Pattern Recognition Letters 55 (2015) 8–14

Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

Outlier robust point correspondence based on GNCCP ☆

Xu Yang, Hong Qiao, Zhi-Yong Liu*

State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China

ARTICLE INFO

Article history:
Received 7 March 2014
Available online 19 January 2015

Keywords:
Feature correspondence
Graph matching
GNCCP

ABSTRACT

Graph matching is a fundamental problem in pattern recognition and computer vision. In this paper we introduce a novel graph matching algorithm to find the specified number of best vertex assignments between two labeled weighted graphs. The problem is first explicitly formulated as the minimization of a quadratic objective function and then solved by an optimization algorithm based on the recently proposed graduated nonconvexity and concavity procedure (GNCCP). Simulations on both synthetic data and real world images witness the effectiveness of the proposed method.

1. Introduction and related works

Graph matching is a key problem in graph theory and combinatorial optimization. It also plays a central role in many pattern recognition (PR) and computer vision (CV) tasks such as feature correspondence [1], symbol retrieval [2], and object recognition [3]. Graph matching is well beyond point correspondence problem. For instance, graph matching can be generalized to approximately tackle the Markov random fields (MRF) inference problem [4–6]. But in this paper we will mainly focus on the point correspondence problem for comparison convenience. Note some inspirations provided in this paper are applicable in related problems.

A point set G can be represented by a graph $\mathcal{G}$, by representing the key points with vertices, and representing pairwise relations with edges. When further assigning a label, e.g. SIFT descriptor, to every vertex, and assigning a weight, e.g. distance, to every edge, the graph is known as the labeled weighted graph [7–9]. Note the labeled weighted graph is much the same with the widely known attributed relational graph (ARG) [10]. The only difference may be the nomenclature.

The correspondence between two point sets $G = \{g_i\}_{i=1}^{M}$ and $H = \{h_j\}_{j=1}^{N}$ is then equivalent to matching vertices in two graphs $\mathcal{G}$ and $\mathcal{H}$. When considering only the point-wise similarity, i.e. first-order similarity, the matching problem can be formulated as

$$\begin{aligned}\mathbf{x}^* &= \arg\min_{x} \sum_{i}^{M}\sum_{j}^{N} c_{ij}x_{ij} \\ &= \arg\min_{\mathbf{X}} \mathrm{tr}(\mathbf{C}^T\mathbf{X}) = \arg\min_{\mathbf{x}} \mathbf{x}^T\mathbf{c},\end{aligned}$$

$$\text{s.t. } x_{ij} \in \{0,1\}, \sum_{j}^{N} x_{ij} = 1, \sum_{i}^{M} x_{ij} \le 1, M \le N. \tag{1}$$

The real number $c_{ij} \in \mathbb{R}$ measures the dissimilarity between vertex i in G and vertex j in H. The integer $x_{ij} \in \{0,1\}$ denotes an assignment where $x_{ij} = 1$ indicates assigning vertex i in $\mathcal{G}$ to vertex j in $\mathcal{H}$. $\mathbf{C} = \{c_{ij}\}^{M\times N}$ and $\mathbf{X} = \{x_{ij}\}^{M\times N}$ are the matrix forms, and $\mathbf{c} \in \mathbb{R}^{MN}$ and $\mathbf{x} \in \{0,1\}^{MN}$ are the row-wise vectorized replicas of $\mathbf{C}$ and $\mathbf{X}$. The constraint $\sum_j^N x_{ij} = 1$, $\sum_i^M x_{ij} \le 1$ is known as the one-to-one constraint which is a usual assumption in graph matching problem [9]. And if it degenerates to $\sum_j^N x_{ij} = 1$, $\sum_i^M x_{ij} = 1$, $M = N$, $\mathbf{X}$ then degenerates to a permutation matrix [7,11,12]. The above problem (1) can be efficiently solved by interior point method [13], or by Hungarian algorithm in the permutation matrix situation [14].

However, when considering the pairwise constraints, i.e. second-order similarity, the matching problem becomes much more complicated, which is a typical NP-hard problem [15]. The approximate methods are thus necessary for efficiency reasons. These methods are usually divided into exact graph matching algorithms and inexact ones [15–17]. For comprehensive surveys the readers are referred to Refs. [15,18,19]. Simply speaking, the exact graph matching algorithms often relax the problem to specific types of graphs like trees, and then solve it by discrete optimization methods like branch and bound. These algorithms are suitable for applications demanding strict structure preservation, e.g. cheminformatics. However, they usually involve exponential complexity in the worst case, and are sensitive to noise and structure transformations. Thus it is inappropriate to apply them in PR and CV tasks because of noisy situations caused by object distortions, occlusions, etc. Therefore most research efforts in these two communities are devoted to the inexact methods which generally formulate the matching problem as the minimization of matching costs, and then find the local minimum by optimization techniques.

☆ This paper has been recommended for acceptance by R. Davies.

* Corresponding author. Tel.: +86 10 82544617.
E-mail addresses: xu.yang@ia.ac.cn (X. Yang), hong.qiao@ia.ac.cn (H. Qiao), zhiyong.liu@ia.ac.cn (Z.-Y. Liu).

http://dx.doi.org/10.1016/j.patrec.2014.12.011
0167-8655/

From the perspective of matching cost, or say the objective function, the inexact methods can be divided into the following two categories [9,20]:

The first category adopts the $F(A_G - \mathbf{X}A_H\mathbf{X}^T)$-like cost where $A_G \in \mathbb{R}^{M\times M}$ and $A_H \in \mathbb{R}^{N\times N}$ are adjacency matrices associated with $\mathcal{G}$ and $\mathcal{H}$ respectively. It is essentially a quadratic cost measure, corresponding to the second-order similarity. To minimize such an objective, Ref. [11] made use of eigendecompositions of A_G and A_H, which is regarded as the first spectral graph matching algorithm. Ref. [21] transformed this problem to the linear programming problem with the help of matrix l_1-norm $F(A_G - \mathbf{X}A_H\mathbf{X}) = \|A_G - \mathbf{X}A_H\mathbf{X}\|_1$ and the permutation matrix property

$$\mathbf{X}\mathbf{X}^T = \mathbf{X}^T\mathbf{X} = \mathbf{I}_{N\times N}. \tag{2}$$

Refs. [7,12] also utilized (2) yet set $F(A_G - \mathbf{X}A_H\mathbf{X}^T) = \|A_G - \mathbf{X}A_H\mathbf{X}^T\|_F^2$ where $\|\cdot\|_F$ is Frobenius matrix norm, and got a quadratic objective function. These methods exhibit some superiorities of this formulation such as storage saving. However, when $M \neq N$, Eq. (2) no longer holds, which leads to a fourth order cost function [8]. And furthermore, if only $L < \min(M, N)$ best assignments between G and H are needed, which makes sense when outliers exist in both images, the matching cost may be modified as $F(A_G \circ \mathbf{X}\mathbf{1}_{N\times N}\mathbf{X} - \mathbf{X}A_H\mathbf{X})$ and result in an even more complex fourth order objective function [22]. The higher order not only leads to a complicated gradient formulae but also increases computational costs. Besides, it is inconsistent with the quadratic nature of the problem.

The second category adopts the $\mathbf{x}^T\mathbf{K}\mathbf{x}$-like cost where $\mathbf{K} \in \mathbb{R}^{MN\times MN}$ (or $\mathbf{A} \in \mathbb{R}^{MN\times MN}$) is a dissimilarity matrix (affinity matrix) between $\mathcal{G}$ and $\mathcal{H}$. In spite of certain storage increase caused by the large $MN \times MN$ matrix $\mathbf{K}$, this matching cost benefits from the concise and general formulation by directly utilizing the quadratic form, where any kind of dissimilarity measure could be directly applied on $\mathbf{K}$. In the past decade, much attention has been paid to this formulation [9]. Ref. [23] proposed a famous spectral technique to efficiently solve the problem by computing the leading eigenvector of $\mathbf{A}$. Based on Ref. [23], Ref. [24] also utilized a spectral decomposition method which explicitly incorporated the one-to-one or one-to-many constraints. Other algorithms utilizing this formulation include the graduated assignment [25], dual decomposition [26,27], semidefinite programming [28], reweighted random walks [29,30], and path following [9]. However, the above methods (partly) suffer from the following drawbacks. The first one is the *winner-take-all projection*. Directly projecting the continuous solution back to the discrete domain by for example the Hungarian algorithm, may introduce significant additional error [7,12,31]. Second, only a few algorithms can deal with the *L best assignment* problem, and those which can usually first match all the vertices and then select the L best assignments by ranking techniques [14,23]. However, such a two-step scheme is not totally equivalent to the original problem, that is its optimal solution by optimally solving both steps may not be the best one for the *L best assignment* problem. Besides, some algorithms are limited to symmetric $\mathbf{K}$ ($\mathbf{A}$), e.g. the spectral decomposition based algorithms [14,23,31]. Symmetric $\mathbf{K}$ implies undirected graph matching, while it is known that direct graph matching is often more robust [12,20].

In this paper, we propose a graph matching algorithm applicable on $\mathbf{x}^T\mathbf{K}\mathbf{x}$ with the following novelties. First, by generalizing the graduated nonconvexity and concavity procedure (GNCCP), the algorithm could enjoy the advantages of graduated projection which has been shown to greatly improve the matching performance [7,8,31], and the applicability on either symmetric $\mathbf{K}$ or asymmetric $\mathbf{K}$. Second, the algorithm directly targets at the L best assignment problem, instead of a two-step scheme.

Below, the proposed method will be introduced in Section 2, and experimentally evaluated in Section 3. Finally Section 4 concludes this paper with the future work.

2. Proposed method

2.1. Formulation

With the notations in (1), the matching problem is formulated as

$$\begin{aligned} &\min \mathbf{x}^T\mathbf{K}\mathbf{x}, \\ &\text{s.t. } \mathbf{W}\mathbf{x} \le b, \quad \mathbf{1}_{MN}^T\mathbf{x} = L, \quad \mathbf{x} \in \{0,1\}^{MN}, \end{aligned} \tag{3}$$

where $\mathbf{W} = \begin{bmatrix} \mathbf{I}_M \otimes \mathbf{1}_N^T \\ \mathbf{1}_M^T \otimes \mathbf{I}_N \end{bmatrix}$, $\mathbf{b} = \begin{bmatrix} \mathbf{1}_M \\ \mathbf{1}_N \end{bmatrix}$ and $\mathbf{I}$ is the identity matrix. The sign $\otimes$ denotes Kronecker product of two matrices. The domain is denoted by $\mathcal{D}$. These constraints explicitly require only L best assignments should be selected. When $L = M \le N$, the problem degenerates to the subgraph matching problem, and when $L = M = N$, the problem degenerates to the equal-sized graph matching problem, i.e. graph matching on graphs with equal sizes.

The dissimilarity matrix $\mathbf{K}$ is constructed as

$$\begin{aligned} \mathbf{K}_{i',j'} &= \mathbf{K}_{(i_G-1)N+i_H,\ (j_G-1)N+j_H} \\ &= \begin{cases} d(i_G, i_H), & \text{if } i_G = j_G,\ i_H = j_H, \\ d(i_Gj_G, i_Hj_H), & \text{otherwise}, \end{cases} \end{aligned} \tag{4}$$

where the diagonal item $d(i_G, i_H)$ measures the vertex dissimilarity between vertices i_G and i_H, and the off-diagonal item $d(i_Gj_G, i_Hj_H)$ measures the dissimilarity between edges i_Gj_G and i_Hj_H. Note vertex dissimilarity is not always the same as in (1), because, by setting $d(i_G, i_H) = c_{i_Gi_H}$, here the vertex dissimilarity takes the form $\mathbf{x}^T\mathrm{diag}(\mathbf{c})\mathbf{x}$, while in (1) it takes the form $\mathbf{c}^T\mathbf{x}$. They are the same only when $\mathbf{x} \in \{0,1\}^{MN}$.

2.2. Optimization technique

To solve (3), we introduce an optimization method based on GNCCP. The GNCCP has been proved to exactly realize a type of the convex–concave relaxation procedure (CCRP) [7,12,32,33], but in a much simpler manner. It does not explicitly involve the convex or concave relaxation functions, which are usually difficult to construct [7,9,12] and greatly hinder the real applications of CCRP.

The GNCCP based optimization method takes the following form,

$$\begin{aligned} &\min F_\zeta(\mathbf{x}) = \begin{cases} (1-\zeta)\mathbf{x}^T\mathbf{K}\mathbf{x} + \zeta\mathbf{x}^T\mathbf{x} & \text{if } 1 \ge \zeta \ge 0, \\ (1+\zeta)\mathbf{x}^T\mathbf{K}\mathbf{x} + \zeta\mathbf{x}^T\mathbf{x} & \text{if } 0 > \zeta \ge -1, \end{cases} \\ &\text{s.t. } \mathbf{x} \in \mathcal{C}, \end{aligned} \tag{5}$$

where $\mathcal{C}$ is the convex hull of $\mathcal{D}$ based on Theorem 1.

Theorem 1. *The convex hull of $\mathcal{D}$ is $\mathcal{C}$, which is defined by,*

$$\mathcal{C} := \{\mathbf{x} | \mathbf{W}\mathbf{x} \le \mathbf{b}, \mathbf{1}_{MN}^T\mathbf{x} = L, \mathbf{x} \ge \mathbf{0}_{MN}\}$$

Proof. See Appendix 1 in the supplementary material for details. □

By gradually decreasing ζ from 1 to -1, F_ζ implicitly realizes the transmission from convex relaxation of (3) to the concave relaxation. Since the minima of the concave function over the convex hull locate in its extremes, the solution for (5) finally approaches $\mathcal{D}$.

For each specific ζ, (5) is solved by the Frank–Wolfe (FW) algorithm [34]—a powerful tool for nonlinear optimization, summarized in Algorithm 1.

In each iteration, we need to compute the optimal descent direction $\mathbf{y} - \mathbf{x}_{\text{old}}$ and the step size α. $\mathbf{y}$ could be obtained by linear programming techniques, e.g. interior point method [13], where the gradient $\nabla F_\zeta(\mathbf{x}_{\text{old}})$ is given by

$$\nabla F_\zeta(\mathbf{x}) = \begin{cases} (1-\zeta)(\mathbf{K}+\mathbf{K}^T)\mathbf{x} + 2\zeta\mathbf{x} & \text{if } 1 \ge \zeta \ge 0, \\ (1+\zeta)(\mathbf{K}+\mathbf{K}^T)\mathbf{x} + 2\zeta\mathbf{x} & \text{if } 0 > \zeta \ge -1, \end{cases} \tag{6}$$

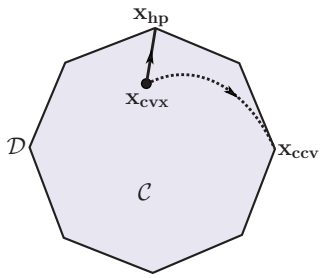

Fig. 1. Hard-cut projection and graduated projection. $\mathbf{x}_{cvx}$ is the minimum point of the convex relaxation. The solid line shows the hard-cut projection finds the nearest point $\mathbf{x}_{hp}$ to $\mathbf{x}_{cvx}$ in $\mathcal{D}$, while the dotted line shows the graduated projection follows a path from $\mathbf{x}_{cvx}$ gradually to $\mathbf{x}_{ccv}$.

Algorithm 1 Frank–Wolfe algorithm.

Given: $\mathbf{x}_{\text{old}}$
1: **repeat**
2: $\mathbf{y} = \arg\min \nabla F_\zeta(\mathbf{x}_{\text{old}})^T\mathbf{y}$, s.t.$\mathbf{y} \in \mathcal{C}$
3: $\alpha = \arg\min F_\zeta(\mathbf{x}_{\text{old}} + \alpha(\mathbf{y} - \mathbf{x}_{\text{old}}))$, s.t.$0 \le \alpha \le 1$
4: $\mathbf{x}_{\text{new}} = \mathbf{x}_{\text{old}} + \alpha(\mathbf{y} - \mathbf{x}_{\text{old}})$
5: $\mathbf{x}_{\text{old}} = \mathbf{x}_{\text{new}}$
6: **until** $\nabla F_\zeta(P)^T(\mathbf{x}_{\text{new}} - \mathbf{y}) < \varepsilon |F_\zeta(\mathbf{x}_{\text{new}}) + \nabla F_\zeta(\mathbf{x}_{\text{new}})^T(\mathbf{y} - \mathbf{x}_{\text{new}})|$, where ε is a small positive constant
Output: $\mathbf{x}_{\text{new}}$

If $\mathcal{G}$ and $\mathcal{H}$ are both undirected graphs, i.e. $\mathbf{K}$ is symmetric, $\nabla F_\zeta(\mathbf{x}_{\text{old}})$ is simply

$$\nabla F_\zeta(\mathbf{x}) = \begin{cases} 2(1-\zeta)\mathbf{K}\mathbf{x} + 2\zeta\mathbf{x} & \text{if } 1 \ge \zeta \ge 0, \\ 2(1+\zeta)\mathbf{K}\mathbf{x} + 2\zeta\mathbf{x} & \text{if } 0 > \zeta \ge -1. \end{cases} \tag{7}$$

The step α can be found by inexact line search, e.g. backtracking method [13].

Based on FW, the overall graph matching algorithm is summarized in Algorithm 2, where the step size $d\zeta$ is adjusted following the way in Ref. [7].

Discussion. With the help of GNCCP, the proposed method is able to deal with the L best assignment problem, and meanwhile it has rare limitations on $\mathbf{K}$. The reason why the graduated projection outperforms the winner-take-all projection is that the graduated projection is guided by the convex relaxation and concave relaxation, which both have the same optimal points with (3) for $\mathbf{x} \in \mathcal{D}$, while winner-take-all projection follows a hard-cut way without any cues from the original objective. The comparison is sketched in Fig. 1.

The complexity of the proposed method is mainly determined by the linear programming problem in FW as follows:

$$\begin{aligned} &\mathbf{y} = \arg\min \nabla F_\zeta(\mathbf{x}_{\text{old}})^T\mathbf{y}, \\ &\text{s.t. } \mathbf{y} \in \mathcal{D}. \end{aligned} \tag{8}$$

Algorithm 2 GNCCP based graph matching method.

Given: Two feature sets G and H
1: Initialize $\mathbf{x}_0 = 1_{MN}\frac{L}{MN}$, $\zeta = 1$
2: Construct the dissimilarity matrix $\mathbf{K}$ by (4)
3: $\mathbf{x}_{\text{old}} = \mathbf{x}_0$
4: **repeat**
5: FW algorithm
6: $\mathbf{x}_{\text{old}} = \mathbf{x}_{\text{new}}$
7: $\zeta = \zeta - d\zeta$
8: **until** $\zeta < -1 \vee \mathbf{x} \in \mathcal{D}$
Output: $\mathbf{x}_{\text{new}}$

As far as we know, interior point method involving $\mathcal{O}(M^3N^3)$ complexity is the most efficient method. To make it more efficient, as an alternative, the Hungarian algorithm [35] with $\mathcal{O}(M^2N)$ complexity is used to approximately solve (8), by first solving the following problem obtaining a partial permutation matrix containing M **1**s,

$$\begin{aligned} &\mathbf{y} = \arg\min \nabla F_\zeta(\mathbf{x}_{\text{old}})^T\mathbf{y}, \\ &\text{s.t. } (\mathbf{I}_M \otimes \mathbf{1}_N^T)\mathbf{x} = \mathbf{1}_M, (\mathbf{1}_M^T \otimes \mathbf{I}_N)\mathbf{x} \le \mathbf{1}_N, \mathbf{x} \ge 0, \end{aligned} \tag{9}$$

and then selecting L smallest items from them. Such an efficient version will be experimentally evaluated in the next section.

3. Experimental results

In this section, we apply the proposed method on both synthetic data and real world images. The algorithms included for comparison are listed below, which all utilize the $\mathbf{x}^T\mathbf{K}\mathbf{x}$ formulation.

Spectral method (SM) [23]. By abandoning all the original mapping constraints and adding the unary constraint $\mathbf{x}^T\mathbf{x} = 1$, the relaxed problem is equivalent to maximizing a Rayleigh quotient as follows:

$$\max \frac{\mathbf{x}^T\mathbf{A}\mathbf{x}}{\mathbf{x}^T\mathbf{x}},$$

which can be solved by finding the principal eigenvector of $\mathbf{A}$ where $\mathbf{A}$ is the affinity matrix.

Graduated assignment (GA) [25]. GA first relaxes $\mathbf{x}$ to a doubly stochastic matrix, and then solve the continuous optimization problem by a gradient descent method. It introduces a softassign strategy as $\mathbf{x}_i = \exp(\beta Q_i)$ where the Q_i is the partial gradient, which is followed by the Sinkhorn method (repeated row-wise and column-wise normalization) to normalize $\mathbf{x}$. By gradually adjusting $\beta_t = \alpha^{t-1}\beta_0$, $\mathbf{x}$ slowly moves towards the discrete domain. Here we set $\beta_0 = N$, $\beta_{\max} = 200$, $\alpha = 1.075$.

Reweighted random walks (RRWM) [29]. RRWM can be treated as a generalization of SM by incorporating mapping constraints into the spectral decomposition iterations. Specifically, in each iteration, it incorporates the Sinkhorn normalization by a *reweighted jump* in a similar way as the personalized Pagerank algorithm [36,37]. A final discretization is utilized to project the continuous solution to the discrete domain. We utilize the public coeds provided by Ref. [29] to implement RRWM.

Probabilistic spectral matching (PGM) [14]. PGM interprets $\mathbf{x}^T\mathbf{A}\mathbf{x}$ in a probabilistic way, and guides the continuous solution to discrete domain using a modified *power method* by iteratively updating $\mathbf{A}$ and $\mathbf{x}$. By treating x_i, $0 \le x_i \le 1$ as the probability for an assignment, the best L assignments are selected by ranking the probabilities.

Last, the proposed method is denoted by OUR and the efficient version is denoted by OURE.

To adapt SM, GA, RRWM, and PGM to the L best assignment problem, their continuous solutions $\mathbf{x}_c \in \mathcal{C}$ is back projected to $\mathbf{x}_d \in \mathcal{D}$ by linear programming techniques.

$$\begin{aligned} &\mathbf{x}_d = \arg\max \mathbf{x}_c^T\mathbf{x}_d, \\ &\text{s.t. } \mathbf{x}_d \in \mathcal{D}. \end{aligned}$$

Please note that in literature, the integer projected fixed point (IPFP) method [31,38] and factorized graph matching (FGM) [9] algorithm are also state-of-the-art graph matching algorithms. Similar to GNCCP, they also involve graduated projections from their continuous solutions to discrete domain. In this paper, they are not included for comparison because they in their current versions are only applicable on the subgraph matching problems, and are difficult to be generalized to the L best assignment problem.

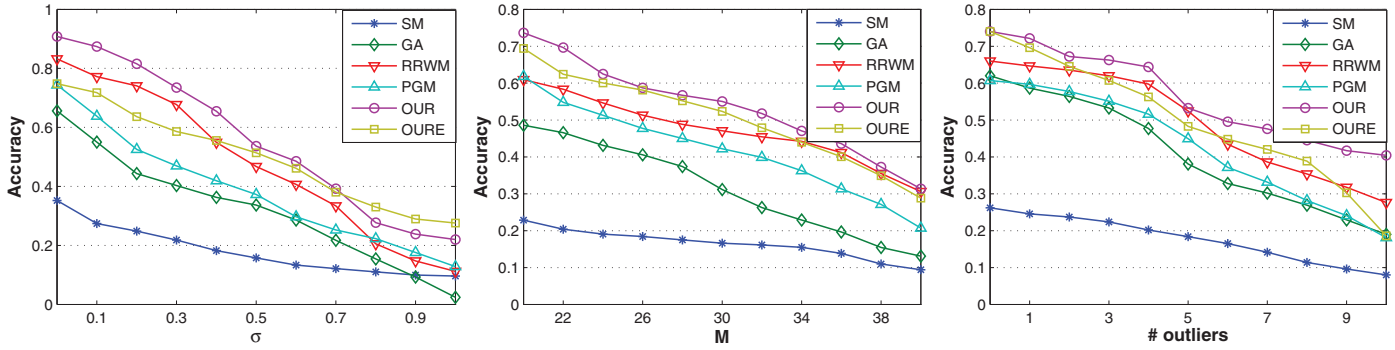

Fig. 2. Results on random synthetic points. The plots from left to right respectively illustrate the comparison results with respect to noise level, problem size, and outlier number.

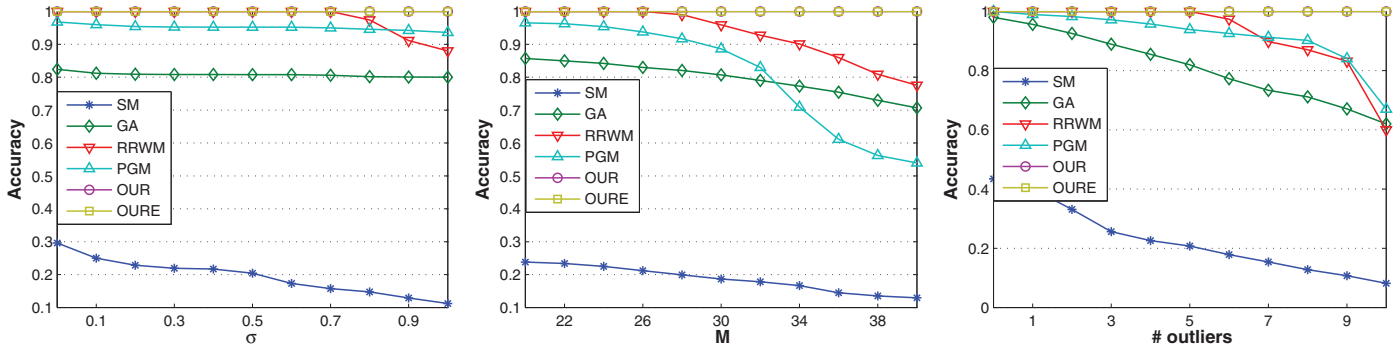

Fig. 3. Results on random synthetic adjacency matrices. The plots from left to right respectively illustrate the comparison results with respect to noise level, problem size, and outlier number.

3.1. Synthetic data

To evaluate the effectiveness and generality of the proposed method, two groups of experiments are carried out on synthetic data, with the first group to match randomly generated points, and the second group to match randomly generated adjacency matrices.

In the point matching group, we follow the experimental settings in previous works [14,25]. The graph is constructed as follows: first randomly generate two point sets $G = \{g_i\}_{i=1}^{M}$, $H = \{h_j\}_{j=1}^{N}$, $g_i, h_j \sim U[0,1]^{1\times 2}$ by uniform sampling, and randomly generate a discrete vector $\mathbf{x}^{gt} \in \mathcal{D}$ as the ground truth correspondence; second update G by

$$g_i = h_j + \eta, \qquad \eta \sim N(0, \sigma 1), \\ \text{if } \mathbf{x}^{gt}_{(i-1)N+j} = 1, \tag{10}$$

where η is the additive noise; last construct $\mathbf{K}$ as in (4).

In the adjacency matrix matching group, to evaluate the generality on arbitrary dissimilarity measure, $\mathbf{K}$ is built as follows: first randomly generate two adjacency matrices, $A_G, A_H \sim U[0,1]^{M\times N}$ and randomly generate $\mathbf{x}^{gt} \in \mathcal{D}$; second update A_G by

$$A_G(i_G, j_G) = A_H(i_H, j_H) + \rho, \qquad \rho \sim N(0, \sigma 2), \\ \text{if } \mathbf{x}^{gt}_{(i_G-1)N+i_H} = 1 \wedge \mathbf{x}^{gt}_{(j_G-1)N+j_H} = 1 \tag{11}$$

where ρ is the additive noise; last construct $\mathbf{K}$. Compared with the point matching, randomly generated adjacency matrices result in the *arbitrary dissimilarity measure*, where even the triangle inequality may not hold.

The accuracies of different algorithms are compared with respect to noise level, problem size and outlier number in both groups. The noise level is increased from $\sigma 1 = 0$ to $\sigma 1 = 0.1$, and from $\sigma 2 = 0$ to $\sigma 2 = 0.1$; the problem size is increased from $M = 20$ to $M = 40$ and $N = M + 5$; and the outlier number is increased from 0 to 10. When one variable takes effect, the other two are set to be the mean values. For instance, we set problem size $M = 30$ and outlier number $\#outliers = 5$ in the noise comparison.

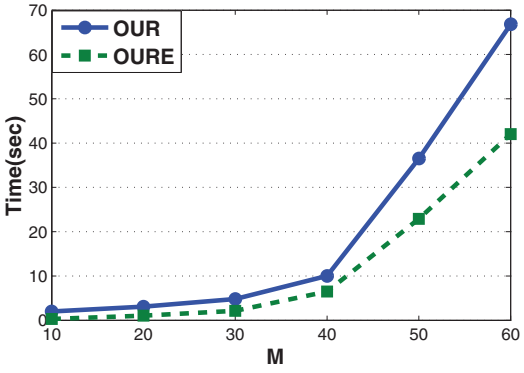

Fig. 4. Time cost comparison between OUR and OURE.

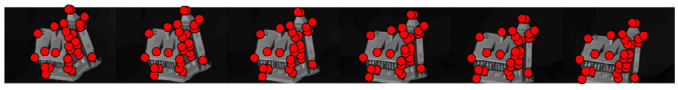

Fig. 5. Examples of House sequence.

The point matching results are shown in Fig. 2. Generally OUR outperforms other algorithms in the three comparisons. OUR outperforms OURE most of the time, except in the noise comparison, where OURE sometimes outperforming OUR may be owe to the inconsistency between objective value and accuracy with large noise, that is, a lower objective value may, by contrast, result in a lower accuracy. Besides, the accuracies of OUR and OURE are equal when the outlier number is 0 because OUR and OURE are equivalent when $L = M$. Fig. 3 shows the results when matching the adjacency matrices. The accuracies for GA, RRWM, PGM, OUR, and OURE are higher than in the

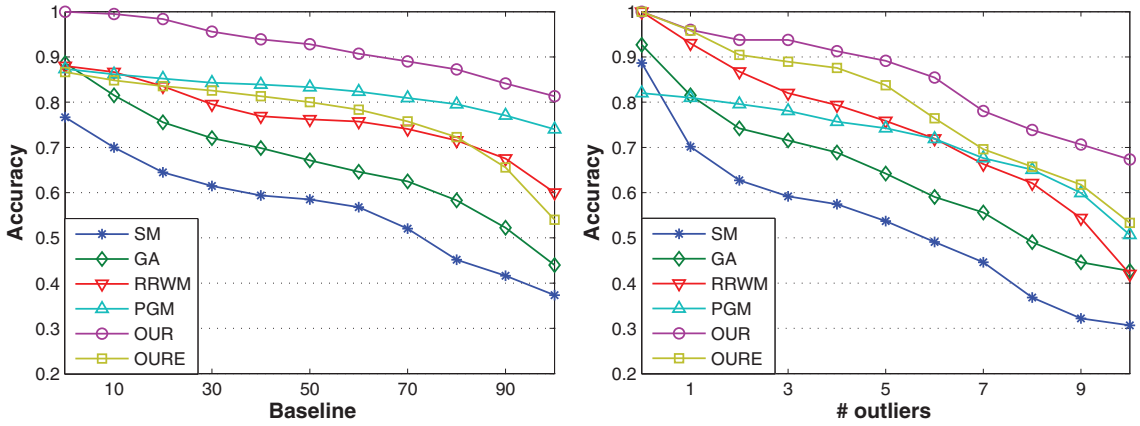

Fig. 6. Results on House sequence. The left and right plots respectively illustrate the comparison results with respect to baseline width and outlier number.

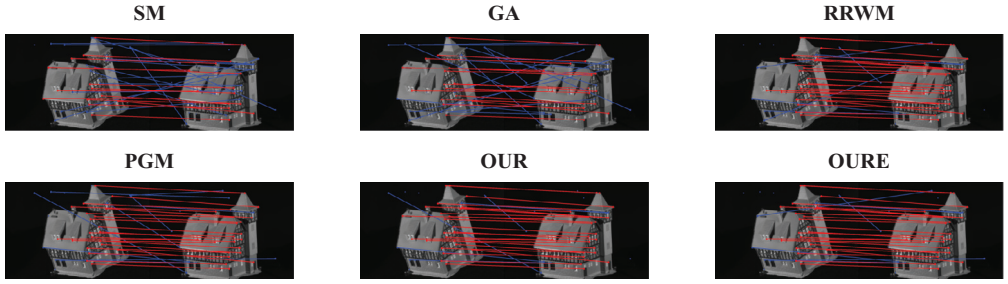

Fig. 7. Matching samples on House sequence. Red lines denote correct assignments, and blue lines denote incorrect assignments. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

point matching situation, because when $\sigma 2 = \sigma 1$, ρ brings in less uncertainty than η. That SM gets lower accuracy than in point matching situation is mainly attributed to its unsuitability to the L best assignment problem and the arbitrarily defined dissimilarity measure. The time cost comparison between OUR and OURE is shown in Fig. 4, where OURE brings in considerable time cost reduction.

3.2. House sequence

We also apply the proposed algorithm to the CMU House sequence[1] which is used in many works [7,9,14] as a benchmark dataset. The sequence contains 111 frames with a varying baseline. Each frame in the sequence is labeled with the same 30 landmark points as in Ref. [26], and some samples are shown in Fig. 5.

The accuracies are first compared as the baseline width is increased from 0 to 100, where larger baseline width implies more difficult matching. Then it is compared as the number of randomly sampled outliers increases from 0 to 10 given the 30 ground truth points. We set the outlier number to be 5 in the baseline comparison and set baseline width to be 50 in the outlier comparison. The result is quantitatively analyzed in Fig. 6, which witnesses the robust performance of the proposed method against geometric transformation and outliers. Some matching samples are shown in Fig. 7.

3.3. Handwritten Chinese character dataset

This experiment is to assess the affect of L selection, which is conducted on a handwritten Chinese character dataset. The dataset contains four characters with each one consisting of 10 samples fetched from Ref. [39], as illustrate in Fig. 8. For *Character 1–2*, we manually label $L = 28$ ground truth points, and for *Character 3–4* we manually label $L = 23$ ground truth points. Besides, for every character sample, we randomly label 10 outliers. In graph construction, the adjacency relations between points are built, by taking into account both character skeletons and randomness. First, edges between all vertices are randomly generated with 0.1 edge density. Second, edges between the inliers, i.e. ground truth points, which coincide with the character skeletons are generated. Finally, the structure is disturbed following a similar way as in Ref. [7] by randomly adding and removing $\frac{1}{2}\sigma \#Edge$ edges from each graph, where σ denotes the noise level set to be 0.1 and $\#Edge$ denotes the number of edges. Some examples with labeled points and constructed adjacency relations are illustrated by Fig. 8.

In implementation, we set $M = L + 5$ and $N = L + 10$, where L is the number of ground truth points. The parameter $\bar{L}$, instead of L as the algorithm input, is increased from $L - 5$ to $L + 5$ by a step size 1. For instance, for *Character 1–2*, there is $L = 28$, $M = 33$, $N = 38$, and $\bar{L}$ is increased from 23 to 33. Due to the varying $\bar{L}$, the only criterion *accuracy* is insufficient for result description. Hence we utilize two criterions, *accuracy* and *recall rate*. For each matching, the *accuracy* measures how many assignments are correct among all the assignments obtained by an algorithm; the *recall rate* measures how many assignments are correctly *recalled* given all the ground truth assignments. Specifically, they are respectively defined as follows:

$$\text{Accuracy} = \frac{\#\text{ correct assignments}}{\bar{L}} \tag{12}$$

$$\text{Recall rate} = \frac{\#\text{ correct assignments}}{L}, \tag{13}$$

where # correct assignments denote the number of correct assignments obtained by an algorithm.

The results are depicted in Fig. 9. We can observe that the accuracies and recall rates of OUR and OURE are more sensitive to $\bar{L}$, compared with the two-step algorithms. Specifically, as $\bar{L}$ increases,

[1] Available at http://vasc.ri.cmu.edu//idb/html/motion/house/index.html.

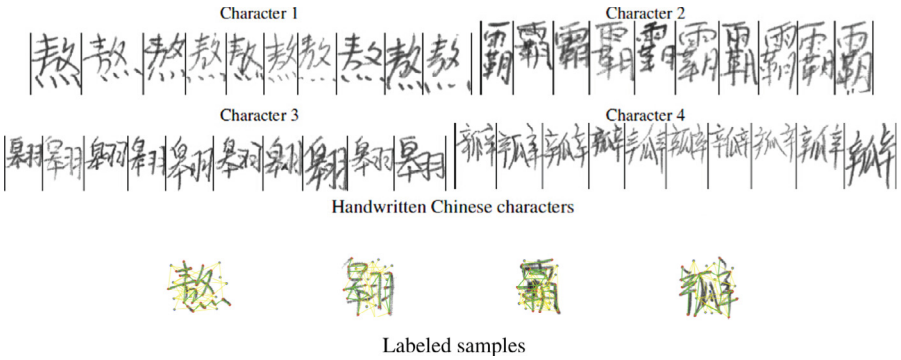

Fig. 8. Handwritten Chinese character dataset. Red points denote inliers and blue points denote outliers. Yellow lines denote edges randomly generated, and green lines denote edges coinciding with character skeletons. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

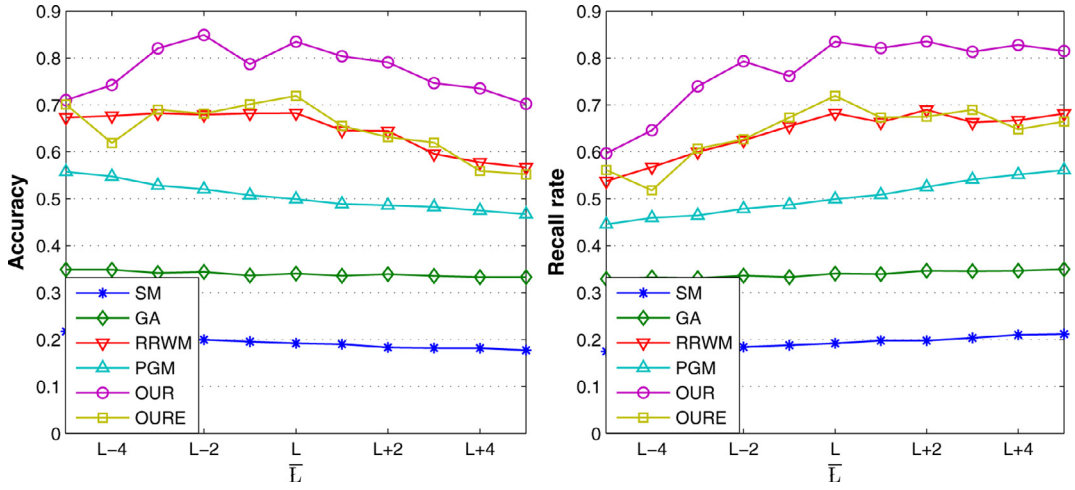

Fig. 9. Results on handwritten Chinese character dataset. The left and right plots respectively utilize accuracy and recall rate as criterions.

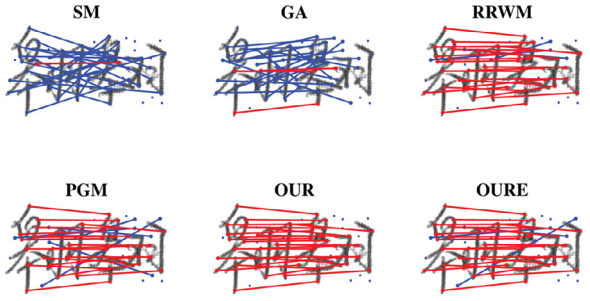

Fig. 10. Matching samples on handwritten Chinese character dataset. Red lines denote correct assignments, and blue lines denote incorrect assignments. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

when $\bar{L} \leq L$, the accuracies of OUR and OURE maintain in high levels while the recall rates increase; when $\bar{L} \geq L$, the accuracies decrease while the recall rates stop increasing, which may be due to the inevitable outlier matching.

Based on the above observations, in realistic applications, if an estimation $\tilde{L}$ of the inlier number is available, it is better to set $\bar{L}$ a bit smaller than $\tilde{L}$, to achieve a relative high accuracy with small recall rate deduction.

Some matching samples achieved by different algorithms are shown in Fig. 10.

4. Conclusion and future work

We propose a novel algorithm to find L best assignments between two point sets, by first formulating the problem as the minimization of $\mathbf{x}^T\mathbf{K}\mathbf{x}$ and then generalizing the GNCCP to approximately solve it. Moreover, an efficient version of the proposed method is introduced. Last, experiments on synthetic data and real world images validate the effectiveness of the proposed method.

One major limitation of the method is that L should be prespecified. Though the specification may be convenient in some computer applications [40], sometimes we want it to be automatically selected, which is our future work.

Acknowledgments

The authors would like to thank the associate editor and anonymous reviewers whose comments greatly improved the manuscript. This work was supported by the National Science Foundation of China, China (NSFC) (grant nos. 61375005, 61033011, 61210009, 61101221) and by National Key Technology R&D Program, China (grant 2012BAI34B02).

Supplementary Material

Supplementary material associated with this article can be found, in the online version, at doi:http://dx.doi.org/10.1016/j.patrec.2014.12.011

References

[1] X. Yang, H. Qiao, Z.Y. Liu, Partial correspondence based on subgraph matching, Neurocomputing 122 (25) (2013) 193–197.
[2] A. Fischer, C.Y. Suen, V. Frinken, K. Riesen, H. Bunke, A fast matching algorithm for graph-based handwriting recognition, in: Graph-Based Representations in Pattern Recognition, 2013, pp. 194–203.
[3] P. Le Bodic, P. Héroux, S. Adam, Y. Lecourtier, An integer linear program for substitution-tolerant subgraph isomorphism and its use for symbol spotting in technical drawings, Pattern Recognit. 45 (12) (2012) 4214–4224.
[4] M. Leordeanu, M. Hebert, Efficient map approximation for dense energy functions, in: Proceedings of International Conference on Machine Learning, ACM, 2006, pp. 545–552.
[5] T. Cour, J.B. Shi, Solving markov random fields with spectral relaxation, in: International Conference on Artificial Intelligence and Statistics, 2007, pp. 75–82.
[6] Z.Y. Liu, H. Qiao, J.H. Su, MAP inference with MRF by graduated non-convexity and concavity procedure, in: Proceedings of International Conference on Neural Information Processing, 2014, pp. 404–412.
[7] M. Zaslavskiy, F. Bach, J.P. Vert, A path following algorithm for the graph matching problem, IEEE Trans. Pattern Anal. Mach. Intell. 31 (12) (2009) 2227–2242.
[8] Z.Y. Liu, H. Qiao, GNCCP-graduated nonconvexity and concavity procedure, IEEE Trans. Pattern Anal. Mach. Intell. 36 (6) (2014) 1258–1267.
[9] F. Zhou, F. De la Torre, Factorized graph matching, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2012, pp. 127–134.
[10] D.H. Kim, I.D. Yun, S.U. Lee, Attributed relational graph matching based on the nested assignment structure, Pattern Recognit. 43 (3) (2010) 914–928.
[11] S. Umeyama, An eigendecomposition approach to weighted graph matching problems, IEEE Trans. Pattern Anal. Mach. Intell. 10 (5) (1988) 695–703.
[12] Z.Y. Liu, H. Qiao, L. Xu, An extended path following algorithm for graph-matching problem, IEEE Trans. Pattern Anal. Mach. Intell. 34 (7) (2012) 1451–1456.
[13] S. Boyd, L. Vandenberghe, Convex Optimization, Cambridge University Press, 2004.
[14] A. Egozi, Y. Keller, H. Guterman, A probabilistic approach to spectral graph matching, IEEE Trans. Pattern Anal. Mach. Intell. 35 (1) (2013) 18–27.
[15] D. Conte, P. Foggia, C. Sansone, M. Vento, Thirty years of graph matching in pattern recognition, Int. J. Pattern Recognit. Artif. Intell. 18 (03) (2004) 265–298.
[16] P. Foggia, G. Percannella, M. Vento, Graph matching and learning in pattern recognition in the last 10 years, Int. J. Pattern Recognit. Artif. Intell. 28 (01) (2014).
[17] T. Caelli, S. Kosinov, An eigenspace projection clustering method for inexact graph matching, IEEE Trans. Pattern Anal. Mach. Intell. 26 (4) (2004) 515–519.
[18] D. Conte, P. Foggia, C. Sansone, M. Vento, How and why pattern recognition and computer vision applications use graphs, in: Applied Graph Theory in Computer Vision and Pattern Recognition, Studies in Computational Intelligence, Springer, Berlin/Heidelberg, 2007, pp. 85–135.
[19] M. Vento, A long trip in the charming world of graphs for pattern recognition, Pattern Recognit. 48 (2) (2015) 291–301.
[20] F. Zhou, F. De la Torre, Deformable graph matching, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2013, pp. 2922–2929.
[21] H.A. Almohamad, S.O. Duffuaa, A linear programming approach for the weighted graph matching problem, IEEE Trans. Pattern Anal. Mach. Intell. 15 (5) (1993) 522–525.
[22] X. Yang, H. Qiao, Z.Y. Liu, A weighted common subgraph matching algorithm, IEEE Trans. Neural Netw. Learn. Syst. Minor revision (2015), arXiv:1411.0763.
[23] M. Leordeanu, M. Hebert, A spectral technique for correspondence problems using pairwise constraints, in: Proceedings of IEEE International Conference on Computer Vision, 2005, pp. 1482–1489.
[24] T. Cour, P. Srinivasan, J.B. Shi, Balanced graph matching, in: Proceedings of Neural Information Processing Systems, 2006.
[25] S. Gold, A. Rangarajan, A graduated assignment algorithm for graph matching, IEEE Trans. Pattern Anal. Mach. Intell. 18 (4) (1996) 377–388.
[26] L. Torresani, V. Kolmogorov, C. Rother, A dual decomposition approach to feature correspondence, IEEE Trans. Pattern Anal. Mach. Intell. 35 (2) (2013) 259–271.
[27] S. Nowozin, C.H. Lampert, Structured learning and prediction in computer vision, Foundations Trends Comput. Graph. Vis. 6 (3–4) (2011) 185–365.
[28] C. Schellewald, C. Schnörr, Probabilistic subgraph matching based on convex relaxation, in: Proceedings of Energy Minimization Methods in Computer Vision and Pattern Recognition, 2005, pp. 171–186.
[29] M. Cho, J. Lee, K.M. Lee, Reweighted random walks for graph matching, in: Proceedings of European Conference on Computer Vision, 2010, pp. 492–505.
[30] M. Cho, K.M. Lee, Progressive graph matching: Making a move of graphs via probabilistic voting, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2012, pp. 398–405.
[31] M. Leordeanu, M. Hebert, R. Sukthankar, An integer projected fixed point method for graph matching and map inference, in: Proceedings of Advances in Neural Information Processing Systems, 2009, pp. 1114–1122.
[32] Z.Y. Liu, H. Qiao, A convex-concave relaxation procedure based subgraph matching algorithm, J. Mach. Learn. Res.: W&CP 25 (2012) 237–252.
[33] Z.Y. Liu, H. Qiao, X. Yang, S.C.H. Hoi, Graph matching by simplified convex-concave relaxation procedure, Int. J. Comput. Vis. 109 (3) (2014) 169–186.
[34] M. Frank, P. Wolfe, An algorithm for quadratic programming, Naval Res. Logistics Q. 3 (1–2) (1956) 95–110.
[35] H.W. Kuhn, The hungarian method for the assignment problem, Naval Res. Logistics Q. 2 (1–2) (1955) 83–97.
[36] T.H. Haveliwala, Topic-sensitive pagerank, in: Proceedings of International Conference on World Wide Web, 2002, pp. 517–526.
[37] G. Jeh, J. Widom, Scaling personalized web search, in: Proceedings of International Conference on World Wide Web, 2003, pp. 271–279.
[38] M. Leordeanu, R. Sukthankar, M. Hebert, Unsupervised learning for graph matching, Int. J. Comput. Vis. 96 (1) (2012) 28–45.
[39] C.L. Liu, F. Yin, D.H. Wang, Q.F. Wang, Casia online and offline Chinese handwriting databases, in: Proceedings of International Conference on Document Analysis and Recognition, 2011, pp. 37–41.
[40] J. Maciel, J. Costeira, Robust point correspondence by concave minimization, Image Vis. Comput. 20 (9) (2002) 683–690.