mathbf{p}_{bc}$ using $\mathbf{p}_{ab}$. Note one can reverse $\mathbf{p}_{bc}^T\mathbf{M}_{bc}\mathbf{p}_{bc}$ into $\mathbf{p}_{cb}^T\mathbf{M}_{cb}\mathbf{p}_{cb}$ without changing the property of the objective function 9, which allows for the following reformulation by replacing $\mathbf{p}_{cb}$ with $\mathbf{p}_{ab}$:

$$\mathbf{p}_{ab}^T\mathbf{M}_{ab}\mathbf{p}_{ab} + \mathbf{p}_{ac}^T\mathbf{M}_{ac}\mathbf{p}_{ac} + \sum_{i,j=1}^{N^2}\mathbf{p}_{ac}^i\mathbf{p}_{ac}^j\mathbf{p}_{ab}^T\mathbf{M}_{cb}^Q(i,j)\mathbf{p}_{ab} \quad (14)$$

where $\mathbf{M}_{cb}^Q = \sum_{i=1}^{N}(\mathbf{Q}_{Ni} \otimes \mathbf{Q}_{Ni})^T\mathbf{M}_{cb}\sum_{i=1}^{N}(\mathbf{Q}_{Ni} \otimes \mathbf{Q}_{Ni})$. It should be noted that $\mathbf{p}_{ab}$, $\mathbf{p}_{ac}$ and $\mathbf{p}_{bc}$ are redundant in determining the two mapping relation among three graphs. This also poses the gap for direct applying of the conventional pair matching methods. The above derivation has shown how to replace the redundant variable $\mathbf{p}_{bc}$ in Eq. 9 and reach a more compact objective formulation as in Eq. 13 or Eq. 14.

Algorithm 1 Alternating optimization for graph G_a,G_b,G_c

Input: affinity matrix $\mathbf{M}_{ab}$,$\mathbf{M}_{bc}$, $\mathbf{M}_{ac}$;
Output: *consistent* mappings $\mathbf{p}_{ab}$, $\mathbf{p}_{bc}$, $\mathbf{p}_{ac}$
Initial: $k = 0$; $\mathbf{p}_{ab}^0 = \mathbf{p}_{bc}^0 = \mathbf{p}_{ac}^0 = [\frac{1}{n^2}, ..., \frac{1}{n^2}]^T$
while (k not exceeds iteration maximum) **do**
  k=k+1
  fix the latest updated $\mathbf{p}_{ab}^{k-1}$, and update $\mathbf{p}_{ac}^k$ using (15)
  fix the latest updated $\mathbf{p}_{ac}^k$, and update $\mathbf{p}_{ab}^k$ using (16)
  $\Delta_{ab} = \|\mathbf{p}_{ab}^k - \mathbf{p}_{ab}^{k-1}\|_2$, $\Delta_{ac} = \|\mathbf{p}_{ac}^k - \mathbf{p}_{ac}^{k-1}\|_2$
  if $\Delta_{ab} < \varepsilon, \Delta_{ac} < \varepsilon$: **stop**
end while
Calculate $\mathbf{p}_{bc}$ by $\mathbf{p}_{ab}$ and $\mathbf{p}_{ac}$ using Eq.8.

4.2. Alternating optimization algorithm

In terms of the proposed framework, we design our optimization algorithm by iterating with respect to $\mathbf{p}_{ab}$ and $\mathbf{p}_{ac}$ alternatively, and fixing the other meanwhile. In the k-th iteration, leveraging on the form of objective function by (13), one can fix $\mathbf{p}_{ab}^{k-1}$ and update $\mathbf{p}_{ac}^k$ by maximizing:

$$\max_{\mathbf{p}_{ac}^k} {\mathbf{p}_{ac}^k}^T [\mathbf{M}_{ac} + \sum_{i,j=1}^{N^2} \mathbf{p}_{ab}^{ik} \mathbf{p}_{ab}^{jk} \mathbf{M}_{bc}^Q(i,j)] \mathbf{p}_{ac}^k \qquad (15)$$

Alternatively, using Eq.14, one can fix the updated $\mathbf{p}_{ac}^k$ and renew $\mathbf{p}_{ab}^k$ via optimizing:

$$\max_{\mathbf{p}_{ab}^k} {\mathbf{p}_{ab}^k}^T [\mathbf{M}_{ab} + \sum_{i,j=1}^{N^2} \mathbf{p}_{ac}^{ik} \mathbf{p}_{ac}^{jk} \mathbf{M}_{bc}^Q(i,j)] \mathbf{p}_{ab}^k \qquad (16)$$

By doing so, we update both $\mathbf{p}_{ab}^k$ and $\mathbf{p}_{ac}^k$ in the k-the iteration. Notice that Eq.15 and Eq.16 are both IQP problem, which can be solved by current pair graph matching techniques 'out-of-the-box'.

Based on the above observation, the proposed formulation and algorithm are described in Alg.1. Any pair matching solver can be performed for alternating updating. Here we briefly discuss the convergence property of the proposed algorithm. As the optimized variable $\mathbf{p}$ = [$\mathbf{p}_{ab}$, $\mathbf{p}_{ac}$] is vectorized permutation matrix, forming a solution chain $\mathbf{p}^1$,$\mathbf{p}^2$,... thus there must exist k and s such that $\mathbf{p}^k$=$\mathbf{p}^s$. In other words, the iteration will converge to a looping sequence or a fixed point after finite steps.

4.3. From triple-graph to N-graph

Now we address the problem raised in the beginning for how to generalize to any-order of multiple graph matching. Observing the fact from the triple matching model that $\mathbf{p}_{bc}$ can be represented by $\mathbf{p}_{ab}$ and $\mathbf{p}_{ac}$. Thus for arbitrary N graphs, e.g. a set of 4 graphs a, b, c, d, the pairwise matching correspondences $\mathbf{p}_{bc}$, $\mathbf{p}_{bd}$, $\mathbf{p}_{cd}$ can be represented by the tuples: ($\mathbf{p}_{ab}$, $\mathbf{p}_{ac}$), ($\mathbf{p}_{ab}$, $\mathbf{p}_{ad}$) and ($\mathbf{p}_{ac}$, $\mathbf{p}_{ad}$) respectively. This indicates that $\mathbf{p}_{ab}$, $\mathbf{p}_{ac}$, $\mathbf{p}_{ad}$ are the base variables for optimization. Similar to the triple-graph case, one can fix two of the three variables, and optimize in terms of the left one. In general, given m graphs $G_1, G_2, \ldots, G_m$, one can reformulate the objective function with respect to $\mathbf{p}_{G_1G_2}, \mathbf{p}_{G_1G_3}, \ldots, \mathbf{p}_{G_1G_m}$, totally $m-1$ variables. And for each iteration, fix $m-2$ variables, and optimize with respect to the other in an alternative manner. This observation is important to the computational extensibility of our framework in that as the number of graph increases, the total computational overhead grows in a linear proportion due to each of the $m-1$ variables is iteratively updated by turn. The convergence analysis for the triple-graph case also applies for the study of N-graph. And the N-view graph matching algorithm is summarized in Alg.2.

Algorithm 2 Alternating optimization for N-graph

Input: affinity matrix $\mathbf{M}_{G_1G_2}$,$\mathbf{M}_{G_2G_3}$,..., $\mathbf{M}_{G_{N-1}G_N}$
Output: *consistent* $\mathbf{p}_{G_1G_2}$, $\mathbf{p}_{G_2G_3}$,...,$\mathbf{p}_{G_{N-1}G_N}$
Initial: k=0; $\mathbf{p}_{G_1G_2}^0$=,...,=$\mathbf{p}_{G_{N-1}G_N}^0$=$[\frac{1}{n^2}, ..., \frac{1}{n^2}]^T$
while (k not exceeds iteration maximum) **do**
  k=k+1
  for (i = 2 to N) **do**
    fix the latest updated $\mathbf{p}_{G_jG_1}^{k-1}$, and update $\mathbf{p}_{G_iG_1}^k$
    $\Delta_{G_iG_1} = \|\mathbf{p}_{G_iG_1}^k - \mathbf{p}_{G_iG_1}^{k-1}\|_2$, ($j$=2,...,$N$, $j \neq i$)
  end for
  if $\forall\ \Delta_{G_iG_1} < \varepsilon; i = 2, 3, ..., N$: **stop**
end while

Table 1: Eq.8 establishing ratio of 30 random trials from the synthetic experiment by independent pair-graph matching.

Deformation	.05	0.1	.15	0.2	.25	0.3	.35
	100%	100%	90%	70%	40%	0%	0%
Outlier #	2	4	6	8	10	12	14
	80%	67%	27%	30%	0%	0%	0%
Density level	0.4	0.5	0.6	0.7	0.8	0.9	1
	70%	67%	64%	64%	20%	3%	0%

In summary, we derive a framework for multiple graph matching, where the variables are optimized jointly and the consistency is always satisfied. In each iteration, we cast the objective function into an IQP that can be solved by any pair matching techniques. The formulation is also mathematically and computationally extensible to N graphs.

5. Experiments and discussion

5.1. Protocol on simulation tests

In terms of experimental protocol for graph matching, the online available [3, 8] has been recognized as a baseline evaluation. Following the same procedure as [3], we ran multiple random tests on synthetic data by different levels of

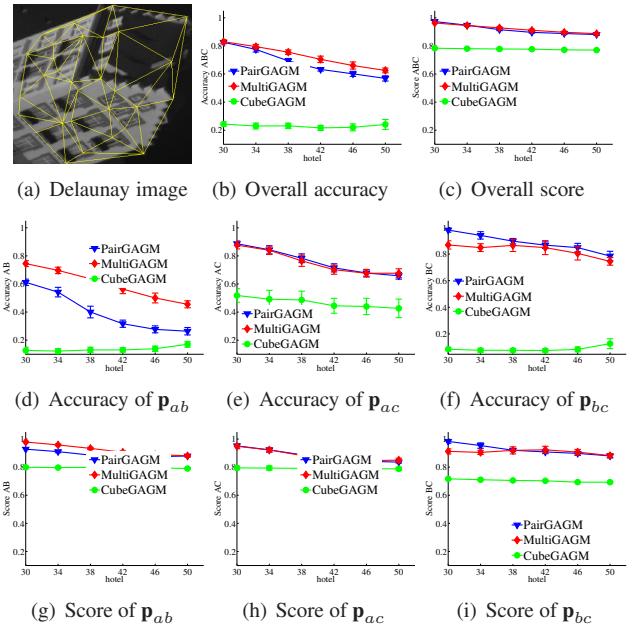
deformation, outliers and edge densities. We also compared the proposed approach with: 1) individual graph matching for each pair by using a particular pair matching algorithm, which might generate inconsistent solutions - in this paper, we term it $\mathbf{p}^{pair}$; and 2) the hypercube framework used in the state-of-the-art [19], which firstly performs a certain pair matching algorithm among graph pairs to build a hypercube, then plus a post-discretization step to obtain the common labelling (can be a sequential greedy algorithm as in this paper or an extended Hungarian method), is performed to obtain the final consistent solutions: $\mathbf{p}^{cube}$. IPFP [10] is chosen as the black-box solver in alternating optimization, in that it is a score non-descending algorithm. And we use different terms: '*Pair*' - method 1), '*Cube*' - method 2) and '*Multi*' - our method, in result illustration.

For each trial in all random graph experiments, the same affinity matrix was shared as the input for the testing algorithms. We build three graphs G_a, G_b and G_c with n_{in} inliers, optional n_{out} outliers and the edges with a density controlled by parameter ρ as the same in [3]. For graph G_a, each edge's attribute d_a^{ij} is assigned by a random value uniformly sampled from [0,1], and the perturbed graph G_b and G_c are created by adding a Gaussian deformation noise ε sampled from $N(0, \sigma^2)$ to d_a^{ij} i.e. $d_b^{ij} = d_a^{ij} + \varepsilon C$ where C is the average of all d_a^{ij} in graph G_a. The affinity matrix is calculated by $\exp(-\|d_a^{ij} - d_b^{xy}\|/\sigma_s^2)$ where ij is the edge between node i and j in one graph and xy is the edge between node x and y in the other graph. σ_s^2 is set to 0.15 same as in [3] for random graph tests. The performances of each method are measured in both accuracy and scores the same as defined in [3]. The maximum iteration in Alg.1 is set to 5. All experiments are repeatedly performed for 30 random trials and the average results together with standard deviations are plotted.

5.2. Observations and analysis

The average accuracy and score out of 30 random trials are shown in Fig.3 for matching across *three* graphs; and Fig.4 for the *four* graph case. The curves are reflecting different degrees of synthetically imposed deformation, outlier, and edge density. For deformation test, the edge density is set to 0.7. For outlier test and density test, we add additional deformation with ε=0.05 and ε=0.2 respectively. The experimental results trigger some discussions:

Discrepancy exists between score and accuracy: solutions with better accuracy may produce lower scores under two particular graphs. One can find such examples in Fig.3, Fig.4 for both three-graph matching and four-graph, especially the disturbance is significant. In our analysis, this might be due to that as graphs are corrupted, the associated local affinity matrix is misleading and makes the individual matching $\mathbf{p}^{pair}$ inaccurate (although being an optimum in score). While our method can propagate the global information from other affinity matrix to compensate such deviation. As a result, it jointly optimizes the mapping variables by simultaneously exploring the affinity matrix across pairs, and become less sensitive to the local noise.



(a) Delaunay image (b) Overall accuracy (c) Overall score

(d) Accuracy of $\mathbf{p}_{ab}$ (e) Accuracy of $\mathbf{p}_{ac}$ (f) Accuracy of $\mathbf{p}_{bc}$

(g) Score of $\mathbf{p}_{ab}$ (h) Score of $\mathbf{p}_{ac}$ (i) Score of $\mathbf{p}_{bc}$

Figure 2: Performance evaluation on CMU hotel sequence dataset with increasing frame gap. Delaunay triangulation is used for graph construction on each raw image as exemplified in the left top sub-graph. GAGM [7] is adopted as the solver for pairwise graph matching.

Accuracy is boosted especially in the presence of outliers and deformation: as the alternating optimization algorithm works in an expectation maximization fashion, the local solution quality in each iteration will largely influence the overall performance and IPFP is found the most competitive one regarding accuracy, in all testing as shown in synthetic-graph matching. Due to the space limitation, in this paper we only provide the results using IPFP as the black-box pair matching solver.

Individual pair matching is inconsistent: Table 1 shows the counts of the trials when Eq. (8) holds, which is out of the 30 trials on three-graph test in terms of deformation, outlier and edge density as shown in Fig.3. As the disturbance grows, it becomes more difficult for the individual two-view matching solutions $\mathbf{p}_{ab}^{pair}$, $\mathbf{p}_{ac}^{pair}$, $\mathbf{p}_{bc}^{pair}$ to satisfy the establishment of Eq.8.

5.3. Results across image sequence

The proposed method is also evaluated against other methods on the CMU *Hotel* and *House* sequences which has been extensively used in many work [3, 5, 8, 22] etc. 30 landmark feature points were manually tracked and labeled across all frames on which Delaunay triangulation is

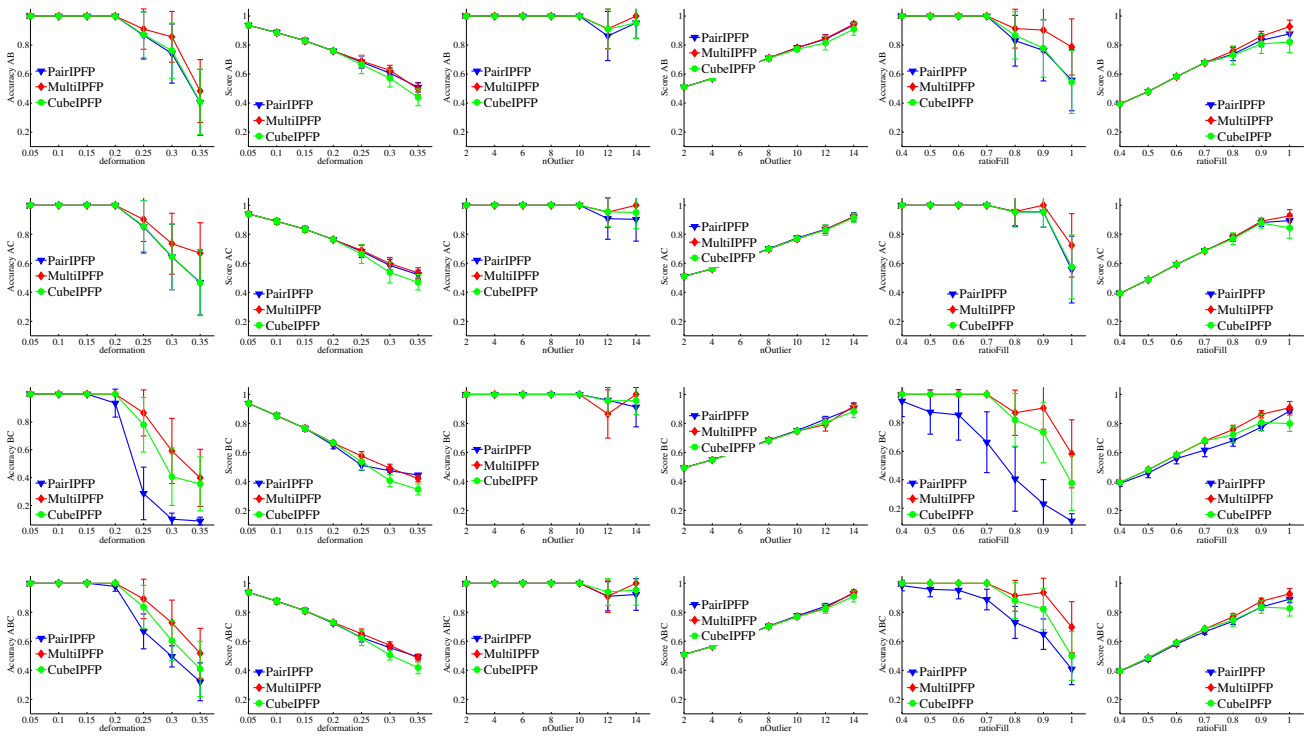


Figure 3: Performance evaluation for matching three synthetic graphs by varying deformation, outlier #, and edge density: average and deviation of accuracy and score out of 30 random trials. Row 1: accuracy and score of $\mathbf{p}_{ab}$; Row 2: accuracy and score of $\mathbf{p}_{ac}$; Row 3: accuracy and score of $\mathbf{p}_{ab}$; Row 4: accuracy and score of mean of $\mathbf{p}_{ab}$,$\mathbf{p}_{ac}$,$\mathbf{p}_{bc}$.

performed to obtain the graph structure. The affinity matrix is set by $\exp(-\|d_a^{ij}\text{-}d_b^{xy}\|/\sigma_s^2)$ where d_a^{ij} (or d_b^{xy}) is the Euclidean distance. To better differentiate the methods, we select the Graduate assignment method [7] as the pair-matching solver as its mechanism is quite different from the IPFP method used in the synthesized test thus the proposed framework can be better verified. And we set the scale parameter σ_s^2 = 0.15 after normalizing the point coordinate between [0,1], as the same with the synthetic experiment. As shown in Fig.2, we vary the sequence gap gap between three frames to verify the robustness of the compared methods. Given frame t for the first graph, the other two graphs are selected from frame $t+0.75*gap$ and $t+1.5*gap$ such that the average sequence gap between two graphs is gap.

6. Conclusion

We propose a novel formulation for robust and consistent multiple graph matching. The merits lie in the extension of the conventional pair matching formulation, and seamless reuse of existing pair matching solvers. Experimental results show the competitiveness of the proposed method.

References

[1] B. Bonev, F. Escolano, M. Lozano, P. Suau, M. Cazorla, and W. Aguilar. Constellations and the unsupervised learning of graphs. In *GbRPR*, 2007.

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

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

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

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

[6] A. Egozi, Y. Keller, and H. Guterman. A probabilistic approach to spectral graph matching. *IEEE Transactions on PAMI*, pages 18–27, 2013.

[7] S. Gold and A. Rangarajan. A graduated assignment algorithm for graph matching. *IEEE Transaction on PAMI*, 1996.

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

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

[10] M. Leordeanu and M. Herbert. An integer projected fixed point method for graph matching and map inference. In *NIPS*, 2009.

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

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

[13] W. Lian and L. Zhang. Robust point matching revisited: A concave optimization approach. In *ECCV*, 2012.

[14] J. Maciel and J. Costeira. A global solution to sparse correspondence problems. *IEEE Transaction on PAMI*, 2003.

[15] S. Ribalta and F. A. Serratosa. A structural and semantic probabilistic model for match- ing and representing a set of graphs. In *GbRPR*, 2009.

[16] A. Sanfeliu, F. Serratosa, and R. Alquézar. Second-order random graphs for modeling sets of attributed graphs and their application to object learning and recognition. *IJPRAI*, 2004.

[17] F. Serratosa, R. Alquézar, and A. Sanfeliu. Function-described graphs for modeling objects represented by attributed graphs. *Pattern Recognition*, 2003.

[18] A. Solé-Ribalta and F. Serratosa. On the computation of the common labelling of a set of attributed graphs. In *CIARP*, 2009.

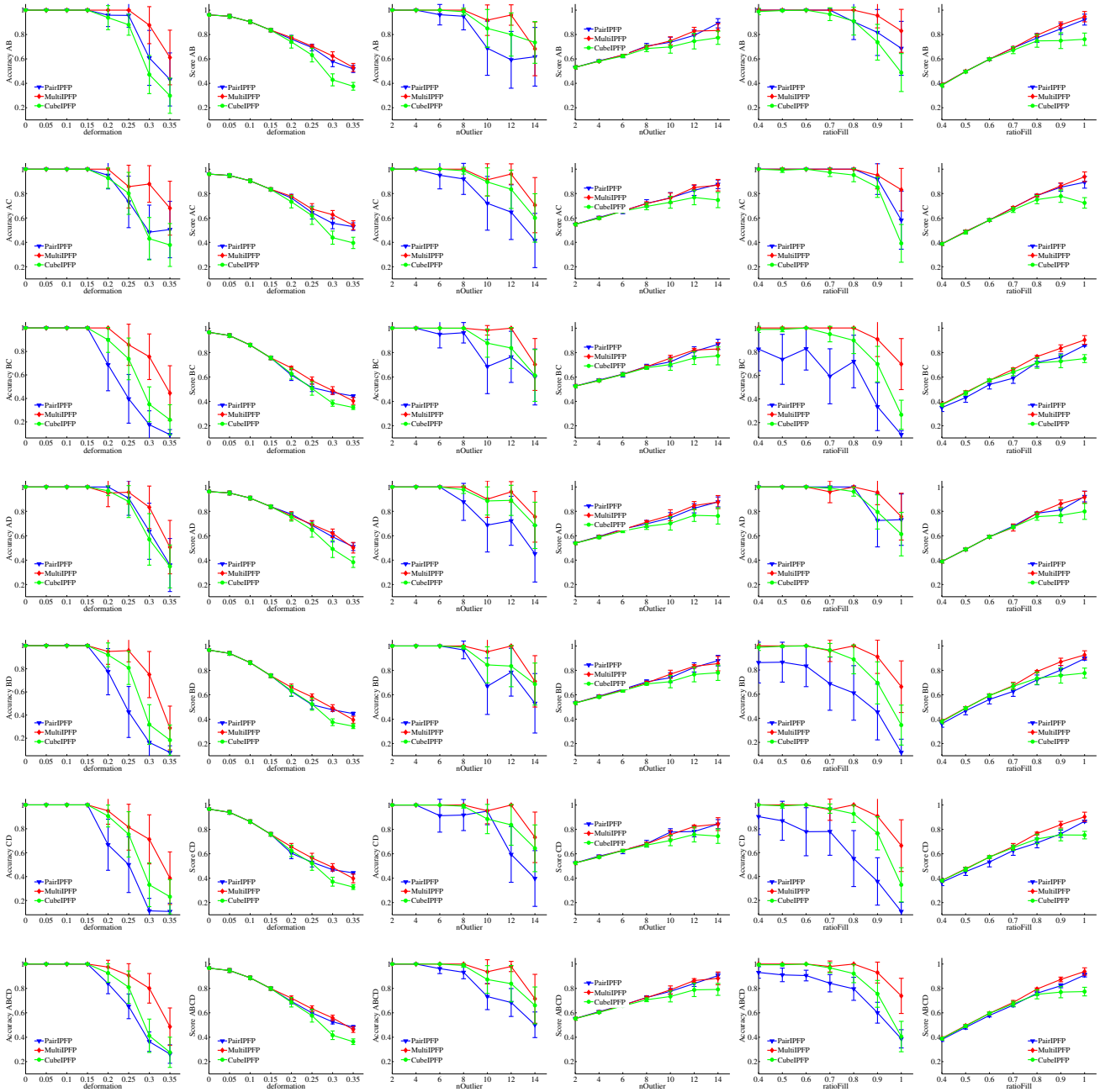


Figure 4: Performance evaluation for matching four synthetic graphs by varying deformation, outlier #, and edge density: average and deviation of accuracy and score out of 30 random trials. Row 1: accuracy and score of $\mathbf{p}_{ab}$; Row 2: accuracy and score of $\mathbf{p}_{ac}$; Row 3: accuracy and score of $\mathbf{p}_{bc}$; Row 4: accuracy and score of $\mathbf{p}_{ad}$; Row 5: accuracy and score of $\mathbf{p}_{bd}$; Row 6: accuracy and score of $\mathbf{p}_{cd}$; Row 7: accuracy and score of mean of $\mathbf{p}_{ab}$,$\mathbf{p}_{ac}$,$\mathbf{p}_{bc}$,$\mathbf{p}_{bc}$,$\mathbf{p}_{bd}$,$\mathbf{p}_{cd}$.

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

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

[21] 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.

[22] L. Torresani, V. Kolmogorov, and C. Rother. Feature correspondence via graph matching: Models and global optimization. In *ECCV*, 2008.

[23] B. van Wyk and M. van Wyk. A pocs-based graph matching algorithm. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 2004.

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

[25] A. Wong and M. You. Entropy and distance of random graphs with application to structural pattern recognition. *IEEE Transactions on PAMI*, 1985.

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