Multi-Graph Matching via Affinity Optimization with Graduated Consistency Regularization

Junchi Yan, Minsu Cho, Hongyuan Zha, Xiaokang Yang, *Senior Member, IEEE*, and Stephen M. Chu

Abstract—This paper addresses the problem of matching common node correspondences among multiple graphs referring to an identical or related structure. This multi-graph matching problem involves two correlated components: i) the local pairwise matching affinity across pairs of graphs; ii) the global matching consistency that measures the uniqueness of the pairwise matchings by different composition orders. Previous studies typically either enforce the matching consistency constraints in the beginning of an iterative optimization, which may propagate matching error both over iterations and across graph pairs; or separate affinity optimization and consistency enforcement into two steps. This paper is motivated by the observation that matching consistency can serve as a regularizer in the affinity objective function especially when the function is biased due to noises or inappropriate modeling. We propose composition-based multi-graph matching methods to incorporate the two aspects by optimizing the affinity score, meanwhile gradually infusing the consistency. We also propose two mechanisms to elicit the common inliers against outliers. Compelling results on synthetic and real images show the competency of our algorithms.

Index Terms—Graph matching, feature correspondence

1 Introduction

Graph matching (GM) [5], [6] has received attentions over decades, and has found wide applications in various problems such as bioinformatics [7], data fusion [8], scene analysis [9], [10], graphics [11] and information retrieval [12]. GM lies at the heart of a range of computer vision tasks—object recognition, shape matching, object tracking, and image labeling among others, which require finding visual correspondences—please refer to [4] for more references. Different from point based matching or registration methods such as RANSAC [13] and Iterative Closet Point (ICP) [14], GM incorporates both unary *node-to-node*, and pairwise *edge-to-edge* structural similarity. By encoding the geometrical information in representation and matching processes, GM methods are in general supposed to be more robust to deformation noise, missing data, and outliers. Due to its well-known NP-complete nature, existing GM methods involve either finding approximate solutions [15], [16], [17] or obtaining the global optima in polynomial time for few types, such as planar graph [18], bounded valence graph [19], and tree structure [20].

Most GM methods focus on establishing one-to-one correspondences between a pair of feature points [16], [21], [22], [23], [24]. The pairwise GM problem can be formulated as a quadratic assignment problem (QAP) that accounts for both individual node matches (unary terms) and pairs of matches (pairwise terms). The general QAP formulation refers to the Lawler's QAP [25] as used in this paper. Another formulation i.e., Koopmans-Beckmann's QAP [26] is a special case of the former one.

In many applications, similar objects do not appear in isolation or in pair, but more frequently in collections. This context potentially allows for higher quality matching and analysis. Given a batch of graphs referring to an identical or related structure, it is required to find the matchings across all graphs. In fact, multi-graph matching is applied to infusing multi-source sensor data [8]. Graphic analysis often requires to model objects by multi-view assembly [27]. The work of [12] applies multi-graph matching to multi-source topic alignment. It is also related to graph clustering, classification and indexing. Emerging multi-graph matching methods [1], [2], [28], [29], [30], [31], [32] are proposed, which will be discussed in the next section.

2 Related Work and Models

2.1 Pairwise Graph Matching

The problem of pairwise GM involving two graphs has been extensively studied in the literature. It is beyond the scope of this paper to introduce an exhaustive body of these works. As it will be detailed later in this paper, our methods and many other multi-graph matching methods [2], [4], [30], [31] build on pairwise GM solvers as a black-box, regardless of the specific context in which these pairwise GM techniques are devised, such as machine learning based approaches [21], [23], [33], [34], [35], hyper-graph matching

- *J. Yan is with the Shanghai Key Laboratory of Media Processing and Transmission, Department of Electronic Engineering, Shanghai Jiao Tong University, Shanghai 200240, China and the IBM Research. E-mail: yanjunchi@sjtu.edu.cn.*
- *M. Cho is with the WILLOW team of INRIA/Ecole Normale Superieure, Paris, France. E-mail: minsu.cho@inria.fr.*
- *H. Zha is with the Software Engineering Institute, East China Normal University, Shanghai 200062, China, and the School of Computational Science and Engineering, College of Computing, Georgia Institute of Technology, Atlanta, Georgia 30332. E-mail: zha@cc.gatech.edu.*
- *X. Yang is with the Shanghai Key Laboratory of Media Processing and Transmission, Department of Electronic Engineering, Shanghai Jiao Tong University, Shanghai 200240, China. E-mail: xkyang@sjtu.edu.cn.*
- *S.M. Chu is with the IBM T.J. Watson Research Center. E-mail: schu@us.ibm.com.*

Manuscript received 5 Jan. 2015; revised 7 May 2015; accepted 1 Sept. 2015. Date of publication 0 . 0000; date of current version 0 . 0000. Recommended for acceptance by T. Brox. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below. Digital Object Identifier no. 10.1109/TPAMI.2015.2477832

approaches [33], [36], [37], [38], [39], [40] and other application-oriented GM systems [3], [41].

Most pairwise GM methods [16], [17], [24], [42], [43], [44] are directly based on the Lawler's QAP. Given two graphs $\mathcal{G}_1$ and $\mathcal{G}_2$ of size n_1 and n_2, an affinity matrix $\mathbf{K} \in \mathbb{R}^{n_1 n_2 \times n_1 n_2}$ is defined such that its elements $\{K_{ia;jb}\}_{i,j=1a,b=1}^{n_1 \quad n_2}$ measure an edge pair affinity $\{(v_i, v_j) \leftrightarrow (v_a, v_b)\}_{i,j=1a,b=1}^{n_1 \quad n_2}$ from two graphs. The diagonal term $\{K_{ia;ia}\}_{i=1,a=1}^{n_1,n_2}$ describes a unary affinity of a node match $\{v_i \leftrightarrow v_a\}_{i=1,a=1}^{n_1,n_2}$. Most GM works [1], [4], [15], [16] conventionally define $\mathbf{K}$ such that element $K_{ia;jb}$ for edge pair $(v_i, v_j) \leftrightarrow (v_a, v_b)$ is located at the $((a-1)n_1 + i)$th row and $((b-1)n_2 + j)$th column of $\mathbf{K}$.

We use assignment matrix $\mathbf{X}$ to establish the one-to-one correspondence such that $X_{ia} = 1$ if node v_i matches v_a, and 0 otherwise. The problem of GM involves finding $\mathbf{X}$ such that the sum of the node and edge compatibility is maximized. Without loss of generality, similar to [15], [16], by assuming $n_1 \geq n_2$, it leads to the widely used constrained quadratic assignment problem:

$$\begin{aligned} \mathbf{X}^* &= \arg\max_{\mathbf{X}} \mathrm{vec}(\mathbf{X})^T \mathbf{K} \mathrm{vec}(\mathbf{X}) \\ s.t. \quad & \mathbf{X}\mathbf{1}_{n_2} \leq \mathbf{1}_{n_1} \quad \mathbf{1}_{n_1}^T \mathbf{X} = \mathbf{1}_{n_2}^T \quad \mathbf{X} \in \{0,1\}^{n_1 \times n_2}. \end{aligned} \tag{1}$$

The constraints refer to two-way one-to-one mapping: a node from $\mathcal{G}_1$ can match at most one node in $\mathcal{G}_2$ and every node in $\mathcal{G}_2$ correspondes to one node in $\mathcal{G}_1$. There is neither one-to-many nor many-to-many matching.

The above formulation is as general as it allows two graphs to have different number of nodes ($n_1 \neq n_2$). A common protocol adopted by the previous studies [4], [15], [42] is converting $\mathbf{X}$ from an assignment matrix ($\mathbf{X}\mathbf{1}_{n_2} \leq \mathbf{1}_{n_1}$) to a permutation matrix ($\mathbf{X}\mathbf{1}_{n_2} = \mathbf{1}_{n_1}$) by adding dummy nodes to one graph (i.e., adding slack variables to the assignment matrix and augment the affinity matrix by zeros) in case $n_1 \neq n_2$. This is a common technique from linear programming and is widely adopted by, e.g. [15], [42], such that is supposed can handle superfluous nodes in a statistically robust manner (see the last paragraph of Section 2.3 in [42]). By taking this step, the graphs are of equal sizes. This preprocessing also opens up the applicability of existing multi-graph methods [30], [32], [45] as they all assume that all graphs are of equal sizes. Therefore, the following formulation is derived which assumes $n_1 = n_2 = n$:

$$\begin{aligned} \mathbf{X}^* &= \arg\max_{\mathbf{X}} \mathrm{vec}(\mathbf{X})^T \mathbf{K} \mathrm{vec}(\mathbf{X}) \\ s.t. \quad & \mathbf{X}\mathbf{1}_{n_2} = \mathbf{1}_{n_1} \quad \mathbf{1}_{n_1}^T \mathbf{X} = \mathbf{1}_{n_2}^T \quad \mathbf{X} \in \{0,1\}^{n_1 \times n_2}. \end{aligned} \tag{2}$$

Here $\mathbf{X}$ is a square permutation matrix by augmenting the raw matching matrix with slack columns if $n_1 > n_2$. This technique can also be applied in our composition-based method as will be shown later in the paper.

2.2 Multi-Graph Matching

Recently, matching a collection of related graphs becomes an emerging topic from various research communities including computer vision [4], [31], machine learning [2], [30], pattern recognition [28], [32], [45], graphics [11] and information retrieval [12], among others. We divide the state-of-the-arts into two categories concerning how the affinity and matching consistency are explored.

i) *affinity*-driven approaches [4], [31], [32], [46]: usually a set of basis pairwise matchings is first generated which derive the matchings for each pair. Then, an objective regarding the overall pairwise matching affinity score is maximized by different algorithms. Solé-Ribalta and Serratosa [32] extend Graduated Assignment [42] for two-graph to multiple graphs. Each graph is associated with an assignment matrix mapping the nodes to a virtual node set. Then, the variable set is updated in a deterministic annealing manner to maximize the overall pairwise affinity score. Yan et al. [4], [31] adaptively find a reference graph $\mathcal{G}_r$, which induces a compact basis matching variable set $\{\mathbf{X}_{kr}\}_{k=1,\neq r}^{N}$ over N graphs. An alternating optimization framework is devised to update these basis variables in a rotating manner. The early work [46] by Gavril selects a set of basis variables from initial pairwise matching solutions via finding a maximum spanning tree (MST) on a super graph *w.r.t* the overall affinity.

ii) *consistency*-driven approaches [2], [28], [30], [45]: these methods usually are comprised of two steps. First, a pairwise GM solver is employed to obtain the matchings over all (or a portion of) graph pairs. Second, a spectrum smoothing technique is devised to enforce global matching consistency in the sense that two sequential pairwise matching in different composition orders shall lead to two identical solutions. Specifically, Solé-Ribalta and Serratosa [28], [45] use a hyper-cube tensor to represent the N-node matching likelihood, each from N different graphs. Then a greedy method is used to binarize the final solutions satisfying the one-to-one two way constraints. Pachauri et al. [30] employ spectral analysis via eigenvector decomposition on the initial pairwise matching solutions, and obtain a set of new consistent matchings. This method relies on the assumption that the initial pairwise matchings are corrupted by Gaussian-Wigner noise which is too ideal in reality. Chen et al. [2] address the 'partial similarity' problem when only a part of nodes can find their correspondences in other graphs. They formulate the problem into a tractable convex programming problem, and solve it by a tailored first-order Alternating Direction Method of Multipliers.

Common to the above approaches is that they either enforce consistency early by using a compact set of basis variables for pairwise matchings [4], [28], [31], [45], which inherently runs at the risk of propagating errors across iterations and graphs; or assume the initial pairwise matchings are obtained with corruptions by a certain pairwise GM method, and perform spectral smoothing on the initial matchings, in fact ignoring affinity information in the procedure [2], [28], [30], [45].

Specifically, a widely used multi-graph matching formulation [4], [31], [32] is as follows. Under the assumption of invertible matching relations $\mathbf{X}_{ij} = \mathbf{X}_{ji}^T$, it enforces matching consistency over all pairwise matchings:

$$\mathbf{X}_{ij}^{*} = \arg\max_{\mathbf{X}_{ij}} \sum_{i=1}^{N-1}\sum_{j=i+1}^{N} \mathrm{vec}(\mathbf{X}_{ij})^T \mathbf{K}_{ij}\mathrm{vec}(\mathbf{X}_{ij}) \quad (3)$$
$$s.t. \quad \mathbf{I}_{n_i}^T\mathbf{X}_{ij} = \mathbf{1}_{n_j}^T, \mathbf{X}_{ij}\mathbf{I}_{n_j} = \mathbf{1}_{n_i}, \mathbf{X}_{ij} = \mathbf{X}_{ik}\mathbf{X}_{kj}{}_{k=1,\neq i,j}^{N}.$$

The above formulation assumes each graph only contain the matchable common inlier nodes. In case there is notable ratio of unmatchable outliers, this paper further devises an effective common inlier eliciting mechanism.

2.3 Philosophy of This Paper

Different from the peer methods, we adopt a composition based affinity optimizing procedure, which is gradually regularized by matching consistency. The rationale is that the affinity is indicative to ground truth in early iterations. Nevertheless, it becomes less informative as the optimized affinity saturates. In fact, the affinity function is often biased to true accuracy due to: i) arbitrary noises over graphs; ii) inherent difficulty in modeling the affinity function via a compact parametric model. In contrast, consistency becomes a useful regularizer to improve the accuracy. In this spirit, in the presence of many outliers, we further propose node-wise consistency and affinity driven mechanisms to elicit the common inliers. Our methods are simple and general such that it can work with any types of graphs, and involve only one major parameter λ. Extensive empirical results illustrate the efficacy of our methods as concluded in Section 5.

3 Proposed Algorithms

Given the initial matching configuration $\mathbb{X}^{(0)}$ by a pairwise matcher, e.g. [15], [16], one can further derive a new pairwise matching $\mathbf{X}_{ij}^{(1)}$ by the composition of a few 'good' matchings i.e., $\mathbf{X}_{ij}^{(1)} = \mathbf{X}_{ik_1}^{(0)}\mathbf{X}_{k_1k_2}^{(0)}\dots\mathbf{X}_{k_{s-1}k_s}^{(0)}\mathbf{X}_{k_sj}^{(0)}$ to replace the original $\mathbf{X}_{ij}^{(0)}$. And the matching accuracy is improved or ideally maximized, by means of interpolating the sequence $\mathcal{G}_i, \mathcal{G}_{k_1}, \dots, \mathcal{G}_{k_{s-1}}, \mathcal{G}_{k_s}, \mathcal{G}_j$. The key problem of this composition framework is how to set up the appropriate compositional replacements.

This paper devises a mechanism for iteratively finding the appropriate new composition to replace the old one without knowing their true accuracy. We will start with a baseline Algorithm 1, and two variants for comparison. Then a graduated consistency-regularized method Algorithm 2 is highlighted with two efficient variants as depicted in Algorithm 3. We first introduce several notations and definitions.

3.1 Notations and Definitions

Throughout the paper, $\mathbb{R}$ denotes the real number domain. Bold capital letters denote a matrix $\mathbf{X}$, bold lower-case letters a column vector $\mathbf{x}$, and hollow bold letters a set $\mathbb{X}$. All non-bold letters represent scalars. $\mathbf{X}^T$, vec($\mathbf{X}$) is the transpose and column-wise vectorized version of $\mathbf{X}$; tr($\mathbf{X}$) is the trace of $\mathbf{X}$. $\mathbf{I}_n \in \mathbb{R}^{n\times n}$ is an identity matrix and subscript n is omitted when it can be inferred from context. $\mathbf{1}_{m\times n}, \mathbf{0}_{m\times n} \in \mathbb{R}^{m\times n}$ is the matrix with elements being all ones or zeros, respectively. $\mathbf{1}_n$ is the abbreviation for $\mathbf{1}_{n\times 1}$. $|\mathbb{X}|$ is the cardinality of the set $\mathbb{X}$, and $\|\mathbf{X}\|_F = \mathrm{tr}(\mathbf{X}^T\mathbf{X})$ denotes the Frobenious norm.

This paper uses $\mathbb{X} = \{\mathbf{X}_{ij}\}_{i=1,j=i+1}^{N-1,N}$ to denote the set of pairwise matching matrices over N graphs $\{\mathcal{G}_k\}_{k=1}^{N}$, which is also termed *matching configuration*. $\mathbb{K} = \{\mathbf{K}_{ij}\}_{i,j=1}^{N}$ denotes the affinity matrix set. We also use $\mathbb{X}_c$ to denote a solution set which is fully consistent (the subscript 'c') as there is no contradictory matchings for $\{\mathbf{X}_{ij} \neq \mathbf{X}_{ik}\mathbf{X}_{kj}\}_{i,j,k=1}^{N}$. If not otherwise stated explicitly, we use N for the number of considered graphs, n and m for the number of nodes and edges, respectively.

So far we have not presented a definition for consistency. Similar concepts have been used in [2], [4], [28], [30], [31] and mentioned earlier in this paper. Now we give the formal definitions for consistency induced by the pairwise matchings $\mathbb{X}$. In addition, we also define two variants of a super graph and a concept related to matching composition. We do not claim the credit for the novelty of these definitions though the key metric for unary consistency is first proposed in the conference version [1] of this paper. Rather, we hope to provide a systematic view on the idea of consistency.

Definition 1. *Given N graphs $\{\mathcal{G}_k\}_{k=1}^{N}$ and the pairwise matching configuration $\mathbb{X} = \{\mathbf{X}_{ij}\}_{i=1,j=i+1}^{N-1,N}$, the unary consistency of graph $\mathcal{G}_k$ is defined as*

$$C_u(k,\mathbb{X}) = 1 - \frac{\sum_{i=1}^{N-1}\sum_{j=i+1}^{N}\|\mathbf{X}_{ij}-\mathbf{X}_{ik}\mathbf{X}_{kj}\|_F/2}{nN(N-1)/2} \in (0,1].$$

Definition 2. *Given graphs $\{\mathcal{G}_k\}_{k=1}^{N}$ and matching configuration $\mathbb{X}$, for any pair $\mathcal{G}_i$ and $\mathcal{G}_j$, the pairwise consistency is defined as* $C_p(\mathbf{X}_{ij},\mathbb{X}) = 1 - \frac{\sum_{k=1}^{N}\|\mathbf{X}_{ij}-\mathbf{X}_{ik}\mathbf{X}_{kj}\|_F/2}{nN} \in (0,1]$.

Definition 3. *Given N graphs $\{\mathcal{G}_k\}_{k=1}^{N}$ and $\mathbb{X}$, we call $\mathbb{X}$ is fully consistent if and only if $\frac{\sum_{i=1,j=i+1}^{N-1,N} C_p(\mathbf{X}_{ij},\mathbb{X})}{N(N-1)/2} = 1$ and (or) $\frac{\sum_{k=1}^{N} C_u(k,\mathbb{X})}{N} = 1$. The following equation always holds: $\frac{\sum_{k=1}^{N} C_u(k,\mathbb{X})}{N} = \frac{\sum_{i=1,j=i+1}^{N-1,N} C_p(\mathbf{X}_{ij},\mathbb{X})}{N(N-1)/2}$. We further define the overall consistency of $\mathbb{X}$ as $C(\mathbb{X}) = \frac{\sum_{k=1}^{N} C_u(k,\mathbb{X})}{N} \in (0,1]$.*

Definition 4. *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 $C_{\mathrm{n}}(u^k,\mathbb{X}) = 1 - \frac{\sum_{i=1}^{N-1}\sum_{j=i+1}^{N}\|\mathbf{Y}(u^k,:)\|_F/2}{N(N-1)/2} \in (0,1]$ where $\mathbf{Y} = \mathbf{X}_{kj} - \mathbf{X}_{ki}\mathbf{X}_{ij}$ and $\mathbf{Y}(u^k,:)$ is the u^kth row of matrix $\mathbf{Y}$.*

Definition 5. *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 $S_n(u^k,\mathbb{X},\mathbb{K}) = \sum_{i=1,\neq k}^{N}\mathrm{vec}(\mathbf{X}_{ki}^{u^k})^T\mathbf{K}_{ki}\mathrm{vec}(\mathbf{X}_{ki})$, where $\mathbf{X}^{u^k}$ denotes the matrix $\mathbf{X}$ with zeros except for the u^kth rows as is.*

Definition 6. *Given $\{\mathcal{G}_k\}_{k=1}^{N}$ and $\mathbb{X}$, the affinity-wise super graph $\mathcal{G}_{sup}^{a}$ is defined as an undirected weighted graph s.t. each node k represents $\mathcal{G}_k$, and edge weight e_{ij} is the affinity score $J_{ij}(\mathbf{X}_{ij}) = \mathrm{vec}(\mathbf{X}_{ij})^T\mathbf{K}_{ij}\mathrm{vec}(\mathbf{X}_{ij})$ induced by $\mathbb{X}$.*

Definition 7. *Given $\{\mathcal{G}_k\}_{k=1}^{N}$ and $\mathbb{X}$, the consistency-wise super graph $\mathcal{G}_{sup}^{c}$ is defined as an undirected weighted graph s.t. node k represents $\mathcal{G}_k$, and the weight of edge e_{ij} is the pairwise consistency $C_p(\mathbf{X}_{ij},\mathbb{X}) = 1 - \frac{\sum_{k=1}^{N}\|\mathbf{X}_{ij}-\mathbf{X}_{ik}\mathbf{X}_{kj}\|_F/2}{nN}$.*

Definition 8. *The path $Z_{ij}(k_1, k_2, \ldots, k_s)$ from $\mathcal{G}_i$ to $\mathcal{G}_j$ is defined as the chain $\mathcal{G}_i \rightarrow \mathcal{G}_{k_1} \rightarrow \cdots \rightarrow \mathcal{G}_{k_s} \rightarrow \mathcal{G}_j$ which induces: $\mathbf{Y}_{ij}(k_1,\ k_2, \ldots, k_s) \triangleq \mathbf{X}_{ik_1}\mathbf{X}_{k_1k_2}\ldots\mathbf{X}_{k_sj}$. Its order s_{ij} is the number of the intermediate graphs between $\mathcal{G}_i$, $\mathcal{G}_j$. The path score is $J_{ij}(k_1, k_2, \ldots, k_s) = vec(\mathbf{Y}_{ij})^T\mathbf{K}_{ij}vec(\mathbf{Y}_{ij})$.*

We present several comments to the above definitions. The unary consistency $C_u(k, \mathbb{X})$ in Definition (1), which makes its debut in the conference version [1] of this paper, is a function *w.r.t.* the graph index k and $\mathbb{X}$. In contrast, the pairwise consistency $C_p(\mathbf{X}_{ij}, \mathbb{X})$ in Definition (2) is generalized to the one *w.r.t.* two variables of $\mathbf{X}_{ij}$ and $\mathbb{X}$. Note $\mathbf{X}_{ij}$ does not necessarily belong to $\mathbb{X}$. The pairwise consistency metric is symmetric such that $C_p(\mathbf{X}_{ij}, \mathbb{X}) = C_p(\mathbf{X}_{ji}, \mathbb{X})$, which derives Definition (3). Definition (4) and (5) break down to node level consistency/affinity similar to unary ones, and they will be used in the proposed inlier eliciting mechanism.

The super graph, either for affinity-wise $\mathcal{G}^a_{sup}$ in Definition (6), or consistency-wise $\mathcal{G}^c_{sup}$ in Definition (7), is a connected (not necessarily fully connected) graph as a portion of pairs are matched by a certain means, and a *maximum spanning tree* [46] can be found with no less total weight of every other spanning tree.

Definition (1) and Definition (2) are used by [4] to set the alternating variable updating order. In contrast, this paper considers the role of consistency as a regularizer for the affinity optimization. The overall consistency for $\mathbb{X}$ as in Definition (3), further tells the relation between unary consistency and pairwise consistency. A similar overall 'inconsistency' metric appears in [32], [47].

The affinity-wise super graph $\mathcal{G}^a_{sup}$ in Definition (6) is similar to the one in [48], of which the author proposes to build a graph where each shape is a vertex, and edges between shapes are weighted by the cost of the best matching. We put it in the scenario for the problem of weighted multi-graph matching. Furthermore, we give a similar definition for a consistency variant $\mathcal{G}^a_{sup}$ in terms of consistency-wise edge weights in Definition (7).

Finally, Definition (8) is related to the idea of approximate path selection as used in our algorithms. Obviously, when $\mathbb{X}$ is fully consistent, any two paths between two given graphs would induce the equal solution.

Based on the above definitions and discussion, we present our main approaches in the rest of this section.

3.2 Composition Based Affinity Optimization

Since most GM methods aim to maximize an objective function regarding with affinity score [4], [15], [16], [17], [31], we also follow this methodology in this section. Our basic rationale is that the node matching regarding with the highest affinity score between two graphs can be found along a higher-order path, as related to Definition (8), instead of the direct (zero-order) pairwise matching. We formalize this idea as follows:

Without loss of generality, assume the affinity-wise super graph $\mathcal{G}^a_{sup}$ is fully connected and induced by the configuration $\mathbb{X}$. Given $\mathcal{G}_i$, $\mathcal{G}_j$, all loop-free matching paths on the super graph can form a set of loop-free paths $\mathbb{Z}_{ij} = \{Z_{ij}(k_1, \ldots, k_s)\}_{s=1}^{N-2}$, whose cardinality $|\mathbb{Z}_{ij}| = \sum_{s=1}^{N-2} s!$. Hence the overhead for finding the highest affinity score solution is intractable since $\sum_{s=1}^{N-2} s!$ times of compositions need be computed. Given the matching configuration $\mathbb{X}^{(t-1)}$, one alternative is approximating the optimal Z^*_{ij} by a series of iterations involving the first-order paths. The problem of finding the optimal third-party graph $\mathcal{G}_k$ at iteration t becomes:

$$k^* = \arg\max_{k=1}^{N} J\big(\mathbf{X}_{ik}^{(t-1)}\mathbf{X}_{kj}^{(t-1)}\big). \tag{4}$$

This can be regarded as approximately compositing the iteratively generated piecewise paths, into a higher-order one that consists of multiple intermediate graphs.

Algorithm 1. Composition Based Affinity Optimization **CAO**

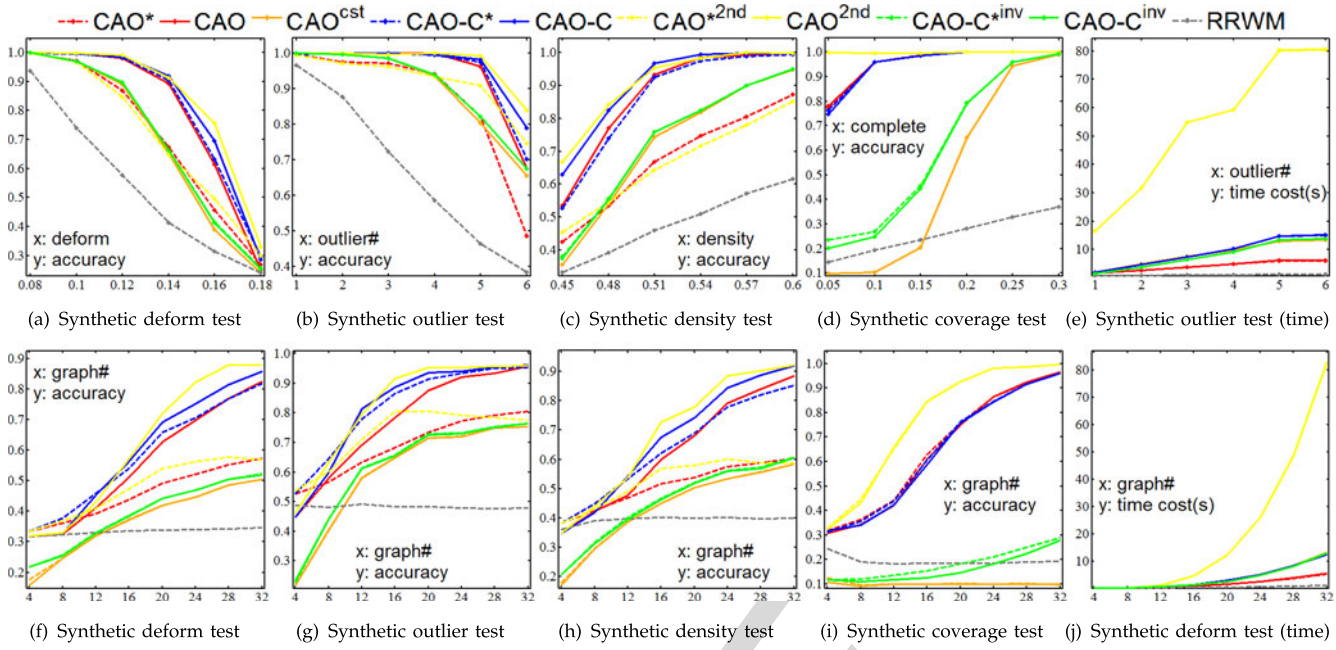
Require: $\{\mathbf{K}_{ij}\}_{i=1,j=i+1}^{N-1,N}$, T, $\gamma \in (0, 1)$;
1: Perform pairwise matching to obtain initial $\mathbb{X}^{(0)}$;
2: Calculate $J^{(0)} = \sum_{i=1,j=i+1}^{N-1,N} \mathrm{vec}(\mathbf{X}_{ij}^{(0)})^T\mathbf{K}_{ij}\mathrm{vec}(\mathbf{X}_{ij}^{(0)})$;
3: **for** $t = 1 : T$ **do**
4: **for all** $i = 1, 2, \ldots, N-1; j = i+1, \ldots, N$ **do**
5: update $\mathbf{X}_{ij}^{(t)} = \mathbf{X}_{ik}^{(t-1)}\mathbf{X}_{kj}^{(t-1)}$ by solving Eq. (4);
6: **end for**
7: **end for**
8: **if** $C(\mathbb{X}^{(t)}) = 1$, **return** $\mathbb{X}^*_c = \mathbb{X}^{(t)}$;
9: **if** $C(\mathbb{X}^{(t)}) < \gamma$ **then**
10: Build the super graph $\mathcal{G}^a_{sup}$ by pairwise affinity $J(\mathbf{X}_{ij}^{(t)})$ and find a maximum span tree to generate $\mathbb{X}^*_c$;
11: **else**
12: If $n \geq N$, build the super graph $\mathcal{G}^c_{sup}$ by $C_p(\mathbf{X}_{ij}^{(t)}, \mathbb{X}^{(t)})$ and find a maximum span tree to generate $\mathbb{X}^*_c$; Otherwise, perform the smoothing method [30] to obtain $\mathbb{X}^*_c$;
13: **end if**

The above idea is concretized into an iterative algorithm termed as *Composition based Affinity Optimization (CAO)* as described in the chart of Algorithm 1. At iteration t, every $\mathbf{X}_{ij}^{(t)}$ is updated by seeking the path with order $s = 1$ via maximizing the affinity score according to Eq. (4). The efficacy of such a composition driven affinity optimization strategy can be justified by an intuitive analysis: even though matching $\mathcal{G}_i$ and $\mathcal{G}_j$ is inherently ambiguous when both are largely corrupted, it is still possible to improve matching accuracy by an intermediate graph $\mathcal{G}_k$ that can find the quality pairwise matches between $\{\mathcal{G}_i, \mathcal{G}_k\}$ and $\{\mathcal{G}_k, \mathcal{G}_j\}$ respectively. The following proposition depicts the convergence of Algorithm 1.

Proposition 1. *Algorithm 1 (CAO) is ensured to converge to a stationary configuration $\mathbb{X}^*$ after a finite number of iterations.*

Proof. For each $J(\mathbf{X}_{ij})$, it forms a non-descending sequence over iterations which is bounded by the upper bound $\widetilde{J}_{ij} = \max\{\mathrm{vec}(\mathbf{Y})^T\mathbf{K}_{ij}\mathrm{vec}(\mathbf{Y}), \mathbf{Y} \in \mathbb{P}_n\}$ in the discrete permutation space $\mathbb{P}_n$: $J(\mathbf{X}_{ij}^{(0)}) \leq J(\mathbf{X}_{ij}^{(1)}) \leq \cdots \leq J(\mathbf{X}_{ij}^{(t)}) \leq \cdots \leq \widetilde{J}_{ij}$. Thus $\mathbb{X}^{(t)}$ will converge. □

In fact, Algorithm 1 (CAO) cannot guarantee the convergent $\mathbb{X}^*$ satisfying full consistency i.e., $C(\mathbb{X}^*) = 1$ in Definition (3). Thus, after the iteration procedure, a post step for enforcing full consistency is *optionally* performed.



(a) Synthetic deform test (b) Synthetic outlier test (c) Synthetic density test (d) Synthetic coverage test (e) Synthetic outlier test (time)

(f) Synthetic deform test (g) Synthetic outlier test (h) Synthetic density test (i) Synthetic coverage test (j) Synthetic deform test (time)

Fig. 1. Comparison of CAO, CAO$^{\text{cst}}$, CAO$^{\text{2nd}}$, CAO-C and CAO-C$^{\text{inv}}$ on the synthetic random graphs. RRWM [16] is used as the pairwise matcher to generate the initial configuration ('RRWM'). 'CAO*' denotes CAO dismissing the step of L9-13, so for other methods (best viewed in color). Refer to Table 1 for settings.

For a heuristical implementation, when the resultant $C(\mathbb{X}^*)$ is less than a given threshold γ, which suggests the consistency metric is not reliable, the affinity-wise super graph $\mathcal{G}^a_{sup}$ in Definition (6) is built to generate the final fully consistent $\mathbb{X}^*_c$. Otherwise, we utilize consistency to smooth the final solution in two cases: i) when $N > n$ i.e., the number of graphs is larger than the number of nodes, the method in [30] is adopted to obtain $\mathbb{X}^*_c$; ii) otherwise, the consistency-wise super graph $\mathcal{G}_c$ in Definition (7) is built to obtain $\mathbb{X}^*_c$. We apply this strategy in all our algorithms for post-processing to achieve full-consistency. Yet how to realize full consistency in post-step is not the focus of this paper and it has also been addressed by other works such as [2], [30]. Moreover, in the presence of a considerable number of outliers, enforcing the full consistency may cause performance degeneration due to unmatchable outliers as will be shown in our experiments (Fig. 3).

In order to obtain a comprehensive view of the affinity optimization model, we consider two additional variants of CAO (Algorithm 1). One uses pairwise consistency instead of affinity to update solutions. The other replaces the first-order path selection with a second-order one. In fact, there exist other random search strategies aside from the fist-order one described here, and these random search variants can be derived by our framework.

The first variant uses the following equation, regarding with the pairwise consistency in Definition (2), to replace Eq. (4) in the step of updating $\mathbf{X}^{(t)}_{ij}$:

$$k^* = \arg\max_{k=1}^{N} C_p\big(\mathbf{X}^{(t-1)}_{ik}\mathbf{X}^{(t-1)}_{kj}, \mathbb{X}^{(t-1)}\big). \tag{5}$$

Unlike CAO, this consistency-driven variant CAO$^{\text{cst}}$ (note superscript added) cannot ensure to converge to a stationary configuration. In particular, the consistency optimization mechanism at each round, can only guarantee $C_p(\mathbf{X}^{(t)}_{ij}, \mathbb{X}^{(t-1)}) \geq C_p(\mathbf{X}^{(t-1)}_{ij}, \mathbb{X}^{(t-1)})$ for any pair of i, j. Nevertheless, it cannot ensure $C_p(\mathbf{X}^{(t)}_{ij}, \mathbb{X}^{(t)}) \geq C_p(\mathbf{X}^{(t-1)}_{ij}, \mathbb{X}^{(t-1)})$. Thus the non-decreasing property $C_p(\mathbf{X}^{(0)}_{ij}, \mathbb{X}^{(0)}) \leq C_p(\mathbf{X}^{(1)}_{ij}, \mathbb{X}^{(1)}) \leq \dots$ does not hold, although they are bounded by $C_p(\mathbf{X}^{(t)}_{ij}, \mathbb{X}^{(t)}) \leq 1$. For similar reasons, one cannot ensure $C(\mathbb{X}^{(t)}) \geq C(\mathbb{X}^{(t-1)})$ where $C(\cdot)$ is defined in Definition (3). As the solution space is discrete and finite, CAO$^{\text{cst}}$ either converges to a stationary point or forms a looping solution path. The former case is much more often observed in our tests.

The second-order variant CAO$^{\text{2nd}}$ is derived by replacing $\mathbf{X}^{(t)}_{ij} = \mathbf{X}^{(t-1)}_{ik}\mathbf{X}^{(t-1)}_{kj}$ with $\mathbf{X}^{(t)}_{ij} = \mathbf{X}^{(t-1)}_{iv}\mathbf{X}^{(t-1)}_{vu}\mathbf{X}^{(t-1)}_{uj}$:

$$u^*, v^* = \arg\max_{u,v=1}^{N} \mathbf{y}^T\mathbf{K}_{ij}\mathbf{y}, \quad \mathbf{y} = \mathrm{vec}\big(\mathbf{X}^{(t-1)}_{iv}\mathbf{X}^{(t-1)}_{vu}\mathbf{X}^{(t-1)}_{uj}\big). \tag{6}$$

This method has better exploration capability than the first-order one, at the expense of growing searching space from $O(N)$ to $O(N^2)$ in terms of traversing the rest graphs (or graph pairs for the second-order case). As the same with CAO, the second-order method can also guarantee convergence due to its score non-decreasing property. These three methods are evaluated in Fig. 1 together with Algorithm 2 (CAO-C) as will be introduced later. See Table 1 for the settings.

One can see that CAO outperforms CAO$^{\text{cst}}$ notably. This is because CAO$^{\text{cst}}$ is based on the assumption that correct matchings are dominant and the meaningful correspondences can be realized along many different paths of matchings. This assumption breaks given an unsatisfactory initial $\mathbb{X}^{(0)}$. For CAO$^{\text{2nd}}$, it outperforms the first-order method, while its overhead becomes significantly larger when more graphs are involved. Note Algorithm 2 (CAO-C) and its variant CAO-C$^{\text{inv}}$ also appear in Fig. 1. We leave them to the next section.

Algorithm 2. Composition Based Affinity Optimization via Graduated Consistency Regularization **CAO-C**

Require: $\{\mathbf{K}_{ij}\}_{i=1,j=i+1}^{N-1,N}$; T, $\lambda^{(T_0)}=\lambda^{(0)}$, $\{\lambda^{(t)}\}_{t=1}^{T_0-1}=0$, γ;
1: Perform pairwise matching to obtain initial $\mathbb{X}^{(0)}$;
2: Calculate $J^{(0)}=\sum_{i=1,j=i+1}^{N-1,N}\mathrm{vec}(\mathbf{X}_{ij}^{(0)})^T\mathbf{K}_{ij}\mathrm{vec}(\mathbf{X}_{ij}^{(0)})$;
3: **for** $t=1:T$ **do**
4: **for all** $i=1,2,\ldots,N-1; j=i+1,\ldots,N$ **do**
5: update $\mathbf{X}_{ij}^{(t)}=\mathbf{X}_{ik}^{(t-1)}\mathbf{X}_{kj}^{(t-1)}$ by solving Eq. (7);
6: **end for**
7: **if** $t>T_0$, $\lambda^{(t)}=min(1,\beta\lambda^{(t-1)})$;
8: **end for**
9: Perform the same post-processing as in Algorithm 1 (L9-13).

3.3 Graduated Consistency Regularization

Our key rationale is viewing the consistency as a *regularizer* for affinity maximization. Note that maximizing pairwise affinity score among all pairs cannot always ensure the consistency constraint. There are two facts from which the bias between affinity score and true accuracy comes: i) corruption associated with the raw feature for constructing the affinity function, such as outliers, missing measurements due to sparse edge sampling, and deformation etc.; ii) the difficulty in parameterizing the affinity function using unary and pairwise features. As a result, there can be a case that for some pairs of graphs, the true ground truth matching configuration does not correspond to the highest affinity score. Thus purely maximizing the overall affinity is biased to accuracy, though maximizing overall consistency alone (Algorithm 1$^{\mathrm{cst}}$) is even more biased as has been studied in the previous section. This is roughly analogous to loss function modeling in machine learning, where one not only considers empirical loss on the training dataset, but also employs a regularizer to account for over-fitting.

One shall note as a baseline method, CAO separates affinity score maximization and consistency smoothing into two independent steps. It is yet appealing to tackle the two aspects jointly, which is motivated by two facts: i) For the initial assignment matrix $\mathbb{X}^{(0)}$ obtained by the pairwise graph matching solver, its scores are more informative for the true accuracy; ii) After several iterations of affinity improvement, affinity becomes less discriminative and consistency becomes more indicative.

In this spirit, we infuse the matching consistency by a weighted term, whose weight λ is gradually increased. As a result, the evaluation function at each iteration is changed from Eq. (4) to a weighted one[1] between Eq. (4) and Eq. (5) by setting $\mathbf{Y}_{ikj}^{(t-1)}=\mathbf{X}_{ik}^{(t-1)}\mathbf{X}_{kj}^{(t-1)}$:

$$k^*=\arg\max_{k=1}^{N}(1-\lambda)J\big(\mathbf{Y}_{ikj}^{(t-1)}\big)+\lambda C_p\big(\mathbf{Y}_{ikj}^{(t-1)},\mathbb{X}^{(t-1)}\big). \quad (7)$$

This method is detailed in the chart of Algorithm 2: *Composition based Affinity Optimization via Graduated Consistency Regularization* **CAO-C**.

1. Note the affinity is not directly comparable with either the unary or pairwise consistency as the latter fall in $[0,1]$ while the former is arbitrary depending on how the affinity matrix is set. In our implementation, the affinity score is further normalized by $J(\mathbf{X}_{ij})=J(\mathbf{X}_{ij})/\max_{ij}\{J(\mathbf{X}_{ij}^{(0)})\}$ where the denominator is a constant. This convention is also used when J^{ψ} is introduced in Section 3.5.

Similar to CAO$^{\mathrm{cst}}$, CAO-C in general cannot guarantee to converge to a stationary configuration, since the non-decreasing property of the sequence $\{C_p(\mathbf{X}_{ij}^{(t)},\mathbb{X}^{(t)})\}_{t=0}^{\infty}$ does not always hold. Our empirical tests show this method often converges to a stationary $\mathbb{X}^*$ after 10 iterations or so. In our experiments, we stop it when the number of iterations arrives at a certain threshold T.

To make our study more comprehensive, like CAO$^{\mathrm{cst}}$ to CAO, we further devise a counterpart to CAO-C by swapping the role of consistency and affinity: the iteration is driven by choosing the anchor graph that lifts the pairwise consistency by Eq. (5) for CAO$^{\mathrm{cst}}$. The affinity weight is increased akin to the consistency used in CAO-C. Hence, the evaluation function becomes:

$$k^*=\arg\max_{k=1}^{N}\lambda J\big(\mathbf{Y}_{ikj}^{(t-1)}\big)+(1-\lambda)C_p\big(\mathbf{Y}_{ikj}^{(t-1)},\mathbb{X}^{(t-1)}\big). \quad (8)$$

We call this algorithm CAO-C$^{\mathrm{inv}}$ where the superscript denotes for 'inverse'. The parameter settings λ^0 and β are identical to CAO-C. We omit the algorithm chart for CAO-C$^{\mathrm{inv}}$ since it is akin to CAO-C. It is outperformed by CAO-C as shown in Fig. 1, which validates our idea.

3.4 Two Efficient Variants

The step of computing $C_p(\mathbf{X}_{ik}^{(t-1)}\mathbf{X}_{kj}^{(t-1)},\mathbb{X}^{(t-1)})$ in CAO-C is repeated for each new k because it depends on the generated candidate $\mathbf{X}_{ik}^{(t-1)}\mathbf{X}_{kj}^{(t-1)}$. Therefore, its complexity for the consistency term is $O(N^2n^3)$ for updating each $\mathbf{X}_{ij}^{(t)}$. The overall cost is $O(N^4n^3)$ per iteration. To reduce this overhead, we are motivated to use two efficient consistency metrics to delegate $C_p(\mathbf{X}_{ik}^{(t-1)}\mathbf{X}_{kj}^{(t-1)},\mathbb{X}^{(t-1)})$.

The first proxy metric is the unary consistency as described in Definition (1). Note this metric is also what we adopt in our preliminary work [1] where it is referred as 'Algorithm 2'. Its evaluation function is:

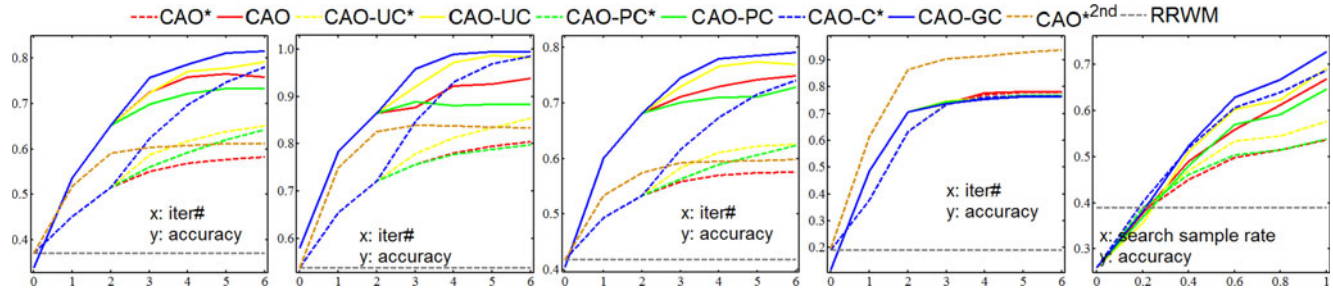
$$k^*=\arg\max_{k=1}^{N}(1-\lambda)J\big(\mathbf{X}_{ik}^{(t-1)}\mathbf{X}_{kj}^{(t-1)}\big)+\lambda C_u\big(k,\mathbb{X}^{(t-1)}\big). \quad (9)$$

The merit of this metric is all $\{C_u(k,\mathbb{X}^{(t-1)})\}_{k=1}^{N}$ can be pre-computed at each iteration only for once.

Alternatively, we devise another consistency estimator by first pre-computing all $\{C_p(\mathbf{X}_{ij}^{(t-1)},\mathbb{X}^{(t-1)})\}_{i=1,j=i+1}^{N-1,N}$ in the beginning of each iteration $t-1$, and then adopting Eq. (10) in the hope of approximating $C_p(\mathbf{X}_{ik}^{(t-1)}\mathbf{X}_{kj}^{(t-1)},\mathbb{X}^{(t-1)})$ for updating $\mathbf{X}_{ij}^{(t)}$:

$$k^*=\arg\max_{k=1}^{N}(1-\lambda)J\big(\mathbf{X}_{ik}^{(t-1)}\mathbf{X}_{kj}^{(t-1)}\big)+\lambda\sqrt{C_p\big(\mathbf{X}_{ik}^{(t-1)},\mathbb{X}^{(t-1)}\big)C_p\big(\mathbf{X}_{kj}^{(t-1)},\mathbb{X}^{(t-1)}\big)}. \quad (10)$$

We term the two variants as *Composition based Affinity Optimization via Efficient Graduated Unary (or Pairwise) Consistency Regularization* as depicted in the chart of Algorithm 3. In this paper, they are further abbreviated by CAO-UC and CAO-PC, where the added characters 'U', 'P' denote for 'unary' and 'pairwise' respectively. CAO-UC has a convergence property as stated in below.



(a) Synthetic deform test (b) Synthetic outlier test (c) Synthetic density test (d) Synthetic coverage test (e) Synthetic sampling test

Fig. 2. Performance on synthetic random graphs by varying the maximum iteration threshold T for $T_0 = 2$ used in the algorithm charts (best viewed in color). Refer to Table 2 for settings.

Algorithm 3. Composition Based Affinity Optimization via Efficient Graduated Unary (or Pairwise) Consistency Regularization **CAO-UC, CAO-PC**

Require: $\{\mathbf{K}_{ij}\}_{i=1,j=i+1}^{N-1,N}$, T, γ, $\lambda^{(T_0)} = \lambda^{(0)}$, $\{\lambda^{(t)}\}_{t=1}^{T_0-1} = 0$;
1: Perform pairwise matching to obtain initial $\mathbb{X}^{(0)}$;
2: Calculate $J^{(0)} = \sum_{i=1,j=i+1}^{N-1,N} \mathrm{vec}(\mathbf{X}_{ij}^{(0)})^T \mathbf{K}_{ij} \mathrm{vec}(\mathbf{X}_{ij}^{(0)})$;
3: **for** $t = 1 : T$ **do**
4: Compute $\{C_u(k, \mathbb{X}^{(t-1)})\}_{k=1}^{N}$ for CAO-UC;
5: Or update $\{C_p(\mathbf{X}_{ij}^{(t-1)}, \mathbb{X}^{(t-1)})\}_{i=1,j=i+1}^{N-1,N}$ for CAO-PC;
6: **for all** $i = 1, 2, \ldots, N-1; j = i+1, \ldots, N$ **do**
7: Update $\mathbf{X}_{ij}^{(t)} = \mathbf{X}_{ik}^{(t-1)} \mathbf{X}_{kj}^{(t-1)}$ by Eq. (9) for CAO-UC;
8: Or update $\mathbf{X}_{ij}^{(t)} = \mathbf{X}_{ik}^{(t-1)} \mathbf{X}_{kj}^{(t-1)}$ by Eq. (10) for CAO-PC;
9: **end for**
10: **if** $t > T_0$, $\lambda^{(t)} = min(1, \beta\lambda^{(t-1)})$;
11: **end for**
12: Perform the same post-processing as in Algorithm 1 (L9-13).

Proposition 2. *CAO-UC will converge to a stationary* $\mathbb{X}^*$.

Proof. Given two graphs $\mathcal{G}_i$, $\mathcal{G}_j$ of n nodes for each, first, define the set of score difference $\{\Delta J_{ij}\}$ as $\Delta J_{ij} = |\mathrm{vec}(\mathbf{X})^T \mathbf{K}_{ij} \mathrm{vec}(\mathbf{X}) - \mathrm{vec}(\mathbf{Y})^T \mathbf{K}_{ij} \mathrm{vec}(\mathbf{Y})|$, $\forall \mathbf{X}, \mathbf{Y}$, between any two assignment matrices $\mathbf{X}$, $\mathbf{Y} \in \mathbb{R}^{n \times n}$ in the enumerable permutation space. Suppose the largest value of difference is $\Delta\widetilde{J}_{ij}$ which is constant given $\mathbf{K}_{ij}$. We further define the largest difference denoted by $\Delta\widetilde{J}_{\mathcal{G}} = \max\{\Delta\widetilde{J}_{ij}\}_{i=1,j=i+1}^{N-1,N}$ for all pairs in the graph.

For a certain iteration in CAO-UC, suppose the most consistent graph in Definition (1) is $\mathcal{G}_a$, and the second one is $\mathcal{G}_b$. At iteration t it will finally satisfy the condition: $\Delta C_u^{(t)} = C_u(a, \mathbb{X}^{(t)}) - C_u(b, \mathbb{X}^{(t)}) > \frac{(1-\lambda)}{\lambda} \Delta\widetilde{J}_{\mathcal{G}}$ as $\lambda^{(t)} \to 1$ by $\lambda^{(t)} = min(\rho\lambda^{(t-1)}, 1)$.[2] Then, the following inequality will hold for any $k \neq a$ and $\{\mathcal{G}_i\}_{i=1}^{N-1}, \{\mathcal{G}_j\}_{j=i+1}^{N}$:

$$
\begin{aligned}
&(1-\lambda) J\big(\mathbf{X}_{ia}^{(t)} \mathbf{X}_{aj}^{(t)}\big) + \lambda C_u\big(a, \mathbb{X}^{(t-1)}\big) \\
&= (1-\lambda) J\big(\mathbf{X}_{ia}^{(t)} \mathbf{X}_{aj}^{(t)}\big) + \lambda C_u(b, \mathbb{X}^{(t-1)}) + \lambda \Delta C_u^{(t)} \\
&= (1-\lambda)\big(J\big(\mathbf{X}_{ia}^{(t)} \mathbf{X}_{aj}^{(t)}\big) - J\big(\mathbf{X}_{ik}^{(t)} \mathbf{X}_{kj}^{(t)}\big)\big) + (1-\lambda) J\big(\mathbf{X}_{ik}^{(t)} \mathbf{X}_{kj}^{(t)}\big) \\
&\quad + \lambda C_u(b, \mathbb{X}^{(t-1)}) + \lambda \Delta C_u^{(t)} \\
&\geq \lambda \Delta C_u^{(t)} - \Delta\widetilde{J}_{\mathcal{G}} + (1-\lambda) J\big(\mathbf{X}_{ik}^{(t)} \mathbf{X}_{kj}^{(t)}\big) + \lambda C_u(k, \mathbb{X}^{(t-1)}) \\
&> (1-\lambda) J\big(\mathbf{X}_{ik}^{(t)} \mathbf{X}_{kj}^{(t)}\big) + \lambda C_u(k, \mathbb{X}^{(t-1)}).
\end{aligned} \tag{11}
$$

As a result, all $\{\mathbf{X}_{ij}^{(t+1)}\}_{i=1,j=i+1}^{N-1,N}$ will be updated by $\mathbf{X}_{ij}^{(t+1)} = \mathbf{X}_{ia}^{(t)} \mathbf{X}_{aj}^{(t)}$. In fact, at iteration t, $\mathbb{X}^{(t)}$ becomes fully consistent in Definition (3), and cannot be lifted by either affinity or consistency metrics used in this paper. □

TABLE 1
Settings for Synthetic Test in Fig. 1, Fig. 3

parameter settings	results
$N = 30, n_i = 10, n_o = 0, \rho = .9, \sigma^2 = .05, c = 1, T = 6$	Fig. 1a, Fig. 3a
$N = 30, \varepsilon = .05, n_i = 8, \rho = 1, \sigma^2 = .05, c = 1, T = 6$	Fig. 1b, Fig. 3b
$N = 30, n_i = 10, n_o = 0, \varepsilon = .05, \sigma^2 = .05, c = 1, T = 6$	Fig. 1c, Fig. 3c
$N = 30, n_i = 10, n_o = 0, \varepsilon = .05, \rho = 1, \sigma^2 = .05, T = 6$	Fig. 1d, Fig. 3d
$n_i = 10, n_o = 0, \varepsilon = .15, \rho = .9, \sigma^2 = .05, c = 1, T = 6$	Fig. 1f, Fig. 3f
$n_i = 6, n_o = 4, \varepsilon = 0, \rho = 1, \sigma^2 = .05, c = 1, T = 6$	Fig. 1g, Fig. 3g
$n_i = 10, n_o = 0, \varepsilon = 0, \rho = .5, \sigma^2 = .05, c = 1, T = 6$	Fig. 1h, Fig. 3h
$n_i = 10, n_o = 0, \varepsilon = .05, \rho = 1, \sigma^2 = .05, c = .1, T = 6$	Fig. 1i, Fig. 3i

The iterative procedure in CAO-UC may run out of iterations before it converges to a stationary and consistent configuration. A post-processing step is needed in line with Algorithm 1 (CAO) and Algorithm 2 (CAO-C) if necessary.

CAO-PC also often converges to a stationary $\mathbb{X}^*$ similar to CAO-C, yet there is no theoretical guarantee. Fig. 2 illustrates how the overall matching accuracy is lifted via the proposed algorithms (settings in Table 2).

In summary, Algorithm 1, Algorithm 2, Algorithm 3 can be conceptually unified in one framework by three steps: i) initialize pairwise matchings $\mathbb{X}$ via a two-graph matcher; ii) iterative affinity optimization by different mechanisms of setting the trade-off parameter λ and computing the consistency scores; iii) post-process $\mathbb{X}$ if needed. By this perspective, the state-of-the-arts [2], [30] only perform step i and iii, while [4], [31] always impose hard consistency in step ii and hence dispense with step iii. Moreover, the methods [4], [31] need to solve a two-graph matching problem per iteration, while our methods only evaluate the score by a composed solution, which is more efficient and general.

2. In case $C_u(a, \mathbb{X}^{(t)}) = C_u(b, \mathbb{X}^{(t)})$, without loss of generality, one can choose any of them as the largest one, and choose the next $\mathcal{G}_c$ as the second largest if $C_u(a, \mathbb{X}^{(t)}) > C_u(c, \mathbb{X}^{(t)})$.

TABLE 2
Settings for Synthetic Test in Fig. 2

test mode	parameter settings	results
deform	$N=20$, $n_i=10$, $n_o=0$, $\varepsilon=.15$, $\rho=.9$, $c=1$, $\sigma^2=.05$	Fig. 2a
outlier	$N=20$, $n_i=6$, $n_o=4$, $\varepsilon=0$, $\rho=1$, $c=1$, $\sigma^2=.05$	Fig. 2b
density	$N=20$, $n_i=10$, $n_o=0$, $\varepsilon=0$, $\rho=.5$, $c=1$, $\sigma^2=.05$	Fig. 2c
coverage	$N=20$, $n_i=10$, $n_o=0$, $\varepsilon=.05$, $\rho=1$, $c=.1$, $\sigma^2=.05$	Fig. 2d
sampling	$N=20$, $n_i=8$, $n_o=2$, $\varepsilon=.1$, $\rho=1$, $c=1$, $\sigma^2=.05$	Fig. 2e

3.5 Eliciting Affinity and Consistency for Inliers

One limitation of many previous studies for both pairwise and multi-graph matching, including those mentioned in this paper, is that they enforce all nodes in one graph to have matchings from the other(s), either by explicitly using a permutation matrix imposing strict node-to-node correspondences [15], [49], or implicitly doing so by generating node-to-node matchings as many as possible such that the objective affinity score is optimized [22], [23], [34], [36], [37]. A common practice when applying a pairwise GM method is assuming one of two graphs as a reference graph, such that each of its nodes can find a match in the other testing graph.

Several existing works also consider the presence of outliers. The Graduated Assignment method [22] attempts to handle outlier nodes by assigning them to additional slack variables. In the presence of a large number of outliers, this approach is not widely adopted due to its sensitivity to parameters for the slack variables. The very recent work [3] proposes a general approach via a max-pooling mechanism suited for the scenario of matching two graphs with a large number of outliers. A less relevant work is by Torresani et al. [50] which also allows nodes to be assigned in an unmatchable status in an energy function induced on the Markov Random Field. However, its complex objective function is designed that account for various similarity measurements e.g. appearance descriptors, occlusions, spatial proximity etc. beyond the general setting of the affinity matrix considered in this paper. Moreover, the associated parameters need to be learned with labeled correspondence ground truths which impedes its applicability. Other pairwise GM methods e.g. [41] integrate the point detection and matching in a synergic manner, while this paper assumes the graphs are given being not tailored to visual problems.

As a building-block, any of these pairwise GM methods can be used for our approaches and other state-of-the-arts [2], [4], [30], [31] to initialize $\mathbb{X}^{(0)}$. Nevertheless, the context of multiple graphs opens the space for a more robust mechanism to prune outliers such that only the affinity and consistency relevant to inliers are considered. We aim to devise a general outlier-rejection mechanism independent of the data such as node distribution, graph attributes and noises type etc.

We describe our 'inlier nodes' eliciting method using the node-wise consistency $C_n(u^k,\mathbb{X})$ in Definition (4). Another alternative is using the node-wise affinity $S_n(u^k,\mathbb{X},\mathbb{K})$ via Definition (5), which can be applied in a similar fashion thus its description is omitted here.

First, given the matching configuration $\mathbb{X}$, each node in one graph is scored by the node-wise consistency in Definition (4). Assume the number of common inliers i.e., n_i for all graphs is known, which is available when a reference template is given, or estimated by other means e.g. [2]. How to estimate n_i is not the focus of this paper.

Then we make two revisions for all of our methods: i) the pairwise affinity term $\mathrm{vec}(\mathbf{X})^T\mathbf{K}\mathrm{vec}(\mathbf{X})$ is modified by keeping the rows of $\mathbf{X}$ that correspond to the first n_i (the number of inliers) nodes in descending order by their node-wise consistency score $C_n(u^k,\mathbb{X})$ in Definition (4), and zeroing the rest of rows. This is because the affinity or consistency between outliers is irrelevant to the semantic matching accuracy and shall be excluded in the optimization procedure. For node-wise consistency, we use $\psi_c(\mathbf{X},\mathbb{X},n_i)$ (or $\psi_a(\mathbf{X},\mathbb{X},\mathbb{K},n_i)$ for node-wise affinity) to denote this 'mask' operation on $\mathbf{X}$ as it is determined by both the input configuration $\mathbb{X}$ and n_i (also $\mathbb{K}$ in case of the affinity metric). Then the affinity score is rewritten as $J^{\psi_c}=\mathrm{vec}(\psi_c(\mathbf{X},\mathbb{X},n_i))^T\mathbf{K}\mathrm{vec}(\psi_c(\mathbf{X},\mathbb{X},n_i))$; ii) the consistency term is modified by the following two equations for unary and pairwise consistencies:

$$C_u^{\psi_c}(k,\mathbb{X},n_i) = 1-\frac{\sum_{i=1}^{N-1}\sum_{j=i+1}^{N}\|\psi_c(\mathbf{X}_{ij}-\mathbf{X}_{ik}\mathbf{X}_{kj},\mathbb{X},n_i)\|_F}{n_iN(N-1)} \tag{12}$$

$$C_p^{\psi_c}(\mathbf{X}_{ij},\mathbb{X},n_i) = 1-\frac{\sum_{k=1}^{N}\|\psi_c(\mathbf{X}_{ij}-\mathbf{X}_{ik}\mathbf{X}_{kj},\mathbb{X},n_i)\|_F}{2n_iN}. \tag{13}$$

We use the above variants for affinity score and consistency to replace the original J, C_u and C_p in the evaluation functions: Eq. (4), Eq. (7), Eq. (9), Eq. (10). Similar steps are performed by using the node-wise affinity.

4 Experiments and Discussion

The experiments are performed on both synthetic and real-image data. The synthetic test is controlled by varying the noise level of deformation, outlier, edge density and initial pairwise matching coverage. The real images are tested by varying viewing angles, scales, shapes etc.. The matching accuracy over all graphs, is calculated by averaging all pairwise matching accuracy $\frac{\sum_{i=1}^{N-1}\sum_{j=i+1}^{N}\mathrm{Acc}_{ij}}{N(N-1)/2}$. Each Acc_{ij} computes the accuracy between $\mathbf{X}_{ij}^{\mathrm{alg}}$ and ground truth $\mathbf{X}_{ij}^{\mathrm{tru}}$: $\mathrm{Acc}_{ij}=\frac{\mathrm{tr}(\mathbf{X}_{ij}^{\mathrm{alg}}\mathbf{X}_{ij}^{\mathrm{tru}})}{\mathrm{tr}(\mathbf{1}_{n_j\times n_i}\mathbf{X}_{ij}^{\mathrm{tru}})}$. In line with [4], [15], we only calculate the accuracy for common inliers and ignore the meaningless correspondences between outliers. The above protocol is widely adopted by related works such as [15], [16].

If not otherwise specified, the parameters of our methods are universally set as: $T_0=2$, $T=6$, $\lambda^{(0)}=0.2$, $\gamma=0.3$, $\beta=1.1$. Note when $t\le T_0$, only affinity is improved without consistency regularization.

4.1 Dataset Description and Affinity Setting

4.1.1 Synthetic Random Graph Matching

The random graph test follows the common protocol of [15], [16], [22], [31], [44]. For each trial, a reference graph with n_i nodes is created by assigning a random weight to its

TABLE 3
Main Parameters for Experimental Settings

inlier#	outlier#	estimated n_i	deform	density	coverage
n_i	n_o	n_{est}	ε	ρ	c

edge, uniformly sampled from the interval $[0, 1]$. Then the 'perturbed' graphs are created by adding a Gaussian deformation disturbance to the edge weight q^r_{ij}, which is sampled from $N(0, \varepsilon)$ i.e., $q^p_{ij} = q^r_{ij} + N(0, \varepsilon)$ where the superscript 'p' and 'r' denotes for 'perturb' and 'reference' respectively. Each 'perturbed' graph is further added by n_o outliers, which can also be helpful to make the graphs of equal sizes when the input graphs are of different sizes. Its edge density is controlled by the density parameter $\rho \in [0, 1]$ via random sampling. The edge affinity is computed by $K_{ia,jb} = \exp(-\frac{(q_{ij}-q_{ab})^2}{\sigma^2})$ where σ^2 is the edge similarity sensitivity parameter. No single-node feature is used and the unary affinity $K_{ia,ia}$ is set to zero, leaving the matching score entirely to the pairwise geometric information. In addition, we control the 'coverage' of the initial matching configuration $\mathbb{X}$ by parameter $c \in [0, 1]$, which denotes the rate of matching pairs generated by the pairwise matching solver, and the rest $1 - c$ portion of pairs are assigned with a randomly generated matching solution. A description of these parameters is listed in Table 3.

4.1.2 Synthetic Random Point Set Matching

The random point set matching is also explored as tested in [3], [43]. First, n_i inliers $\{\mathbf{p}_k\}_{k=1}^{n_i}$ are randomly generated on the 2-D plane via Gaussian distribution $N(0, 1)$ as the reference point set. Then each point is copied with Gaussian noise $N(0, \varepsilon)$ to generate N random points by further adding n_o outliers via $N(0, 1)$. The edge weight is set by the Euclidean distances $q_{ij} = \|\mathbf{p}_i - \mathbf{p}_j\|$ for each graph. The subsequent steps are the same with the random graph matching case. This dataset is used for the synthetic testing in comparison with MPM [3] under a relatively large number of outliers. This is because the max-pooling method MPM is designed for geometric relation rather than arbitrary edge weights as in the random graph matching setting.

4.1.3 CMU-POSE Sequence

It contains four sequences. Two sequences are from the CMU house (30 landmarks, 101 frames), hotel (30 landmarks, 111 frames) sequence (http://vasc.ri.cmu.edu//idb/html/motion/) which are commonly used in [15], [16], [23], [31], [34]. The other two sequences are sampled from the 'VolvoC70' and the 'houseblack' (both 19 landmarks, 225 frames) covering a range of 70 degrees of viewing angles from the POSE dataset [51]. We use this data for the 'partial similarity' test. We select $n_i = 10$ landmarks out of all n_{ant} annotated landmarks, and randomly choose $n_o = 4$ nodes from the rest $n_{ant} - n_i$ nodes as 'outliers'. We perform edge sampling following the same way as [15] by constructing the sparse delaunay triangulation among the points (no distinction to inliers or outliers). The affinity matrix is set by $K_{ia,jb} = \exp(\frac{(d_{ij}-d_{ab})^2}{-\sigma^2})$ where d_{ij}, d_{ab} are the Euclidean distances between two points normalized to $[0, 1]$ by dividing the largest edge length. The diagonal is set zero as the same as [15], [16].

4.1.4 WILLOW-ObjectClass

This dataset (ObjectClass for short) released in [34] is constructed using images from Caltech-256 and PASCAL VOC2007. Each object category contains different number of images: 109 Face, 50 Duck, 66 Wine bottle, 40 Motorbike, and 40 Car images. For each image, 10 feature points are manually labeled on the target object. We further add n_o random outliers detected by a SIFT detector in our outlier test. The edge sampling for the affinity matrix follows the same way as [15] by constructing the sparse delaunay triangulation among the landmarks as done in the CMU-POSE sequence test. We use the protocol of [15], [34] that sets the final affinity matrix reweighted by the edge length affinity and angle affinity: $K_{ia,jb} = \beta K^{\text{len}}_{ia,jb} + (1 - \beta)K^{\text{ang}}_{ia,jb}$ where $\beta \in [0, 1]$ is the weighting parameter. This is because this dataset contains more ambiguities for structural symmetry as pointed out in [4], than the sequence dataset if only length information is used. The angle for each edge is computed as the absolute angle between the edge and the horizontal line as used in [15]. The edge affinity and angle affinity are calculated as in the CMU-POSE test, and leave unary terms zero.

Fig. 7 illustrates the visual results on real images.

4.2 Comparing Methods and Time Complexity

The implementations of all comparing methods are the authors' Matlab code and all tests run on a laptop (2.9 G Intel Core I7 and 8 G memory) with a single thread.

4.2.1 Re-Weighted Random Walk Matching (RRWM)

Since many existing multi-graph matching methods [2], [4], [28], [30], [31], [45] require a pairwise GM solver in different ways, we choose RRWM [16] due to its cost-effectiveness. We set its parameters $\alpha = 0.2$, $\beta = 30$.

4.2.2 Max-Pooling Matching (MPM)

MPM [3] is an outlier-tolerant method computing the affinity score of each candidate match via maximal support from nearby matches. This method is tested in the presence of more outliers.

Several state-of-the-art multi-graph matching methods are evaluated, and all methods start with an initial matching configuration $\mathbb{X}^{(0)}$ obtained by RRWM.[3]

4.2.3 Alternating Optimization for Multi-Graph Matching (MatchOpt)

MatchOpt [4] transforms the multi-graph matching problem into a pairwise one by a star structure on which the alternating updating is carried out. Both factorized and

3. Due to space limitation, we have to omit a few peer multi-graph and point-set matching methods, including graduated assignment based common labeling [32], [47], the joint feature matching [52] and [11]. The method in [32], [47] has been evaluated in [4], and [2] is a more advanced method from [11]. The feature matching method [52] only considers unary features descriptors rather than the second-order or higher-order information as formulated in the GM problem.

TABLE 4
Complexity Comparison of the State of the Art

algorithm	time complexity
CAO, CAO-UC, CAO-PC	$O(N^3n^3 + N^2\tau_{\text{pair}})$
CAO-C	$O(N^4n + N^3n^3 + N^2\tau_{\text{pair}})$
MatchLift [2]	$O(N^3n^3 + N^2\tau_{\text{pair}})$
MatchOpt [4]	$O(N^2n^4 + N^3n + N^2\tau_{\text{pair}})$
MatchSync [30]	$O(N^2n^3 + N^2\tau_{\text{pair}})$

non-factorized models are devised. Here the latter is compared as it has been shown in [4] more cost-effective. We set its iteration threshold $T = 4$.

4.2.4 Match Lifting via Convex Relaxation (MatchLift)

MatchLift [2] adopts a first-order approximate algorithm. The main cost $O(N^3n^3)$ is computing the eigenvalues. This solver typically requires 10-20 rounds iterations to reach an optimum, and thus we set $T = 20$.

4.2.5 Permutation Synchronization (MatchSync)

MatchSync [30] uses spectral analysis to the grouped 'target matrix' $\mathcal{T}$ of size $nN \times nN$ (Eq. (7) in [30]) stacked by all initial $\{\mathbf{X}_{ij}^{(0)}\}_{i,j=1}^{N}$. Its largest cost $O(n^3N^2)$ refers to the one-shot SVD to find the n leading eigenvectors of '$\mathcal{T}$'. As MatchSync is applicable only when $n \leq N$, thus the Maximum Spanning Tree is used on the consistency-wise super graph $\mathcal{G}_{sup}^{c}$ if $n > N$.

The overall time complexity for all compared methods is summarized in Table 4. Now we consider the time complexity of our methods in several aspects.

First, computing the pairwise score $J(\mathbf{X}_{ik}\mathbf{X}_{kj})$ in all our methods by trying $N-2$ anchor graph $\{\mathcal{G}_k\}_{k=1,\neq i,j}^{N}$ is repeated for $\frac{N(N-1)}{2}$ pairs, at the unit cost of $O(n^3)$ as $\mathbf{X}_{ij}$ is a sparse permutation matrix. Thus the related overhead is $O(N^3n^3)$, which is also the complexity of CAO. Computing the initial $\mathbb{X}^{(0)}$ requires $O(N^2\tau_{\text{pair}})$ depending on the pairwise matching solver.

The second category refers to the computing of the pairwise consistency $\{C_p(\mathbf{X}_{ik}\mathbf{X}_{kj}, \mathbb{X})\}_{i,j,k=1}^{N}$ in CAO-C. The complexity of C_p is $O(Nn)$, thus similar to the first category, the total overhead per iteration is $O(N^4n)$.

The third one refers to the off-line computing unary and pairwise consistency as used in CAO-UC or CAO-PC. The cost for computing each C_u can be reduced to $O(N^2n)$, thus the overall cost of computing $\{C_u(k, \mathbb{X})\}_{k=1}^{N}$ for N graphs is $O(N^3n)$. Thus the overall complexity of CAO-UC is $O(N^3n^3) + O(N^3n) + O(N^2\tau_{\text{pair}})$. On the other hand, computing the off-line pairwise consistency $\{C_p(\mathbf{X}_{ij}^{(t-1)}, \mathbb{X}^{(t-1)})\}_{i=1,j=i+1}^{N-1,N}$ costs $O(N^3n)$ at the expense of $O(Nn)$ for each C_p. The overall complexity of CAO-PC is the same with CAO-UC.

Finally, the inlier-eliciting $\psi_c(\cdot, \mathbb{X}, n_i)$ involves computing the node-wise consistency $\{C_n(u^k, \mathbb{X})\}_{u^k=1,k=1}^{n_i,N}$ for each node per graph, thus the total overhead per iteration is $O(N^3n)$ by $O(N^2n)$ per graph. For $\psi_a(\cdot, \mathbb{X}, \mathbb{K}, n_i)$, the node-wise affinity in Definition (5) is $O(N^2n^3)$. The overhead of ψ_c, ψ_a can both be absorbed in the overall complexity as shown in Table 4.

4.3 Experimental Results and Discussion

For the 'RRWM' and 'MPM' curves in the plots, they are generated by applying the two methods on each pair of graphs independently. 'CAO*' denotes performing CAO without running the post step L9-13 in the chart of Algorithm 1 and so for other methods. Fifty random trials are performed for all synthetic tests. And we plot the average results in the figures. For real image data, 20 random trials are sampled from the image set for each setting. For clarity, standard deviations are not plotted as they are found stable over curves.

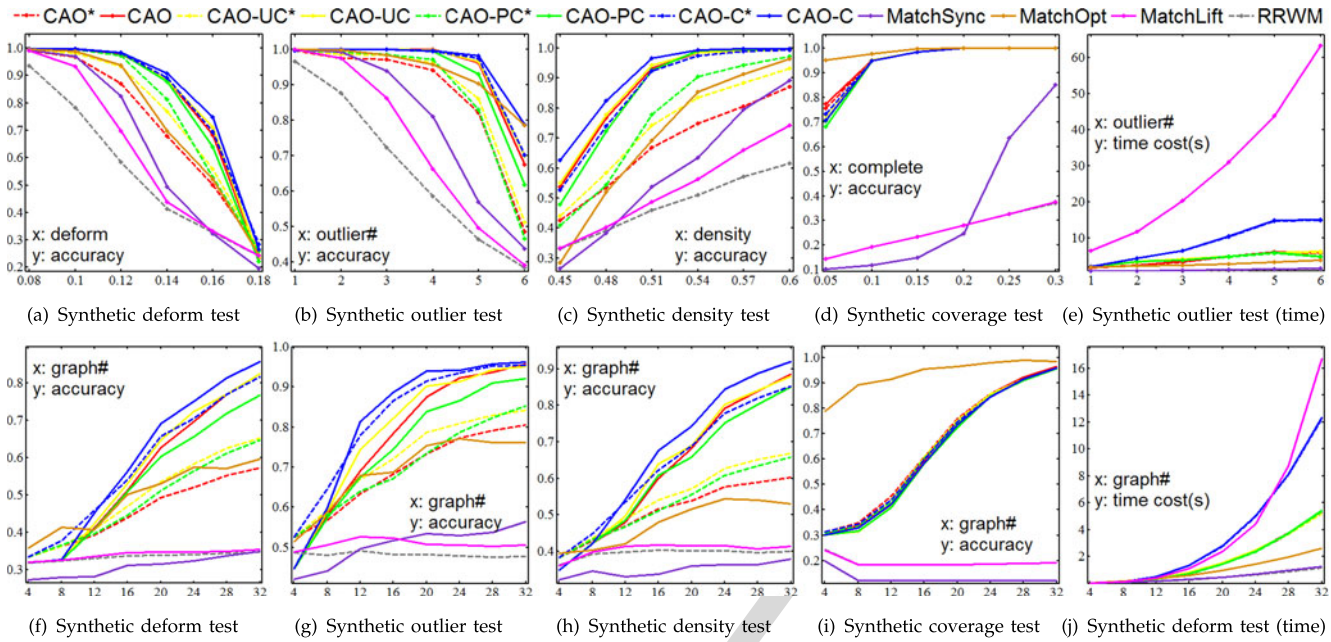
4.3.1 In Case of Few Outliers

Fig. 1 gives a comparison with controlled noise level, for several putative methods in this paper besides the main algorithms: {CAO, CAO$^{\text{cst}}$, CAO$^{\text{2nd}}$} in Section 3.2 and {CAO-C, CAO-C$^{\text{inv}}$} in Section 3.3. Fig. 2 shows the accuracy optimization behavior as a function of T and based on this plot, we set $T = 6$ in all our tests. There is little noise in the affinity function for the coverage test in Fig. 2d, thus pure affinity-driven approaches outperform others. In other cases, CAO and CAO$^{\text{2nd}}$ under-perform others and the latter even slightly degenerates as T grows. Fig. 2e suggests the relation between the accuracy and the sampling rate r for the $(N-2) * r$ graphs used from all graphs over the exhaustive searching space, to find the best anchor graph. Since our framework is applicable to other random search strategies, which might also depend on the specific applications. Here we do not dwell on how to design an efficient random search technique. We summarize the results as follows:

i) The proposed main method CAO-C outperforms the baseline CAO and state-of-the-arts in most cases for both synthetic random graphs (Fig. 3) and real images (Fig. 4), except for 'coverage' test (Figs. 3d and 3i). In 'coverage' test, many pairwise matchings in the initial $\mathbb{X}^{(0)}$ are randomly generated and a consistent $\mathbb{X}^*$ is biased to the true accuracy. When $\mathbb{X}^{(0)}$ is largely corrupted as controlled by coverage rate c, the methods utilizing affinity information including [4] and ours, outperform MatchLift [2] and MatchSync [30] that only use the initial matching configuration $\mathbb{X}^{(0)}$ as input.

ii) As shown in the bottom row of Fig. 3 in addition with Fig. 4, the accuracy in general increases as the number of graphs N grows, though there is some fluctuations especially for the test on 'car' as shown in Fig. 4a. We think this is due to the inherent matching difficulty of the sampled images, which is evidenced by the baseline 'RRWM' whose performance also fluctuates. We find this phenomenon is more pronounced as shown in the bottom row of Fig. 4 due to the larger viewing range when more frames are sampled. In fact the *relative* accuracy improvement against the RRWM baseline is increasing as N grows. This is in line with the intuition that more graphs can help dismiss local ambiguity.

iii) For the 'partial similarity' CMU-POSE data test which is often the case in reality, as shown in the bottom row of Fig. 4, without enforcing full consistency, i.e., the steps of (L9-13) described in Algorithm 1 (dashed curve) for the proposed algorithms often obtains higher accuracy than using this constraint (solid



(a) Synthetic deform test (b) Synthetic outlier test (c) Synthetic density test (d) Synthetic coverage test (e) Synthetic outlier test (time)

(f) Synthetic deform test (g) Synthetic outlier test (h) Synthetic density test (i) Synthetic coverage test (j) Synthetic deform test (time)

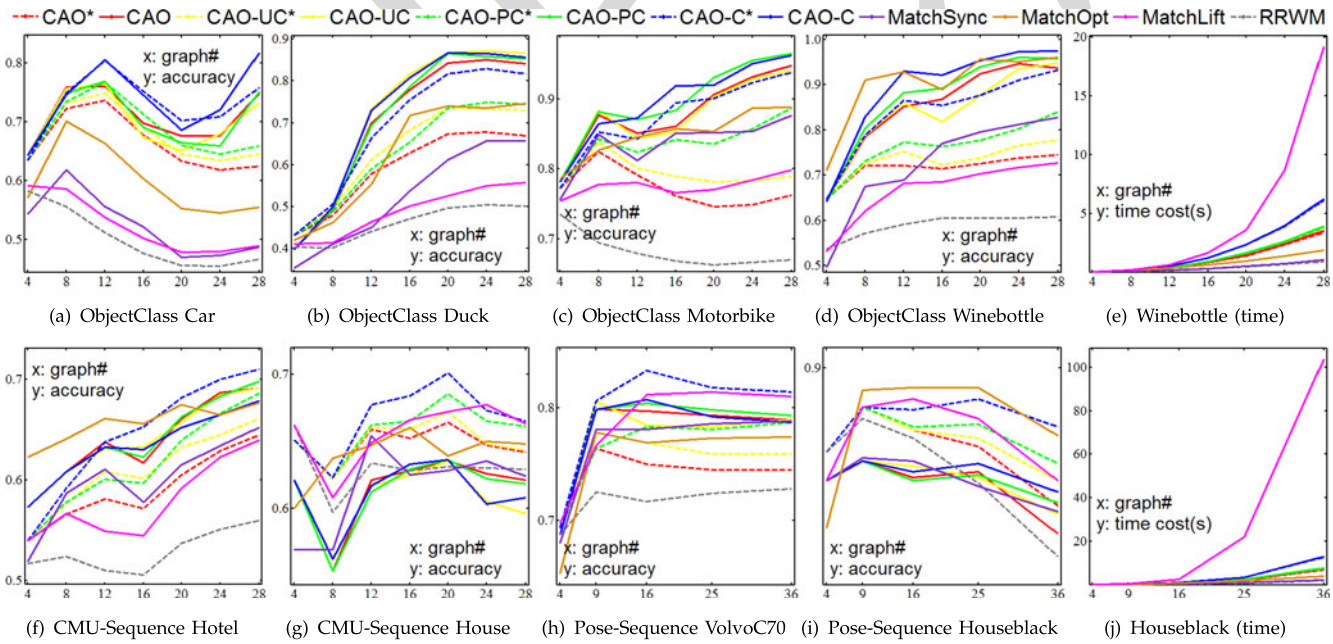
Fig. 3. In case of few outliers: evaluation for CAO, CAO-C and the two variants CAO-UC and CAO-PC, and state of the art on the synthetic random graph dataset by varying the disturbance level (top row), and by varying the number of considered graphs (bottom row). Refer to Table 1 for settings (best viewed in color).

curve). It suggests the advantage of the graduated consistency regularized mechanism against the two-step hard synchronization methodology used in CAO.

4.3.2 *In Case of More Outliers*

Our methods are tailored for the presence of a large number of outliers as described in Section 3.5. The performance is depicted in Figs. 5 and 6, for the tests on synthetic point sets (inline with the setting for the compared method MPM [3]) and images, by varying the number of outliers, graphs and the estimated inliers. No post-synchronization is performed in the outlier tests. We analyze the results as follows:

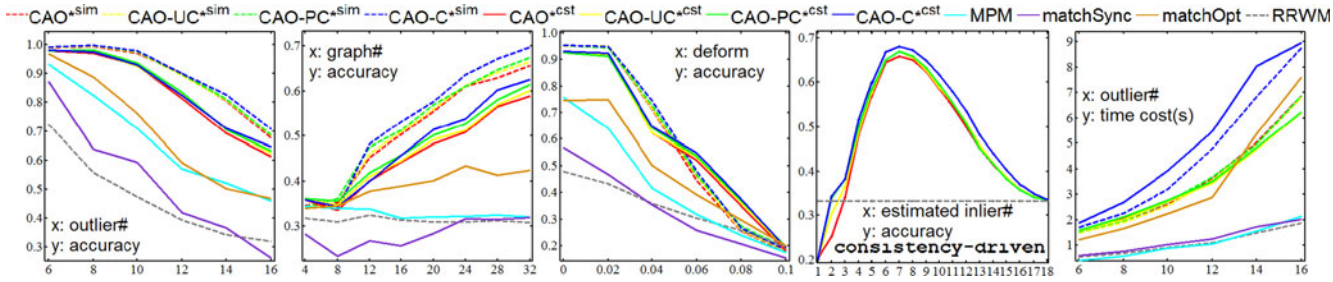
i) In Fig. 6o, we show an example of the hitting rate of the top n_i nodes in descending order by the node-wise consistency (solid curve) and affinity (dashed curve) metrics against true inliers. Given such reasonable hitting rates (> 0.8), the suite of our inlier eliciting methods, i.e., the consistency-driven variant (marked by the superscript 'cst') and the affinity-driven one (marked by 'sim') in general outperforms state-of-the-arts notably. Note in Fig. 6o, the hitting rate for 'face' by the affinity-driven metric is robust against the number of outliers. This is because the landmarks on face form a



(a) ObjectClass Car (b) ObjectClass Duck (c) ObjectClass Motorbike (d) ObjectClass Winebottle (e) Winebottle (time)

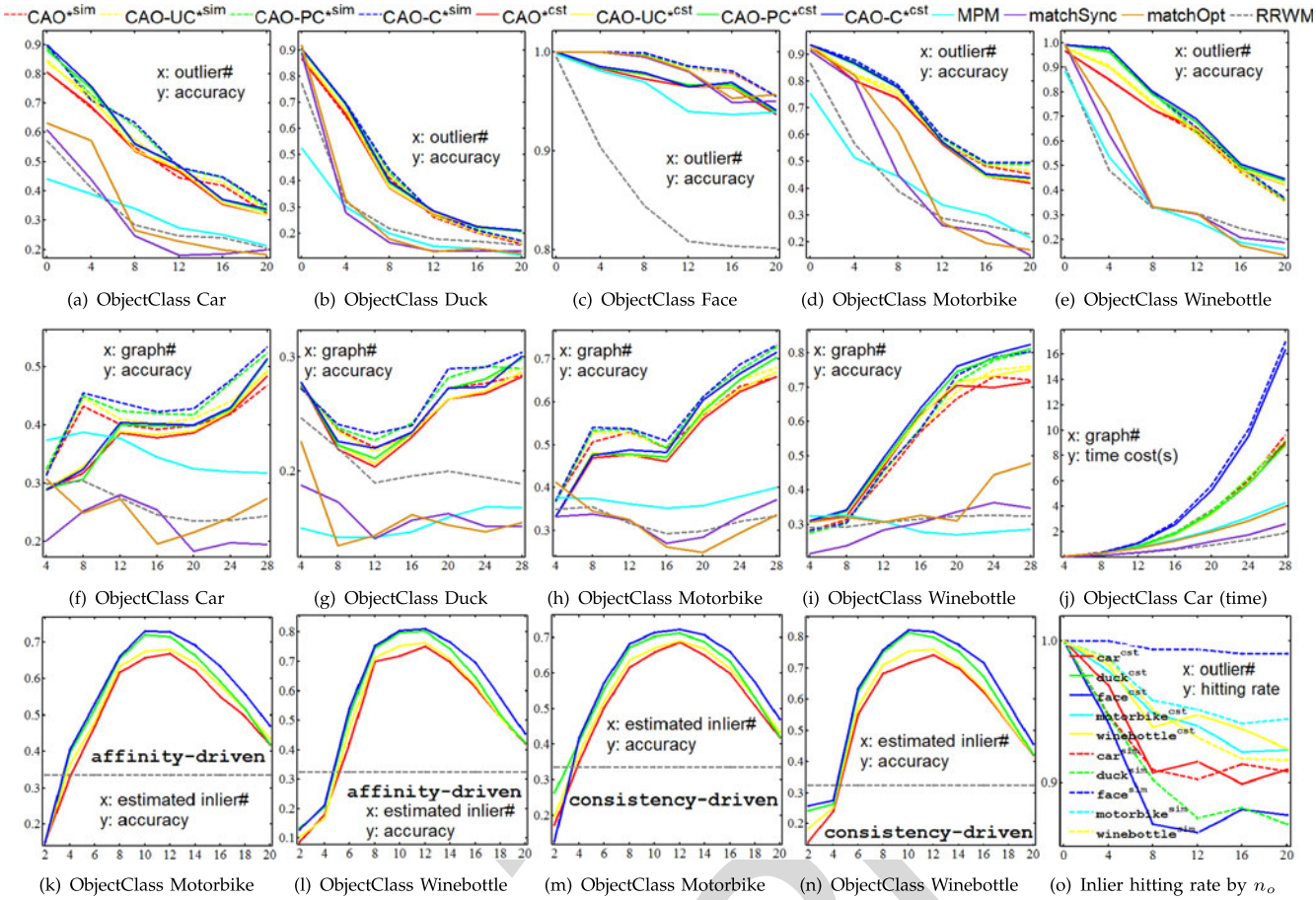
(f) CMU-Sequence Hotel (g) CMU-Sequence House (h) Pose-Sequence VolvoC70 (i) Pose-Sequence Houseblack (j) Houseblack (time)

Fig. 4. In case of few outliers: evaluation for CAO, CAO-C and the two variants CAO-UC and CAO-PC, and state of the art on real images (best viewed in color). Refer to Table 5 for settings.



(a) Synthetic outlier test (b) Outlier test by N (c) Synthetic deform test (d) Outlier test by n_{est} (e) Outlier test (time)

Fig. 5. In the presence of more outliers: random point test for our methods driven by consistency (solid) and affinity (dashed) inlier eliciting mechanism (best viewed in color). Refer to Table 6 for settings.



(a) ObjectClass Car (b) ObjectClass Duck (c) ObjectClass Face (d) ObjectClass Motorbike (e) ObjectClass Winebottle

(f) ObjectClass Car (g) ObjectClass Duck (h) ObjectClass Motorbike (i) ObjectClass Winebottle (j) ObjectClass Car (time)

(k) ObjectClass Motorbike (l) ObjectClass Winebottle (m) ObjectClass Motorbike (n) ObjectClass Winebottle (o) Inlier hitting rate by n_o

Fig. 6. In the presence of more outliers: real image test for our methods driven by consistency (solid) and affinity (dashed) inlier eliciting mechanism (best viewed in color). Refer to Table 7 for settings.

very distinctive structure thus its affinity is more informative.

ii) The sensitive test *w.r.t.* the disturbance of n_i is illustrated in Fig. 5d and the bottom row of Fig. 6, for synthetic data and real images respectively. The eliciting mechanism boosts the accuracy compared with our original algorithms which in fact set n_i equal to $n = 20$ with no discrimination to outliers. Moreover, the accuracy decline is relatively smooth around the exact $n_i = 10$ in these plots which suggests the robustness of our methods given a rough estimation of n_i.

TABLE 5
Parameter Settings for Real-Image Test in Fig. 4

parameter settings	dataset
$c = 1, \sigma^2 = .1, \beta = .9, n_i = 10, n_o = 2$	WillowObjectClass
$c = 1, \sigma^2 = .05, \beta = 0, n_i = 10, n_o = 4$	CMU-POSE

TABLE 6
Settings for Random Point Set Test in Fig. 5

parameter settings	results
$N = 20, \varepsilon = .02, n_i = 6, \rho = 1, \sigma^2 = .05, c = 1, T = 6$	Fig. 5a
$\varepsilon = .05, n_i = 6, n_o = 12, \rho = 1, \sigma^2 = .05, c = 1, T = 6$	Fig. 5b
$N = 20, n_i = 6, n_o = 12, \rho = .9, \sigma^2 = .05, c = 1, T = 6$	Fig. 5c
$N = 20, n_i = 6, n_o = 12, \varepsilon = .05, \rho = 1, \sigma^2 = .05, c = 1, T = 6$	Fig. 5d

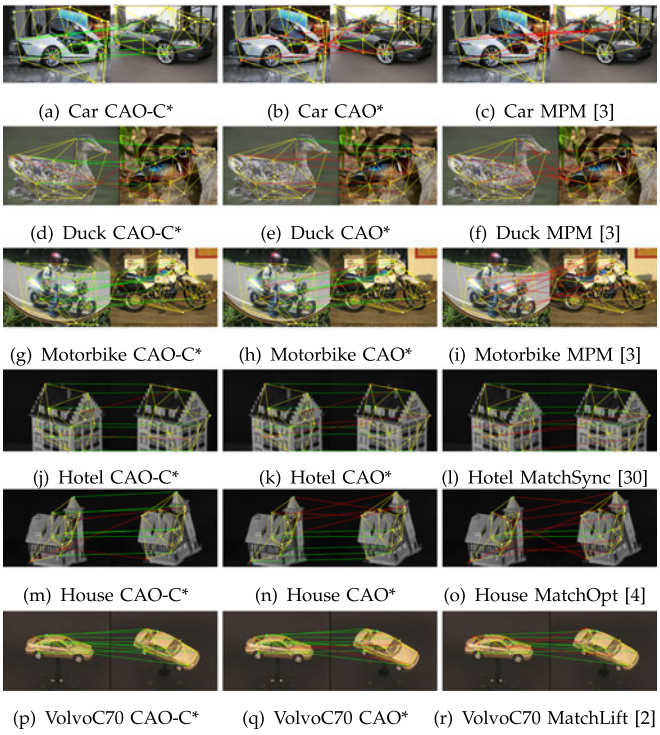


Fig. 7. Examples of visual results on public dataset. Top three rows correspond to the second row in Fig. 6, and the bottom three rows refer to the second row in Fig. 4. Inlier correspondences are drawn by green and red for correct and wrong matchings respectively. Outliers are colored in white (best viewed in color).

iii) Two run-time overhead examples are plot in Fig. 5e for the synthetic test and in Fig. 6j for real images. CAO-C* is more costive as we find it converges slower and need more overhead in each iteration as discussed in the complexity analysis. Also, all the proposed algorithms can be parallelized more easily than other iterative methods like matchOpt, matchSync, since updating $\mathbf{X}_{ij}^{(t-1)}$ can be done independently.

iv) MPM outperforms the baseline RRWM but does not perform as robustly as our methods. We think this is because MPM is suited under moderate deformation in addition with the outliers. This often does not hold in reality and neither in our settings—Tables 6 and 7.

v) In the synthetic point data test for Figs. 5a and 5b, the affinity-driven inlier eliciting variant outperforms the consistency-driven counterpart given the deformation $\varepsilon < .05$. However, as shown in Fig. 5c when ε grows by fixing the number of outliers, the consistency-driven mechanism outperforms when $\varepsilon > .05$. This is consistent with the motivation of this paper: consistency becomes helpful in the presence of a large amount of noises given a small number of outliers.

5 Conclusion

This paper proposes *effective*, *simple* and *general* multi-graph matching algorithms by incorporating affinity and consistency via a composition based optimization procedure. The outlier-tolerance variants are designed by eliciting the affinity and consistency associated with inliers. Experimental results suggest that i) Algorithm 2 (CAO-C) in general achieves higher accuracy compared with state-of-the-arts; ii) Algorithm 3 (CAO-UC, CAO-PC) improve the cost-effectiveness of Algorithm 1 (CAO) especially on the real images under arbitrary noises.

TABLE 7
Settings of Willow-ObjectClass Test in Fig. 6

parameter settings	results
$c = 1, \sigma^2 = .1, \beta = .9, n_{est} = n_i = 10, N = 28$	top
$c = 1, \sigma^2 = .1, \beta = .9, n_{est} = n_i = 10, n_o = 10$	middle
$c = 1, \sigma^2 = .1, \beta = .9, n_i = 10, n_o = 10, N = 28$	bottom

Acknowledgments

The work is supported by US National Science Foundation (NSF) DMS-1317424, NSFC (61527804, 61129001, 61221001), STCSM (15JC1401700, 14XD1402100, 13511504501) and the 111 Program (B07022). A preliminary version appeared in [1]. The current paper makes several extensions and improvements: i) a new graduated consistency-regularized affinity optimization algorithm (CAO-C in Algorithm 2) achieving more accurate matching results, meanwhile helps reinterpret the method (CAO-UC in Algorithm 3) in [1]; ii) an inlier eliciting mechanism against considerable outliers based on node-wise consistency and affinity; iii) more technical details of the algorithms and convergence discussion which are not fully described in [1]; iv) extensive evaluations that involve more various settings and additional datasets. In addition, more emerging state-of-the-arts [2], [3], [4] are compared especially after the year 2013. The source code will be made public available.

References

[1] J. Yan, Y. Li, W. Liu, H. Zha, X. Yang, and S. M. Chu, "Graduated consistency-regularized optimization for multi-graph matching," in *Proc. 13th Eur. Conf. Comput. Vis.*, 2014, pp. 407–422.

[2] Y. Chen, L. Guibas, and Q. Huang, "Near-optimal joint object matching via convex relaxation," in *Proc. 31st Int. Conf. Mach. Learn.*, 2014, pp. 100–108.

[3] M. Cho, J. Sun, O. Duchenne, and J. Ponce, "Finding matches in a haystack: A max-pooling strategy for graph matching in the presence of outliers," in *Proc. Conf. Comput. Vis. Pattern Recog.*, 2014, pp. 2091–2098.

[4] J. Yan, J. Wang, H. Zha, and X. Yang, "Consistency-driven alternating optimization for multi-graph matching: A unified approach," *IEEE Trans. Image Process.*, vol. 24, no. 3, pp. 994–1009, Mar. 2015.

[5] D. Conte, P. Foggia, C. Sansone, and M. Vento, "Thirty years of graph matching in pattern recognition," *Int. J. Pattern Recog. Artif. Intell.*, vol. 18, no. 3, pp. 265–298, 2004.

[6] P. Foggia, G. Percannella, and M. Vento, "Graph matching and learning in pattern recognition in the last 10 years," *Int. J. Pattern Recog. Artif. Intell.*, vol. 28, no. 1, pp. 1–40, 2014.

[7] M. Zaslavskiy, F. Bach, and J.-P. Vert, "Global alignment of protein-protein interaction networks by graph matching methods," *Bioinformatics*, vol. 25, no. 12, pp. i259–1267, 2009.

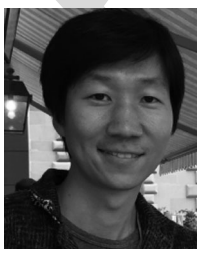
[8] M. L. Williams, R. C. Wilson, and E. Hancock, "Multiple graph matching with Bayesian inference," *Pattern Recog. Lett.*, vol. 18, no. 11–13, pp. 1275–1281, Nov. 1997.

[9] L. Liu, W. Lin, and Y. Zhong, "Traffic flow matching with clique and triplet cues," in *Proc. IEEE Int. Workshop Multimedia Signal Process.*, 2015, pp. 1–6.

[10] W. Wang, W. Lin, Y. Chen, J. Wu, J. Wang, and B. Sheng, "Finding coherent motions and semantic regions in crowd scenes: A diffusion and clustering approach," in *Proc. 13th Eur. Conf. Comput. Vis.*, 2014, pp. 756–77.

[11] Q. Huang and L. Guibas, "Consistent shape maps via semidefinite programming," in *Proc. Eurograph. Symp. Geom. Process.*, 2013, pp. 177–186.
[12] S. Liu, X. Wang, J. Chen, J. Zhu, and B. Guo, "Topicpanorama: A full picture of relevant topics," in *Proc. IEEE Conf. Vis. Anal. Sci. Technol.*, 2014, pp. 183–192.
[13] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, 1981.
[14] Z. Zhang, "Iterative point matching for registration of free-form curves and surfaces," in *Proc. Int. J. Comput. Vis.*, vol. 13, no. 2, pp. 11 9–152, 1994.
[15] F. Zhou and F. D. Torre, "Factorized graph matching," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2012, pp. 127–134.
[16] M. Cho, J. Lee, and K. M. Lee, "Reweighted random walks for graph matching," in *Proc. 11th Eur. Conf. Comput. Vis.*, 2010, pp. 492–505.
[17] M. Leordeanu, M. Hebert, and R. Sukthankar, "An integer projected fixed point method for graph matching and map inference," in *Proc. Adv. Neural Inf. Process. Syst. 22*, 2009, pp. 1114–1122.
[18] D. Eppstein, "Subgraph isomorphism in planar graphs and related problems," in *Proc. 6th Annu. ACM-SIAM Symp. Discrete Algorithms*, 1995, pp. 632–640.
[19] E. M. Luks, "Isomorphism of graphs of bounded valence can be tested in polynomial time," *J. Comput. Syst. Sci.*, vol. 25, pp. 42–65, 1982.
[20] A. V. Aho, M. Ganapathi, and S. W. Tjiang, "Code generation using tree matching and dynamic programming," *ACM Trans. Program. Lang. Syst.*, vol. 11, pp. 491–516, 1989.
[21] M. Leordeanu, R. Sukthankar, and M. Hebert, "Unsupervised learning for graph matching," *Int. J. Comput. Vis.*, vol. 96, no. 1, pp. 28–45, Jan. 2012.
[22] S. Gold and A. Rangarajan, "A graduated assignment algorithm for graph matching," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 18, no. 4, pp. 377–388, Apr. 1996.
[23] T. Caetano, J. McAuley, L. Cheng, Q. Le, and A. J. Smola, "Learning graph matching," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 8, pp. 1048–1058, Jun. 2009.
[24] A. Egozi, Y. Keller, and H. Guterman, "A probabilistic approach to spectral graph matching," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 18–27, Jan. 2013.
[25] E. Lawler, "The quadratic assignment problem," *Manage. Sci.*, vol. 9, pp. 586–599, 1963.
[26] T. C. Koopmans and M. Beckmann, "Assignment problems and the location of economic activities," *Econometrica*, vol. 25, pp. 53–76, 1957.
[27] T. Funkhouser, M. Kazhdan, P. Shilane, P. Min, W. Kiefer, A. Tal, and D. Dobkin, "Modeling by example," *ACM Trans. Graph.*, vol. 23, pp. 652–663, 2004.
[28] A. Solé-Ribalta and F. Serratosa, "Models and algorithms for computing the common labelling of a set of attributed graphs," *Comput. Vis. Image Understanding*, vol. 115, pp. 929–945, 2011.
[29] 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 Trans. Graph.*, vol. 31, pp. 1–11, 2012.
[30] D. Pachauri, R. Kondor, and S. Vikas, "Solving the multi-way matching problem by permutation synchronization," in *Proc. Adv. Neural Inf. Process. Syst.*, 2013, pp. 1860–1868.
[31] J. Yan, Y. Tian, H. Zha, X. Yang, Y. Zhang, and S. Chu, "Joint optimization for consistent multiple graph matching," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2013, 1649–1656.
[32] A. Sole-Ribalta and F. Serratosa, "Graduated assignment algorithm for multiple graph matching based on a common labeling," in *IJPRAI*, vol. 27, no. 1 pp. 1–27, 2013.
[33] M. Leordeanu, A. Zanfir, and C. Sminchisescu, "Semi-supervised learning and optimization for hypergraph matching," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2011, pp. 2274–2281.
[34] M. Cho, K. Alahari, and J. Ponce, "Learning graphs to match," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2013, pp. 25–32.
[35] N. Hu, R. M. Rustamov, and L. Guibas, "Graph matching with anchor nodes: A learning approach," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2013, pp. 2906–2913.
[36] M. Chertok and Y. Keller, "Efficient high order matching," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 12, pp. 2205–2215, Dec. 2010.
[37] O. Duchenne, F. Bach, I. Kweon, and J. Ponce, "A tensor-based algorithm for high-order graph matching," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2009, pp. 1980–1987.
[38] J. Lee, M. Cho, and K. M. Lee, "Hyper-graph matching via reweighted random walks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2011, pp. 1633–1640.
[39] R. Zass and A. Shashua, "Probabilistic graph and hypergraph matching," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2008, pp. 1–8.
[40] J. Yan, C. Zhang, H. Zha, W. Liu, X. Yang, and S. M. Chu, "Discrete hyper-graph matching," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2015, pp. 1520–1528.
[41] T. Collins, P. Mesejo, and A. Bartoli, "An analysis of errors in graph-based keypoint matching and proposed solutions," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 138–153.
[42] S. Gold and A. Rangarajan, "Softmax to softassign: Neural network algorithms for combinatorial optimization," *J. Artif. Neural Netw.*, vol. 2, pp. 381–399, 1996.
[43] M. Leordeanu and M. Hebert, "A spectral technique for correspondence problems using pairwise constraints," in *Proc. IEEE 10th Int. Conf. Comput. Vis.*, 2005, pp. 1482–1489.
[44] P. S. T. Cour and J. Shi, "Balanced graph matching," in *Proc. Adv. Neural Inf. Process. Syst. 19*, 2006, pp. 313–320.
[45] A. Solé-Ribalta and F. Serratosa, "On the computation of the common labelling of a set of attributed graphs," in *Proc. 14th Iberoamerican Conf. Pattern Recog.*, Guadalajara, Jalisco, Mexico, Nov. 2009, pp. 137–144.
[46] F. Gavril, "Generating the maximum spanning trees of a weighted graph," *J. Algorithms*, vol. 8, pp. 592–597, 1987.
[47] A. S. Ribalta and F. Serratosa, "Graduated assignment algorithm for finding the common labelling of a set of graphs," in *Proc. SSPR*, 2010, pp. 180–190.
[48] G. Charpiat, "Learning shape metrics based on deformations and transport," in *Proc. IEEE 12th Int. Conf. Comput. Vis. Workshops*, 2009, pp. 328–335.
[49] S. Umeyama, "An eigendecomposition approach to weighted graph matching problems," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 10, no. 5, pp. 695–703, Sep. 1988.
[50] L. Torresani, V. Kolmogorov, and C. Rother, "Feature correspondence via graph matching: Models and global optimization," in *Proc. 10th Eur. Conf. Comput. Vis.*, 2008, pp. 596–609.
[51] F. Viksten, P. Forssén, B. Johansson, and A. Moe, "Comparison of local image descriptors for full 6 degree-of-freedom pose estimation," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2009, pp. 2779–2786.
[52] K. Jia, T. H. Chan, Z. Zeng, G. Wang, T. Zhang, and Y. Ma, "Roml: A robust feature correspondence approach for matching objects in a set of images," *arXiv:1403.7877*, 2014.

Junchi Yan received the MS degree from Shanghai Jiao Tong University, China in 2011 and he is currently working toward the PhD degree at the Department of Electronic Engineering of Shanghai Jiao Tong University, China. He is also a research staff member and master inventor with IBM Research—China. He received the IBM Research Accomplishment and Outstanding Accomplishment Award in 2013, 2014. His research interests are computer vision and machine learning applications.



Minsu Cho received the bachelor and PhD degrees in electrical engineering and computer science from Seoul National University. He is a researcher in the WILLOW team at Inria Paris-Rocquencourt and Ecole Normale Superieure. He has joined Inria as a postdoctoral fellow in 2012 and recently became an Inria starting researcher. His research are computer vision and machine learning, especially in the problems of object discovery, correspondence, and graph matching.

Hongyuan Zha received the PhD degree in scientific computing from Stanford University in 1993. He is a professor at East China Normal University and the School of Computational Science and Engineering, College of Computing, Georgia Institute of Technology. Since then he has been working on information retrieval, machine learning applications, and numerical methods. He receievd the Leslie Fox Prize (1991, second prize) of the Institute of Mathematics and its Applications, the Outstanding Paper Awards of the 26th International Conference on Advances in Neural Information Processing Systems (NIPS 2013), and the Best Student Paper Award (advisor) of the 34th ACM SIGIR International Conference on Information Retrieval (SIGIR 2011). He serves as an associate editor of *IEEE Transactions on Knowledge and Data Engineering*.

Xiaokang Yang (M'00-SM'04) received the BS degree from Xiamen University, Xiamen, China, in 1994, the MS degree from the Chinese Academy of Sciences, Shanghai, China, in 1997, and the PhD degree from Shanghai Jiao Tong University, Shanghai, China, in 2000. He is currently a distinguished professor of the School of Electronic Information and Electrical Engineering, and the deputy director of the Institute of Image Communication and Information Processing, Shanghai Jiao Tong University, Shanghai, China. His research interests include visual signal processing and communication, media analysis and retrieval, and pattern recognition. He is a senior member of the IEEE.

Stephen M. Chu studied physics at Peking University in Beijing, China, and received the MS and PhD degrees in electrical and computer engineering from the University of Illinois at Urbana-Champaign. He is a senior technical staff member at IBM. His team received the top prize in the DARPA-organized speech recognition contest for five consecutive times. His pioneering work in Audio-Visual Speech recognition was featured in the PBS program Scientific American Frontiers with Alan Alda. His research interests cover speech recognition, computer vision, and signal processing.