Adaptive Graph Matching

Xu Yang and Zhi-Yong Liu, *Senior Member, IEEE*

***Abstract*—Establishing correspondence between point sets lays the foundation for many computer vision and pattern recognition tasks. It can be well defined and solved by graph matching. However, outliers may significantly deteriorate its performance, especially when outliers exist in both point sets and meanwhile the inlier number is unknown. In this paper, we propose an adaptive graph matching algorithm to tackle this problem. Specifically, a novel formulation is proposed to make the graph matching model adaptively determine the number of inliers and match them, then by relaxing the discrete domain to its convex hull the discrete optimization problem is relaxed to be a continuous one, and finally a graduated projection scheme is used to get a discrete matching solution. Consequently, the proposed algorithm could realize inlier number estimation, inlier selection, and inlier matching in one optimization framework. Experiments on both synthetic data and real world images witness the effectiveness of the proposed algorithm.**

***Index Terms*—Graduated projection, graph matching, point correspondence, regularization method.**

I. Introduction and Preliminaries

ESTABLISHING correspondence between interest points across images has been a long-standing problem in computer vision and pattern recognition. It also plays a key role in many applications including 2-D and 3-D image registration, object recognition, and video analysis [1]–[3].

Point correspondence could be well defined and effectively solved by graph matching. Specifically, first a point set $G = \{g_i\}_{i=1}^{M}$ is represented by a graph $\mathcal{G}$. A point g_i is represented by a vertex i, where an appearance descriptor l_i around g_i, like the SIFT descriptor, is assigned to i as a label. And the link between g_i and g_j is represented by an edge ij, where a real number (vector) w_{ij} describing for example the length or direction of the link, is assigned to ij as an edge weight. In this paper, by the term *graph* we mean such a labeled weighted graph. Then the correspondence problem between two point sets G and $H = \{h_a\}_{a=1}^{N}$ is transformed to vertex matching between two graphs $\mathcal{G}$ and $\mathcal{H}$. Without loss of generality, it is assumed that $M \leq N$.

Different from earlier graph matching algorithms which did not involve a well-defined objective function [4], modern ones usually formulate graph matching as a quadratic binary programming (QBP) problem, which aim at minimizing the inconsistency between two matched graphs, or equivalently maximizing the similarity between them. By using the maximization problem, it typically takes the following form:

$$\begin{aligned} \max \ & F(\mathcal{G}, \mathcal{H}, \mathbf{X}) \\ \text{s.t.} \ & C_1(\mathbf{X}) \preceq \mathbf{0} \\ & C_2(\mathbf{X}) = \mathbf{0} \\ & \mathbf{X} \in \{0,1\}^{M\times N} \end{aligned} \tag{1}$$

where $\mathbf{0}$ denotes an all-zero vector. The assignments between vertices are mathematically represented by an assignment matrix $\mathbf{X} = \{0,1\}^{M\times N}$, where $\mathbf{X}_{ia} = 1$ means assigning the vertex i in $\mathcal{G}$ to the vertex a in $\mathcal{H}$. The objective function $F(\mathcal{G}, \mathcal{H}, \mathbf{X})$ measures the similarity between graphs $\mathcal{G}$ and $\mathcal{H}$ matched by $\mathbf{X}$. The constraint functions $C_1(\mathbf{X})$ and $C_2(\mathbf{X})$ are usually affine functions [5], depending on particular matching assumptions.

In recent years, a most popular formulation of (1) is as follows [4]–[9]:

$$\begin{aligned} \max \ & \mathbf{x}^T\mathbf{A}\mathbf{x} \\ \text{s.t.} \ & \mathbf{W}\mathbf{x} \preceq \mathbf{1}_{M+N} \\ & \mathbf{x} \in \{0,1\}^{MN} \end{aligned} \tag{2}$$

where the linear transformation matrix of the constraint function is

$$\mathbf{W} = \begin{bmatrix} \mathbf{1}_M^T \otimes \mathbf{I}_N \\ \mathbf{I}_M \otimes \mathbf{1}_N^T \end{bmatrix} \tag{3}$$

by denoting the identity matrix as $\mathbf{I}$ and denoting the all-one vector as $\mathbf{1}$. The notation $\otimes$ denotes the Kronecker product between matrices. The translation term $\mathbf{1}_{M+N}$ is usually put on the right side of $\preceq$. The non-negative matrix $\mathbf{A} \in \mathbb{R}^{MN\times MN}$, known as the affinity matrix [6], can be defined by[1]

$$\begin{aligned} \mathbf{A}_{ia,jb} &= \mathbf{A}_{(i-1)N+a,(j-1)N+b} \\ &= \begin{cases} \mathcal{A}(l_i, l_a), & \text{if } i = j, a = b, \\ \mathcal{A}(w_{ij}, w_{ab}), & \text{if edges } ij \text{ and } ab \text{ exist} \\ 0, & \text{otherwise.} \end{cases} \end{aligned} \tag{4}$$

[1] In (4), the comma in $\mathbf{A}_{ia,jb}$ is used to separate the row number and column number of a matrix, while in other equations it is omitted, e.g., (5).

Manuscript received June 6, 2016; revised April 6, 2017 and April 20, 2017; accepted April 22, 2017. This work was supported in part by the National Natural Science Foundation of China under Grant 61503383, Grant 61633009, Grant U1613213, Grant 61375005, and Grant 61303174, and in part by the National Key Research and Development Plan of China under Grant 2016YFC0300801. This paper was recommended by Associate Editor J. Basak. *(Corresponding author: Zhi-Yong Liu.)*

X. Yang is with the State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China (e-mail: xu.yang@ia.ac.cn).

Z.-Y. Liu is with the State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China, also with the Center for Excellence in Brain Science and Intelligence Technology, Chinese Academy of Sciences, Shanghai 200031, China, and also with the University of Chinese Academy of Sciences, Beijing 100190, China (e-mail: zhiyong.liu@ia.ac.cn).

This paper has supplementary downloadable multimedia material available at http://ieeexplore.ieee.org provided by the authors.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCYB.2017.2697968

The diagonal entry $\mathcal{A}(l_i, l_a)$ measures the label similarity between vertices i in $\mathcal{G}$ and a in $\mathcal{H}$, and the off-diagonal entry $\mathcal{A}(w_{ij}, w_{ab})$ describes the weight consistency between edges ij in $\mathcal{G}$ and ab in $\mathcal{H}$. In accordance to the building way of $\mathbf{A}$ in (4), the assignment matrix $\mathbf{X}$ is row-wisely vectorized to obtain an assignment vector[2] $\mathbf{x}$. In (2), the one-to-one matching constraint $\mathbf{W}\mathbf{x} \preceq \mathbf{1}_{M+N}$ is a common matching assumption, which means a vertex in $\mathcal{G}$ can be assigned to at most one vertex in $\mathcal{H}$, and vice versa. It is more often shown in the matrix form by

$$\sum_{i}^{M} \mathbf{X}_{ia} \leq 1, \sum_{a}^{N} \mathbf{X}_{ia} \leq 1. \tag{5}$$

Note some papers minimize an objective function based on the dissimilarity matrix $\mathbf{K}$ of the same size with $\mathbf{A}$, which is equivalent to a maximization problem by some straightforward transformations. Including the minimization problem, the formulation (2) is widely adopted by many graph matching algorithms, especially by the spectral decomposition-based algorithms [4]–[7].

Though recent papers directly take the form (2), we claim that they only equivalently solve the following solve the following **Problem 0**:

$$\begin{aligned} \max \ & \mathbf{x}^T\mathbf{A}\mathbf{x} \\ \text{s.t.} \ & \mathbf{W}\mathbf{x} \preceq \mathbf{1}_{M+N} \\ & \mathbf{x}^T\mathbf{1}_{MN} = \min(M, N) = M, \ \mathbf{x} \in \{0, 1\}^{MN}. \end{aligned} \tag{6}$$

With the additional equation $\mathbf{x}^T\mathbf{1}_{MN} = M$, the constraints in Problem 0 imply that all the vertices in $\mathcal{G}$ should be matched. For clear understanding they can be rewritten by

$$\left(\mathbf{1}_M^T \otimes \mathbf{I}_N\right)\mathbf{x} \preceq \mathbf{1}_N, \ \left(\mathbf{I}_M \otimes \mathbf{1}_N^T\right)\mathbf{x} = \mathbf{1}_M, \ \mathbf{x} \in \{0, 1\}^{MN} \tag{7}$$

or in matrix form by

$$\sum_{i}^{M} \mathbf{X}_{ia} \leq 1, \sum_{a}^{N} \mathbf{X}_{ia} = 1, \mathbf{X} \in \{0, 1\}^{M\times N}, \ \forall i, a. \tag{8}$$

The reason for this claim lies in the fact that given the nonnegative $\mathbf{A}$, any additional assignment will increase (or at least maintain) the objective value, so the maximum objective value of (2) can always be obtained with the most M assignments. It means that the optimal solution of (2) can always be found in the domain of Problem 0. Thus though the constraint $\mathbf{x}^T\mathbf{1}_{MN} = M$ is not explicitly used in (2), it is implicitly added.

It can be observed in Problem 0 that outliers only exist in the larger graph $\mathcal{H}$, and it means that the recent algorithms using formulation (2) actually solve the *part-in-whole* subgraph matching problem [10] which recognizes $\mathcal{G}$ as a part of $\mathcal{H}$. However, in many practical tasks, especially computer vision tasks, there may be outliers in both graphs [11] due to geometric transformations, occlusions, etc. For this situation, it is more reasonable to define the problem as matching the most similar subgraphs in $\mathcal{G}$ and $\mathcal{H}$. Some algorithms [4], [6], [7], [12] may be generalized to this problem with the help of softened (continuous) assignment vector $\mathbf{x}$. Specifically, these algorithms first match all the vertices and then select assignments by removing low-confidence assignments. The algorithms are named by the two-step algorithms for further references. Such a two-step strategy, however, is not completely consistent with the problem to match the most similar subgraphs [13]. In view of this, we previously proposed a method to match the most similar subgraphs with a fixed number of best assignments in two graphs, as will be introduced in Problem 1. And beyond it in this paper we propose a method named by adaptive graph matching which could further adaptively determine the number of inliers and match them. In conclusion, the widely used formulation (2) can be considered as a special case of Problem 1 based on the above claim. And interestingly, though adopting a similar formulation, the formulation (2) is actually a special case of the adaptive graph matching problem introduced by Problem 2.

Specifically, by some transformations, the problem to find a fixed number of best assignments in [13] and [14] can be formulated as a maximization problem by by **Problem 1**:[3]

$$\begin{aligned} \max \ & \mathbf{x}^T\mathbf{A}\mathbf{x} \\ \text{s.t.} \ & \mathbf{W}\mathbf{x} \preceq \mathbf{1}_{M+N} \\ & \mathbf{x}^T\mathbf{1}_{MN} = L, \ \mathbf{x} \in \{0, 1\}^{MN} \end{aligned} \tag{9}$$

where the prespecified constant L determines the size of the subgraphs in $\mathcal{G}$ and $\mathcal{H}$. The domain of this problem is denoted by $\mathcal{D}^L$, which in matrix form is

$$\begin{aligned} &\sum_{i}^{M} \mathbf{X}_{ia} \leq 1, \sum_{a}^{N} \mathbf{X}_{ia} \leq 1 \\ &\sum_{i}^{M}\sum_{a}^{N} \mathbf{X}_{ia} = L, \mathbf{X} \in \{0, 1\}^{M\times N}, \ \forall i, a. \end{aligned} \tag{10}$$

When $L = M \leq N$, Problem 1 degenerates to Problem 0.

Since Problem 1 is a QBP problem, approximate algorithms which make certain relaxations to the original problem are necessary for efficiency reasons. The continuous methods [5]–[8], [15]–[18] form a quite important group of the approximate methods in the graph matching community, which can usually find a local optimum in reasonable time. The proposed method also belongs to the continuous methods which typically involve first relaxing the discrete domain to be a continuous one, and then projecting the continuous solution to be a discrete one.

For the continuous domain relaxation, there are a number of ways such as the relaxation by fixing the norm $\|\mathbf{x}\| = 1$ [6]. A more reasonable relaxation may be relaxing the discrete domain to its convex hull, because some matching constraints could be included in the optimization process [4], [5]. For instance, the equal-sized graph matching algorithms [16], [19] relax the set of permutation matrices to its convex hull, the

[2] In some papers $\mathbf{x}$ is a column-wise replica of $\mathbf{X}$, and correspondingly $\mathbf{A}$ should be

$$\mathbf{A}_{i+(a-1)M,j+(b-1)M} = \begin{cases} \mathcal{A}(l_i, l_a), & \text{if } i = j, a = b \\ \mathcal{A}(w_{ij}, w_{ab}), & \text{if edges } ij \text{ and } ab \text{ exist} \\ 0, & \text{otherwise.} \end{cases}$$

[3] The relations between Problems 0–4 are illustrated by a schematic in the supplementary material.

well-known doubly stochastic matrices. For another instance, the algorithms [4], [12], [17] relax the discrete domain of Problem 0, i.e., (8) in the matrix form, to its convex hull

$$\sum_{i}^{M} \mathbf{X}_{ia} \leq 1, \sum_{a}^{N} \mathbf{X}_{ia} = 1, \mathbf{X}_{ia} \in [0, 1], \ \forall i, a. \tag{11}$$

For Problem 1, the convex hull $\mathcal{C}^L$ of the discrete domain $\mathcal{D}^L$ [13], [14] is

$$\mathbf{W}\mathbf{x} \preceq \mathbf{1}_{M+N}, \mathbf{x}^T\mathbf{1}_{MN} = L, \mathbf{x}_i \in [0, 1], \ \forall i \tag{12}$$

which in matrix form is

$$\begin{aligned} &\sum_{i}^{M} \mathbf{X}_{ia} \leq 1, \sum_{a}^{N} \mathbf{X}_{ia} \leq 1 \\ &\sum_{i}^{M}\sum_{a}^{N} \mathbf{X}_{ia} = L, \mathbf{X}_{ia} \in [0, 1], \forall i, a. \end{aligned} \tag{13}$$

This paper also involves the relaxation of the discrete domain in Problem 2 to its convex hull. Note in these relaxations, each extreme point set of the above convex hulls is exactly the corresponding discrete domain, which property is useful in the graduated projection introduced below.

For the discrete domain projection, using the direct projection, e.g., projecting the continuous solution to the nearest discrete point, may introduce significant additional error [1], [16], [19]. Some algorithms [4], [7], [12] introduce updating schemes of the continuous solution before the final direct projection. Further, originating from [8] and [16], graduated discrete domain projection, which usually automatically results in a discrete solution without a final projection, has been observed to greatly improve the matching accuracy. Particularly, the algorithm [16] gradually adjusts the linear combination parameter between a convex relaxation and a concave relaxation of the original objective function, which guarantees a discrete solution can be obtained following a path in the continuous domain. It takes advantage of the property that the minimum point of a concave function over a convex set (convex hull) locates exactly in its extreme set (discrete domain). Following this idea, the graduated projection is applied to the relaxed Problem 0 by algorithms [17] and [18]. Further it is applied to the relaxed Problem 1 in our previous works [13], [14]. However, one main problem of these algorithms is that the number of assignments, i.e., the number of inliers representing object, should be prespecified, which significantly deteriorates the matching performance when the inlier number cannot be figured out.

The main contribution of this paper is to extend the algorithms [13], [14] to tackle the matching problem with unknown inlier number, which consists of the following three extensions. First, by regularizing on the assignment number, a novel formulation (Problem 2) with respect to the redefined assignment vector (matrix) is introduced to make the graph matching model adaptively determine the number of inliers and match them; second, the discrete optimization problem (Problem 2) is relaxed to be a continuous one (Problem 4), where the convex hull of the discrete domain is proved; and third, the graduated projection is generalized to Problem 4, and a quick solution for a linear programming problem, which is closely related to the complexity of optimization algorithm, is provided. Consequently, the proposed algorithm named by adaptive graph matching could realize inlier number estimation, inlier selection, and inlier matching in one optimization framework.

The remaining paper is organized as follows. For the sake of continuity, the adaptive graph matching algorithm is first proposed in Section II, and then followed by some discussions on related works from both model and optimization aspects in Section III. After giving the experimental results in Section IV, finally Section V concludes this paper.

II. Adaptive Graph Matching Model

In this section, the adaptive graph matching problem is first formulated as a regularized QBP problem, and then approximated by a continuous problem by relaxing the discrete domain to its convex hull. The graduated projection-based optimization method is applied to the continuous problem, which automatically terminates at a discrete solution. Finally, some implementation details are discussed.

A. Formulation

To make the inlier number adaptively determined according to the problem, the proposed method imposes a regularization on the assignment number L, with the following with the following **Problem 2**:

$$\begin{aligned} \max \ & \mathbf{x}^T\mathbf{A}\mathbf{x} - \rho\mathbf{x}^T\mathbf{1}_{MN} \\ \text{s.t.} \ & \mathbf{W}\mathbf{x} \preceq \mathbf{1}_{M+N} \\ & \mathbf{x} \in \{0, 1\}^{MN}. \end{aligned} \tag{14}$$

The affinity matrix $\mathbf{A}$ and the linear transformation matrix $\mathbf{W}$ are the same with (2). Different from Problems 1 and 2 drops the constraint $\mathbf{x}^T\mathbf{1}_{MN} = L$ and instead incorporates $-\mathbf{x}^T\mathbf{1}_{MN}$ into the objective as a regularization term controlling the assignment number, with ρ as a positive regularization parameter. The domain of Problem 2 is denoted by $\mathcal{D}$, which is a union of the domain $\mathcal{D}^L$ of Problem 1 by

$$\mathcal{D} = \bigcup_{L=0}^{M} \mathcal{D}^L. \tag{15}$$

An (idealized or say simulative) illustration of $\mathcal{D}$ and its relation with $\mathcal{D}^L$ are given in Fig. 1. On the lowest level, $\mathcal{D}^{L=0}$ is represented by an origin point O because the only vector $\mathbf{x} \in \mathcal{D}^0$ is an all-zero vector. On the highest level, $\mathcal{D}^{L=M}$ is visually represented by the vertex set of a regular hexagon, and its convex hull $\mathcal{C}^{L=M}$ is accordingly represented by the regular hexagon. Analogically, $\mathcal{D}^L, \forall 0 < L < M$ is, respectively, represented by the vertex set of a regular polygon, and its convex hull $\mathcal{C}^L$ is represented by that regular polygon. Note in Fig. 1 $\mathcal{D}^{M-1}$ is deliberately represented by an octagon, with more vertices than the hexagon for $\mathcal{D}^M$, which is because that given M and N the number of vertices $\#_L = (M!N!)/(L!(M-L)!(N-L)!)$ does not monotonically increase with respect to L as shown in Fig. 2.

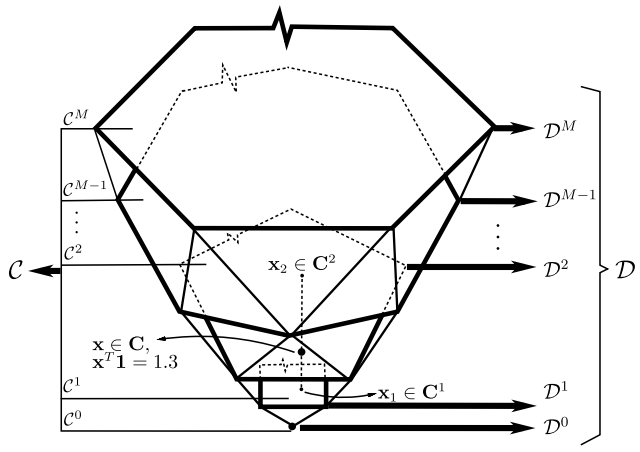


Fig. 1. Illustration of relations between $\mathcal{D}$, $\mathcal{C}$, $\mathcal{D}^L$, and $\mathcal{C}^L$. The point $\mathbf{x}$ is a convex combination of points $\mathbf{x}_1 \in \mathcal{C}^1$ and $\mathbf{x}_2 \in \mathcal{C}^2$. Though $\bar{L} = \mathbf{x}^T\mathbf{1} = 1.3$ is not an integer, implying that $\mathbf{x} \notin \mathcal{C}^L, \forall L$, there is $\mathbf{x} \in \mathcal{C}$.

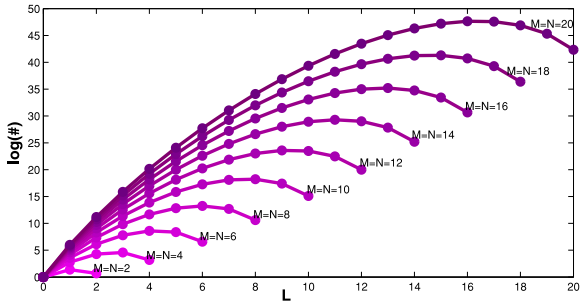


Fig. 2. Point number of $\mathcal{D}^L$ with respect to L, where log(#) denotes the natural logarithm of point number.

Intuitively, $\mathcal{D}$ is the combination of vertices of all the polygons (including O).

When $\rho = 0$, Problem 2 takes the same formulation with (2), and as claimed in Section I, its optimal solution could always be found in $\mathcal{D}^M$ because any additional assignment would increase (or at least maintain) the objective value given the non-negative $\mathbf{A}$. When $\rho > 0$, the regularization term takes effect, and as ρ increases, it demands that less vertices should be matched. The objective function (14) then becomes a tradeoff between the matching affinity and the assignment number. For an intuitive description, the influence of ρ can be shown through Fig. 1 that as ρ increases, the solution $\mathbf{x} \in \mathcal{D}$ is *pushed toward* the origin point O. And thus the physical meaning is that less vertex assignments are preferred while maintaining a reasonable matching affinity. It makes sense because removing outlier assignments will cause little affinity measure decrease while bring considerable increase in the regularization term $-\mathbf{x}^T\mathbf{1}_{MN}$. So if properly regularized, the outlier assignments could be automatically removed by Problem 2.

Note though in this paper we focus on the maximization formulation Problem 2, it can be equivalently reformulated as the minimization of the dissimilarity between $\mathcal{G}$ and $\mathcal{H}$. Since the minimization formulation is also widely used in graph matching, including our previous works [13], [14], the minimization form of Problem 2 is briefly discussed below, given by **Problem 3**:

$$\begin{aligned} \min \ & \mathbf{x}^T\mathbf{K}\mathbf{x} - \rho\mathbf{x}^T\mathbf{1}_{MN} \\ \text{s.t.} \ & \mathbf{x} \in \mathcal{D}. \end{aligned} \tag{16}$$

The non-negative dissimilarity matrix [12], [14] $\mathbf{K} \in \mathbb{R}^{MN\times MN}$ is of the same size with $\mathbf{A}$, which can be obtained by the operation $\mathbf{K} = \mathcal{F}(-\mathbf{A})$ with an entry-wise non-negative monotone increasing function[4] $\mathcal{F}(\cdot)$. When $\rho = 0$, the lowest matching cost is obtained when there are no assignments, i.e., $\mathbf{x} = \mathbf{0}_{MN}$. Incorporating the regularization term when $\rho > 0$, implies that more vertex assignments are preferred while preserving a low matching cost.

Since Problem 2 is a typical QBP problem, to make it computationally tractable, next a continuous method is introduced to approximately solve it. It first relaxes the discrete domain to the convex hull to get a continuous optimization problem (Problem 4), and then utilizes a graduated projection scheme to push the continuous solution gradually back to the discrete domain.

B. Continuous Relaxation to Convex Hull

As introduced in Section I, relaxing the discrete domain to its convex hull is an important way for the continuous relaxation, and the property that the discrete domain is exactly the extreme point set of its convex hull is essential to some graduated projection algorithms [8], [16]–[19]. To get the convex hull of the discrete domain $\mathcal{D}$, two lemmas are first introduced. The first Lemma 1 is also known as Minkowski theorem [20].

Lemma 1: Any compact convex set is the convex hull of its extreme points.

And the second Lemma 2 [13] is as follows.

Lemma 2: The convex hull of $\mathcal{D}^L$ is $\mathcal{C}^L$, and $\mathcal{D}^L$ is the extreme point set of $\mathcal{C}^L$, where $\mathcal{D}^L$ and $\mathcal{C}^L$ in matrix form are, respectively, defined by (10) and (13).

Based on these two lemmas, the convex hull of $\mathcal{D}$ can be obtained by the following theorem.

Theorem 1: The convex hull of $\mathcal{D}$ is $\mathcal{C}$, and $\mathcal{D}$ is the extreme point set of $\mathcal{C}$, where $\mathcal{D}$ is the domain of Problem 2 and $\mathcal{C}$ is defined as follows:

$$\mathbf{W}\mathbf{x} \preceq \mathbf{1}_{M+N} \ \mathbf{x} \succeq 0. \tag{17}$$

Proof Sketch: For representation convenience, we equivalently utilize the matrix forms of $\mathcal{D}$ and $\mathcal{C}$ by

$$\mathcal{D}_{\mathcal{M}} \coloneqq \left\{ \mathbf{X} \middle| \sum_{i=1}^{M} \mathbf{X}_{ia} \le 1, \ \sum_{a=1}^{N} \mathbf{X}_{ab} \le 1, \mathbf{X}_{ia} \in [0,1] \right\} \tag{18}$$

$$\mathcal{C}_{\mathcal{M}} \coloneqq \left\{ \mathbf{X} \middle| \sum_{i=1}^{M} \mathbf{X}_{ia} \le 1, \ \sum_{a=1}^{N} \mathbf{X}_{ab} \le 1, \mathbf{X} \in \{0,1\}^{M\times N} \right\}. \tag{19}$$

The basic idea is to prove that $\mathcal{D}_{\mathcal{M}}$ is exactly the same with the extreme point set of the convex set $\mathcal{C}_{\mathcal{M}}$, and from Lemma 1 it can be derived that $\mathcal{C}_{\mathcal{M}}$ is the convex hull of $\mathcal{D}_{\mathcal{M}}$. The proof includes the following six steps, i.e., 1–6, which is organized in a structural way [21]. Particularly, when the substep number

[4]Some explanations on the entry-wise non-negative monotone increasing function are given in the supplementary material.

is too long, it is simplified by putting the number of its first several digits in the bracket. For instance, the substep number 4.2.5.5.1 is abbreviated by [4].1., which could be understood through the context.

Proof:

1. Prove that $\mathcal{C}_\mathcal{M}$ is a convex set.
   Assume: $\mathbf{X}', \mathbf{X}'' \in \mathcal{C}_\mathcal{M}$, $\theta \in [0, 1]$.
   Prove: $\mathbf{X} \in \mathcal{C}_\mathcal{M}$, where $\mathbf{X} = \theta\mathbf{X}' + (1-\theta)\mathbf{X}''$.
   1.1. Prove $\sum_{i=1}^{M} \mathbf{X}_{ia} \leq 1$.
   1.1.1. $\sum_{i=1}^{M} \mathbf{X}'_{ia} \leq 1$, $\sum_{i=1}^{M} \mathbf{X}'_{ia} \leq 1$, and $\theta \in [0, 1]$ (by the assumption).
   1.1.2. $\sum_{i=1}^{M} \mathbf{X}_{ia} = \sum_{i=1}^{M} (\theta\mathbf{X}'_{ia} + (1-\theta)\mathbf{X}''_{ia}) = \theta \sum_{i=1}^{M} \mathbf{X}'_{ia} + (1-\theta) \sum_{i=1}^{M} \mathbf{X}''_{ia} \leq 1$.
   1.1.3. Q.E.D.
   1.2. Prove $\sum_{a=1}^{N} \mathbf{X}_{ia} \leq 1$.
   The proof is similar to 1.1.
   1.3. Prove $\mathbf{X}_{ia} \in [0, 1]$.
   1.3.1. $\mathbf{X}'_{ia} \in [0, 1]$, $\mathbf{X}'_{ia} \in [0, 1]$, and $\theta \in [0, 1]$ (by the assumption).
   1.3.2. $\mathbf{X}_{ia} = \theta\mathbf{X}'_{ia} + (1-\theta)\mathbf{X}''_{ia} \leq 1$.
   1.3.3. Q.E.D.
   1.4. Q.E.D.
2. Prove that $\mathcal{D}_\mathcal{M} \subseteq \mathcal{C}_\mathcal{M}$.
   It is straightforward that $\mathcal{D}_\mathcal{M} \subseteq \mathcal{C}_\mathcal{M}$ and the proof is omitted.
3. Prove that $\mathcal{D}_\mathcal{M} \subseteq \text{ext}(\mathcal{C}_\mathcal{M})$, where $\text{ext}(\mathcal{C}_\mathcal{M})$ denotes the extreme point set of $\mathcal{C}_\mathcal{M}$.
   Proof Sketch: Given the definition [20] of an extreme point in a convex set, it only needs to prove that any matrix $\mathbf{X} \in \mathcal{D}_\mathcal{M}$, which also belongs to $\mathcal{C}_\mathcal{M}$ by **2**, cannot be represented by the convex combination of other matrices in $\mathcal{C}_\mathcal{M}$.
   Assume:
   a) $\mathbf{X} \in \mathcal{D}_\mathcal{M}$, $\mathbf{X}^t \in \mathcal{C}_\mathcal{M}$, $\mathbf{X}^t \neq \mathbf{X}$, $t \in \{1, \ldots, T\}$.
   b) $w^t \in (0, 1)$, $\sum_{t=1}^{T} w^t = 1$.
   c) $\mathbf{X} = \sum_{t=1}^{T} w^t \mathbf{X}^t$.
   Prove: False.
   3.1. Since $\mathbf{X} \neq \mathbf{X}^t$, for any $\mathbf{X}^t$, there exists $\mathbf{X}_{ia} \neq \mathbf{X}^t_{ia}$.
   3.2. *Case:* $\mathbf{X}_{ia} = 1$.
   3.2.1. $\mathbf{X}^t_{ia} < 1$ (by step 3.1).
   3.2.2. $\mathbf{X}_{ia} = \sum_{t=1}^{T} \mathbf{X}^t_{ia} \leq (1-w^t)1 + w^t\mathbf{X}^t_{ia} < 1$.
   3.2.3. A contradiction between steps 3.2. and 3.2.2.
   3.3. *Case:* $\mathbf{X}_{ia} = 0$.
   3.3.1. $\mathbf{X}^t_{ia} > 0$ (by step 3.1).
   3.3.2. $\mathbf{X}_{ia} = \sum_{t=1}^{T} \mathbf{X}^t_{ia} \geq (1-w^t)0 + w^t\mathbf{X}^t_{ia} \geq 0$.
   3.3.3. A contradiction between steps 3.3. and 3.3.2.
   3.4. Q.E.D. (both cases are contradictions).
4. Prove that for any $\mathbf{X} \in \mathcal{C}_\mathcal{M} \backslash \mathcal{D}_\mathcal{M}$, there is $\mathbf{X} \notin \text{ext}(\mathcal{C}_\mathcal{M})$, where $\mathcal{C}_\mathcal{M} \backslash \mathcal{D}_\mathcal{M}$ denotes the relative complement of $\mathcal{D}_\mathcal{M}$ in $\mathcal{C}_\mathcal{M}$.
   Proof Sketch: For any $\mathbf{X} \in \mathcal{C}_\mathcal{M} \backslash \mathcal{D}_\mathcal{M}$, if it can be represented by the convex combination of other matrices, then there must be $\mathbf{X} \notin \text{ext}(\mathcal{C}_\mathcal{M})$.
   Assume: $\mathbf{X} \in \mathcal{C}_\mathcal{M} \backslash \mathcal{D}_\mathcal{M}$.
   Prove: There exist $\mathbf{X}', \mathbf{X}'' \in \mathcal{C}_\mathcal{M}, \mathbf{X}' \neq \mathbf{X}, \mathbf{X}'' \neq \mathbf{X}$, so that $\mathbf{X} = 0.5\mathbf{X}' + 0.5\mathbf{X}''$.
   4.1. *Case:* $\bar{L} \in \mathcal{N}$, where $\bar{L} = \sum_{i=1}^{M} \sum_{a=1}^{N} \mathbf{X}_{ia}$ and $\mathcal{N}$ denotes the set of non-negative integers.
   4.1.1. $\mathbf{X} \in \mathcal{C}^{\bar{L}}$ (by the definition of $\mathcal{C}^{\bar{L}}$).
   4.1.2. There exist such $\mathbf{X}', \mathbf{X}''$ (based on Lemma 2).
   4.1.3. Q.E.D.
   4.2. *Case:* $\bar{L} \in \mathcal{R}_+ \backslash \mathcal{N}$, where $\mathcal{R}_+$ denotes the set of non-negative reals.
   4.2.1. There exists $\mathbf{X}_{ia} \in (0, 1)$.
   4.2.2. There exists $i \in \{1, \ldots, M\}$ so that $\sum_a \mathbf{X}_{ia} \in (0, 1)$ (by step 4.2).
   4.2.3. There exists $a \in \{1, \ldots, N\}$ so that $\sum_i \mathbf{X}_{ia} \in (0, 1)$ (by step 4.2).
   4.2.4. *Case:* There exists at least one entry $\mathbf{X}_{ia} \in (0, 1)$, so that $\sum_a \mathbf{X}_{ia} \in (0, 1)$, and $\sum_i \mathbf{X}_{ia} \in (0, 1)$.
   [3].1. Let $\mathbf{X}'_{ia} = \mathbf{X}_{ia} - \nu$ where ν is a small positive constant.
   [3].2. Let $\mathbf{X}''_{ia} = \mathbf{X}_{ia} + \nu$.
   [3].3. $\mathbf{X} = 0.5\mathbf{X}' + 0.5\mathbf{X}''$.
   [3].4. Q.E.D.
   4.2.5. *Case:* For all $\mathbf{X}_{ia} \in (0, 1)$, there are $\sum_a \mathbf{X}_{ia} = 1$. or $\sum_i \mathbf{X}_{ia} = 1$
   [3].1. Let $R = \{r_1, \ldots, r_m\}$ denotes the collection of row numbers, where $\sum_a \mathbf{X}_{ia} \in (0, 1), \forall i \in R$ (by step 4.2.2).
   [3].2. For any $\mathbf{X}_{ia}, i \in R$, there is $\sum_i \mathbf{X}_{ia} = 1$ (by step 4.2.5).
   [3].3. Let $C = \{c_1, \ldots, c_n\}$ denotes the collection of column numbers, where $\sum_i \mathbf{X}_{ia} \in (0, 1), \forall a \in C$ (by step 4.2.3).
   [3].4. For any $\mathbf{X}_{ia}, a \in C$, there is $\sum_a \mathbf{X}_{ia} = 1$ (by step 4.2.5).
   [3].5. Locate an entry sequence $\mathcal{X} = \{\mathbf{X}_{i_1a_1}, \ldots, \mathbf{X}_{i_Sa_S}\}$, $\mathbf{X}_{i_sa_s} \in (0, 1)$ by searching row and column alternately.
   [4].1. Locate any $\mathbf{X}_{i_1a_1} \in (0, 1)$, where $i_1 \in R$.
   [4].2. Locate any $\mathbf{X}_{i_2a_2} \in (0, 1)$, where $a_2 = a_1$ (by step 4.2.5.2).
   [4].3. Locate any $\mathbf{X}_{i_3a_3} \in (0, 1)$, where $i_3 = i_2$ (by step 4.2.5.4).
   $\vdots$
   [4].S. Locate $\mathbf{X}_{i_Sa_S} \in (0, 1)$ and terminate when $i_S \in R$ or $a_S \in C$.
   [3].6 *Case:* $i_S \in R$ (i.e., S is an even number).
   [4].1. Let $\mathbf{X}'$ be

$$\begin{aligned}
\mathbf{X}'_{i_1a_1} &= \mathbf{X}_{i_1a_1} - \nu \\
\mathbf{X}'_{i_2a_2} &= \mathbf{X}_{i_2a_2} + \nu \\
&\cdots \\
\mathbf{X}'_{i_{S-1}a_{S-1}} &= \mathbf{X}_{i_{S-1}a_{S-1}} - \nu \\
\mathbf{X}'_{i_Sa_S} &= \mathbf{X}_{i_Sa_S} + \nu.
\end{aligned}$$

   [4].2. Let $\mathbf{X}''$ be

$$\begin{aligned}
\mathbf{X}''_{i_1a_1} &= \mathbf{X}_{i_1a_1} + \nu \\
\mathbf{X}''_{i_2a_2} &= \mathbf{X}_{i_2a_2} - \nu \\
&\cdots \\
\mathbf{X}''_{i_{S-1}a_{S-1}} &= \mathbf{X}_{i_{S-1}a_{S-1}} + \nu \\
\mathbf{X}''_{i_Sa_S} &= \mathbf{X}_{i_Sa_S} - \nu.
\end{aligned}$$

   [4].3. $\mathbf{X} = 0.5\mathbf{X}' + 0.5\mathbf{X}''$.
   [4].4. Q.E.D.
   [3].7 *Case:* $a_S \in C$ (i.e., S is an odd number).
   [4].1. Let $\mathbf{X}'$ be

$$\mathbf{X}'_{i_1a_1} = \mathbf{X}_{i_1a_1} - \nu$$

$$\begin{aligned}\mathbf{X}'_{i_2a_2} &= \mathbf{X}_{i_2a_2} + \nu\\ &\cdots\\ \mathbf{X}'_{i_{S-1}a_{S-1}} &= \mathbf{X}_{i_{S-1}a_{S-1}} + \nu\\ \mathbf{X}'_{i_Sa_S} &= \mathbf{X}_{i_Sa_S} - \nu.\end{aligned}$$

[4].2. Let $\mathbf{X}''$ be

$$\begin{aligned}\mathbf{X}''_{i_1a_1} &= \mathbf{X}_{i_1a_1} + \nu\\ \mathbf{X}''_{i_2a_2} &= \mathbf{X}_{i_2a_2} - \nu\\ &\cdots\\ \mathbf{X}''_{i_{S-1}a_{S-1}} &= \mathbf{X}_{i_{S-1}a_{S-1}} - \nu\\ \mathbf{X}''_{i_Sa_S} &= \mathbf{X}_{i_Sa_S} + \nu.\end{aligned}$$

[4].3. $\mathbf{X} = 0.5\mathbf{X}' + 0.5\mathbf{X}''$.

[4].4. Q.E.D.

5. Prove $\mathcal{D}_{\mathcal{M}} = \text{ext}(\mathcal{C}_{\mathcal{M}})$ (based on steps 3 and 4).

6. Q.E.D. (based on step 5 and Lemma 1). ■

In $\mathcal{C}$, $\overline{L} = \mathbf{x}^T\mathbf{1}$ is a real ranging from 0 to M rather than an integer. Different from the relation between $\mathcal{D}$ and $\mathcal{D}^L$ by (15), the union of $\mathcal{C}^L$ is only a proper subset of $\mathcal{C}$, that is

$$\mathcal{C} \supset \bigcup_{L=0}^{M} \mathcal{C}^L. \tag{20}$$

An (idealized or say simulative) illustration of $\mathcal{C}$ and its relation with $\mathcal{C}^L$ are also shown in Fig. 1. Besides all the regular polygons representing $\mathcal{C}^L$, $\mathcal{C}$ also contains the *space* between these polygons.

By relaxing the domain from $\mathcal{D}$ to $\mathcal{C}$, the Problem 2 is approximated by the following **Problem 4**:

$$\begin{aligned}\max\ & \mathbf{x}^T\mathbf{A}\mathbf{x} - \rho\mathbf{x}^T\mathbf{1}_{MN}\\ \text{s.t.}\ & \mathbf{x} \in \mathcal{C}.\end{aligned} \tag{21}$$

If $\mathbf{A}$ is positive definite, Problem 4 is a convex optimization problem for which effective optimization algorithms exist, such as the interior point method [22]. However, in realistic tasks $\mathbf{A}$ is usually neither positive definite nor negative definite, resulting in a more complex optimization problem. Besides, though we use Problem 4 to approximate Problem 2, a discrete solution rather than the continuous solution for Problem 4 is necessary. Next we show how to tackle these issues by the graduated projection scheme.

C. Optimization

When the objective function (21) is nonconvex and nonconcave, an intuitive method for Problem 4 is to first get the convex relaxation of (21) and then solve the relaxed problem by convex optimization techniques. Since the continuous solution locates in $\mathcal{C}$ rather than $\mathcal{D}$, to finally get the matching result, it needs to project the continuous solution back to $\mathcal{D}$. Though the projection can be efficiently realized by for example the Hungarian algorithm, it may introduce significant additional error because such a direct projection is independent of the objective function [16].

Below a graduated projection scheme is applied to Problem 4, which generalizes the graduated nonconvexity and concavity procedure (GNCCP). The GNCCP originates from the path following algorithms [16], [18], [19] which by gradually adjusting the linear combination parameter between a convex relaxation and a concave relaxation of the objective function, push the continuous solution to be a discrete one following a path. A major advantage of the GNCCP over the previous path following algorithms is that it implicitly realizes the linear combination in a simple formulation without needing to figure out the specific forms of the convex relaxation and the concave relaxation. Therefore, it is convenient to be extended.

Specifically, for adaptive graph matching the GNCCP can take the following form:

$$\begin{aligned}\max\ & F_\zeta(\mathbf{X}) = \begin{cases}(1+\zeta)F(\mathbf{x}) + \zeta\left(\mathbf{x}^T\mathbf{x} - \mathbf{x}^T\mathbf{1}_{MN}\right)\\ \qquad \text{if } \zeta^{\min} \le \zeta \le 0\\ (1-\zeta)F(\mathbf{x}) + \zeta\left(\mathbf{x}^T\mathbf{x} - \mathbf{x}^T\mathbf{1}_{MN}\right)\\ \qquad \text{if } 0 < \zeta \le \zeta^{\max}\end{cases}\\ \text{s.t.}\ & \mathbf{x} \in \mathcal{C}\end{aligned} \tag{22}$$

where $F(\mathbf{x}) = \mathbf{x}^T\mathbf{A}\mathbf{x} - \rho\mathbf{x}^T\mathbf{1}_{MN}$ is the original objective function (21). The initialization of the upper and lower thresholds $\zeta^{\max} \in [0, 1]$ and $\zeta^{\min} \in [-1, 0]$ is discussed in Section II-D1. Briefly speaking, when $\zeta \le \zeta^{\min}$, $F_\zeta(\mathbf{x})$ is concave, and when $\zeta \ge \zeta^{\max}$, $F_\zeta(\mathbf{x})$ becomes convex. By gradually increasing ζ from $\zeta^{\min}$ to $\zeta^{\max}$, maximizing $F_\zeta(\mathbf{x})$ is gradually transformed from a convex optimization problem to be a concave optimization problem, and correspondingly the continuous solution $\mathbf{x} \in \mathcal{C}$ is pushed into $\mathcal{D}$ which is the extreme point set of $\mathcal{C}$, as shown by Theorem 1.

For each specific ζ, $F_\zeta(\mathbf{x})$ is maximized by the Frank–Wolfe algorithm [23], [24], an effective nonlinear optimization algorithm. In each iteration, it needs to compute the ascent direction $\mathbf{y} - \mathbf{x}_{\text{old}}$ and step size α. In the computation of the ascent direction, $\mathbf{y}$ is obtained by solving the linear programming problem $\mathbf{y} = \arg\max \nabla F_\zeta(\mathbf{x}_{\text{old}})^T\mathbf{y}$, s.t. $\mathbf{y} \in \mathcal{C}$, for which a quick solution is provided in Section II-D2. The gradient $\nabla F_\zeta(\mathbf{x}_{\text{old}})$ given by

$$\nabla F_\zeta(\mathbf{x}) = \begin{cases}(1+\zeta)\nabla F(\mathbf{x}) + \zeta(2\mathbf{x} - \mathbf{1}_{MN})\\ \qquad \text{if } \zeta^{\min} \le \zeta \le 0\\ (1-\zeta)\nabla F(\mathbf{x}) + \zeta(2\mathbf{x} - \mathbf{1}_{MN})\\ \qquad \text{if } 0 < \zeta \le \zeta^{\max}\end{cases} \tag{23}$$

where

$$\nabla F(\mathbf{x}) = \left(\mathbf{A} + \mathbf{A}^T\right)\mathbf{x} - \rho\mathbf{1}_{MN}. \tag{24}$$

For undirected graph matching, there is $\nabla F(\mathbf{x}) = 2\mathbf{A}\mathbf{x} - \rho\mathbf{1}_{MN}$ because $\mathbf{A}$ is symmetric. The step size α can be obtained by inexact line search, e.g., the backtracking method [22].

The optimization method for adaptive graph matching is summarized in Algorithm 1.

D. Some Implementation Details and Discussion

1) Initialization of ζ^{min}, ζ^{max}, and $\boldsymbol{x}_0$: Any values making $F_\zeta(\mathbf{x}), \zeta \le \zeta^{\min}$ concave and $F_\zeta(\mathbf{x}), \zeta \ge \zeta^{\max}$ convex, can be used to initialize $\zeta^{\min}$ and $\zeta^{\max}$. The most convenient initialization is to set $\zeta^{\min} = -1$ and $\zeta^{\max} = 1$ as in [17]. However, for efficiency reasons, it is better to set $\zeta^{\min} = \zeta^{\min}_{\text{sup}}$ and $\zeta^{\max} = \zeta^{\max}_{\text{inf}}$, where $\zeta^{\min}_{\text{sup}}$ is the supremum of $\zeta^{\min}$ and $\zeta^{\max}_{\text{inf}}$ is the infimum of $\zeta^{\max}$. The calculation of $\zeta^{\min}_{\text{sup}}$ and

Algorithm 1 Adaptive Graph Matching

Given: the affinity matrix $\mathbf{A}$

Initialize ζ^{min}, ζ^{max}, and $\mathbf{x}_0$
$\mathbf{x}_{\text{old}} = \mathbf{x}_0$, $\zeta = \zeta^{min}$
repeat
  repeat
    $\mathbf{y} = \arg\max \nabla F_\zeta(\mathbf{x}_{\text{old}})^T \mathbf{y}$, s.t. $\mathbf{y} \in \mathcal{C}$
    $\alpha = \arg\max F_\zeta(\mathbf{x}_{\text{old}} + \alpha(\mathbf{y} - \mathbf{x}_{\text{old}}))$, s.t. $0 \le \alpha \le 1$
    $\mathbf{x}_{\text{new}} = \mathbf{x}_{\text{old}} + \alpha(\mathbf{y} - \mathbf{x}_{\text{old}})$
    $\mathbf{x}_{\text{old}} = \mathbf{x}_{\text{new}}$
  until $\nabla F_\zeta(\mathbf{x})^T (\mathbf{y} - \mathbf{x}_{\text{new}}) < \varepsilon |F_\zeta(\mathbf{x}_{\text{new}}) - \nabla F_\zeta(\mathbf{x}_{\text{new}})^T (\mathbf{y} - \mathbf{x}_{\text{new}})|$, where ε is a small positive constant
  $\zeta = \zeta + d\zeta$, where $d\zeta$ is the step size
until $(\zeta > \zeta^{max}) \vee (\mathbf{x}_{\text{new}} \in \mathcal{D})$, where $\vee$ denotes the OR operation

Output: $\mathbf{x}_{\text{new}}$

$\zeta_{\text{inf}}^{\max}$ is related to the concavity and convexity of $F_\zeta(\mathbf{x})$, whose Hessian matrix is

$$\mathbf{H}_\zeta = \begin{cases} (1+\zeta)\mathbf{H} + \zeta\mathbf{I}, & \text{if } \zeta^{\min} \le \zeta \le 0 \\ (1-\zeta)\mathbf{H} + \zeta\mathbf{I}, & \text{if } 0 < \zeta \le \zeta^{\max} \end{cases} \tag{25}$$

where $\mathbf{H}$ denotes the Hessian matrix of $F(\mathbf{x})$. To make $\mathbf{H}_\zeta = (1+\zeta)\mathbf{H} + \zeta\mathbf{I}$, $\zeta \le \zeta^{\min}$ negative definite, there must be

$$\zeta \le \frac{-\lambda_{\max}}{\lambda_{\max} + 1} \tag{26}$$

where $\lambda_{\max}$ denotes the largest eigenvalue of $\mathbf{H}$ [17]. Therefore, there is

$$\zeta_{\text{sup}}^{\min} = \min\left\{0, \frac{-\lambda_{\max}}{\lambda_{\max} + 1}\right\} \tag{27}$$

where $\zeta_{\text{sup}}^{\min} = 0$ only when $F(\mathbf{x})$ itself is a concave function. Similarly, there is

$$\zeta_{\text{inf}}^{\max} = \max\left\{0, \frac{\lambda_{\min}}{\lambda_{\min} - 1}\right\} \tag{28}$$

where $\lambda_{\min}$ denotes the smallest eigenvalue of $\mathbf{H}$, and $\zeta_{\text{inf}}^{\max} = 0$ only when $F(\mathbf{x})$ itself is a convex function. Thanks to the quadratic form of $F(\mathbf{x})$, it is easy to get $\mathbf{H} = \mathbf{A}^T + \mathbf{A}$, and then $\zeta_{\text{inf}}^{\min}$ and $\zeta_{\text{sup}}^{\min}$ can be efficiently obtained by the Arnoldi iteration [25].

$\mathbf{x}_0$ can be simply initialized by $\mathbf{x}_0 = \mathbf{1}_{MN}/N$ or $\mathbf{x}_0 = \mathbf{0}_{MN}$, especially when $\zeta^{\min}$ is initialized by $\zeta^{\min} = -1$. When $\zeta^{\min}$ is initialized by $\zeta^{\min} = \zeta_{\text{sup}}^{\min}$, $\mathbf{x}$ can be initialized by solving a constrained convex quadratic optimization problem by for instance the Newton algorithm, which could well accelerate the optimization process [16].

2) Complexity and Quick Solution: The previous algorithms [4]–[7], [9], [12] based on the affinity matrix $\mathbf{A}$ usually involve a $\max(\mathcal{O}(pq), \mathcal{O}(N^3))$ computational complexity, where p and q are the edge numbers of the two graphs. $\mathcal{O}(pq)$ is generally determined by the computation over $\mathbf{A}$, such as $\mathbf{Ax}$, which may be even as large as $\mathcal{O}(N^4)$ on fully connected graphs. $\mathcal{O}(N^3)$ is generally determined by solving the linear programming problem such as $\mathbf{y} = \arg\max \nabla F(\mathbf{x})^T \mathbf{y}$ or $\mathbf{y} = \arg\max \mathbf{x}^T \mathbf{y}$ by for instance the Hungarian algorithm. When $L < \min(M, N)$, the situation becomes more complex because the Hungarian algorithm or other efficient linear assignment algorithms [26] are inapplicable, while the general linear programming methods like interior point method usually involve an $\mathcal{O}(N^6)$ computational complexity. Beyond the approximate quick solution in [14], we propose an exact quick solution for the linear programming problem, which maintains the computational complexity of the proposed adaptive graph matching algorithm as $\max(\mathcal{O}(pq), \mathcal{O}(N^3))$.

The quick solution comes from the observation that for the linear programming problem

$$\begin{aligned} &\mathbf{y} = \arg\max \ \nabla F_\zeta(\mathbf{x})^T \mathbf{y} \\ &\text{s.t. } \mathbf{y} \in \mathcal{C} \end{aligned} \tag{29}$$

its solution $\mathbf{y}$ must belong to the $\mathcal{D}$, which corresponds to an optimal combination of only positive values in $\nabla F_\zeta(\mathbf{x})$. Specifically, the quick solution takes the following steps.

1) Perform $\nabla_P = E_1(\nabla F_\zeta(\mathbf{x}))$, where $\nabla_P \in \mathbf{1}^{MN}$ is of the same size with $\nabla F_\zeta(\mathbf{x})$, and $E_1(\cdot)$ is an entry-wise operator taking the following form:

$$(\nabla_P)_i = \begin{cases} \left(\nabla F_\zeta\right)_i, & \text{if } \left(\nabla\left(F_\zeta\right)_i > 0\right. \\ 0, & \text{if } \left(\nabla F_\zeta\right)_i \le 0. \end{cases} \tag{30}$$

2) Get a partial permutation matrix $\overline{\mathbf{y}} \in \mathcal{D}^M$ by solving the linear assignment problem

$$\begin{aligned} &\overline{\mathbf{y}} = \arg\max \nabla_P^T \mathbf{y}_1 \\ &\text{s.t.} \quad \overline{\mathbf{y}} \in \mathcal{C}^M. \end{aligned} \tag{31}$$

3) Perform $\mathbf{y} = E_2(\overline{\mathbf{y}}, \nabla_P)$, where $E_2(\cdot)$ is an entry-wise operator taking the following form:

$$\mathbf{y}_i = \begin{cases} \overline{\mathbf{y}}_i, & \text{if } \left(\overline{\mathbf{y}}_i = 1\right) \wedge \left((\nabla_P)_i > 0\right) \\ 0, & \text{otherwise} \end{cases} \tag{32}$$

where $\wedge$ denotes the AND operation.

The final solution of (29) is $\mathbf{y}$. Solving the linear assignment problem in the second step involves the highest complexity which is $\mathcal{O}(N^3)$ in the worst case.[5]

3) Selection of Parameter ρ: Recall that the objective function (14) is a tradeoff between the two terms $\mathbf{x}^T\mathbf{Ax}$ and $\mathbf{x}^T\mathbf{1}$. Therefore, the value of ρ can be estimated by balancing the two terms when the number of assignments varies. Consider the case when an extra assignment is added in $\mathbf{x}$ by setting the ith entry to be 1, i.e., $\mathbf{x}_i = 0 \to \mathbf{x}_i = 1, \forall i$. It is straightforward that the decrement $\mathfrak{D}_2$ of the second term $\mathbf{x}^T\mathbf{1}$ is

$$\mathfrak{D}_2 = \mathbf{x}_i = 1. \tag{33}$$

And the increment $\mathfrak{I}_1$ of the first term $\mathbf{x}^T\mathbf{Ax}$ is

$$\mathfrak{I}_1 = \mathbf{x}_i\mathbf{A}_{i:}\mathbf{x} + \mathbf{x}^T\mathbf{A}_{:i}\mathbf{x}_i - \mathbf{x}_i\mathbf{A}_{ii}\mathbf{x}_i \tag{34}$$

$$= \sum_{\substack{j=1 \\ \mathbf{x}_j=1}}^{MN} \mathbf{A}_{ij} + \sum_{\substack{j=1 \\ \mathbf{x}_j=1}}^{MN} \mathbf{A}_{ji} - \mathbf{A}_{ii} \tag{35}$$

where $\mathbf{A}_{i:}$ and $\mathbf{A}_{:i}$, respectively, denote the ith row and ith column of $\mathbf{A}$.

[5] A toy example for the quick solution is presented in the supplementary material.

Given the mean value $\overline{\mathbf{A}}_{::}$ of the matrix $\mathbf{A}$ as follows:

$$\overline{\mathbf{A}}_{::} = \frac{\sum_{i=1}^{MN}\sum_{j=1}^{MN}\mathbf{A}_{ij}}{(MN)^2} \tag{36}$$

the increment (38) can be approximated by

$$\Im_1 \approx 2(\mathbf{x}^T\mathbf{1})\overline{\mathbf{A}}_{::} - \mathbf{A}_{::}. \tag{37}$$

Since after setting $\mathbf{x}_i = 1$, $\mathbf{x}^T\mathbf{1}$ varies from 1 to M, it is approximated by the mean value $(1 + M/2)$. Then there is

$$\Im_1 \approx M\overline{\mathbf{A}}_{::}. \tag{38}$$

To balance of the increment (38) and decrement (33), there should be

$$\Im_1 \approx \rho\mathfrak{D}_2. \tag{39}$$

Then we get a selection guideline for ρ by

$$\rho \leftarrow M\overline{\mathbf{A}}_{::}. \tag{40}$$

In practice it is better to set ρ to be a bit larger than $M\overline{\mathbf{A}}_{::}$, such as $2M\overline{\mathbf{A}}_{::}$.

In (40), the selection of ρ is directly related to the mean value $\overline{\mathbf{A}}_{::}$ with respect to all the entries in $\mathbf{A}$. An interesting deduction based on this observation is that the selection of ρ is also closely related to the edge density, given that the sparsity of the affinity matrix $\mathbf{A}$ is determined by the edge density. The explanation is as follows. In (36), the numerator can be rewritten as the sum of nonzero entries in $\mathbf{A}$, and there is

$$\overline{\mathbf{A}}_{::} = \frac{\sum_{i=1}^{MN}\sum_{\substack{j=1\\ \mathbf{A}_{ij}\neq 0}}^{MN}\mathbf{A}_{ij}}{(MN)^2} = \overline{\mathbf{A}}_{\mathrm{nz}}\frac{\sum_{i=1}^{MN}\sum_{\substack{j=1\\ \mathbf{A}_{ij}\neq 0}}^{MN}1}{(MN)^2} \tag{41}$$

where $\overline{\mathbf{A}}_{\mathrm{nz}}$ denotes the mean value with respect to the nonzero entries in $\mathbf{A}$, that is

$$\overline{\mathbf{A}}_{\mathrm{nz}} = \frac{\sum_{i=1}^{MN}\sum_{\substack{j=1\\ \mathbf{A}ij\neq 0}}^{MN}\mathbf{A}_{ij}}{\sum_{i=1}^{MN}\sum_{\substack{j=1\\ \mathbf{A}ij\neq 0}}^{MN}1}. \tag{42}$$

Note that the number of nonzero entries is exactly

$$\sum_{i=1}^{MN}\sum_{\substack{j=1\\ \mathbf{A}ij\neq 0}}^{MN}1 = pq + MN \tag{43}$$

where pq is number of off-diagonal nonzero entries encoding edge similarities in $\mathbf{A}$, and MN is the number of diagonal entries encoding vertex similarities. Since for any connected graph the edge number (p or q) involves at least a linear increase, usually a high order increase, with respect the vertex number (i.e., M or N), ignoring the number of diagonal entries MN there is

$$\overline{\mathbf{A}}_{::} = \overline{\mathbf{A}}_{\mathrm{nz}}\frac{p}{M^2}\frac{q}{N^2}. \tag{44}$$

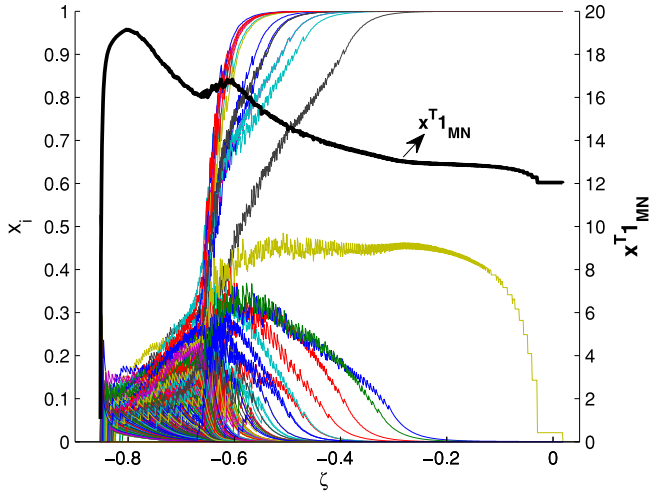


Fig. 3. Optimization process of a matching example with $M = N = 20$ and $L = 12$. The fine curves denote the updating of $\mathbf{x}_i$, corresponding to the left y-axis, and the thick curve denotes the updating of $\mathbf{x}^T\mathbf{1}$, corresponding to the right y-axis.

By defining the edge density of a graph $\mathcal{G}$ as the ratio between the edge number and the number of possible edges

$$E(\mathcal{G}) = \frac{2p}{M(M-1)} \tag{45}$$

it can be derived from (40) and (44) that the selection of ρ is positively related to the edge densities of the two input graphs $\mathcal{G}$ and $\mathcal{H}$, that is

$$\rho \propto E(\mathcal{G})E(\mathcal{H}). \tag{46}$$

4) Other Discussion: Different from [17], in (22) we utilize the term $\mathbf{x}^T\mathbf{x} - \mathbf{x}^T\mathbf{1}_{MN}$ rather than $\mathbf{x}^T\mathbf{x}$ (i.e., $\mathbf{tr}(\mathbf{X}^T\mathbf{X})$ in [17]) to guarantee $F_\zeta(\mathbf{x})$ and $F(\mathbf{x})$ own the same maximum points in $\mathcal{D}$.

The optimization process is illustrated in Fig. 3. As ζ increases, $\mathbf{x}_i$ is gradually pushed to 0 or 1, which is similar to the previous path following algorithms (see [18, Fig. 6(c)] for comparison). On the other hand, $\overline{L} = \mathbf{x}^T\mathbf{1}$ varies in the optimization process and finally reaches an integer, 12 in this example, while in previous path following algorithms [16], [18], [19] it is always a constant.

III. Related Works

From the model aspect, the graduated assignment algorithm [12], [27] uses a dissimilarity matrix $\mathbf{K}$-based model, which can be equivalently transformed to an affinity matrix-based model, and the model is optimized by a deterministic annealing method. Some spectral algorithms [6], [7] also use the affinity matrix, and relax the discrete problem to be a continuous spectral decomposition problem and optimize it by the standard or improved power iteration. An important improvement of the spectral algorithm [6] is to introduce the integer projection scheme [8] by the same authors. The reweighted random walk matching algorithm [4] interprets and solves the graph matching problem from the random walk perspective, which can be treated as a generalization of the spectral

algorithm [6]. The same authors also introduced the max-pooling scheme [28] in the power iteration. The factorized graph matching algorithm [18] factorizes the affinity matrix for the sake of storage efficiency, and generalizes the path following algorithms [16], [19] to optimize the problem. Further the same authors improved the factorized graph matching algorithm by estimating the affine transformation parameters in the optimization process, which is named by deformable graph matching [29]. Since the above algorithms use or implicitly use the model (2), to some extent their graph matching models can be considered as special cases of the adaptive graph matching model (14).

An early robust point matching algorithm [30] uses a similar additional term as $\mathbf{x}^T\mathbf{1}$ in their model. Different from the proposed method aiming at unknown inlier number, the additional term in [30] is only used to guard against null assignments, which results in M assignments, i.e., $\mathbf{x} \in \mathcal{D}_M$. The dual decomposition-based algorithm incorporates an additional term controlling the assignment number in its model, which is then optimized by the dual decomposition technique. Different from the proposed method simultaneously estimating the inlier number and matching the inliers, this method belongs to the two-step algorithms, in which the final assignments are selected by ranking the assignment confidence. Another thing about this method is that the local exhaust search or the branch-and-bound search used in its optimization usually lead to a *significant lower* running speed, as observed in [7].

From the optimization aspect, the adaptive graph matching problem is optimized by generalizing the GNCCP [13], [14], [17], which originates from the path following algorithms [16], [18], [19], [29], as introduced in Section I. This group of algorithms are related to above mentioned integer projected fixed point method [8], [9] which also involves the graduated projection in the optimization process.

IV. Experimental Results

In this section, the proposed method is comparatively evaluated on both synthetic data and real world images. On the synthetic data, the performance of the proposed method is first evaluated with respect to noise level, problem size, outlier number, and edge density, which is followed by the test of selection guidelines of ρ, and the running time comparison. Besides, an experimental case study on the subgraph matching problem, in which outliers exist in only one graph, is also performed on the synthetic data. On the real world images, three datasets are utilized, which are, respectively, the famous *House* sequence, the widely used *Car* and *Motorbike* matching dataset, and a handwritten Chinese character dataset.

The proposed algorithm denoted by ADM is mainly compared with the two-step algorithms including SM [6], GA [12], RRWM [4], and PGM [7]. Besides, some graduated projection-based algorithms, including IPFP [8], FGM [18], and SGM [14], [17], are compared in the case study on the subgraph matching problem, given that the other graduated projection-based algorithms except ADM are inapplicable to the general Problem 4. Among all these algorithms, GA, PGM, and SGM are implemented by us, and SM, RRWM,[6] IPFP, and FGM[7] are implemented by public codes.

Since the estimated inlier number may be unequal to the ground truth one, two comparison criteria, *recall rate* or *accuracy*, are adopted [31], [32]. The recall rate measures how many assignments are correctly recalled from the ground truth assignments, that is

$$\text{recall rate} = \frac{\#\text{ correct assignments}}{\#\text{ ground truth assignments}} \tag{47}$$

and accuracy refers to the ratio between the number of the correct assignments and that of the selected assignments, that is

$$\text{accuracy} = \frac{\#\text{ correct assignments}}{\#\text{ selected assignments}}. \tag{48}$$

Since the assignment number of the two-step algorithms cannot be adaptively determined, the number of selected assignments in their accuracies is set to be equal to ADM.

In the experiments, the similarity measures $\mathcal{A}(l_i, l_a)$ and $\mathcal{A}(w_{ij}, w_{ab})$ in (4) are defined by

$$\mathcal{A}(l_i, l_a) = \exp\left(\frac{-\|l_i - l_a\|^2}{\sigma_l}\right) \tag{49}$$

$$\mathcal{A}(w_{ij}, w_{ab}) = \exp\left(-\frac{\|w_{ij} - w_{ab}\|^2}{\sigma_w}\right) \tag{50}$$

where σ_l and σ_w are kernel width parameters and are empirically set to be $\sigma_l = 0.15$ and $\sigma_w = 0.15$.

A. Synthetic Points

The first experiment is performed on the synthetic points, which are generated in a similar way as in [14]. First, two point sets $G = \{g_i\}_{i=1}^{M}$ and $H = \{h_i\}_{j=1}^{N}$ on a 2-D-plane are randomly generated by uniform sampling. Second, an assignment vector $\mathbf{X}^{\text{gt}} \in \mathcal{D}^L$ with L ground truth assignments is randomly generated. Finally, the first point set G is updated by permutating H with $\mathbf{X}^{\text{gt}}$ as follows:

$$g_i = h_j + \eta,\ \eta \sim N\left(0, \sigma^2\right),\ \text{if } \mathbf{X}_{ij}^{\text{gt}} = 1 \tag{51}$$

where η denotes the additive gaussian noise. The sparse graph structure is constructed by adjusting the edge density, and it is also disturbed by the noise η following the way in [16]. That is $(1/2)\sigma p$ edges are randomly added to and removed from each sparse graph, where p denotes the edge number. The normalized length is utilized as the edge weight.

The comparisons are performed with respect to noise level (σ), problem size (N), outlier number (# outliers), and edge density ($E(\mathcal{G})$). Specifically, in the first comparison σ is increased from 0 to 0.2 by a step size of 0.02, with the other variables fixed by $N = 30$, # outliers $= 5$, and $E(\mathcal{G}) = 0.1$. In the second comparison N is increased from 20 to 40 by a step size of 2, with $\sigma = 0.1$, # outliers $= 5$, and $E(\mathcal{G}) = 0.1$. In the third comparison # outliers is increased from 0 to 10 by a step size of 1, with $\sigma = 0.1$, $N = 30$, and $E(\mathcal{G}) = 0.1$.

[6]Codes for SM and RRWM are available at http://cv.snu.ac.kr/research/∼RRWM/.

[7]Codes for IPFP and FGM are available at http://www.f-zhou.com/gm.html.

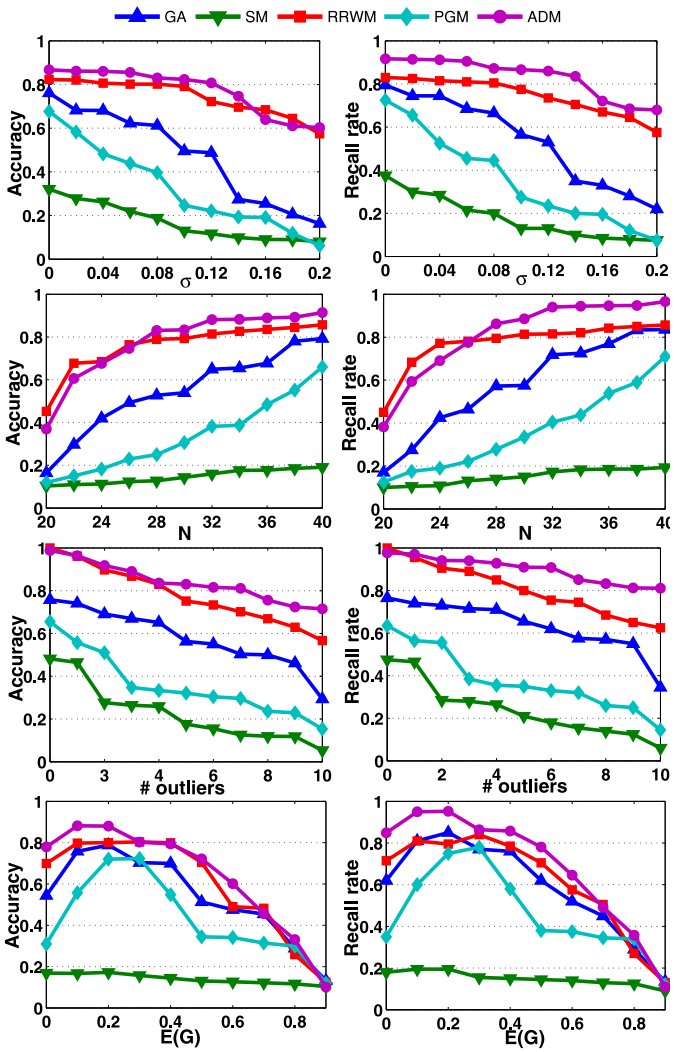


Fig. 4. Quantitative comparison results on synthetic points. The comparisons with respect to noise level, problem size, outlier number, and edge density are, respectively, shown in the four rows. The criteria accuracy and recall rate are, respectively, used in the left and right columns.

In the fourth comparison $E(\mathcal{G})$ is increased from 0.1 to 1 by a step size of 0.1, with $\sigma = 0.1$, $N = 30$, and # outliers $= 5$. In all these comparisons, it is set that $M = N-$# outliers and $L = N-2$# outliers. The parameter ρ is set by (40), i.e., $M\overline{\mathbf{A}}_{::}$.

The quantitative comparison results are shown in Fig. 4. Two interesting observations are from the second comparison and the fourth comparison. For the second one, the accuracies increase as the problem size increases, which may be because that by our settings the ratio of the outliers decreases at the same time. For the fourth one, the accuracies first increase and then decrease as the edge density increases. The reason is that when the graphs are too sparse the structural information may be insufficient, and when the graphs are too dense the excessive noise is introduced. From the four comparisons, it can be observed generally ADM outperforms the other algorithms on both accuracy and recall rate. And by comparing its accuracy and recall rate, it can be observed in general the accuracy is slightly smaller than the recall rate, which means statistically the number of selected assignments is larger than the number

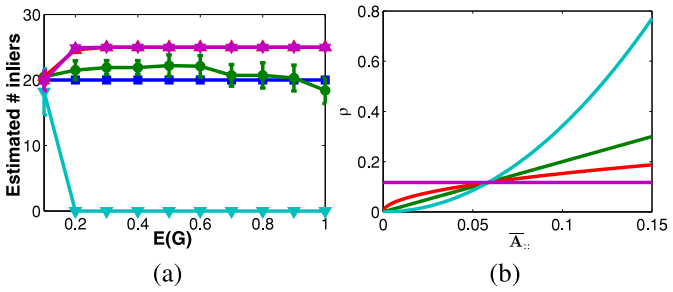


Fig. 5. Comparison on selection guidelines of ρ. (a) Compares the estimated inlier numbers by different selection guidelines of ρ with respect to the edge density. (b) Illustrates the values of ρ by different selection guidelines with respect to $\overline{\mathbf{A}}_{::}$.

of the ground truth assignments. It implies that the selection of ρ in these comparisons, i.e., $M\overline{\mathbf{A}}_{::}$, should be increased a bit.

Using the same setting with the fourth comparison above, an additional experiment is performed to assess the selection of ρ. Besides the selection guideline (40) rewritten by

$$\rho_1 \leftarrow c_1 M\overline{\mathbf{A}}_{::} \tag{52}$$

other selection guidelines for comparison include

$$\rho_2 \leftarrow c_2 M\overline{\mathbf{A}}_{::}^{\frac{1}{2}} \tag{53}$$

$$\rho_3 \leftarrow c_3 M\overline{\mathbf{A}}_{::}^{2} \tag{54}$$

and

$$\rho_4 \leftarrow c_4 M. \tag{55}$$

The constant c_1 is set to be 1. And c_2–c_4 are set by making ρ_1–ρ_4 equal when the edge density starts at 0.1, which corresponds to the intersection point in Fig. 5(b). Recall that the value of $\overline{\mathbf{A}}_{::}$ is positively related to the edge density. The comparison results given in Fig. 5 witness the robustness of the proposed guideline (52) with respect to the edge density. It may be counterintuitive that the inlier number estimated by ρ_2 is usually larger than ρ_3 because intuitively ρ_2 should be larger than ρ_3 and therefore should result in a smaller inlier number estimation. The reason is that as shown in Fig. 5(b) ρ_2 is actually smaller than ρ_3 by the calculation method of c_2 and c_3.

The running time with respect to the problem size is compared in Fig. 6. The plot is in a logarithmic style, and the growth rates of the curves indicate the computational complexities of different algorithms. Specifically, the growth rates of GA and PGM are about 4.9 ± 0.3, and that of RRWM is about 3.5 ± 0.6, and those of SM and ADM are about 4.1 ± 0.4.

1) Case Study on Subgraph Matching: By setting ρ small enough, the inlier number estimated by ADM would always be M, i.e., $\mathbf{x} \in \mathcal{D}^M$, and thus the general case Problem 2 is degenerated to the *part-in-whole* subgraph matching problem in which outliers exist in at most one graph. Below an experimental case study on this problem is done by setting $\rho = 0$ for ADM. In this case, some other graduated projection-based algorithms including IPFP [8], FGM [18], and SGM [17], are available for comparison. And in this case *recall rate* and *accuracy* defined in (47) and (48) are the same, and therefore

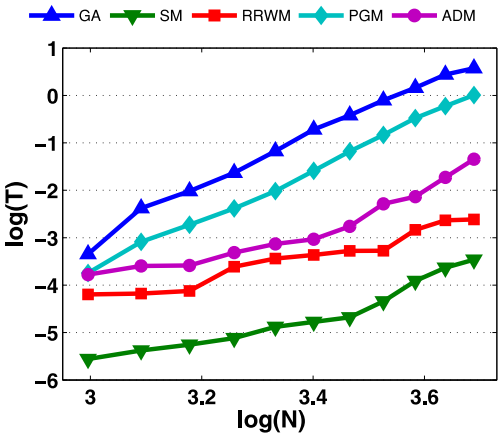


Fig. 6. Running time comparison with respect to problem size. The plot is shown in a logarithmic manner, where T in the y-axis denotes the running time in seconds.

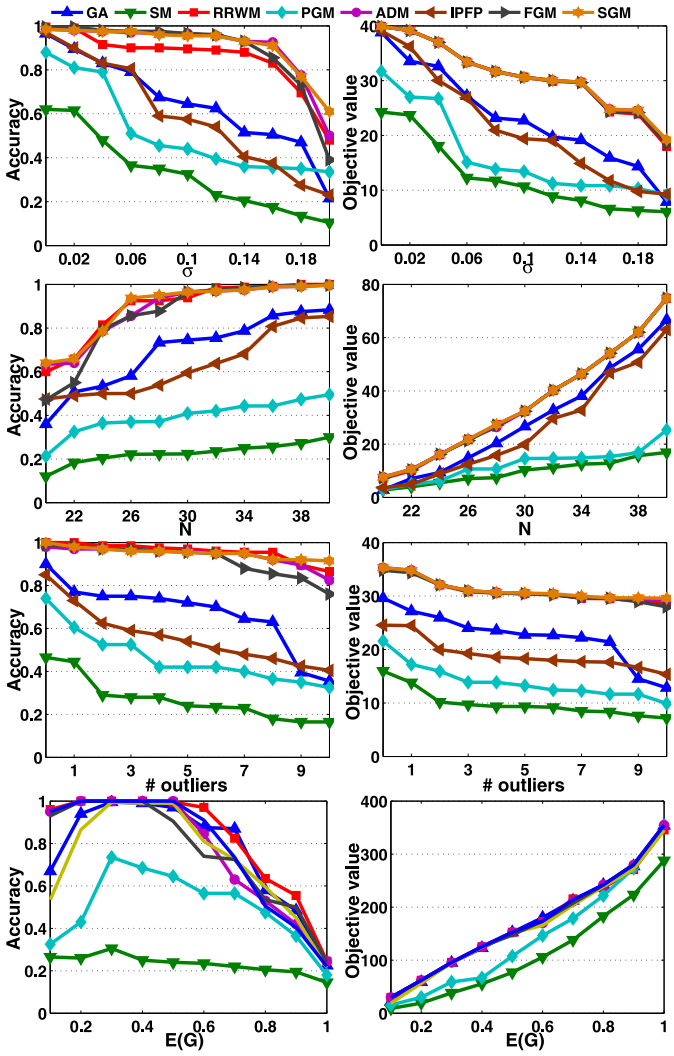


Fig. 7. Quantitative comparison results of case study on subgraph matching. The comparisons with respect to noise level, problem size, outlier number, and edge density are, respectively, shown in the four rows. The criteria accuracy and objective value are, respectively, used in the left and right columns.

accuracy is used as one criterion. Meanwhile, when $\rho = 0$ all the algorithms actually use the same objective function (2), and therefore *objective value* is adopted as another criterion.

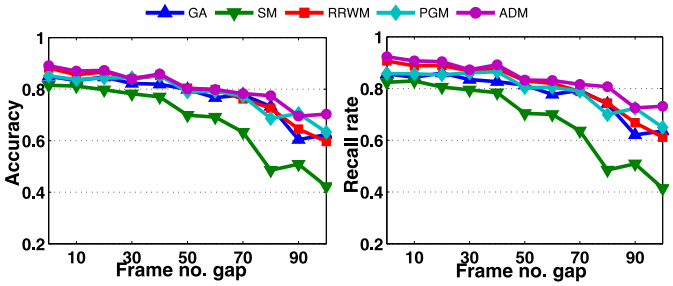


Fig. 8. Quantitative comparison results on *House* sequence. The criteria accuracy and recall rate are, respectively, used in the left and right columns.

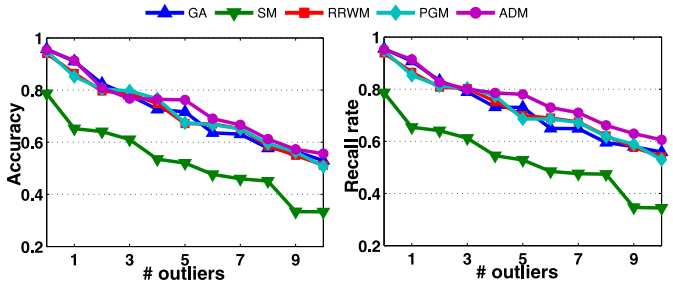


Fig. 9. Quantitative comparison results on *Motorbike* and *Car* images. The criteria accuracy and recall rate are, respectively, used in the left and right columns.

The generation of the synthetic points, the construction of graphs, and the comparison settings are the same with the above experiment except for setting $L = M$. The experimental results are shown in Fig. 7. It can be observed that the *part-in-whole* subgraph matching problem is easier than the general case, given that all the methods get higher accuracies than those in Fig. 4. Generally the performance of ADM is comparable with other graduated projection-based methods, such as FGM and SGM, even better than IPFP. However, they are inapplicable to the general adaptive graph matching problem. The inconsistency between the accuracy and objective value is also observed, that is a higher accuracy may be sometimes associated with a lower objective value. It is because the ground truth assignments may not always correspond to the lowest objective value.

B. House Sequence

The first experiment on the real world images is carried out on the CMU *House* sequence,[8] which consists of 111 frames sampled from a 3-D-rotating *House* video clip. All the frames are manually labeled with 30 landmark points as in [33], where a larger gap between frames (denoted by frame no. gap) implies a more difficult matching. The frame no. gap is increased from 0 to 100 by a step size of 10. For each frame no. gap, ten image pairs are randomly chosen, and thus 110 image pairs are obtained. Since this paper deals with the situation where outliers exist in both images, the matching problem is made as follows. First 20 ground truth assignments are randomly chosen and the corresponding points are preserved in both images; then the other ten assignments are divided into

[8]The *House* sequence dataset is available at http://vasc.ri.cmu.edu//idb/html/motion/house/index.html.

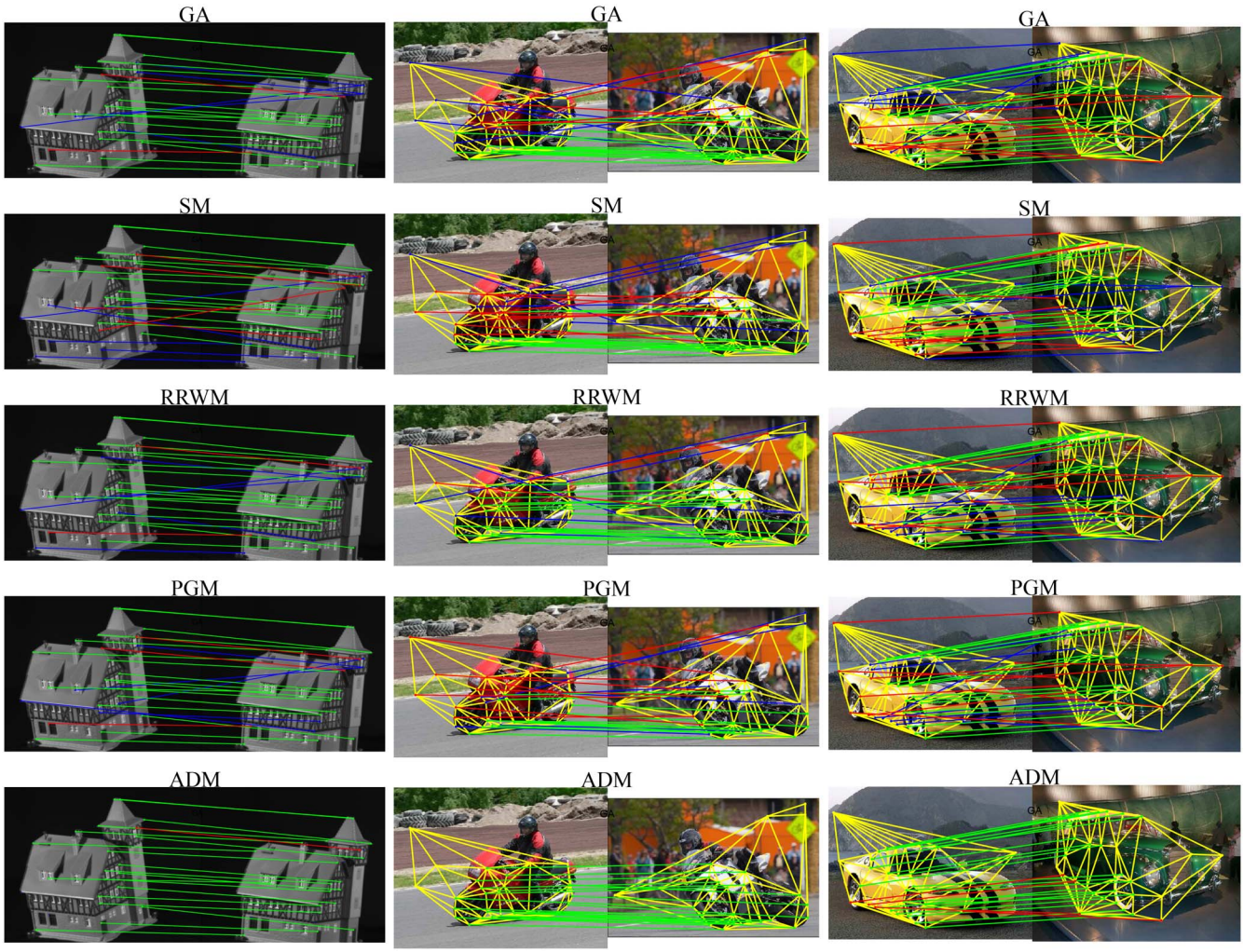


Fig. 10. Typical *House*, *Motorbike*, and *Car* matching instances by different algorithms. In the *House* matching, the numbers of ground truth assignments and selected assignments are, respectively, 20 and 20. In the *Motorbike* matching, the numbers are, respectively, 21 and 20. And in the *Car* matching, the numbers are, respectively, 28 and 26. The correct and wrong assignments are shown in green and red, respectively. For the two-step algorithms, the unselected assignments with low assignment confidences are shown in blue.

two parts with size $M' + N' = 10$ and the corresponding points are preserved separately; finally, two point sets with size $M = 20 + M'$ and $N = 20 + N'$ are obtained. Without loss of generality, it is guaranteed $M \leq N$ in the division. The normalized length and normalized direction are used as edge weights and the graph structure is constructed by full connection. The parameter ρ is set to be $M\overline{\mathbf{A}}_{::}$.

The quantitative comparison results are given in Fig. 8. Generally ADM obtains higher accuracy and recall rate than the other algorithms. Note that the previous works [7], [18] show that on the *House* sequence RRWM and PGM get almost 100% accuracy at the frame no. gap of 80, and it is less than 80% in Fig. 8, which implies the difficulty of the adaptive graph matching problem with outliers in both images. Some typical matching instances by different algorithms are given in Fig. 10.

C. Car and Motorbike Images

This dataset consists of five pairs of *Motorbike* images and five pairs of *Car* images fetched from a benchmark matching dataset[9] [8], [9], where the ground truth assignment number varies from 15 to 34. The performances are compared with respect to the outlier number which is increased from 0 to 10 by a step size of 1, and therefore 110 matching pairs are obtained. The normalized length and normalized direction are used as edge weights and the graph structure is built by the Delaunay triangulation [34]. It is set that $\rho = 2M\overline{\mathbf{A}}_{::}$. The quantitative comparison results are given in Fig. 9. By setting ρ a bit larger than $M\overline{\mathbf{A}}_{::}$, the more balanced accuracy and recall rate can be observed, which implies a more accurate inlier number estimation of ADM. When the outlier number is zero, the algorithms GA, RRWM, PGM, and ADM all achieve quite excellent matching performance, similarly to the observations in [14] and [18]. And when the outlier number increases, the accuracies and recall rates all obviously decrease. Generally, on the *Car* and *Motorbike* images, ADM achieves comparable performance with GA, RRWM, and PGM, and slightly better performance with more outliers. Some matching instances are given in Fig. 10.

[9]The *Car* and *Motorbike* dataset is available at http://109.101.234.42/code.phps.

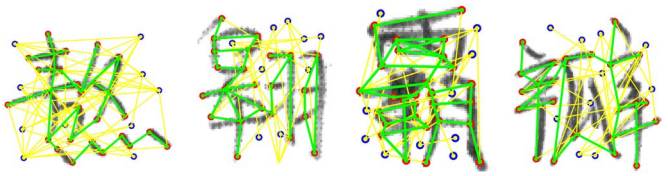


Fig. 11. Handwritten Chinese character samples. The inliers and outliers are, respectively, shown in red and blue. The graph structure by the character skeleton and randomness are, respectively, shown in green and yellow.

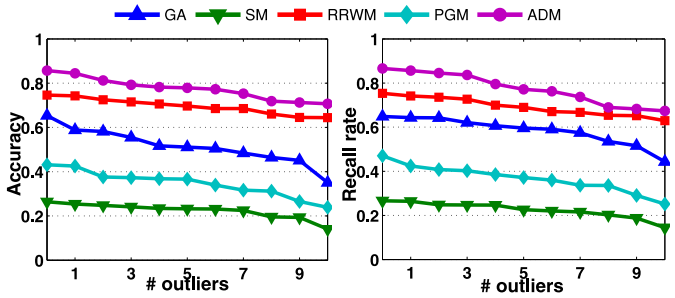


Fig. 12. Quantitative comparison results on handwritten Chinese character images. The criteria accuracy and objective value are, respectively, used in the left and right columns.

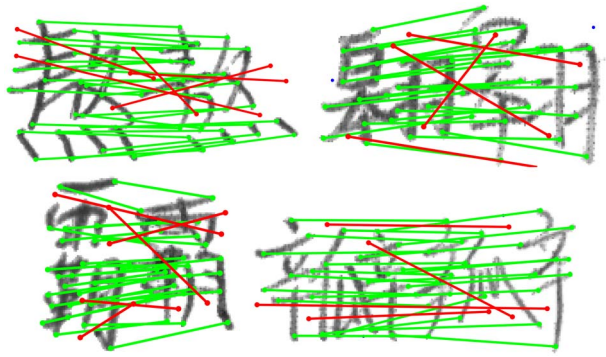


Fig. 13. Typical handwritten Chinese character matching instances by ADM. The correct and wrong assignments are shown in green and red, respectively.

D. Handwritten Chinese Character Images

In the handwritten Chinese character images, there are four characters and each character consists of ten image samples fetched from [35]. Hence there are totally 45 image pairs. For the first and third characters, 28 ground truth points are manually labeled, and for the second and fourth characters, 23 ground truth points are manually labeled. Further, ten outliers in each image are randomly labeled. The graph structure is built by taking into account both character skeleton and randomness, about which please refer to [14] for details. Some samples with labeled points and graph structures are illustrated in Fig. 11. The comparison is performed with respect the outlier number which is increased from 0 to 10 by a step size of 1. The quantitative comparison results are given in Fig. 12, which show the superior performance of ADM. Some matching instances are given in Fig. 13.

V. Conclusion

This paper introduces an adaptive graph matching algorithm to deal with the point correspondence problem when there exist outliers in both point sets and the number of inliers is unknown. Consequently, the proposed method is able to adaptively estimate the number of inliers and figure out their correspondence. From the perspective of objective function, the proposed one is a natural extension of existing graph matching algorithms, and from the perspective of optimization techniques, the proposed method introduces a graduated projection scheme for the objective function. Simulations on both synthetic data and real-world images witness its effectiveness. Related issues include the cluster problem which can be interpreted by graph theory and it often needs to adaptively determine the number of cluster centers.

Acknowledgment

The authors would like to thank Dr. H.-W. Zhang at the Academy of Mathematics and Systems Sciences, Chinese Academy of Sciences for his help on the presentation of mathematical proofs. They would also like to thank the anonymous reviewers whose comments and suggestions have greatly improved this paper.

References

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

[2] J. Tang, L. Shao, X. L. Li, and K. Lu, "A local structural descriptor for image matching via normalized graph Laplacian embedding," *IEEE Trans. Cybern.*, vol. 46, no. 2, pp. 410–420, Feb. 2016.

[3] H. L. Shang, Y. D. Tao, Y. Gao, C. Zhang, and X. L. Wang, "An improved invariant for matching molecular graphs based on VF2 algorithm," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 45, no. 1, pp. 122–128, Jan. 2015.

[4] M. Cho, J. Lee, and K. M. Lee, "Reweighted random walks for graph matching," in *Proc. Eur. Conf. Comput. Vis.*, Heraklion, Greece, 2010, pp. 492–505.

[5] T. Cour, P. Srinivasan, and J. B. Shi, "Balanced graph matching," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 19. Vancouver, BC, Canada, 2006, pp. 313–320.

[6] M. Leordeanu and M. Hebert, "A spectral technique for correspondence problems using pairwise constraints," in *Proc. IEEE Int. Conf. Comput. Vis.*, Beijing, China, 2005, pp. 1482–1489.

[7] 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.

[8] 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.*, Vancouver, BC, Canada, 2009, pp. 1114–1122.

[9] M. Leordeanu, R. Sukthankar, and M. Hebert, "Unsupervised learning for graph matching," *Int. J. Comput. Vis.*, vol. 96, no. 1, pp. 28–45, 2012.

[10] S. Biasotti, S. Marini, M. Spagnuolo, and B. Falcidieno, "Sub-part correspondence by structural descriptors of 3D shapes," *Comput. Aided Design*, vol. 38, no. 9, pp. 1002–1019, 2006.

[11] H. Chui and A. Rangarajan, "A new algorithm for non-rigid point matching," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Hilton Head Island, SC, USA, 2000, pp. 44–51.

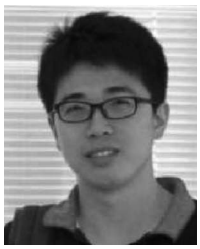
[12] S. Gold and A. Rangarajan, "A graduated assignment algorithm for graph matching," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 18, no. 4, pp. 377–388, Apr. 1996.

[13] X. Yang, H. Qiao, and Z. Y. Liu, "A weighted common subgraph matching algorithm," *IEEE Trans. Neural Netw. Learn. Syst.*, *arXiv preprint arXiv:1411.0763*.

[14] X. Yang, H. Qiao, and Z.-Y. Liu, "Outlier robust point correspondence based on GNCCP," *Pattern Recognit. Lett.*, vol. 55, pp. 8–14, Apr. 2015.

[15] H. A. Almohamad and S. O. Duffuaa, "A linear programming approach for the weighted graph matching problem," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 15, no. 5, pp. 522–525, May 1993.

[16] M. Zaslavskiy, F. Bach, and J.-P. Vert, "A path following algorithm for the graph matching problem," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 12, pp. 2227–2242, Dec. 2009.
[17] Z.-Y. Liu and H. Qiao, "GNCCP—Graduated nonconvexity and concavity procedure," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 6, pp. 1258–1267, Jun. 2014.
[18] F. Zhou and F. De la Torre, "Factorized graph matching," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Providence, RI, USA, 2012, pp. 127–134.
[19] Z.-Y. Liu, H. Qiao, and L. Xu, "An extended path following algorithm for graph-matching problem," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 7, pp. 1451–1456, Jul. 2012.
[20] J. M. Borwein and A. S. Lewis, *Convex Analysis and Nonlinear Optimization: Theory and Examples*, vol. 3. New York, NY, USA: Springer, 2006.
[21] L. Lamport, "How to write a proof," *Amer. Math. Monthly*, vol. 102, no. 7, pp. 600–608, 1995.
[22] S. P. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge, U.K.: Cambridge Univ. Press, 2004.
[23] M. Frank and P. Wolfe, "An algorithm for quadratic programming," *Naval Res. Logistics Quart.*, vol. 3, nos. 1–2, pp. 95–110, 1956.
[24] M. Jaggi, "Revisiting Frank–Wolfe: Projection-free sparse convex optimization," in *Proc. Int. Conf. Mach. Learn.*, Atlanta, GA, USA, 2013, pp. 427–435.
[25] R. B. Lehoucq and D. C. Sorensen, "Deflation techniques for an implicitly restarted Arnoldi iteration," *SIAM J. Matrix Anal. Appl.*, vol. 17, no. 4, pp. 789–821, 1996.
[26] A. Schrijver, *Combinatorial Optimization: Polyhedra and Efficiency*, Heidelberg, Germany: Springer Verlag, 2003.
[27] Y. Tian *et al.*, "On the convergence of graph matching: Graduated assignment revisited," in *Proc. Eur. Conf. Comput. Vis.*, Florence, Italy, 2012, pp. 821–835.
[28] 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. IEEE Conf. Comput. Vis. Pattern Recognit.*, Columbus, OH, USA, 2014, pp. 2091–2098.
[29] F. Zhou and F. De la Torre, "Deformable graph matching," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Portland, OR, USA, 2013, pp. 2922–2929.
[30] H. Chui and A. Rangarajan, "A new point matching algorithm for non-rigid registration," *Comput. Vis. Image Understanding*, vol. 89, nos. 2–3, pp. 114–141, 2003.
[31] Y. Wang *et al.*, "Semisupervised tripled dictionary learning for standard-dose PET image prediction using low-dose PET and multimodal MRI," *IEEE Trans. Biomed. Eng.*, vol. 64, no. 3, pp. 569–579, Mar. 2017.
[32] C. Zu *et al.*, "Robust multi-atlas label propagation by deep sparse representation," *Pattern Recognit.*, vol. 63, pp. 511–517, Mar. 2017.
[33] T. S. Caetano, J. J. McAuley, L. Cheng, Q. V. Le, and A. J. Smola, "Learning graph matching," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 6, pp. 1048–1058, Jun. 2009.
[34] P. Su and R. L. S. Drysdale, "A comparison of sequential Delaunay triangulation algorithms," in *Proc. Annu. Symp. Comput. Geom.*, Vancouver, BC, Canada, 1995, pp. 61–70.
[35] C. L. Liu, F. Yin, D. H. Wang, and Q. F. Wang, "CASIA online and offline Chinese handwriting databases," in *Proc. Int. Conf. Document Anal. Recognit.*, Beijing, China, 2011, pp. 37–41.



Xu Yang received the B.S. degree in automation from the Ocean University of China, Qingdao, China, in 2009, and the Ph.D. degree in pattern recognition and intelligent systems from the Institute of Automation, Chinese Academy of Sciences, Beijing, China, in 2014.

He is an Assistant Professor with the State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences. His current research interests include computer vision, graph algorithms, and robotics.

Zhi-Yong Liu (SM'15) received the B.S. degree from Tianjin University, Tianjin, China, in 1997, the M.E. degree from the Institute of Automation, Chinese Academy of Sciences, Beijing, China, in 2000, and the Ph.D. degree from the Chinese University of Hong Kong, Hong Kong, in 2003.

He is a Professor with the State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, where he was an Associate Professor, from 2005 to 2013, and has been a Professor, since 2013. He was a Post-Doctoral Fellow with the Chinese University of Hong Kong. He has published over 50 technical papers at prestigious international journals, including the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, IJCV, *Pattern Recognition*, and *Neural Computation*. His current research interests include machine learning, optimization technique, pattern recognition, computer vision, and graph theory.