Graduated Consistency-Regularized Optimization for Multi-graph Matching

Junchi Yan[1,2], Yin Li[3], Wei Liu[4],
Hongyuan Zha[3,5], Xiaokang Yang[1], and Stephen Mingyu Chu[2]

[1] Shanghai Jiao Tong University, 200240 Shanghai, China
yanjunchi@sjtu.edu.cn
[2] IBM Research - China, 201203 Shanghai China
[3] Georgia Institute of Technology, 30032 Atlanta, USA
[4] IBM T.J. Watson Research Center, 10598 New York, USA
[5] East China Normal University, 200062 Shanghai, China

Abstract. Graph matching has a wide spectrum of computer vision applications such as finding feature point correspondences across images. The problem of graph matching is generally NP-hard, so most existing work pursues suboptimal solutions between two graphs. This paper investigates a more general problem of matching N attributed graphs to each other, i.e. labeling their common node correspondences such that a certain compatibility/affinity objective is optimized. This multi-graph matching problem involves two key ingredients affecting the overall accuracy: a) the pairwise affinity matching score between two local graphs, and b) global matching consistency that measures the uniqueness and consistency of the pairwise matching results by different sequential matching orders. Previous work typically either enforces the matching consistency constraints in the beginning of iterative optimization, which may propagate matching error both over iterations and across different graph pairs; or separates score optimizing and consistency synchronization in two steps. This paper is motivated by the observation that affinity score and consistency are mutually affected and shall be tackled jointly to capture their correlation behavior. As such, we propose a novel multi-graph matching algorithm to incorporate the two aspects by iteratively approximating the global-optimal affinity score, meanwhile gradually infusing the consistency as a regularizer, which improves the performance of the initial solutions obtained by existing pairwise graph matching solvers. The proposed algorithm with a theoretically proven convergence shows notable efficacy on both synthetic and public image datasets.

1 Introduction

Due to the powerful characteristics of abstraction, Graph Matching (GM) plays a central role in tremendous computer vision applications such as image registration [39], object categorization [30], action recognition [5], shape matching [29], stereo [20,46], and so forth. The problem of graph matching is to establish a compatible node-to-node mapping among two or more graphs. In computer vision, GM has primarily been used to find correspondences among two or more sets

D. Fleet et al. (Eds.): ECCV 2014, Part I, LNCS 8689, pp. 407–422, 2014.
© Springer International Publishing Switzerland 2014

of local features extracted from images. Different from conventional matching methods such as RANSAC [15] and Iterative Closest Point (ICP) [4], GM goes beyond the first-order node-wise feature as well as the focus of location information [51], which incorporates more distinctive pairwise [43,47] or higher-order [50,13,28,33] node interactions for matching structural objects. Consequently, GM has attracted considerable research attention [12,17,22,34,16] for decades, yet remains computationally challenging due to its combinatorial nature.

Current graph matching methods mostly focused on the two-graph scenario such as [21,38,9,41,52,43,10,8,47]. In particular, there are several research groups focusing on graph matching and related problems over the past years, such as Professor Horst Bunke's group in Bern [3], Professor Edwin R. Hancock's group in York [2], and Professor Francesc Serratosa's group in Tarragona [1], to name a few. Readers are referred to the recent comparison review paper [22] for more details therein. However, on one hand, many practical computer vision tasks need matching multiple images or point sets, which is a building block for various applications that involve multi-view registration or matching. On the other hand, it is generally recognized [47,44,40,36] that simultaneously exploring all pairwise affinity information across the whole pool of graphs $\mathbb{G} = \{G_1, \cdots, G_N\}$ may improve the matching accuracy. Such an improvement is accomplished through avoiding trapping to local optima, or a false optima away from the semantic true correspondence due to the large deformation, appearing in the pairwise case since only affinity between two local graphs is explored. Therefore, it is appealing to design effective and efficient multi-graph matching algorithms beyond conventional pairwise matching solvers.

The multi-graph matching problem has been basically solved in a sequential manner [37], where each step executes a pairwise matching of two graphs. Ideally, the pairwise matching sequence can be designed by different orders that cover all graphs in a path, *e.g.*, $G_1 \rightarrow G_2 \rightarrow \ldots \rightarrow G_N$. However, whatever path order is chosen, a single error in the corresponding sequence will typically create a large number of erroneous pairwise matches. To fully explore the information across the whole graph pool $\mathbb{G}$, it is perhaps more robust to compute all or part of pairwise matching results independently, and then leave the calculation of the final solution to several post-steps [36,24]. Compared with computing a pairwise matching chain like $G_1 \rightarrow G_2 \rightarrow \ldots \rightarrow G_N$, such an exhaustive matching strategy would cause the problem of *redundancy*, or in another word, *inconsistency*, as the node mapping between two graphs cannot be uniquely determined by different pairwise matching paths. Formally, we use the term "cycle-consistency" as introduced and described in [48,26,24], i.e., that composition of correspondences between two graphs should be independent of the connecting path chosen.

2 Problem Statement

2.1 Graph Matching Formulations

For self-completeness, first we briefly introduce the widely used objective function of graph matching in the context of two graphs. Concretely, given two

graphs $G_1(V_1, E_1, A_1)$ and $G_2(V_2, E_2, A_2)$, where V denotes nodes, E, edges and A, attributes, there is an affinity matrix $\mathbf{K} \in \mathbb{R}^{n_1 \times n_2}$, whereby its elements $K_{ia;jb}$ measure the affinity with the edge pair candidate $\{v_1(i), v_1(j)\}$ vs. $\{v_2(a), v_2(b)\}$. The diagonal elements $K_{ia;ia}$ represent the unary affinity of a node pair candidate $\{v_1(i), v_2(a)\}$. By introducing a permutation[1] matrix $\mathbf{P} \in \{0,1\}^{n_1 \times n_2}$ whereby P_{ia}=1 if node $v_1(i)$ matches node $v_2(a)$ (P_{ia}=0 otherwise). It can be concisely formulated as a constrained quadratic assignment problem (QAP [35]). It goes beyond the linear assignment problem that can be efficiently solved by the Hungarian method [27]. In general, the QAP is known to be NP-hard [18].

$$\mathbf{P}^* = \arg\max_{\mathbf{P}} \mathrm{vec}(\mathbf{P})^T \mathbf{K} \mathrm{vec}(\mathbf{P}) \quad s.t. \quad \mathbf{I}_{n_1}^T \mathbf{P} = \mathbf{1}_{n_2}^T, \mathbf{P}\mathbf{I}_{n_2} = \mathbf{1}_{n_1}, \mathbf{P} \in \{0,1\}^{n_1 \times n_2}.$$

Here $\mathrm{vec}(\mathbf{P})$ is the vectorized permutation matrix of $\mathbf{P}$, and $\mathrm{vec}(\mathbf{P})^T$ is the transpose version. The constraints refer to the one-to-one node bijection.

One step further, now we consider the formulation for the multi-graph matching. Given N graphs and the pairwise affinity matrix $\mathbf{K}_{ij}(i, j = 1, 2, \ldots, N; i>j)$, a natural extension for multi-graph matching as used in [47] is:

$$\{\mathbf{P}_{ij}\}^* = \arg\max_{\{\mathbf{P}_{ij}\}} \sum_{i,j=1,2,\ldots,N;i>j} \mathrm{vec}(\mathbf{P}_{ij})^T \mathbf{K}_{ij} \mathrm{vec}(\mathbf{P}_{ij}) \tag{1}$$

$$s.t. \quad \mathbf{I}_{n_1}^T \mathbf{P}_{ij} = \mathbf{1}_{n_2}^T \quad \mathbf{P}_{ij}\mathbf{I}_{n_2} = \mathbf{1}_{n_1} \quad \mathbf{P}_{ij} \in \{0,1\}^{n_i \times n_j} \quad \forall i, j = 1, 2, \ldots, N; \quad i > j$$

where $\{\mathbf{P}_{ij}\}$ is the set of pairwise permutation matrix over N graphs. Note that for notational simplicity, without loss of generality, here we omit the weight λ_{ij} for each quadratic term in the objective function as used in [47].

2.2 Notations and Preliminaries

We first give several notations and definitions which will be used throughout this paper. Without loss of generality, given a set of N graphs $\mathbb{G} = \{G_k, \sum_{k=1}^{N}\}$, let the assignment matrix $\mathbf{P}_{ij} \in \mathbb{R}^{n_i \times n_j}$ denote the node-to-node mapping between G_i and G_j. We also define the pairwise affinity matching score J_{ij} that measures the similarity between two aligned graphs as $J_{ij} = \mathrm{vec}(\mathbf{P}_{ij}^T)\mathbf{K}_{ij}\mathrm{vec}(\mathbf{P}_{ij})$. This score definition is widely used in graph matching related work, such as [9,52,43,10,8,47]. By stacking all $\mathbf{P}_{ij}$ into one whole matrix, we define the resulting *matching configuration matrix*[2] as: $\mathbf{W} = \begin{pmatrix} \mathbf{P}_{11} & \cdots & \mathbf{P}_{1N} \\ \vdots & \ddots & \vdots \\ \mathbf{P}_{N1} & \cdots & \mathbf{P}_{NN} \end{pmatrix} \in \mathbb{R}^{nN \times nN}$.

[1] We use"assignment matrix"/"permutation matrix" interchangeably in the paper.

[2] For multi-graph matching tackled in this paper, we seek the common inlier node set (via the one-to-one node matching) shared among all graphs, and admit outliers appearing in graphs i.e. $n_i = n_i^{in} + n_i^{out}$ for each graph G_i. In general, in the case of $n_i \neq n_j$, the variable $\mathbf{P}_{ij}$ in Eq. (1) is a partial assignment matrix. Similar to [52,21,40], we transform the partial assignment matrix $\mathbf{P}$ to a full square assignment matrix by augmenting proper dummy nodes (adding slack variables into $\mathbf{P}_{ij}$ accordingly), so that the one-to-one two-way constraints can always be satisfied. This is a conventional strategy used in solving linear programming, and also is previously adopted by graph matching work such as [45,14,40]. Thus throughout the paper from now on, we assume $n_i = n_j = n$ for all graphs.

Now we introduce several definitions that will facilitate the presentation of the paper: Definition (1) and (2) are used to quantify the consistency metric related to the proposed Alg.2 as we will show later. Definition (3) is regarding with using Maximum Span Tree for multi-graph matching and related to the concept of matching path in Definition (4), which is related to Alg.1 and Alg.2.

Definition 1. *Given a set of graphs* $\mathbb{G}=\{G_k, \sum_{k=1}^{N}\}$*, the consistency of graph* G_k *is defined as* $C(G_k)=1-\frac{\sum_{i=1}^{N-1}\sum_{j=i+1}^{N}(\|\boldsymbol{P}_{ij}-\boldsymbol{P}_{ik}\boldsymbol{P}_{kj}\|_F/2}{n(N-2)(N+1)/2}$ *where* $\boldsymbol{P}_{ij}$ *is the pairwise assignment matrix over the graph set* $\mathbb{G}$*.*

Definition 2. *Given a set of graphs* $\mathbb{G}=\{G_k, \sum_{k=1}^{N}\}$ *and its pairwise assignment matrix set* $\{\boldsymbol{P}_{ij}\}$*, we call the graph set* $\mathbb{G}$ *is fully consistent w.r.t. its pairwise matching configuration* $\boldsymbol{W}$ *if* $\forall i,j,k \leq N$*:* $\boldsymbol{P}_{ij}=\boldsymbol{P}_{ik}\boldsymbol{P}_{kj}$*. Specifically, the consistency measure of* $\mathbb{G}$ *is defined as* $C(\mathbb{G})=\frac{\sum_{k=1}^{N}C(G_k)}{N}$*. Thus* $C(\mathbb{G})=1$ *if and only if* $\mathbb{G}$ *is fully consistent.*

Definition 3. *Given a set of graphs* $\{G_k, \sum_{k=1}^{N}\}$ *and* $\boldsymbol{P}_{ij}$*, the super graph* $\mathcal{G}$ *is defined as an undirected weighted graph such that each node* k *denotes* G_k*, and its edge* e_{ij} *is weighted by the* $J_{ij} = vec(\boldsymbol{P}_{ij})^T \boldsymbol{K}_{ij} vec(\boldsymbol{P}_{ij})$*.*

Definition 4. *The matching path* $T_{ij}(k_1, k_2, \ldots, k_s)$ *from graph* G_i *to* G_j *is defined as a loop-free chain of graphs* $G_i \rightarrow G_{k_1} \rightarrow \cdots \rightarrow G_{k_s} \rightarrow G_j$ *which induces the multiplication of the pairwise matching solutions:* $\boldsymbol{H}_{ij}(k_1, k_2, \ldots, k_s) \triangleq \boldsymbol{P}_{ik_1}\boldsymbol{P}_{k_1k_2}\ldots\boldsymbol{P}_{k_sj}$*. Its order* s *is further defined as the number of the intermediate graphs between the two ending graphs* G_i, G_j*. The score* $J_{ij}(k_1, k_2, \ldots, k_s)$ *of the path is induced by the chaining matching solution* $\boldsymbol{H}_{ij}(k_1, k_2, \ldots, k_s)$ *as* $J_{ij}(k_1, k_2, \ldots, k_s) = vec(\boldsymbol{H}_{ij})^T \boldsymbol{K}_{ij} vec(\boldsymbol{H}_{ij})$*.*

Comments. By assuming full consistency of Definition (2), solving the multi-matching problem reduces to finding $N-1$ different assignment matrix, rather than $\mathrm{O}(N^2)$ that cover all pairwise cases. The consistency measurement regarding a single graphs as defined in Definition (1) reflects its consistency contribution. The super graph is considered as a fully connected graph when each pair of graphs is matched by some means ($e_{ij} > 0$) or set to zero if the pairwise matching is unknown. In general, given a connected super graph (not necessarily fully connected), a *maximum spanning tree* (MST) [19] can be found with more or equal to the weight of every other spanning tree on the super graph.

3 Related Work

As a general problem for matching structural data, graph matching has been extensively studied for decades not only in computer vision, but also in computer science and mathematics [12]. Here we view the problem from several key aspects that account for the main threads of the related work.

Machine Learning for GM: Conventional graph matching methods first compute an affinity matrix and keep the graph unchanged during the entire matching process. Recent work leverage various leaning algorithms for estimating the optimal affinity matrix [6,32,33,23,8], and the methods can fall into either supervised [6] or unsupervised [32] or semi-supervised [33] learning paradigms.

Higher-Order Affinity Modeling: Combing the unary and pairwise edge information has been heavily investigated since such types of matching schemes play a good tradeoff between computational complexity and representation capability [9,21,29,31,42]. More recently, the higher-order (most are third order) information has been encoded to achieve more robust matching paradigms. Several representative hypergraph matching methods have been developed for computer vision applications [7,13,28,33,50] that encode higher-order information to enhance the structural distinctiveness for matching.

Optimization Methods: Most approaches first formulate an objective function, and then employ certain optimization methods to derive optimal solutions [21,29,42], which vary among a wide spectrum of optimization strategies [11]. Some recent work first relax the objective function to convex-concave formulation [52,49]. Then the optimal solutions are achieved using the so-called path following strategy and a modified version of the Frank-Wolfe algorithm [49]. Probabilistic matching paradigms are also developed, which have shown unique power in interpreting and addressing hypergraph matching problems [9,28,50].

Surprisingly, little work in computer vision community has been done for *simultaneously* addressing both matching consistency and matching affinity. The most recent methods still concentrate on one single aspect of the problem - either aiming at maximizing/minimizing the matching affinity score/cost by using a *reference graph* to ensure the matching consistency, like [47,40], or addressing the problem from a spectral smoothing perspective to enhance consistency like [36] while ignoring the affinity score between two local graphs. More specifically, Sole-Ribalta and Serratosa [40] extend the classical Graduated Assignment Graph Matching (GAGM) algorithm [21,43] from pairwise to the multi-graph case, which inherits the robustness of the original method, yet meanwhile, being less efficient as it repeatedly applies the GAGM method across graph pairs iteratively. Yan et. al. [47] propose an iterative optimization method that imposes the rigid matching consistency constraint via a closed form in each iteration. The recent work [36] employs spectral analysis and approximation to eigenvector decomposition on the matching configuration matrix comprised of all initial pairwise matching solutions, and recover the consistent matching solutions. In this paper, we formulate the multi-graph matching problem as a novel graduated regularized optimization procedure, and solve it using an iterative first-order approximating algorithm regarding with the affinity score.

The *main contribution* of this paper lies in the study of the correlation score between affinity score and matching accuracy decays as the score value increases, while consistency becomes a more informative regularizer. Based on this observation, a novel graduated consistency-regularization method is proposed to effectively improve the accuracy from the initial pairwise matching solutions.

4 Graduated Consistency-Regularized Approximating

4.1 Maximizing Matching Score under Consistency Constraints

Note that the formulation (1) does not automatically account for the matching consistency. By introducing the consistency measure associated with the configuration matrix $\mathbf{W}$ for graph set $\mathbb{G}$ as defined in Definition (2), we present the consistency-constrained version of the objective function:

$$\{\mathbf{P}_{ij}\}^* = \arg\max_{\{\mathbf{P}_{ij}\}} \sum_{i,j=1,2,\ldots,N;i>j} \mathrm{vec}(\mathbf{P}_{ij})^T \mathbf{K}_{ij} \mathrm{vec}(\mathbf{P}_{ij}) \tag{2}$$

$$\mathbf{I}_{n_1}^T \mathbf{P}_{ij} = \mathbf{1}_{n_2}^T \quad \mathbf{P}_{ij}\mathbf{I}_{n_2} = \mathbf{1}_{n_1} \quad \mathbf{P}_{ij} = \mathbf{P}_{ik}\mathbf{P}_{kj} \in \{0,1\}^{n_i \times n_j} \quad \forall k = 1, 2, \ldots, N; k \neq i, j$$

We first transform the above objective to a "Lagrange-multiplier" form, which is more suitable for the problem as we will show later:

$$\{\mathbf{P}_{ij}\}^* = \arg\max_{\{\mathbf{P}_{ij}\}} \sum_{i,j=1,2,\ldots,N;i>j} \mathrm{vec}(\mathbf{P}_{ij})^T \mathbf{K}_{ij} \mathrm{vec}(\mathbf{P}_{ij}) + \lambda \sum_{k=1}^{N} \|\mathbf{P}_{ij} - \mathbf{P}_{ik}\mathbf{P}_{kj}\|_F \tag{3}$$

$$\mathbf{I}_{n_1}^T \mathbf{P}_{ij} = \mathbf{1}_{n_2}^T \quad \mathbf{P}_{ij}\mathbf{I}_{n_2} = \mathbf{1}_{n_1} \quad \mathbf{P}_{ij} = \mathbf{P}_{ik}\mathbf{P}_{kj} \in \{0,1\}^{n_i \times n_j} \quad \forall k = 1, 2, \ldots, N; k \neq i, j$$

Note that we do not move the one-to-one bijection constraint to the "Lagrange-multiplier" term, this is due to existing pairwise graph matching solvers are able to handle this constraint, such as relaxing to the permutation matrix's convex hull - a doubly stochastic matrix e.g. [9].

The above objective function is challenging due to NP-hard, and even harder as more consistency constraints need being satisfied here. Thus we are more interested in devising efficient approximating algorithms, which is based on two rationales to the specific matching problem as we will show in the following.

4.2 Maximizing Pairwise Score via Approximating Path Selection

Our first key rationale is that the highest-score matching between two graphs may be found along a higher-order path (refer to Definition 4) instead of the direct (zero-order) pairwise matching. We formalize this idea as follows:

For graph G_i, G_j, all possible (loop-free) matching pathes can form the following (loop-free) path set $\mathbb{T}_{ij} = \{T_{ij}(k_1, \ldots, k_s), s = 1, 2, \ldots, N-2\}$. Its cardinality $|\mathbb{T}_{ij}| = \sum_{s=1}^{N-2} s!$. Thus exhaustively searching for the best solution T_{ij}^* is intimidating as the complexity is exponential in terms of N.

One alternative approach is to approximate T_{ij}^* by a set of consecutive first-order iterations, which involves concatenating the first-order short pathes that are chosen in each iteration, into a higher-order path. This idea is formally described in Alg.1: in each iteration s, the pairwise matching solution $\mathbf{P}_{ij}$ is updated by the highest-score one among the confined *subset* of $\mathbb{T}_{ij}$ with path order $s = 1$ (See line 8-10 in the algorithm chart). This algorithm bears solid convergence property due to the score-ascending procedure will always converge to a fixed value because the score is bounded in the discrete permutation matrix space. In general, such a greedy algorithm cannot ensure the global optimality

Algorithm 1. Iterative Approximating Pairwise Affinity Maximization

Input

One set of N graphs with n nodes: $\mathbf{V}_i = \{v_1, v_2, \ldots, v_n\}, (i = 1, 2, \ldots, N)$;
Pairwise affinity matrix $\mathbf{K}_{ij} (i = 1, 2, \ldots, N; j = i + 1, \ldots, N)$;
Maximum iteration count: S_{max}, initial iteration count: $s = 0$;

Output

Consistent matching configuration matrix $\mathbf{W} \in \mathbb{R}^{Nn \times Nn}$;

Procedure

Perform pairwise graph matching to obtain the putative assignment matrix $\mathbf{P}_{ij}^{(0)} \in \mathbb{R}^{n \times n}$ and the matching configuration matrix $\mathbf{W}^{(0)} \in \mathbb{R}^{Nn \times Nn}$;
Calculate the initial total score $J^{(0)}$ by $J^{(0)} = \sum_{i=1,j=i+1}^{N,N} \mathrm{vec}(\mathbf{P}_{ij}^{(0)})^T \mathbf{K}_{ij} \mathrm{vec}(\mathbf{P}_{ij}^{(0)})$;
while $s \leq S_{max}$ **do**
  for all $i = 1, 2, \ldots, N; j = i + 1, \ldots, N$ **do**
    Update $\mathbf{P}_{ij}^{(s)} = \mathbf{P}_{ik}^{(s-1)} \mathbf{P}_{kj}^{(s-1)}$ for k to maximize $\mathrm{vec}(\mathbf{P}_{ij}^{(s)})^T \mathbf{K}_{ij} \mathrm{vec}(\mathbf{P}_{ij}^{(s)})$
  end for
  Calculate the total score $J^{(s)} = \sum_{i=1,j=i+1}^{N,N} \mathrm{vec}(\mathbf{P}_{ij}^{(s)})^T \mathbf{K}_{ij} \mathrm{vec}(\mathbf{P}_{ij}^{(s)})$;
  If $J_s = J_{s-1}$, **break**;
  s++;
end while
Update $\mathbf{W}^{(s)}$ by updating all $\mathbf{P}_{ij}^{(s)}$;
Impose full consistency by spectral smoothing method [36] when $N > n$, or via building a maximum span tree on the super graph [25] when $N <= n$.

for affinity score. The following proposition shows under certain conditions, Alg.1 can guarantee the global optimality. Proving its correctness is trivial.

Proposition 1. *For any G_i, G_j out of a fully connected super graph, denote the "highest-score" path as T_{ij}^*. If for any sub-segment $\widetilde{T}_{kl}^*$ of T_{ij}^*, it is also the "highest-score" path for the two ending graph G_k, G_l, i.e. $\widetilde{T}_{kl}^* = T_{kl}^*$, then Alg.1 is ensured to find the global optimum for the initial configuration $\boldsymbol{W}$.*

To satisfy the full consistency as defined in Definition (2), after iteration, a synchronization post-step [36] is performed (See the last step in Alg.1). This step can also involve other smoothing methods especially when a small number of graphs are used for matching ($n > N$). A simple alternative is finding maximum spanning tree on the super graph induced by $\mathbf{W}^{(s)}$ and populate other edges (assignment matrix) by multiplication through the maximum span tree.

The above discussion is concretized into an iterative algorithm as described in Alg.1: **Iterative Approximating Pairwise Affinity Maximization (IA-PAM)**. In each iteration, it approximates the best path w.r.t. pairwise matching score between two graphs by the best solution out of the first-order path. This algorithm is ensured to converge when the score stops increasing. Then the post-step of global consistency synchronization is performed by either a spectral smoothing method [36] or maximum span tree as discussed above. The efficacy of such a score-ascending strategy can be exemplified by a concrete analysis:

Algorithm 2. Graduated Consistency-Regularized Affinity Maximization

Input

One set of N graphs with n nodes: $\mathbf{V}_i = \{v_1, v_2, \ldots, v_n\}, (i = 1, 2, \ldots, N)$;
Pairwise affinity matrix $\mathbf{K}_{ij} (i = 1, 2, \ldots, N; j = i + 1, \ldots, N)$;
Maximum iteration count: S_{max}, initial iteration count: $s = 0$;
Consistency weight initialization $\lambda = \lambda_0$ and the weight increasing rate $\rho > 1$;

Output

Consistent matching configuration matrix $\mathbf{W} \in \mathbb{R}^{Nn \times Nn}$;

Procedure

Perform pairwise graph matching to obtain the putative assignment matrix $\mathbf{P}_{ij}^{(0)} \in \mathbb{R}^{n \times n}$ and the matching configuration matrix $\mathbf{W}^{(0)} \in \mathbb{R}^{Nn \times Nn}$;
Set the constant score scale reference $J_{max}^{(0)} = \max_{\mathbf{P}_{ij}^{(0)} \in \mathbf{W}^{(0)}} \mathrm{vec}(\mathbf{P}_{ij}^{(0)})^T \mathbf{K}_{ij} \mathrm{vec}(\mathbf{P}_{ij}^{(0)})$;
while $s \leq S_{max}$ **do**
  for all $i = 1, 2, \ldots, N; j = i + 1, \ldots, N$ **do**
    Set $\mathbf{P}_{ij}^{(s)} = \mathbf{P}_{ik}^{(s-1)} \mathbf{P}_{kj}^{(s-1)}$ for k to maximize the regularized objective: $C(G_k) + (1 - \lambda) \mathrm{vec}(\mathbf{P}_{ij}^{(s)})^T \mathbf{K}_{ij} \mathrm{vec}(\mathbf{P}_{ij}^{(s)}) / J_{max}^{(0)}$
  end for
  Increase λ by $\lambda^{(s+1)} = \rho \lambda^{(s)}$;
  Calculate the total score $J^{(s)} = \sum_{i=1, j=i+1}^{N, N} \mathrm{vec}(\mathbf{P}_{ij}^{(s)})^T \mathbf{K}_{ij} \mathrm{vec}(\mathbf{P}_{ij}^{(s)}) / J_{max}^{(0)}$;
  s++;
end while
Update $\mathbf{W}^{(s)}$ by updating all $\mathbf{P}_{ij}^{(s)}$;
Impose full consistency by spectral smoothing method [36] when $N > n$, or via building a maximum span tree on the super graph [25] when $N <= n$.

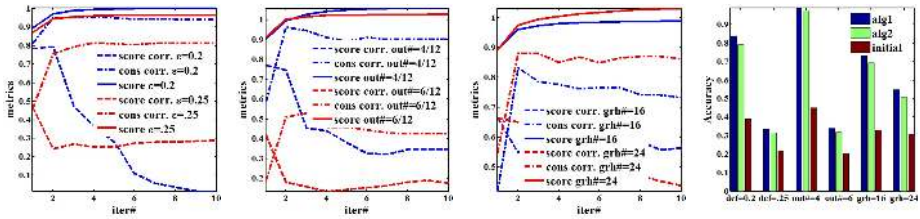
when matching graph G_i and G_j is ambiguous due to both are deformed, it is still possible to recover the correct matching by an intermediate high quality graph that is able to find perfect matches respectively.

4.3 Graduated Consistency-Regularized Optimization Algorithm

Our second key rationale is viewing the consistency constraint as a *regularizer* for affinity score maximization. Note that maximizing pairwise matching score among all pairs cannot ensure the consistency constraint. Moreover, due to outliers and local deformation and the difficulty in setting up the affinity matrix in a parametric manner[3], there can be a case that for some pairs of graphs, the ground truth matching may not produce the highest score. Thus purely maximizing the overall matching score is biased to accuracy.

As a baseline method, Alg.1 separates score maximization and consistency synchronization into two separate steps. It is yet appealing to tackle the two aspects jointly. We make the following statements, for devising a novel algorithm that gradually introduces consistency during the score-ascending procedure:

[3] Currently the affinity function is mostly modeled by parametric functions, the fixed parameters by whatever manual setting [43], or automatically learned from training samples [6,8] etc. may still be unable to best fit the score with accuracy.



(a) Deformation ε (b) Outlier # (c) Graph # (d) Accuracy

Fig. 1. Correlation coefficient value as a function of number of iterations of Alg.1 under different synthetic settings on a set of N=24 graphs with n_{in}=8 inliers: a) varying deformation level $\varepsilon = 0.2$ and $\varepsilon = 0.25$; b) varying number of outliers for n_{out}=4 and n_{out}=6; c) varying number of graphs for N=16 and N=24. For outlier test and graph set size test, ε is set to 0.05 and 0.2 respectively. Dot dashed curve is for the correlation coefficient between consistency and accuracy; dashed curve is for score and accuracy. The solid curve is for the normalized score.

- For the initial assignment matrix $\mathbf{W}^{(0)}$ obtained by the pairwise graph matching solver, its scores are more correlated with the true accuracy.
- After rounds of iterations of score-ascending, score becomes less discriminative for accuracy and consistency becomes a more indicative measurement.

The assertions are verified on synthetic tests using Alg.1 as illustrated in Fig.1: As we continue to iterate, the overall normalized matching score (solid curve) increases, and the correlation coefficient between score and accuracy (dashed curve) drops quickly, while the correlation coefficient for consistency and accuracy (dot dashed curve) still remain at a certain level (above 0.4). This observation is consistent across deformation and outlier tests as show in Fig.1(a) and Fig.1(b) respectively. Moreover, Fig.1(c) shows a relatively larger size of graph set will further improve the correlation between consistency and accuracy. In summary, as affinity score increases along iterating, consistency *gradually* becomes more important as a *regularizer* that help dismiss the biased matching resulting from an unfitted affinity function or due to arbitrary local ambiguities.

Thus we infuse the matching consistency in each iteration by a weighted term that accounts for consistency. In line with the observation from Fig.1, its weight λ gradually increases until the procedure exceeds a certain iteration threshold or converges to a fully consistent configuration. To enhance the overall consistency, the similar post-processing can also be conducted which will stop the score growing immediately. This idea is detailed in Alg.2: **Graduated Consistency-Regularized Pairwise Affinity Maximization (GCRPAM)**. We will evaluate the two proposed algorithms in our experiments. Now we present the convergence property of Alg.2 by Proposition (2) as follows.

Proposition 2. *The iteration procedure of Alg.2 is ensured to converge to a fixed matching configuration $\boldsymbol{W}^*$ after a finite number of iterations.*

Intuitively, the matching configuration becomes more consistent as the weight λ dominates over iterations. A rigorous proof is given in below.

Proof. Given two graphs G_i, G_j with n nodes for each graph, define the set of score difference $\{\Delta S_{ij}\}$ as $\Delta S_{ij} = \text{vec}(\mathbf{P})^T\mathbf{K}_{ij}\text{vec}(\mathbf{P}) - \text{vec}(\mathbf{Q})^T\mathbf{K}_{ij}\text{vec}(\mathbf{Q})$, $\forall \mathbf{P}, \mathbf{Q}$, between two possible assignment matrix $\mathbf{P}$, $\mathbf{Q} \in \mathbb{R}^{n\times n}$ in the numerable permutation space. Suppose the largest value of difference is δS_{ij}^{max} which is constant given the fixed $\mathbf{K}_{ij}$. Moreover, for a certain iteration in Alg.2 (Line 10), suppose the the most consistent graph by definition (1) is G_a, and the second largest is G_b, the iteration will finally arrive $C(G_a) - C(G_b) > \frac{(1-\lambda)}{\lambda}\delta S_{ij}^{max}$ as λ increase close enough to 1. Then, the algorithm will converge by always choosing the most consistent graph G_a for updating, as its consistency score will be further improved in next iteration. Thus we finish the proof.

Finally we discuss the computational complexity of the algorithms. In each iteration of Alg.1 and Alg.2, we re-calculate the pairwise score by $\text{vec}(\mathbf{P})^T\mathbf{K}\text{vec}(\mathbf{P})$ to update each graph pair's correspondence by using each of other graphs as the anchor graph. This complexity is $O(N^3)O(J_1)$ where $O(J_1)$ is the overhead for pairwise score computing. In addition, the Alg.2 requires the calculation of consistency for each graph by using it as the anchor graph to compare the new chaining solution with the original direct matching for all graph pairs. Its complexity is $O(N^3)O(J_2)$ where $O(J_2)$ is the overhead for the multiplication of two permutation matrix for calculating a single graph's consistency as defined in 1. Typically the affinity matrix $\mathbf{K}$ is very sparse as Delaunay triangulation is performed to sparsify the edge density. Thus it is usually significantly faster than the worst case $O(n^3)$ - note the assignment matrix is also sparse. In addition, Step 9 in Alg.1 and Step 10 in Alg.2 can also be speeded up: one can first compute all possible $\mathbf{P}_{ij}$ by choosing different intermediate graph G_k. There is a possible case that for $k_1 \neq k_2$, one has $\mathbf{P}_{ik_1}\mathbf{P}_{k_1j} = \mathbf{P}_{ik_2}\mathbf{P}_{k_2j}$. Thus we only need to compute the affinity score once given two equal derived $\mathbf{P}$. This also enables possible local search and hashing mechanism which we leave for future work.

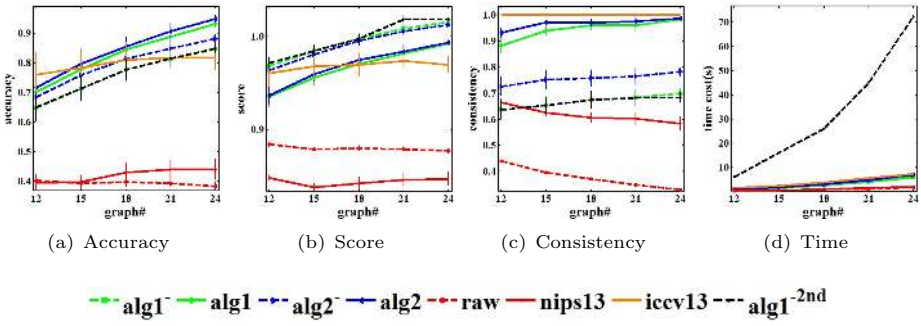
5 Experiments

The experiments involve synthetic simulation and two public real image datasets which follow a standard protocol widely employed by related work such as [9,43] and so forth. We conduct the experiments on a laptop PC with dual cores at 3.02GHz for each. The pairwise graph matching solver is implemented in C++ and the iterative optimization procedure is implemented in Matlab. The comparing methods are [47,36][4] since we focus on the multiple-graph matching problem thus the pairwise matching solvers are in parallel with our work.

5.1 Protocol Description

Graph affinity setting. Following the widely used protocol of [52,47] etc, we use Delaunay method to triangulate the landmarks that are annotated/detected

[4] For space limitation, we did not present the results of [40]. In our test it is slightly better than the pairwise matching in accuracy while being significantly slower.



(a) Accuracy (b) Score (c) Consistency (d) Time

Fig. 2. Synthetic data evaluation via 10 random tests. alg1$^-$ (alg2$^-$) denotes Alg.1 (Alg.2) without post-synchronization, alg1^{-2nd} denotes replacing 1st-order approximation in Alg.1 with 2nd-order path selection. Deformation $\varepsilon = 0.15$, $n_{in} = 10$, $n_{out} = 2$ and edge density $\rho = 1$ (fully connected attributed graphs).

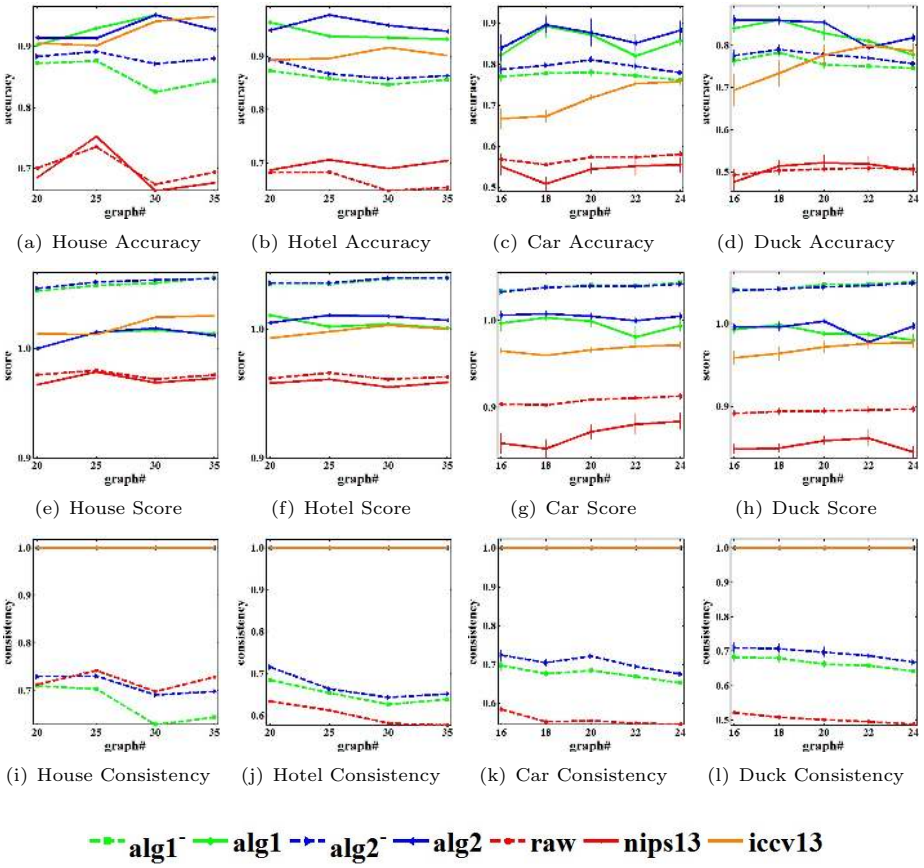
in each image. This setting can also speed up the score calculation in our algorithms as it sparsifies the affinity matrix. The edge length affinity matrix between two graphs is calculated by $K^{edge}_{ij,ab} = e^{-\frac{|q_{ij}-q_{ab}|}{0.15}}$, where q_{ij} (q_{ab}) are the Euclidean distance between two points i, j (a, b) that is further normalized to [0,1] by dividing the largest edge value. For real image test, we further add the edge angle affinity matrix in a similar way such that: $\mathbf{K} = \frac{4}{5}\mathbf{K}^{edge} + \frac{1}{5}\mathbf{K}^{angle}$.

Evaluation Setting. The comparing multi-graph matching methods [47,36] and ours all apply pairwise matching solvers as an out-of-box building block. For space limitation, we focus on Reweighted Random Walks Matching (RRWM) as it has been proven [9] in general more cost-effective[5]. Our methods are set to stop when it converges or exceeds 10 iterations and set the consistency inflating parameter in Alg.2 $\rho = 1.05$ for all tests. There are three main performance metrics: i) accuracy: the number of correctly matched inliers divided by the total number of inliers; ii) average matching affinity score over the whole graph set; iii) consistency as defined in Definition (2). In addition, we testify the comparing methods under two conditions: i) large number of graphs $N > n$; and ii) small number of graphs $N <= n$. Note that for the spectral smooth method [36], it requires the number of graphs shall be larger than the number of nodes, thus for the second case, we build a maximum span tree on the super-graph.

5.2 Dataset Description

Synthetic Dataset. The synthetic test is performed with the aim of testing the robustness against deformation and outlier in a quantitative manner. Specifically, a reference graph with n_{in} nodes is created by assigning random attribute to

[5] We have also tested three widely used pairwise matching solvers including Graduated Assignment (GAGM) [21], Reweighted Random Walks Matching (RRWM) [9] and Integer Projected Fixed Point (IPFP) [31] respectively. It is found that the overall performances are insensitive to the selection of pairwise solver.



(a) House Accuracy (b) Hotel Accuracy (c) Car Accuracy (d) Duck Accuracy

(e) House Score (f) Hotel Score (g) Car Score (h) Duck Score

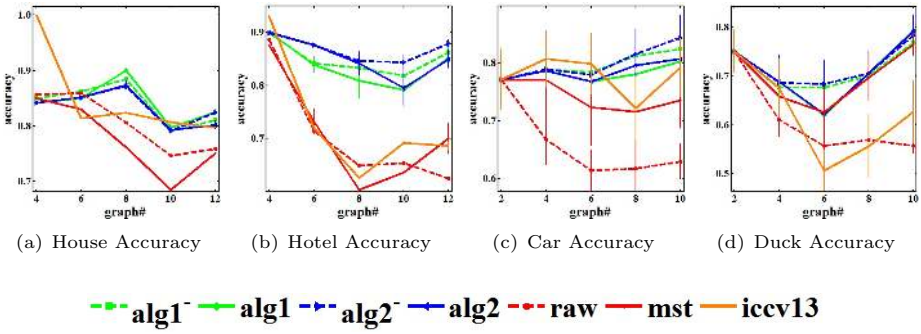
(i) House Consistency (j) Hotel Consistency (k) Car Consistency (l) Duck Consistency

Fig. 3. Evaluation on large sized graph set ($N > n$) by 10 tests of random sampling on two object categories from CMU dataset, and two categories from WILLOW-ObjectClass. $N = 16, 18, 20, 22, 24$; $n_{in} = 10$ and $n_{out} = 3$.

each of its edge, which is uniformly sampled from the interval [0,1]. Based on the created reference graph, the "perturbed" graph set is created by adding a Gaussian noise ε, which is sampled from $N(0, \sigma^2)$, to the edge attribute d_a^{ij} by $d_b^{ij} = d_a^{ij} + \varepsilon$. Each "perturbed" graph is further added n_{out} outliers. This protocol is the same with [9,43,47] etc. Furthermore, we set n_{in} and n_{out} equal across all graphs to make $\mathbf{P}$ a square permutation matrix. This setting is adopted in comparing methods [47,36,40] and so forth.

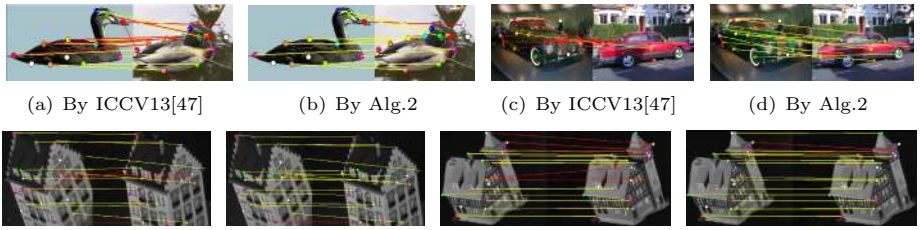
CMU House/Hotel Dataset. The CMU house/hotel image sequence[6] was commonly used to test the performance of graph matching algorithms [9,6,52,43,47,8] etc. The hotel sequence contains 101 frames and the house sequence consists of 111 frames. Thirty landmarks are annotated in each image for both sequences.

[6] `http://vasc.ri.cmu.edu/idb/html/motion/`



(a) House Accuracy (b) Hotel Accuracy (c) Car Accuracy (d) Duck Accuracy

Fig. 4. Evaluation on small sized graph set ($N > n$) by 10 tests of random sampling on two object categories from CMU dataset, and two categories from WILLOW-ObjectClass. $N = 4, 6, 8, 12$; $n_{in} = 10$ and $n_{out} = 3$.



(a) By ICCV13[47] (b) By Alg.2 (c) By ICCV13[47] (d) By Alg.2

(e) By ICCV13[47] (f) By Alg.2 (g) By ICCV13[47] (h) By Alg.2

Fig. 5. Matching illustration by Alg.2 and ICCV13 [47] on Duck and Car of WILLOW-ObjectClass, and the hotel and house of CMU dataset. More yellow and less red denote more accurate matching. White circles denote outliers.

For more details of the dataset, readers are referred to [9,52]. To model the challenges in real world, we use $n_{in} = 10$ markers as the inliers, and randomly chose $n_{out} = 3$ from the rest of the markers as outliers.

WILLOW-ObjectClass dataset. The object class dataset[7] is recently created and used by Cho et al. [8]. Here we perform our tests on two categories of this dataset: Duck (50 images) and Car (40 images) which is constructed with images from Caltech-256 and PASCAL VOC2007 respectively. For each image, $n_{in} = 10$ landmarks were manually labeled on the target object. Moreover, we also add $n_{out} = 3$ outliers of detected points from the background by the SIFT detector.

5.3 Results and Discussion

Tests on Synthetic Dataset. The results on synthetic dataset is shown in Fig.2, which suggest the proposed two methods outperform other methods as the number of graphs grows, under fixed deformation and outlier configuration. The regularized maximization method (Alg.2) further improves the baseline (Alg.1)

[7] `http://www.di.ens.fr/willow/research/graphlearning/`

wherein no regularization is added. Fig.2 also plots the results of using 2nd-order path score maximization policy (termed as Alg.1^{-2nd} in black), from which one can observe the first order method (in green) is cost-effective and almost equally accurate. Note green and black dash lines are almost overlapped to each other in Fig.2(a) while the overhead differs significantly as shown in Fig.2(d).

Tests on Large/Samll Number of Real Image Set. Intuitively (also empirically exemplified in Fig.1(c)), more graphs will increase the robustness of consistency as a correlated indicator towards accuracy. From Fig.3 for large number of image sets test, one can observe the post-synchronization step improves accuracy for both Alg.1 and Alg.2. However, as shown in Fig.4, this improvement decays for small number of graph tests and the role of graduated regularization becomes more important. On the other hand, as shown in Fig.3, the effectiveness of our methods is highlighted on the Willow-Object dataset (Duck and Car) compared with the CMU dataset (House and Hotel). Fig.5 shows some comparing examples from these datasets, from which one can observe the Willow-Object dataset is more challenging for matching with more varying object size and viewing angle. This suggests our methods are more robust for matching less related objects with more deformation perturbation. In our analysis, the robustness comes from our methods are able to flexibly and effectively capture the statistical correlation behavior between score/consistency and accuracy at different score ranges (refer to Fig.1). In the opposite, the comparing method [47] strictly imposes full consistency in the beginning of iteration and is sensitive to the basis reference graph selected for iterative optimization (there are fluctuations in the plot), while the other method [36] imposes synchronization in an one-shot fashion.

6 Conclusion and Future Work

We proposed novel algorithms towards robust multi-graph matching by incorporating both matching scores and matching consistency in an iterative approximating optimization procedure. Their efficacy is demonstrated through convincing experiments conducted on both synthetic and public real image datasets. The underlying rationale is that the two aspects are statistically coupled and can thus be tackled jointly. Our future work include i) study the concept of "partial consistency" that involves part of the nodes; ii) connect unsupervised or semi-supervised machine learning methodologies with multi-graph matching, as the consistency itself can guide the matching and learning.

Acknowledgment. The work was supported in part by NSF IIS-1116886, NSF DMS-1317424, NSFC 61129001,61025005/F010403. The authors would also like to thank reviewers' valuable comments during the whole review process.

References

1. http://deim.urv.cat/~francesc.serratosa/
2. http://www-users.cs.york.ac.uk/~erh/
3. http://www.iam.unibe.ch/fki/staff/prof.-dr.-horst-bunke
4. Besl, P., McKay, N.: A method for registration of 3-d shapes. PAMI (1992)

5. Brendel, W., Todorovic, S.: Learning spatiotemporal graphs of human activities. In: ICCV (2011)
6. Caetano, T., McAuley, J., Cheng, L., Le, Q., Smola, A.J.: Learning graph matching. IEEE Transaction on PAMI 31(6), 1048–1058 (2009)
7. Chertok, M., Keller, Y.: Efficient high order matching. PAMI (2010)
8. Cho, M., Alahari, K., Ponce, J.: Learning graphs to match. In: ICCV (2013)
9. Cho, M., Lee, J., Lee, K.M.: Reweighted random walks for graph matching. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part V. LNCS, vol. 6315, pp. 492–505. Springer, Heidelberg (2010)
10. Cho, M., Lee, K.M.: Progressive graph matching: Making a move of graphs via probabilistic voting. In: CVPR (2012)
11. Cho, M., Sun, J., Duchenne, O., Ponce, J.: Finding matches in a haystack: A max-pooling strategy for graph matching in the presence of outliers. In: CVPR (2014)
12. Conte, D., Foggia, P., Sansone, C., Vento, M.: Thirty years of graph matching in pattern recognition. IJPRAI (2004)
13. Duchenne, O., Bach, F., Kweon, I., Ponce, J.: A tensor-based algorithm for high-order graph matching. In: CVPR (2009)
14. Eshera, M.A., Fu, K.S.: An image understanding system using attributed symbolic representation and inexact graph-matching. PAMI (1986)
15. Fischler, M., Bolles, R.: Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. Comm. of the ACM, 381–395 (1981)
16. Foggia, P., Percannella, G., Vento, M.: Graph matching and learning in pattern recognition in the last 10 years. IJPRAI (2014)
17. Gallagher, B.: Matching structure and semantics: A survey on graph-based pattern matching. In: AAAI, pp. 45–53 (2006)
18. Garey, M.R., Johnson, D.S.: Computers and Intractability; A Guide to the Theory of NP-Completeness. W. H. Freeman and Co., New York (1990)
19. Gavril, F.: Generating the maximum spanning trees of a weighted graph. Journal of Algorithms, 592–597 (1987)
20. Goesele, M., Snavely, N., Curless, B., Hoppe, H., Seitz, S.: Multi-view stereo for community photo collections. In: ICCV (2007)
21. Gold, S., Rangarajan, A.: A graduated assignment algorithm for graph matching. IEEE Transaction on PAMI (1996)
22. Hancock, E.R., Wilson, R.C.: Pattern analysis with graphs: Parallel work at bern and york. Pattern Recognition Letters, 833–841 (2012)
23. Hu, N., Rustamov, R.M., Guibas, L.: Graph mmatching with anchor nodes: a learning approach. In: CVPR (2013)
24. Huang, Q., Zhang, G., Gao, L., Hu, S., Butscher, A., Guibas, L.: An optimization approach for extracting and encoding consistent maps in a shape collection. ACM Transactions on Graphics, TOG (2012)
25. Huang, Q.X., Flory, S., Gelfand, N., Hofer, M., Pottmann, H.: Reassembling fractured objects by geometric matching. ACM Trans. Graph., 569–578 (2006)
26. Kim, V.G., Li, W., Mitra, N.J., DiVerdi, S., Funkhouser, T.: Exploring collections of 3D models using fuzzy correspondences. In: SIGGRAPH (2012)
27. Kuhn, H.W.: The hungarian method for the assignment problem. Export. Naval Research Logistics Quarterly, 83–97 (1955)
28. Lee, J., Cho, M., Lee, K.M.: Hyper-graph matching via reweighted random walks. In: CVPR (2011)
29. Leordeanu, M., Hebert, M.: A spectral technique for correspondence problems using pairwise constraints. In: ICCV (2005)

30. Leordeanu, M., Hebert, M., Sukthankar, R.: Beyond local appearance: Category recognition from pairwise interactions of simple features. In: CVPR (2007)
31. Leordeanu, M., Herbert, M.: An integer projected fixed point method for graph matching and map inference. In: NIPS (2009)
32. Leordeanu, M., Sukthankar, R., Hebert, M.: Unsupervised learning for graph matching. Int. J. Comput. Vis., 28–45 (2012)
33. Leordeanu, M., Zanfir, A., Sminchisescu, C.: Semi-supervised learning and optimization for hypergraph matching. In: ICCV (2011)
34. Livi, L., Rizzi, A.: The graph matching problem. Pattern Anal. Applic., 253–283 (2013)
35. Loiola, E.M., de Abreu, N.M., Boaventura-Netto, P.O., Hahn, P., Querido, T.: A survey for the quadratic assignment problem. EJOR, 657–690 (2007)
36. Pachauri, D., Kondor, R., Vikas, S.: Solving the multi-way matching problem by permutation synchronization. In: NIPS (2013)
37. Pevzner, P.A.: Multiple alignment, communication cost, and graph matching. SIAM JAM (1992)
38. Qiu, H., Hancock, E.R.: Spectral simplification of graphs. In: Pajdla, T., Matas, J. (eds.) ECCV 2004. LNCS, vol. 3024, pp. 114–126. Springer, Heidelberg (2004)
39. Shen, D., Hammer, C.D.: Hierarchical attribute matching mechanism for elastic registration. TMI (2002)
40. Sole-Ribalta, A., Serratosa, F.: Graduated assignment algorithm for multiple graph matching based on a common labeling. IJPRAI (2013)
41. Suh, Y., Cho, M., Lee, K.M.: Graph matching via sequential monte carlo. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012, Part III. LNCS, vol. 7574, pp. 624–637. Springer, Heidelberg (2012)
42. Cour, T., Srinivasan, P., Shi, J.: Balanced graph matching. In: NIPS (2006)
43. Tian, Y., Yan, J., Zhang, H., Zhang, Y., Yang, X., Zha, H.: On the convergence of graph matching: Graduated assignment revisited. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012, Part III. LNCS, vol. 7574, pp. 821–835. Springer, Heidelberg (2012)
44. Williams, M.L., Wilson, R.C., Hancock, E.: Multiple graph matching with bayesian inference. Pattern Recognition Letters, 1275–1281 (1997)
45. Wong, A., You, M.: Entropy and distance of random graphs with application to structural pattern recognition. IEEE Transactions on PAMI (1985)
46. Yan, J., Li, Y., Zheng, E., Liu, Y.: An accelerated human motion tracking system based on voxel reconstruction under complex environments. In: Zha, H., Taniguchi, R.-I., Maybank, S. (eds.) ACCV 2009, Part II. LNCS, vol. 5995, pp. 313–324. Springer, Heidelberg (2010)
47. Yan, J., Tian, Y., Zha, H., Yang, X., Zhang, Y.: Joint optimization for consistent multiple graph matching. In: ICCV (2013)
48. Zach, C., Klopschitz, M., Pollefeys, M.: Disambiguating visual relations using loop constraints, pp. 1246–1433 (2010)
49. Zaslavskiy, M., Bach, F.R., Vert, J.P.: A path following algorithm for the graph matching problem. PAMI (2009)
50. Zass, R., Shashua, A.: Probabilistic graph and hypergraph matching. In: CVPR (2008)
51. Zeng, Z., Chan, T.-H., Jia, K., Xu, D.: Finding correspondence from multiple images via sparse and low-rank decomposition. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012, Part V. LNCS, vol. 7576, pp. 325–339. Springer, Heidelberg (2012)
52. Zhou, F., Torre, F.D.: Factorized graph matching. In: CVPR (2012)