Dual Calibration Mechanism based $L_{2,p}$-Norm for Graph Matching

Yu-Feng Yu, *Member, IEEE*, Guoxia Xu, *Member, IEEE*, Ke-Kun Huang, *Member, IEEE*, Hu Zhu, *Member, IEEE*, Long Chen, *Member, IEEE* and Hao Wang, *Member, IEEE*

***Abstract*—Unbalanced geometric structure caused by variations with deformations, rotations and outliers is a critical issue that hinders correspondence establishment between image pairs in existing graph matching methods. To deal with this problem, in this work, we propose a dual calibration mechanism (DCM) for establishing feature points correspondence in graph matching. In specific, we embed two types of calibration modules in the graph matching, which model the correspondence relationship in point and edge respectively. The point calibration module performs unary alignment over points and the edge calibration module performs local structure alignment over edges. By performing the dual calibration, the feature points correspondence between two images with deformations and rotations variations can be obtained. To enhance the robustness of correspondence establishment, the $L_{2,p}$-norm is employed as the similarity metric in the proposed model, which is a flexible metric due to setting the different p values. Finally, we incorporate the dual calibration and $L_{2,p}$-norm based similarity metric into the graph matching model which can be optimized by an effective algorithm, and theoretically prove the convergence of the presented algorithm. Experimental results in the variety of graph matching tasks such as deformations, rotations and outliers evidence the competitive performance of the presented DCM model over the state-of-the-art approaches.**

***Index Terms*—Calibration mechanism, graph matching, similarity metric**

I. Introduction

Graph matching, which aims to establish correspondences between two geometrical graphs, is an important problem in computer vision and pattern recognition tasks such as object classification [1] [2], shape matching [3] [4] [5], surface registration [6] [7], target tracking [8] [9] [10] , person re-identification [11] and networks alignment [12]. Essentially, a graph structure consists of a set of vertices and a set of edges, in which the vertices represent the unary feature information, and the edges represent the local geometric structure relationship between points. Thus, given two graphs, the purpose of graph matching can be viewed as establishing correspondence between their vertices sets and keeping the consistency between the edges sets simultaneously. In general, the graph matching can be derived to the quadratic assignment problem (Lawler's QAP [13]) which is known as a NP-hard problem and difficult to obtain the global optimum due to the non-convexity of the objective function and the discreteness of the solution space. Therefore, many studies focus on finding approximate solutions or local optimum by relaxing the original problem.

Regarding how to find an acceptable approximate solution or a reasonable local optimum for the graph matching problem, we can divide the graph matching algorithms into three categories. The first category of algorithms use graph embedding methods to relax the objective functions of matching models. The main idea is that the graphs vertices can be mapped onto the feature space, and the mapped vertices in the feature space can be matched to approximately represent the matching of graphs vertices. For instance, Luo *et al.* [14] use EM algorithm and singular value decomposition and propose a structural graph matching model to solve this problem. In which the graphs with different levels of structural corruption can be matched. Bai *et al.* [15] initially embed the vertices of the structure graph into the low-dimensional Euclidean space by using the manifold learning method, and then adopt the semi-positive definite programming optimization algorithm to realize the matching between point sets in the low-dimensional Euclidean space. The method is easy to implement and has strong practicability in real application. However, it is not guaranteed to find the global optimal solution. Tang *et al.* [16] propose a general graph matching method based on the joint embedding model, in which a collaborative representation framework is employed to embed and match the graph vertices. Feng *et al.* [17] use a multiplicity matrix to model the spectral-multiplicity of the graph, and then propose an alternating optimization algorithm to solve the multiplicity matrix and the permutation matrix. In addition, many other embedding based methods such as equidistance embedding [18] [19], subpattern embedding [20] and prototype embedding [21] also have been widely used in graph matching. These methods embed the graphs vertices into the high-dimensional feature space, and then transform the graph matching problem into the points sets matching in the feature space.

The second category relaxes the binary constraint to provide an approximate model for graph matching. For example, Leordeanu *et al.* [22] propose a spectral relaxation-based model. It establishes a new assignment graph, where the points represent the potential matching relationships and the weights of the edges represent the pairwise consistency among the

Y. F. Yu is with the Department of Statistics, Guangzhou University, Guangzhou 510006, China and also with the Department of Computer and Information Science, University of Macau, Macau 999078, China (e-mail: yuyufeng220@163.com).

K. K. Huang is with the School of Mathematics, Jiaying University, Meizhou 514015, China (e-mail: kkcocoon@163.com).

H. Zhu is with the College of Telecommunication and Information Engineering, Nanjing University of Posts and Telecommunications, Nanjing 210003, China (e-mail: peter.hu.zhu@gmail.com).

L. Chen is with the Department of Computer and Information Science, University of Macau, Macau 999078, China (e-mail: longchen@um.edu.mo).

G. Xu and H. Wang are with the Department of Computer Science, Norwegian University of Science and Technology, 2815 Gjovik, Norway (Email: gxxu.re@gmail.com; hawa@ntnu.no).(Corresponding author: Guoxia Xu.)

potential matching relationships. The global optimal solution of the relaxation model is obtained by calculating the principal eigenvector from the affinity-matrix based on the assignment graph. Zhang *et al.* [23] relax the binary constraints to soft matching and propose a K-nearest neighbor pooling matching model. In which the graph matching problem is modeled as a quadratic function and can be solved by relaxing the integer and matching constraints. Nie *et al.* [24] use the high-order clique relations to develop a hyper-clique graph matching model. The proposed model can be optimized by the affinity-preserving reweighted random walks method and the obtained solution is further converted into a binary permutation matrix by Hungarian method. In [25], the matching problem of two given graphs is taken as the point sorting and selection problem, and a reweighted random walks algorithm is proposed to drive the point sorting and selection. In [26], authors adopt the alternating direction method of multipliers, and propose a decomposition model to address the graph matching involving constraints with arbitrary order and potentials. In addition, Jiang *et al.* [27] propose a Lagrangian relaxation-based graph matching model, in which the double random constraints of the graph matching problem are embedded into an objective function, and the multivariate multiplier optimization algorithm is used to solve the relaxation model. Khan *et al.* [28] perform a bidirectional uniform graph matrix sampling to deal with the scalability problem, and use the low-rank CUR decomposition to solve the correspondence of graphs. Egozi *et al.* [29] combine spectral relaxation and probability framework, and propose a probabilistic graph matching algorithm.

The third category methods are based on discrete model to directly search matching solutions in the discrete space. Lee *et al.* [30] utilize the spectral properties of affinity matrix and propose a data-driven Markov Chain Monte Carlo framework to solve the general graph matching problem. Yan *et al.* [31] propose a discrete model to deal with the hypergraph matching problem, in which the high-order assignment problem is reduced to a first-order linear assignment problem in the iterative process. In each iteration, the gradient assignment algorithm is used to find the optimal assignment matrix. In [32], Yan *et al.* design an adaptive relaxation mechanism to ensure that the proposed discrete model converges to a fixed solution. Suh *et al.* [33] propose a robust graph matching algorithm based on Sequence Monte Carlo framework which can effectively explore the solution space under the one-to-one matching constraint. In [34], Adamczewski *et al.* transform the graph matching problem into the equivalent weighted maximum clique problem of the corresponding association graph and propose a penalized association graph framework.

It is worth noting that these methods have excellent performance when dealing with the graph matching problem with balanced geometric structures. However, establishing correspondence between two graphs remains the challenge because of the unbalanced geometric structure caused by variations with deformations, rotations and outliers. Therefore, it is necessary to consider: how to propose a graph matching model to address these variations and enhance the matching performance. To deal with this problem, some works using transformation strategy are proposed such as the variants of iterative closest point [35] [36], Mobius transformation [37] and adaptive transformation [38]. However, the DCM differs from these transformation methods and we make up for their deficiency in the following aspects. 1) Chui *et al.* [35] adopt a mapping function to perform non-rigid transformation of points, and the thin-plate spline is used to fit the mapping function between two point sets. Coherent point drift (CPD) [36] first considers the alignment of two point sets as a probability density estimation problem and embeds the transformation into the log-likelihood function. These methods require a good initial estimate of the transformation and only use point features for graph matching. 2) The Mobius transformation in [37] is used as a relaxation strategy to obtain the simple estimation, but it restricts the unbalance characteristic of general graphs. 3) The adaptive transformation in [38] is introduced to graph matching from the perspective of functional representation. In which the graph matching is formulated as transformation from one point set to the space spanned by another point set, and the pairwise edge features of graphs are directly represented by unary point features. Therefore, it does not consider the important role of edge transformation in graph matching.

Motivated by the above limitations, we propose a dual calibration strategy, which models the correspondence relationship in point and edge respectively. It should be noted that the idea of calibration has been widely used in robotics and automation, such as lidar calibration [39] and camera calibration [40]. In [39], Jiao *et al.* make use of three linearly independent planar surfaces, and propose an automatic algorithm to calibrate dual lidars without any additional sensors and artificial landmarks. In [40], an effective and flexible calibration approach is proposed for dual-camera system. In which the space intersection method is adopted to calculate 3D coordinates and the reconstruction error is used to optimize the calibration parameters. Different from these calibration methods, we employ two calibration modules to perform unary alignment over points and local structure alignment over edges, respectively. The framework of calibration mechanism is listed in Fig. 1. In addition, robustness is a key issue in machine learning and computer vision. The robust similarity metric has been commonly used in many tasks such as feature extraction [41] and dimensional reduction [42], but it is rarely considered in graph matching problem. Thus, presenting a robust similarity metric is significant in the task of graph matching. In this work we propose a novel robust similarity metric to better establish correspondence between two geometrical graphs. In particular, we adopt the $L_{2,p}$-norm to replace L_2-norm as the similarity metric in the proposed graph matching algorithm. As a consequence, it can improve the robustness of the algorithm. Recently, the $L_{2,1}$-norm based methods such as [43] [41] [42] have been widely applied to computer vision and pattern recognition to improve the robustness, but they don't have sufficient flexibility to accommodate different types of data sources. Different from $L_{2,1}$-norm, $L_{2,p}$-norm can be effectively applied to different types of data sources due to the flexibility of setting p value. Finally, the $L_{2,p}$-norm based similarity metric and the dual calibration mechanism are incorporated into a joint framework. The main contributions of this work involve:

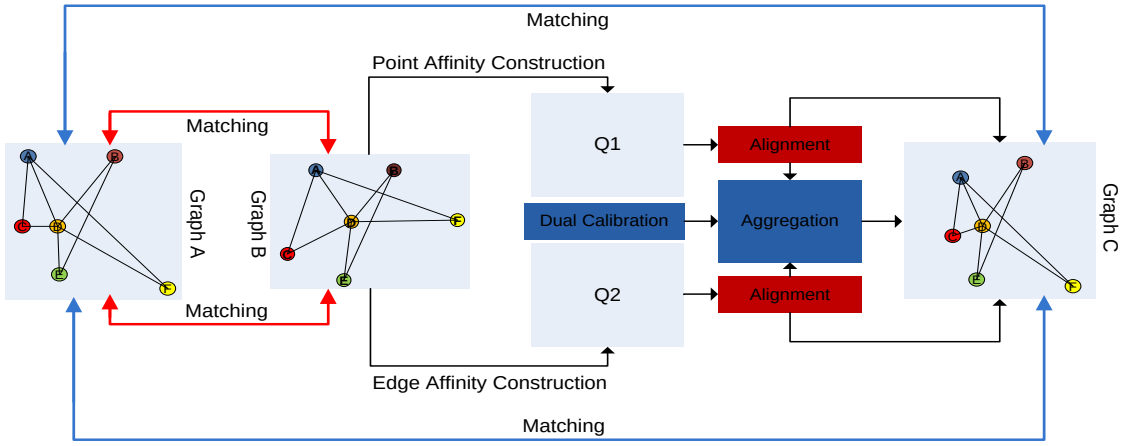


Fig. 1. The framework of calibration mechanism. Note that Graph A is an original image, and Graph B is a rotated and deformed image. It is not easy to directly establish the correspondence of feature points between Graph A and Graph B. We employ dual calibration modules on Graph B to perform unary alignment over points and local structure alignment over edges, then Graph B is transformed to Graph C. Obviously, establishing correspondence between Graph A and Graph C is easier than between Graph A and Graph B.

- A dual calibration strategy which performs unary alignment over points and local structure alignment over edges is proposed to deal with deformations and rotations for improving the performance of the algorithm.
- The similarity metric is no longer using L_2-norm, but instead $L_{2,p}$-norm is used in the proposed model to measure the points and edges similarities, which can improve the robustness of the algorithm in graph matching with unbalanced geometric structures.
- The $L_{2,p}$-norm based similarity metric and the dual calibration mechanism are incorporated into an objective function, and an efficient iterative algorithm is proposed to solve the objective function.

The remainder of this paper is organized as follows. Section II introduces the notations and states the graph matching problem. Section III presents the proposed dual calibration model for graph matching, and derives an efficient algorithm to optimize the model. In Section IV, the proposed model is evaluated on the benchmark databases of both synthetic and real-word images. Finally, we draw the conclusion and discuss the future work in Section V.

II. Problem Statement

In this paper, the column vector is represented by bold lowercase letter, e.g. $\mathbf{x}$. The matrix is represented by bold uppercase letter, e.g. $\mathbf{X}$. The i-th column of matrix $\mathbf{X}$ is represented as $\mathbf{x}_i$, the i-th row and j-th column element of matrix $\mathbf{X}$ is represented as $x_{i,j}$. $\mathbf{X}^T$ represents the transposed matrix of $\mathbf{X}$.

Given two graphs $\mathcal{G}_1 = (\mathcal{V}_1, \mathcal{E}_1, \mathbf{H}_1, \mathbf{F}_1)$ and $\mathcal{G}_2 = (\mathcal{V}_2, \mathcal{E}_2, \mathbf{H}_2, \mathbf{F}_2)$, extracted from the model and target images respectively. Here $\mathcal{V}_1$ and $\mathcal{V}_2$ are point sets, and the corresponding point numbers are n_1 and n_2, respectively. $\mathcal{E}_1$ and $\mathcal{E}_1$ are edge sets, and the number of edges are m_1 and m_2, respectively. $\mathbf{H}_1$, $\mathbf{H}_2$, $\mathbf{F}_1$, and $\mathbf{F}_2$ are four point-edge incidence matrices, which are defined as [44]. The goal of graph matching is to find the correspondence relationship of the point sets in $\mathcal{G}_1$ and $\mathcal{G}_2$. Supposing the elements of $\mathbf{X}$ are made up of 0 or 1, the graph matching problem can be converted to find an $n_1 \times n_2$ dimensional binary matching matrix $\mathbf{X}$. If point $v^1_{i_1} \in \mathcal{V}_1$ is matched with point $v^2_{i_2} \in \mathcal{V}_2$, then $x_{i_1 i_2} = 1$, otherwise $x_{i_1 i_2} = 0$. On the other hand, assuming the unary similarity of a point pair is $s^{\mathcal{V}}_{i_1 i_2}$ and the similarity of an edge pair is $s^{\mathcal{E}}_{k_1 k_2}$, then the graph matching problem can be formulated as the maximization of the following objection function:

$$\begin{aligned} J(\mathbf{X}) &= \sum_{i_1,i_2} s^{\mathcal{V}}_{i_1 i_2} x_{i_1 i_2} + \sum_{\substack{i_1 \neq j_1, i_2 \neq j_2 \\ a^1_{i_1 k_1} b^1_{j_1 k_1} = 1 \\ a^2_{i_2 k_2} b^2_{j_2 k_2} = 1}} s^{\mathcal{E}}_{k_1 k_2} x_{i_1 i_2} x_{j_1 j_2} \\ &= \mathbf{x}^T \mathbf{S} \mathbf{x}, \end{aligned} \tag{1}$$

$$\text{s.t.} \begin{cases} \sum_{i_1}^{n_1} x_{i_1 i_2} \leq 1, & \forall\ i_2 = 1, 2, \cdots, n_2 \\ \sum_{i_2}^{n_2} x_{i_1 i_2} = 1, & \forall\ i_1 = 1, 2, \cdots, n_1 \\ x_{i_1 i_2} \in \{0, 1\}, & \forall\ i_1, i_2 \end{cases}$$

where $\mathbf{x}$ denotes the vectorized replica of the matching matrix $\mathbf{X}$, and the matrix $\mathbf{S} \in \mathbb{R}^{n_1 n_2 \times n_1 n_2}$ represents the global similarity, which is calculated as:

$$s_{i_1 i_2 j_1 j_2} = \begin{cases} s^{\mathcal{V}}_{i_1 i_2}, & \text{if } i_1 = j_1 \text{ and } i_2 = j_2 \\ s^{\mathcal{E}}_{k_1 k_2}, & \text{if } \substack{i_1 \neq j_1, i_2 \neq j_2 \text{ and} \\ h^1_{i_1 k_1} f^1_{j_1 k_1} h^2_{i_2 k_2} f^2_{j_2 k_2} = 1} \\ 0, & \text{otherwise} \end{cases}$$

Generally, the model in (1) is regarded as a quadratic assignment problem [13] [45], and it is a NP-hard problem due to the combinatorial property of $x_{i_1 i_2} \in \{0, 1\}$. To deal with the problem, many methods based relaxation strategies [25] [46] are proposed to seek an approximate solution in the continuous domain.

Note that this graph matching model has high computational cost when addressing the matching problem of large graphs. To handle this problem, Zhou *et al* [47] divide the global similarity matrix $\mathbf{S}$ into six small/sparse matrices, which are unary and edge-pair similarity matrices and four point-edge incidence

matrices. The factorization can be computed as:

$$\mathbf{S} = \text{diag}(\text{vec}(\mathbf{S}_{\mathcal{V}})) + (\mathbf{H}_2 \otimes \mathbf{H}_1)\text{diag}(\text{vec}(\mathbf{S}_{\mathcal{E}}))(\mathbf{F}_2 \otimes \mathbf{F}_1)^T, \tag{2}$$

here $\otimes$ is the Kronecker product and vec$(\cdot)$ denotes the vector form of a matrix. $\mathbf{S}_{\mathcal{V}} \in \mathbb{R}^{n_1 \times n_2}$ represents the similarity matrix of point pairs, and the element of the i_1-th row and the i_2-th column is $s^{\mathcal{V}}_{i_1 i_2}$. $\mathbf{S}_{\mathcal{E}} \in \mathbb{R}^{m_1 \times m_2}$ denotes similarity matrix of edges, which is constructed by $s^{\mathcal{E}}_{k_1 k_2}$ ($k_1 = 1, 2, \cdots, m_1$ and $k_2 = 1, 2, \cdots, m_2$). Putting (2) into (1), the graph matching problem can be formulated with the following objective function:

$$\max_{\mathbf{X}} J(\mathbf{X}) = \text{tr}(\mathbf{S}_{\mathcal{V}}^T \mathbf{X}) + \text{tr}(\mathbf{S}_{\mathcal{E}}^T \mathbf{Y}) \tag{3}$$

$$\text{s.t.} \begin{cases} \sum_{i_1}^{n_1} x_{i_1 i_2} \leq 1, & \forall\ i_2 = 1, 2, \cdots, n_2 \\ \sum_{i_2}^{n_2} x_{i_1 i_2} = 1, & \forall\ i_1 = 1, 2, \cdots, n_1 \\ x_{i_1 i_2} \in \{0, 1\}, & \forall\ i_1, i_2 \end{cases}$$

here $\mathbf{Y} = (\mathbf{H}_1^T \mathbf{X} \mathbf{H}_2 \odot \mathbf{F}_1^T \mathbf{X} \mathbf{F}_2) \in \{0, 1\}^{m_1 \times m_2}$ is an auxiliary matrix. Each element of the matrix $y_{c_1 c_2}$ represents the correspondence relationship between the edges in graph $\mathcal{G}_1$ and graph $\mathcal{G}_2$. That is, if the c_1-th edge in $\mathcal{G}_1$ and the c_2-th edge in $\mathcal{G}_2$ are matched, $y_{c_1 c_2} = 1$, otherwise $y_{c_1 c_2} = 0$. The symbol $\odot$ is the Hadamard product.

III. Proposed joint framework and algorithm

In this section, we introduce the details about the proposed method for graph matching. The main content will be separated into the following several parts including dual calibration mechanism, an alternatively iterative algorithm, convergence and computational complexity analysis.

A. Dual Calibration Mechanism

As mentioned above, establishing correspondence relationship between two graphs with the unbalanced geometric structure caused by variations with deformations and rotations is difficult. In addition, the task of graph matching suffering from outliers is also a challenge. Using the model (1) and (3) directly to handle these graph matching problems is limited. Therefore, in this section, we propose a dual calibration strategy to reduce the influence of deformations and rotations, and employ the joint $L_{2,p}$-norm to measure the points and edges similarities for improving the robustness.

We assume $\mathbf{V}_1 = [\mathbf{v}_1^1, \mathbf{v}_2^1, \cdots, \mathbf{v}_{n_1}^1]^T \in \mathbb{R}^{n_1 \times d}$ and $\mathbf{V}_2 = [\mathbf{v}_1^2, \mathbf{v}_2^2, \cdots, \mathbf{v}_{n_2}^2]^T \in \mathbb{R}^{n_2 \times d}$ are point feature matrices which are taken from point sets $\mathcal{V}_1$ and $\mathcal{V}_2$, respectively. $\mathbf{E}_1 = [\mathbf{e}_1^1, \mathbf{e}_2^1, \cdots, \mathbf{e}_{m_1}^1]^T \in \mathbb{R}^{m_1 \times d}$ and $\mathbf{E}_2 = [\mathbf{e}_1^2, \mathbf{e}_2^2, \cdots, \mathbf{e}_{m_2}^2]^T \in \mathbb{R}^{m_2 \times d}$ are edge feature matrices whose elements are taken from edge sets $\mathcal{E}_1$ and $\mathcal{E}_2$, respectively. In this paper, we consider the two calibration modules are $f_1(\cdot)$ and $f_2(\cdot)$, respectively. After performing calibration strategy, the point and edge features with deformations and rotations can be adjusted. The adjusted point and edge features are then computed as $f_1(\mathbf{v}) = \mathbf{v}\mathbf{W}_1 + \mathbf{p}_1$ and $f_2(\mathbf{e}) = \mathbf{e}\mathbf{W}_2 + \mathbf{p}_2$, respectively. Here $\mathbf{W}_1$ and $\mathbf{W}_2$ are two $d \times d$ dimensional calibration matrices, and $\mathbf{p}_1$ and $\mathbf{p}_2$ are two d dimensional vectors. Assume $\mathbf{Q}_1 = [\mathbf{W}_1; \mathbf{p}_1] \in \mathbb{R}^{d+1 \times d}$ and $\tilde{\mathbf{v}} = [\mathbf{v}, 1] \in \mathbb{R}^{1 \times d+1}$, then we have $f_1(\mathbf{v}) = \tilde{\mathbf{v}}\mathbf{Q}_1$. Similarly, we have $f_2(\mathbf{e}) = \tilde{\mathbf{e}}\mathbf{Q}_2$ if $\mathbf{Q}_2 = [\mathbf{W}_2; \mathbf{p}_2] \in \mathbb{R}^{d+1 \times d}$ and $\tilde{\mathbf{e}} = [\mathbf{e}, 1] \in \mathbb{R}^{1 \times d+1}$. This way, the point similarity $s^{\mathcal{V}}_{i_1 i_2}$ and edge similarity $s^{\mathcal{E}}_{k_1 k_2}$ can be defined as:

$$s^{\mathcal{V}}_{i_1 i_2} = \text{S}im\left(\mathbf{v}_{i_1}^1, f_1(\mathbf{v}_{i_2}^2)\right), \tag{4}$$

and

$$s^{E}_{k_1 k_2} = \text{S}im\left(\mathbf{e}_{k_1}^1, f_2(\mathbf{e}_{k_2}^2)\right). \tag{5}$$

Employing a robust similarity metric to achieve better performance is significant in graph matching. Generally, L_2-norm is used to measure the similarity. However, it is not robust to graph matching with unbalanced geometric structures. In addition, the $L_{2,1}$-norm has been widely used as the similarity metric due to the robustness, but it does not have sufficient flexibility to accommodate different types of data sources. Different from $L_{2,1}$-norm, $L_{2,p}$-norm can be effectively applied to different types of data sources due to the flexibility of setting p value. We therefore employ $L_{2,p}$-norm as the similarity metric, and define the point similarity and edge similarity as:

$$s^{\mathcal{V}}_{i_1 i_2} = -\|\mathbf{v}_{i_1}^1 - f_1(\mathbf{v}_{i_2}^2)\|_2^p, \tag{6}$$

and

$$s^{\mathcal{E}}_{k_1 k_2} = -\|\mathbf{e}_{k_1}^1 - f_2(\mathbf{e}_{k_2}^2)\|_2^p \tag{7}$$

where $0 < p \leq 2$. Combining (6), (7) and (3), we can develop a robust model as follows:

$$\begin{aligned} \max_{\mathbf{X}, \mathbf{Q}_1, \mathbf{Q}_2} J(\mathbf{X}, \mathbf{Q}_1, \mathbf{Q}_2) = & \sum_{i_1}^{n_1} \sum_{i_2}^{n_2} -\|\mathbf{v}_{i_1}^1 - \tilde{\mathbf{v}}_{i_2}^2 \mathbf{Q}_1\|_2^p x_{i_1 i_2} \\ & + \mu_1 \sum_{k_1}^{m_1} \sum_{k_2}^{m_2} -\|\mathbf{e}_{k_1}^1 - \tilde{\mathbf{e}}_{k_2}^2 \mathbf{Q}_2\|_2^p y_{k_1 k_2} \\ & + \mu_2 \langle \mathbf{1}, \mathbf{Y} \rangle \end{aligned} \tag{8}$$

$$\text{s.t.} \begin{cases} \sum_{i_1}^{n_1} x_{i_1 i_2} \leq 1, & \forall\ i_2 = 1, 2, \cdots, n_2 \\ \sum_{i_2}^{n_2} x_{i_1 i_2} = 1, & \forall\ i_1 = 1, 2, \cdots, n_1 \\ x_{i_1 i_2} \in \{0, 1\}, & \forall\ i_1, i_2 \end{cases}$$

here μ_1 is a trade-off parameter that controls the importance and influence between the point and edge consistency. μ_2 is a tuning parameter. The first term of the robust model in (8) aims to maximize the point similarity after performing point alignment. Combination of the second and third terms is to align the edges and exploit the correspondence relationship between the edges in the graph $\mathcal{G}_1$ and graph $\mathcal{G}_2$.

B. Optimization Algorithm

In this section, we develop a two steps iteration algorithm to solve the proposed graph matching model. First, we fix $\mathbf{Q}_1$ and $\mathbf{Q}_2$ to compute the matching matrix $\mathbf{X}$, then fix $\mathbf{X}$ to update calibration matrices $\mathbf{Q}_1$ and $\mathbf{Q}_2$. Iterating the two steps until the convergence of the proposed model is reached.

We first set the initial calibration matrices as $\mathbf{Q}_1 = \mathbf{I}_1$ and $\mathbf{Q}_2 = \mathbf{I}_2$. After fixing $\mathbf{Q}_1$ and $\mathbf{Q}_2$, and by the simple algebra, the proposed graph matching model in (8) becomes

$$\max_{\mathbf{X}} J(\mathbf{X}) = \text{tr}(\mathbf{S}_{\mathcal{V}}^T \mathbf{X}) + \mu_1 \text{tr}(\mathbf{S}_{\mathcal{E}}^T \mathbf{Y}) + \mu_2 \text{tr}(\mathbf{1}^T \mathbf{Y}) \tag{9}$$

$$\text{s.t.} \begin{cases} \sum_{i_1}^{n_1} x_{i_1 i_2} \leq 1, & \forall\ i_2 = 1, 2, \cdots, n_2 \\ \sum_{i_2}^{n_2} x_{i_1 i_2} = 1, & \forall\ i_1 = 1, 2, \cdots, n_1 \\ x_{i_1 i_2} \in \{0, 1\}, & \forall\ i_1, i_2 \end{cases}$$

As can be noticed, the objective function in (9) is non-convex. However, it can be relaxed to a convex-concave problem and then be optimized by the path-following algorithm presented in [47].

Conversely, when the matching matrix $\mathbf{X}$ is computed, $\langle \mathbf{1}, \mathbf{Y} \rangle$ and matching constraints are constants and can be ignored. The graph matching is then reduced to minimize the following function:

$$J(\mathbf{Q}_1, \mathbf{Q}_2) = \sum_{i_1}^{n_1}\sum_{i_2}^{n_2} \|\mathbf{v}_{i_1}^1 - \tilde{\mathbf{v}}_{i_2}^2 \mathbf{Q}_1\|_2^p x_{i_1 i_2} + \mu_1 \sum_{k_1}^{m_1}\sum_{k_2}^{m_2} \|\mathbf{e}_{k_1}^1 - \tilde{\mathbf{e}}_{k_2}^2 \mathbf{Q}_2\|_2^p y_{k_1 k_2}. \tag{10}$$

By simple algebra, we get

$$\begin{aligned}
&\sum_{i_1}^{n_1}\sum_{i_2}^{n_2} \|\mathbf{v}_{i_1}^1 - \tilde{\mathbf{v}}_{i_2}^2 \mathbf{Q}_1\|_2^2 \|\mathbf{v}_{i_1}^1 - \tilde{\mathbf{v}}_{i_2}^2 \mathbf{Q}_1\|_2^{p-2} x_{i_1 i_2} \\
&= \sum_{i_1}^{n_1}\sum_{i_2}^{n_2} \mathrm{tr}\Big[(\mathbf{v}_{i_1}^1 - \tilde{\mathbf{v}}_{i_2}^2 \mathbf{Q}_1)^T (\mathbf{v}_{i_1}^1 - \tilde{\mathbf{v}}_{i_2}^2 \mathbf{Q}_1)\Big] z_{i_1 i_2}^1 x_{i_1 i_2} \\
&= \sum_{i_1}^{n_1}\sum_{i_2}^{n_2} \mathrm{tr}\Big[{\mathbf{v}_{i_1}^1}^T \mathbf{v}_{i_1}^1 - 2\mathbf{Q}_1^T {\tilde{\mathbf{v}}_{i_2}^2}^T \mathbf{v}_{i_1}^1 \\
&\quad + \mathbf{Q}_1^T {\tilde{\mathbf{v}}_{i_2}^2}^T \tilde{\mathbf{v}}_{i_2}^2 \mathbf{Q}_1\Big] z_{i_1 i_2}^1 x_{i_1 i_2},
\end{aligned} \tag{11}$$

where $z_{i_1 i_2}^1 = \|\mathbf{v}_{i_1}^1 - \tilde{\mathbf{v}}_{i_2}^2 \mathbf{Q}_1\|_2^{p-2}$. Suppose $\mathbf{Z}_1$ is an $n_1 \times n_2$ dimensional matrix, and the element of the i_1-th row and the i_2-th column is $z_{i_1 i_2}^1$. Defining $\mathbf{K}_1 = \mathbf{Z}_1 \odot \mathbf{X}$. In addition, we also denote two diagonal matrices $\mathbf{A}_1$ and $\mathbf{B}_1$, of which the diagonal elements are computed as

$$a_{i_2 i_2}^1 = \sum_{i_1}^{n_1} z_{i_1 i_2}^1 x_{i_1 i_2}, \tag{12}$$

and

$$b_{i_1 i_1}^1 = \sum_{i_2}^{n_2} z_{i_1 i_2}^1 x_{i_1 i_2}. \tag{13}$$

Combining the above representations, we have

$$\begin{aligned}
&\sum_{i_1}^{n_1}\sum_{i_2}^{n_2} \|\mathbf{v}_{i_1}^1 - \tilde{\mathbf{v}}_{i_2}^2 \mathbf{Q}_1\|_2^p x_{i_1 i_2} \\
&= \mathrm{tr}\Big[\mathbf{V}_1^T \mathbf{B}_1 \mathbf{V}_1 - 2\mathbf{Q}_1^T {\tilde{\mathbf{V}}_2}^T \mathbf{K}_1^T \mathbf{V}_1 + \mathbf{Q}_1^T {\tilde{\mathbf{V}}_2}^T \mathbf{A}_1 \tilde{\mathbf{V}}_2 \mathbf{Q}_1\Big].
\end{aligned} \tag{14}$$

Similarly, we compute $z_{k_1 k_2}^2 = \|\mathbf{e}_{k_1}^1 - \tilde{\mathbf{e}}_{k_2}^2 \mathbf{Q}_2\|_2^{p-2}$ and assume $\mathbf{Z}_2$ is an $m_1 \times m_2$ dimensional matrix with elements $z_{k_1 k_2}^2$ ($k_1 = 1, \cdots, m_1; k_2 = 1, \cdots, m_2$). Moreover, we also define two diagonal matrices $\mathbf{A}_2$ and $\mathbf{B}_2$, of which the diagonal elements are computed as

$$a_{k_2 k_2}^2 = \sum_{k_1}^{m_1} z_{k_1 k_2}^2 y_{k_1 k_2}, \tag{15}$$

and

$$b_{k_1 k_1}^2 = \sum_{k_2}^{m_2} z_{k_1 k_2}^2 y_{k_1 k_2}. \tag{16}$$

Algorithm 1 DCM algorithm

Input: Two graphs $\mathcal{G}_1 = \{\mathcal{V}_1, \mathcal{E}_1, \mathbf{H}_1, \mathbf{F}_1\}$ and $\mathcal{G}_2 = \{\mathcal{V}_2, \mathcal{E}_2, \mathbf{H}_2, \mathbf{F}_2\}$.
Output: Matching matrix $\mathbf{X}$.
1: Initialize $\mathbf{Q}_1 = \mathbf{I}_1$ and $\mathbf{Q}_2 = \mathbf{I}_2$;
2:**while** Not convergent **do**
3: Update the matching matrix $\mathbf{X}$ in (9) via the path-following algorithm;
4: Calculate $\mathbf{Z}_1$, $\mathbf{Z}_2$, $\mathbf{A}_1$, $\mathbf{A}_2$, $\mathbf{K}_1$ and $\mathbf{K}_2$;
5: Calculate calibration matrices $\mathbf{Q}_1$ and $\mathbf{Q}_2$ using (19) and (20).
6: **end while**

With these definitions and denoting $\mathbf{K}_2 = \mathbf{Z}_2 \odot \mathbf{Y}$, we have

$$\begin{aligned}
&\sum_{k_1}^{m_1}\sum_{k_2}^{m_2} \|\mathbf{e}_{k_1}^1 - \tilde{\mathbf{e}}_{k_2}^2 \mathbf{Q}_2\|_2^p y_{k_1 k_2} \\
&= \mathrm{tr}\Big[\mathbf{E}_1^T \mathbf{B}_2 \mathbf{E}_1 - 2\mathbf{Q}_2^T {\tilde{\mathbf{E}}_2}^T \mathbf{K}_2^T \mathbf{E}_1 + \mathbf{Q}_2^T {\tilde{\mathbf{E}}_2}^T \mathbf{A}_2 \tilde{\mathbf{E}}_2 \mathbf{Q}_2\Big].
\end{aligned} \tag{17}$$

Putting (14) and (17) into (10), we have

$$\begin{aligned}
J(\mathbf{Q}_1, \mathbf{Q}_2) = \mathrm{tr}\Big[&\mathbf{V}_1^T \mathbf{B}_1 \mathbf{V}_1 - 2\mathbf{Q}_1^T {\tilde{\mathbf{V}}_2}^T \mathbf{K}_1^T \mathbf{V}_1 \\
&+ \mathbf{Q}_1^T {\tilde{\mathbf{V}}_2}^T \mathbf{A}_1 \tilde{\mathbf{V}}_2 \mathbf{Q}_1\Big] + \mu_1 \mathrm{tr}\Big[\mathbf{E}_1^T \mathbf{B}_2 \mathbf{E}_1 \\
&- 2\mathbf{Q}_2^T {\tilde{\mathbf{E}}_2}^T \mathbf{K}_2^T \mathbf{E}_1 + \mathbf{Q}_2^T {\tilde{\mathbf{E}}_2}^T \mathbf{A}_2 \tilde{\mathbf{E}}_2 \mathbf{Q}_2\Big].
\end{aligned} \tag{18}$$

Now we consider how to solve the calibration matrices $\mathbf{Q}_1$ and $\mathbf{Q}_2$ of the model in (18). We aim to get the calibration matrices $\mathbf{Q}_1$ and $\mathbf{Q}_2$ by minimizing the objective function. It can be seen that $\mathbf{K}_1$, $\mathbf{A}_1$, $\mathbf{B}_1$ are relative to $\mathbf{Q}_1$, and $\mathbf{K}_2$, $\mathbf{A}_2$, $\mathbf{B}_2$ are relative to $\mathbf{Q}_2$. Therefore, it does not have closed-form solution. Here we employ an alternatively updating strategy. First, we fix $\mathbf{Q}_2$ to update $\mathbf{Q}_1$. In this process, we utilize the last updated $\mathbf{Q}_1$ to compute $\mathbf{K}_1$, $\mathbf{A}_1$ and $\mathbf{B}_1$. That is, $\mathbf{K}_1^t$, $\mathbf{A}_1^t$ and $\mathbf{B}_1^t$ are known in the $(t+1)$-th iteration, then we can compute $\mathbf{Q}_1$ by minimizing the model (18). Conversely, we adopt the same strategy to update $\mathbf{Q}_2$. Finally, performing the derivative of $J(\mathbf{Q}_1,\mathbf{Q}_2)$ with respect to $\mathbf{Q}_1$ and $\mathbf{Q}_2$ as zero, respectively, we have

$$\mathbf{Q}_1 = ({\tilde{\mathbf{V}}_2}^T \mathbf{A}_1 \tilde{\mathbf{V}}_2)^{-1} ({\tilde{\mathbf{V}}_2}^T \mathbf{K}_1^T \mathbf{V}_1), \tag{19}$$

and

$$\mathbf{Q}_2 = ({\tilde{\mathbf{E}}_2}^T \mathbf{A}_2 \tilde{\mathbf{E}}_2)^{-1} ({\tilde{\mathbf{E}}_2}^T \mathbf{K}_2^T \mathbf{E}_1). \tag{20}$$

The pseudo code of optimizing the proposed graph matching model in (8) is listed in Algorithm 1.

C. Convergence Analysis

In this section, we prove the convergence of the proposed algorithm. First, we re-write the graph matching model and

have the following equivalent objective function:

$$\begin{aligned}
&J(\mathbf{X},\mathbf{Q}_1,\mathbf{Z}_1,\mathbf{Q}_2,\mathbf{Z}_2)\\
&=\sum_{i_1}^{n_1}\sum_{i_2}^{n_2}\mathrm{tr}\Big[(\mathbf{v}_{i_1}^1-\tilde{\mathbf{v}}_{i_2}^2\mathbf{Q}_1)^T z_{i_1i_2}^1 x_{i_1i_2}(\mathbf{v}_{i_1}^1-\tilde{\mathbf{v}}_{i_2}^2\mathbf{Q}_1)\Big]\\
&+\mu_1\sum_{k_1}^{m_1}\sum_{k_2}^{m_2}\mathrm{tr}\Big[(\mathbf{e}_{k_1}^1-\tilde{\mathbf{e}}_{k_2}^2\mathbf{Q}_2)^T z_{k_1k_2}^2 y_{k_1k_2}(\mathbf{e}_{k_1}^1-\tilde{\mathbf{e}}_{k_2}^2\mathbf{Q}_2)\Big]\\
&-\mu_2\mathrm{tr}(\mathbf{1}^T\mathbf{Y}).
\end{aligned}\tag{21}$$

Then we have

Proposition: At each iteration of Algorithm 1, we have

$$\begin{aligned}
&J(\mathbf{X}^{t+1},\mathbf{Q}_1^{t+1},\mathbf{Z}_1^{t+1},\mathbf{Q}_2^{t+1},\mathbf{Z}_2^{t+1})\\
&\le J(\mathbf{X}^{t},\mathbf{Q}_1^{t},\mathbf{Z}_1^{t},\mathbf{Q}_2^{t},\mathbf{Z}_2^{t}).
\end{aligned}\tag{22}$$

It means the value of $J(\mathbf{X},\mathbf{Q}_1,\mathbf{Z}_1,\mathbf{Q}_2,\mathbf{Z}_2)$ monotonically decreases along with the iteration.

Proof: Suppose the objective function is $J(\mathbf{X}^t,\mathbf{Q}_1^t,\mathbf{Z}_1^t,\mathbf{Q}_2^t,\mathbf{Z}_2^t)$ in the t-th iteration, after updating the matching matrix $\mathbf{X}$ in the $(t+1)$ iteration, we have the following inequality:

$$J(\mathbf{X}^{t+1},\mathbf{Q}_1^{t},\mathbf{Z}_1^{t},\mathbf{Q}_2^{t},\mathbf{Z}_2^{t})\le J(\mathbf{X}^{t},\mathbf{Q}_1^{t},\mathbf{Z}_1^{t},\mathbf{Q}_2^{t},\mathbf{Z}_2^{t}).\tag{23}$$

After obtaining $\mathbf{X}^{t+1}$, we can compute $\mathbf{Q}_1^{t+1}$ by minimizing the model in (18). Thus, we get

$$J(\mathbf{X}^{t+1},\mathbf{Q}_1^{t+1},\mathbf{Z}_1^{t},\mathbf{Q}_2^{t},\mathbf{Z}_2^{t})\le J(\mathbf{X}^{t+1},\mathbf{Q}_1^{t},\mathbf{Z}_1^{t},\mathbf{Q}_2^{t},\mathbf{Z}_2^{t}).\tag{24}$$

According to (21) and (24), we have $J(\mathbf{X}^{t+1},\mathbf{Q}_1^{t+1},\mathbf{Z}_1^{t})\le J(\mathbf{X}^{t+1},\mathbf{Q}_1^{t},\mathbf{Z}_1^{t})$. That is

$$\begin{aligned}
&\sum_{i_1}^{n_1}\sum_{i_2}^{n_2}\mathrm{tr}\Big[(\mathbf{v}_{i_1}^1-\tilde{\mathbf{v}}_{i_2}^2\mathbf{Q}_1^{t+1})^T z_{i_1i_2}^{1,t} x_{i_1i_2}^{t+1}(\mathbf{v}_{i_1}^1-\tilde{\mathbf{v}}_{i_2}^2\mathbf{Q}_1^{t+1})\Big]\\
&\le\sum_{i_1}^{n_1}\sum_{i_2}^{n_2}\mathrm{tr}\Big[(\mathbf{v}_{i_1}^1-\tilde{\mathbf{v}}_{i_2}^2\mathbf{Q}_1^{t})^T z_{i_1i_2}^{1,t} x_{i_1i_2}^{t+1}(\mathbf{v}_{i_1}^1-\tilde{\mathbf{v}}_{i_2}^2\mathbf{Q}_1^{t})\Big].
\end{aligned}\tag{25}$$

Substituting $z_{i_1i_2}^{1,t}$ by $\|\mathbf{v}_{i_1}^1-\tilde{\mathbf{v}}_{i_2}^2\mathbf{Q}_1^t\|_2^{p-2}$, (25) can be transformed to

$$\begin{aligned}
&\sum_{i_1}^{n_1}\sum_{i_2}^{n_2}\frac{\Big\|(\mathbf{v}_{i_1}^1-\tilde{\mathbf{v}}_{i_2}^2\mathbf{Q}_1^{t+1})\Big\|_2^2}{\Big\|(\mathbf{v}_{i_1}^1-\tilde{\mathbf{v}}_{i_2}^2\mathbf{Q}_1^{t})\Big\|_2^2}\Big\|(\mathbf{v}_{i_1}^1-\tilde{\mathbf{v}}_{i_2}^2\mathbf{Q}_1^{t})\Big\|_2^p x_{i_1i_2}^{t+1}\\
&\le\sum_{i_1}^{n_1}\sum_{i_2}^{n_2}\Big\|(\mathbf{v}_{i_1}^1-\tilde{\mathbf{v}}_{i_2}^2\mathbf{Q}_1^{t})\Big\|_2^p x_{i_1i_2}^{t+1}.
\end{aligned}\tag{26}$$

Based on [48], we have the property: for all $x>0$ and $0<p\le 2$, if $h(x)=x^p-\frac{p}{2}x^2+\frac{p}{2}-1$, then $h(x)\le 0$. Using the property and assuming $x=\Big\|(\mathbf{v}_{i_1}^1-\tilde{\mathbf{v}}_{i_2}^2\mathbf{Q}_1^{t+1})\Big\|_2/\Big\|(\mathbf{v}_{i_1}^1-\tilde{\mathbf{v}}_{i_2}^2\mathbf{Q}_1^{t})\Big\|_2$, we have

$$\frac{\Big\|(\mathbf{v}_{i_1}^1-\tilde{\mathbf{v}}_{i_2}^2\mathbf{Q}_1^{t+1})\Big\|_2^p}{\Big\|(\mathbf{v}_{i_1}^1-\tilde{\mathbf{v}}_{i_2}^2\mathbf{Q}_1^{t})\Big\|_2^p}-\frac{p}{2}\frac{\Big\|(\mathbf{v}_{i_1}^1-\tilde{\mathbf{v}}_{i_2}^2\mathbf{Q}_1^{t+1})\Big\|_2^2}{\Big\|(\mathbf{v}_{i_1}^1-\tilde{\mathbf{v}}_{i_2}^2\mathbf{Q}_1^{t})\Big\|_2^2}+\frac{p}{2}-1\le 0.\tag{27}$$

Taking a simple algebra, we get

$$\begin{aligned}
&\Big\|(\mathbf{v}_{i_1}^1-\tilde{\mathbf{v}}_{i_2}^2\mathbf{Q}_1^{t+1})\Big\|_2^p+(\frac{p}{2}-1)\Big\|(\mathbf{v}_{i_1}^1-\tilde{\mathbf{v}}_{i_2}^2\mathbf{Q}_1^{t})\Big\|_2^p\\
&\le\frac{p}{2}\frac{\Big\|(\mathbf{v}_{i_1}^1-\tilde{\mathbf{v}}_{i_2}^2\mathbf{Q}_1^{t+1})\Big\|_2^2}{\Big\|(\mathbf{v}_{i_1}^1-\tilde{\mathbf{v}}_{i_2}^2\mathbf{Q}_1^{t})\Big\|_2^2}\Big\|(\mathbf{v}_{i_1}^1-\tilde{\mathbf{v}}_{i_2}^2\mathbf{Q}_1^{t})\Big\|_2^p.
\end{aligned}\tag{28}$$

Multiplying $x_{i_1i_2}^{t+1}$ on both sides of (28), and summing for each i_1 and i_2, then we have

$$\begin{aligned}
&\sum_{i_1}^{n_1}\sum_{i_2}^{n_2}\Big\|(\mathbf{v}_{i_1}^1-\tilde{\mathbf{v}}_{i_2}^2\mathbf{Q}_1^{t+1})\Big\|_2^p x_{i_1i_2}^{t+1}\\
&+(\frac{p}{2}-1)\sum_{i_1}^{n_1}\sum_{i_2}^{n_2}\Big\|(\mathbf{v}_{i_1}^1-\tilde{\mathbf{v}}_{i_2}^2\mathbf{Q}_1^{t})\Big\|_2^p x_{i_1i_2}^{t+1}\\
&\le\frac{p}{2}\sum_{i_1}^{n_1}\sum_{i_2}^{n_2}\frac{\Big\|(\mathbf{v}_{i_1}^1-\tilde{\mathbf{v}}_{i_2}^2\mathbf{Q}_1^{t+1})\Big\|_2^2}{\Big\|(\mathbf{v}_{i_1}^1-\tilde{\mathbf{v}}_{i_2}^2\mathbf{Q}_1^{t})\Big\|_2^2}\Big\|(\mathbf{v}_{i_1}^1-\tilde{\mathbf{v}}_{i_2}^2\mathbf{Q}_1^{t})\Big\|_2^p x_{i_1i_2}^{t+1}.
\end{aligned}\tag{29}$$

Combining (26) and (29), and performing a simple transformation, we get

$$\begin{aligned}
&\sum_{i_1}^{n_1}\sum_{i_2}^{n_2}\Big\|(\mathbf{v}_{i_1}^1-\tilde{\mathbf{v}}_{i_2}^2\mathbf{Q}_1^{t+1})\Big\|_2^p x_{i_1i_2}^{t+1}\\
&\le\sum_{i_1}^{n_1}\sum_{i_2}^{n_2}\Big\|(\mathbf{v}_{i_1}^1-\tilde{\mathbf{v}}_{i_2}^2\mathbf{Q}_1^{t})\Big\|_2^p x_{i_1i_2}^{t+1}.
\end{aligned}\tag{30}$$

This indicates

$$\begin{aligned}
&\sum_{i_1}^{n_1}\sum_{i_2}^{n_2}\mathrm{tr}\Big[(\mathbf{v}_{i_1}^1-\tilde{\mathbf{v}}_{i_2}^2\mathbf{Q}_1^{t+1})^T z_{i_1i_2}^{1,t+1} x_{i_1i_2}^{t+1}(\mathbf{v}_{i_1}^1-\tilde{\mathbf{v}}_{i_2}^2\mathbf{Q}_1^{t+1})\Big]\\
&\le\sum_{i_1}^{n_1}\sum_{i_2}^{n_2}\mathrm{tr}\Big[(\mathbf{v}_{i_1}^1-\tilde{\mathbf{v}}_{i_2}^2\mathbf{Q}_1^{t})^T z_{i_1i_2}^{1,t} x_{i_1i_2}^{t+1}(\mathbf{v}_{i_1}^1-\tilde{\mathbf{v}}_{i_2}^2\mathbf{Q}_1^{t})\Big],
\end{aligned}\tag{31}$$

thus we have $J(\mathbf{X}^{t+1},\mathbf{Q}_1^{t+1},\mathbf{Z}_1^{t+1})\le J(\mathbf{X}^{t+1},\mathbf{Q}_1^{t},\mathbf{Z}_1^{t})$. Adding the second and third terms of (21) on both sides of (31), we get

$$\begin{aligned}
&J(\mathbf{X}^{t+1},\mathbf{Q}_1^{t+1},\mathbf{Z}_1^{t+1},\mathbf{Q}_2^{t},\mathbf{Z}_2^{t})\\
&\le J(\mathbf{X}^{t+1},\mathbf{Q}_1^{t},\mathbf{Z}_1^{t},\mathbf{Q}_2^{t},\mathbf{Z}_2^{t}).
\end{aligned}\tag{32}$$

Similarly, we compute $\mathbf{Q}_2^{t+1}$ by minimizing the objective function in (18). Thus, we get

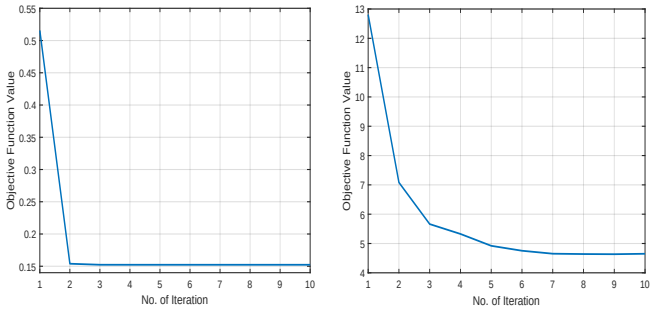
$$\begin{aligned}
&J(\mathbf{X}^{t+1},\mathbf{Q}_1^{t+1},\mathbf{Z}_1^{t+1},\mathbf{Q}_2^{t+1},\mathbf{Z}_2^{t})\\
&\le J(\mathbf{X}^{t+1},\mathbf{Q}_1^{t+1},\mathbf{Z}_1^{t+1},\mathbf{Q}_2^{t},\mathbf{Z}_2^{t}).
\end{aligned}\tag{33}$$

According to (33), and performing the same operations from (25) to (32), we have

$$\begin{aligned}
&J(\mathbf{X}^{t+1},\mathbf{Q}_1^{t+1},\mathbf{Z}_1^{t+1},\mathbf{Q}_2^{t+1},\mathbf{Z}_2^{t+1})\\
&\le J(\mathbf{X}^{t+1},\mathbf{Q}_1^{t+1},\mathbf{Z}_1^{t+1},\mathbf{Q}_2^{t},\mathbf{Z}_2^{t}).
\end{aligned}\tag{34}$$

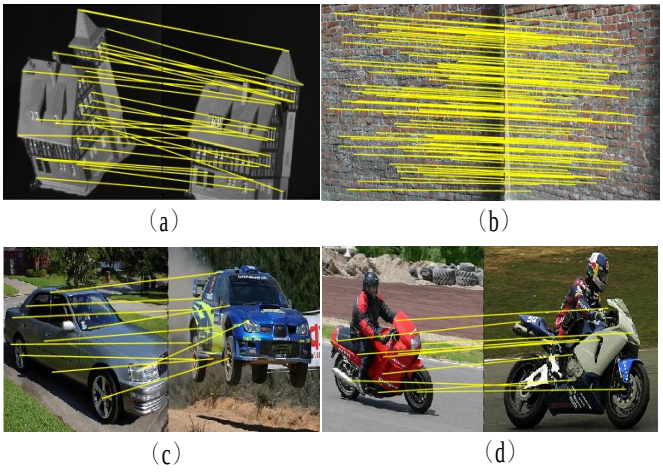
Combining (23), (32) and (34), we have

$$\begin{aligned}
&J(\mathbf{X}^{t+1},\mathbf{Q}_1^{t+1},\mathbf{Z}_1^{t+1},\mathbf{Q}_2^{t+1},\mathbf{Z}_2^{t+1})\\
&\le J(\mathbf{X}^{t},\mathbf{Q}_1^{t},\mathbf{Z}_1^{t},\mathbf{Q}_2^{t},\mathbf{Z}_2^{t}),
\end{aligned}\tag{35}$$



(a) (b)

Fig. 2. Convergence curves of the proposed DCM on the different graph matching databases. (a) CMU database: Hotel. (b) Affine Covariant Regions database: Gaffiti.



(a) (b) (c) (d)

Fig. 3. The examples of real images from CMU database, Affine Covariant Regions database, and WILLOW-ObjectClass database. (a) CMU hotel image pair: 30 points. (b) Affine Covariant Regions wall image pair: 130 points. (c) WILLOW-ObjectClass car image pair: 10 points. (d) WILLOW-ObjectClass motorbike image pair: 10 points.

which indicates that Algorithm 1 converges. ■

To test the convergence speed of DCM, we display the convergence curves on the CMU and Affine Covariant Regions databases in Fig. 2. It can be observed that the presented DCM converges quickly within less than 10 iterations.

D. Complexity Analysis

As stated in Algorithm 1, we update the matching matrix $\mathbf{X}$ in (9) via the path-following algorithm, which takes $O(T(\zeta + 2m_1m_2) + m_1m_2^2)$ operations. Here $\zeta = n_1^3$ if $n_1 > n_2$, otherwise $\zeta = n_2^3$, which is the time complexity of the Hungarian algorithm. In addition, n_1 and n_2 are the number of points in graph $\mathcal{G}_1$ and $\mathcal{G}_2$, m_1 and m_2 are the number of edges in graph $\mathcal{G}_1$ and $\mathcal{G}_2$, and T is the number of iterations. On the other hand, the main complexity is to calculate calibration matrices $\mathbf{Q}_1$ and $\mathbf{Q}_2$. It needs to update matrices $\tilde{\mathbf{V}}_2{}^T\mathbf{A}_1\tilde{\mathbf{V}}_2$, $\tilde{\mathbf{E}}_2{}^T\mathbf{A}_2\tilde{\mathbf{E}}_2$, $\tilde{\mathbf{V}}_2{}^T\mathbf{K}_1^T\mathbf{V}_1$ and $\tilde{\mathbf{E}}_2{}^T\mathbf{K}_2^T\mathbf{E}_1$. In which the time complexity is $O(d(n_2^2+m_2^2+n_1n_2+m_1m_2))$.

IV. Experimental Results

In this section, we design experiments on standard benchmark databases including synthetics database [49], Affine Covariant Regions database [50], WILLOW-ObjectClass database [1], and CMU-House/Hotel database [51]. Fig. 3 displays the examples of real images. The coordinate of each point is used as the point feature, and the coordinate difference of the connected points is used as the edge feature. The performance of the presented algorithm is evaluated by comparing the matching accuracy with the state-of-the-art matching approaches. We brief describe them as follows:

1) Graduated Assignment (GA) [52] is a graduated assignment algorithm for graph matching, which is fast and accurate. By combining graduated non-convexity, two-way (assignment) constraints, and sparsity, large improvements in accuracy and speed are achieved.
2) Re-weighted Random Walk Matching (RRWM) [25] introduces a random walk view on the graph matching problem and the core algorithm utilizes the point selection on an association graph to represent candidate correspondence between the two graphs. The solution is obtained by simulating random walks with re-weighting jumps enforcing the matching constraints on the association graph.
3) Integer Projected Fixed Point (IPFP) [53] can solve both graph matching and MAP inference efficiently. The solution is optimized in the quadratic score of discrete domain and shows an excellent result either by itself or by starting from the solution returned by any graph matching algorithms.
4) Spectral Matching (SM) [54] is an efficient spectral method for graph matching. It gets the optimal solution by using the principal eigenvectors of the adjacency matrix and imposing the mapping constraints required by the overall correspondence mapping.
5) Probabilistic Graph Matching (PM) [55] formalizes a soft matching criterion and algebraic relation between the hyper-edge weight matrix and the desired vertex-to-vertex probabilistic matching. It can scale naturally from graphs to hypergraphs.
6) Spectral Matching with Affine Constraints (SMAC) [56] uses a new spectral relaxation technique to get the approximate solution for matching problem.
7) Composition based Affinity Optimization Model (CAO) [57] elicits the affinity and consistency associated with inliers to design the outlier, and incorporates the affinity and consistency to address multi-graph matching problem. In our experiments, we employ the single step of multi-graph matching for the case of bipartite graph.
8) Robust Graph Matching (RGM) [43] embeds the joint transformation into the graph matching model to perform unary matching and local structure matching simultaneously, and the $L_{2,1}$-norm is used to measure the similarity.
9) Adaptive Discrete Graph Matching (DGA) [32] exploits the discrete domain by linear assignment approximation to iteratively update the solution and then devises an adaptive relaxation mechanism to jump out the degrading

case.

10) Factorized Graph Matching (FGM) [44] factorizes the large pairwise affinity matrix into smaller matrices that encode the local structure of each graph and the pairwise affinity between edges for reducing computational complexity. It has two versions, of which FGM-U only uses the undirected edge features to match graphs and FGM-D employs directed edge features for graph matching.
11) Alternating Direction Graph Matching (ADGM) [58] using the alternating direction method of multipliers develops a decomposition framework, which can solve graph matching involving the constraints of arbitrary order and arbitrary potentials.

A. Performance Evaluation on the Synthetic Database

In this part, we make a comparison between the proposed DCM model and the state-of-the-art methods on the synthetic database (the Chinese character 'blessing' and tropical fish image) [49]. Each image of Chinese character 'blessing' contains 105 points and tropical fish is composed of 98 points. To illustrate the robustness of our method under the influence of deformations and noise, we conduct two series of experiments to match points of each model image with points of the corresponding target image. In the first series, the target image is generated by adding Gaussian random noise with different levels to the original model image. That is $\mathbf{T} = \mathbf{M} + \sigma N(0, 1)$. Here $\mathbf{M}$ is the coordinate matrix of the points in the original model image and $\mathbf{T}$ is the coordinate matrix of the points in the target image. σ is a scale parametric. We set it as $\sigma \in \{0.01, 0.02, 0.03, 0.04, 0.05\}$ to generate five different noise levels. The examples of different noise levels are shown in Fig. 4, and the matching results are listed in Table I and Table II. In the second series, we follow the experimental setting in [59], and each model image is added the different levels of nonrigid deformations to generate the corresponding target image. The matching results are shown in Table III and Table IV.

As can be seen in Table I and Table II, it is reasonable that the accuracies decrease with the increasing of noise level. Nevertheless, taking an overall view of matching accuracies, the robustness of DCM is superior to other algorithms. What we can find from Table III and Table IV is that the presented DCM achieves higher accuracies than other algorithms on the database with different levels of deformation variations. Especially the performance in the Chinese character 'blessing' sequence. DCM achieves the best performance over all other algorithms in terms of matching accuracy. The main reason is that the proposed dual calibration strategy balances the structure information heuristically between unary and pair, and the metric learning leads to the robust correspondence relationship of graphs.

B. Performance Evaluation on the Affine Covariant Regions Database

In this section, we conduct experiments using wall sequence from the Affine Covariant Regions database [50] for evaluating the performance of the graph matching approaches. The wall sequence contains six images. The six images belong to a same object and have various image transformations such as rotation, scaling and stretching. In our experiment, we set the first one as the model and the rest as the targets. That is to say, this sequence is actually composed by five matching pairs. For each matching pair, we extract the feature point coordinates by using the VLFeat library [60] with the SIFT descriptor [61]. In which each image has 130 points in wall sequence. In the first row of Table V, we summarize the matching results of the presented DCM and other compared methods. One can observe that the performance of the DCM is much better than others, and the matching accuracy is 100.0%.

C. Performance Evaluation on the WILLOW-ObjectClass Database

In this section, we conduct experiments using car, duck, and motorbike sequences from the Willow-ObjectClass database [1] to evaluate the performance of the presented DCM. There are 50 duck images, 40 motorbike images and 40 car images. Following the setting in [62], we utilize ten manually labeled points in each image for test. Table V shows the average accuracies. We can see that the matching accuracies of the compared methods in duck sequence are lower than in motorbike and car sequences. The main reason is that duck sequence suffers from larger viewpoint variations than motorbike and car sequences, which leads establishing correspondence between two graphs is more difficult. In addition, one can see the proposed DCM obtains the good performance, and the accuracies are 99.0%, 91.6%, and 100.0%, respectively.

D. Performance Evaluation on the CMU-House/Hotel Database

The CMU House/Hotel dataset [51] has two sequences, of which house sequence consists of 111 frames and hotel sequence contains 101 frames. With the same setting in [26], we use 30 manually labeled points in each frame to test. The comparison methods are evaluated in two ways. First, we set frame 0 as the model image, and the rests as target images. We match all 30 points (no outliers) and compute the average accuracies by pairing the model with 10 targets, for example, frame 0 and frames 1-10, frame 0 and frames 11-20, and so on. Matching results are shown in Fig. 5 (a) and (b). It shows that the performance of the proposed DCM is superior to the other compared approaches. This is probably because we adopt the dual calibration strategy, which is effective to graph matching with rotation variation. In the second way, we choose N points as outliers and the rests as inliers. That is, for each pair, we match $30 - N$ points in the model image with 30 points in the target image. Matching results are displayed in Fig. 5 (c) and (d). It can be found that our DCM is more robust than the compared methods. The probable reason is that we employ $L_{2,p}$-norm to measure the points and edges similarities, which can reduce the influence of outliers.

E. Parameter Sensitivity Analysis

In this section, we evaluate the effect of the different parameters to the proposed DCM algorithm.

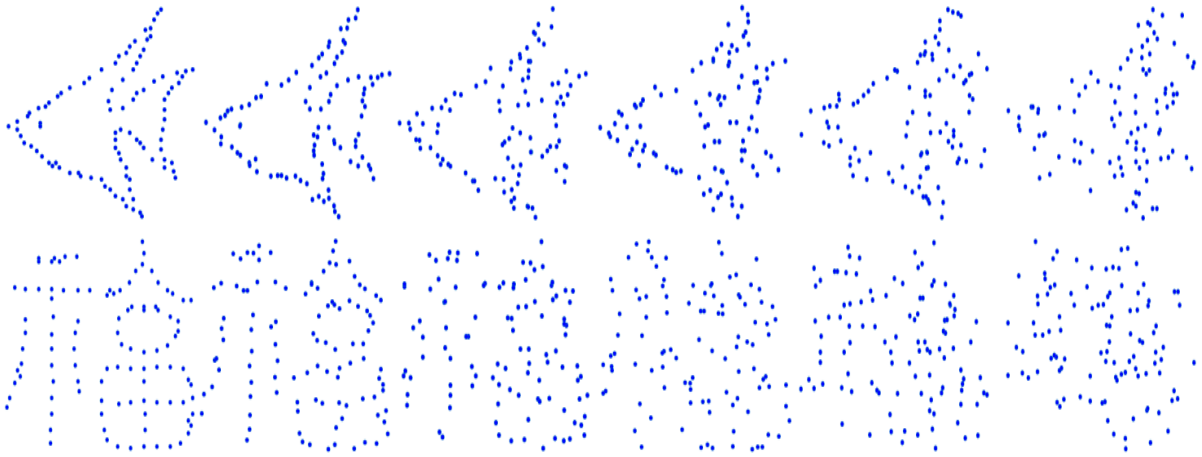


Fig. 4. The visual examples of different noise levels on the synthetic database. The first column represents original images. The second to sixth columns represent the images with five different noise levels, and there are level 1 to level 5, respectively.

TABLE I
MATCHING RESULTS OF ALL METHODS IN THE CHINESE CHARACTER BLESSING SEQUENCE WITH DIFFERENT LEVELS OF NOISE (%).

Method	SM	PM	GA	RRWM	IPFP-U	IPFP-S	SMAC	ADGM	FGM-U	FGM-D	RGM	DGA	CAO	DCM
Level 1	42.8	29.0	41.5	81.1	80.2	74.7	87.8	95.1	83.0	79.3	95.1	75.9	72.2	**96.4**
Level 2	22.2	12.1	22.5	43.4	40.3	34.6	50.1	63.6	43.2	40.7	77.9	35.8	27.6	**78.5**
Level 3	17.2	7.1	15.8	35.2	27.2	25.2	34.6	41.1	29.5	30.0	56.9	20.9	28.2	**58.1**
Level 4	12.4	4.0	12.4	23.9	22.1	19.7	23.4	27.7	22.7	23.1	40.3	16.1	16.6	**41.8**
Level 5	8.5	3.2	9.1	19.1	19.0	14.0	15.3	23.4	17.4	17.7	**31.1**	12.6	14.8	30.6

TABLE II
MATCHING RESULTS OF ALL METHODS IN THE TROPICAL FISH SEQUENCE WITH DIFFERENT LEVELS OF NOISE (%).

Method	SM	PM	GA	RRWM	IPFP-U	IPFP-S	SMAC	ADGM	FGM-U	FGM-D	RGM	DGA	CAO	DCM
Level 1	33.9	29.2	35.3	71.7	72.7	67.0	76.2	94.7	74.0	73.1	94.3	70.9	61.0	**96.3**
Level 2	16.5	10.1	22.4	43.6	39.1	27.6	51.7	64.3	45.4	40.8	72.7	33.2	31.8	**73.7**
Level 3	10.5	6.1	14.7	28.4	26.9	16.6	33.7	37.1	28.1	24.8	**53.6**	22.7	21.7	49.4
Level 4	10.1	5.4	12.4	21.8	20.6	15.0	26.5	22.9	21.9	22.4	**38.2**	15.0	18.3	37.9
Level 5	9.2	3.6	10.5	18.5	17.1	12.6	18.4	21.0	17.1	12.6	**29.7**	13.1	13.0	28.7

TABLE III
MATCHING RESULTS OF ALL METHODS IN THE CHINESE CHARACTER BLESSING SEQUENCE WITH DIFFERENT LEVELS OF DEFORMATION (%).

Method	SM	PM	GA	RRWM	IPFP-U	IPFP-S	SMAC	ADGM	FGM-U	FGM-D	RGM	DGA	CAO	DCM
Level 1	66.5	64.2	67.8	97.9	97.6	98.8	95.1	50.7	63.9	98.8	100.0	97.1	96.6	**100.0**
Level 2	33.2	37.2	41.9	82.2	84.6	71.7	63.1	20.0	11.4	61.8	89.9	72.6	72.4	**95.0**
Level 3	20.3	25.0	31.1	63.1	60.6	40.1	39.3	15.4	8.5	37.4	85.5	46.4	52.8	**89.7**
Level 4	17.8	27.5	34.6	66.4	67.6	58.3	41.3	14.3	15.6	46.6	70.4	55.5	59.3	**77.1**
Level 5	9.6	14.9	15.7	24.0	24.6	13.3	20.6	12.6	11.6	20.0	45.5	18.7	24.2	**45.7**

1) Parameter p: We set $p \in \{0.2, 0.4, \cdots, 1.8, 2.0\}$ and conduct the experiments on four databases to evaluate the performance of the proposed DCM. The matching results are shown in Fig.6 when choosing different values of the parameter p. We can obtain the best selections of p according to different databases and different matching tasks flexibly and adaptively. In particular, it could achieve the best performance when the values of p are set as 1.4 and 0.4 on the synthetic database with deformation variation and noise variation, respectively. The best results of the proposed DCM are obtained when $p = 1.0$ on affine covariant regions and willow-objectclass databases. The best performance can be achieved when $p \geq 1.2$ on the CMU-House/Hotel database.

2) Parameters μ_1 and μ_2: We further evaluate the effect of the different values of μ_1 and μ_2 on the matching accuracy. Here we empirically set $\mu_1, \mu_2 \in \{0.001, 0.01, 0.1, 1, 10, 100\}$ and conduct the experiments on four databases. The matching accuracies of the proposed DCM are shown in Fig. 7. According to the experimental results, one can see that the performance is not changeless to different values of parameters. In this paper we set the values of parameter pair $\{\mu_1, \mu_2\}$ as (0.01, 0.01), (0.1, 0.1), (0.01, 0.01) and (1, 0.1) for Synthetic database, WILLOW-ObjectClass database, Affine Covariant Regions database, and CMU-House/Hotel database.

TABLE IV
MATCHING RESULTS OF ALL METHODS IN THE TROPICAL FISH SEQUENCE WITH DIFFERENT LEVELS OF DEFORMATION (%).

Method	SM	PM	GA	RRWM	IPFP-U	IPFP-S	SMAC	ADGM	FGM-U	FGM-D	RGM	DGA	CAO	DCM
Level 1	88.7	84.0	86.5	100.0	100.0	98.8	99.7	86.5	100.0	100.0	100.0	100.0	100.0	**100.0**
Level 2	67.9	64.0	68.8	99.7	97.9	98.3	90.9	46.4	78.8	98.9	100.0	97.7	91.4	**100.0**
Level 3	47.1	51.4	55.4	91.2	87.6	76.9	77.9	33.0	50.6	76.9	98.8	90.8	88.2	**100.0**
Level 4	40.0	46.4	50.2	85.3	85.7	75.8	67.4	27.8	45.3	81.1	**99.8**	79.4	77.1	99.3
Level 5	24.4	31.1	36.6	73.6	73.4	55.9	46.1	17.1	19.1	63.1	**99.5**	57.7	56.3	97.2

TABLE V
MATCHING RESULTS ON THE AFFINE COVARIANT REGIONS DATABASE AND WILLOW-OBJECTCLASS DATABASE (%).

Method	SM	PM	GA	RRWM	IPFP-U	IPFP-S	SMAC	ADGM	FGM-U	FGM-D	RGM	DGA	CAO	DCM
Wall	66.6	58.5	67.1	97.9	95.7	91.4	84.0	53.8	93.6	94.0	100.0	93.6	95.2	**100.0**
Duck	59.6	49.6	71.6	69.2	68.8	64.8	71.2	86.0	75.6	67.2	88.8	60.4	72.0	**91.6**
Car	79.5	67.0	75.0	83.5	82.0	80.5	85.5	99.0	89.0	89.0	99.0	75.5	84.5	**99.0**
Motorbike	87.0	77.5	88.0	99.0	96.0	97.0	92.5	100.0	99.0	96.5	100.0	91.5	99.0	**100.0**

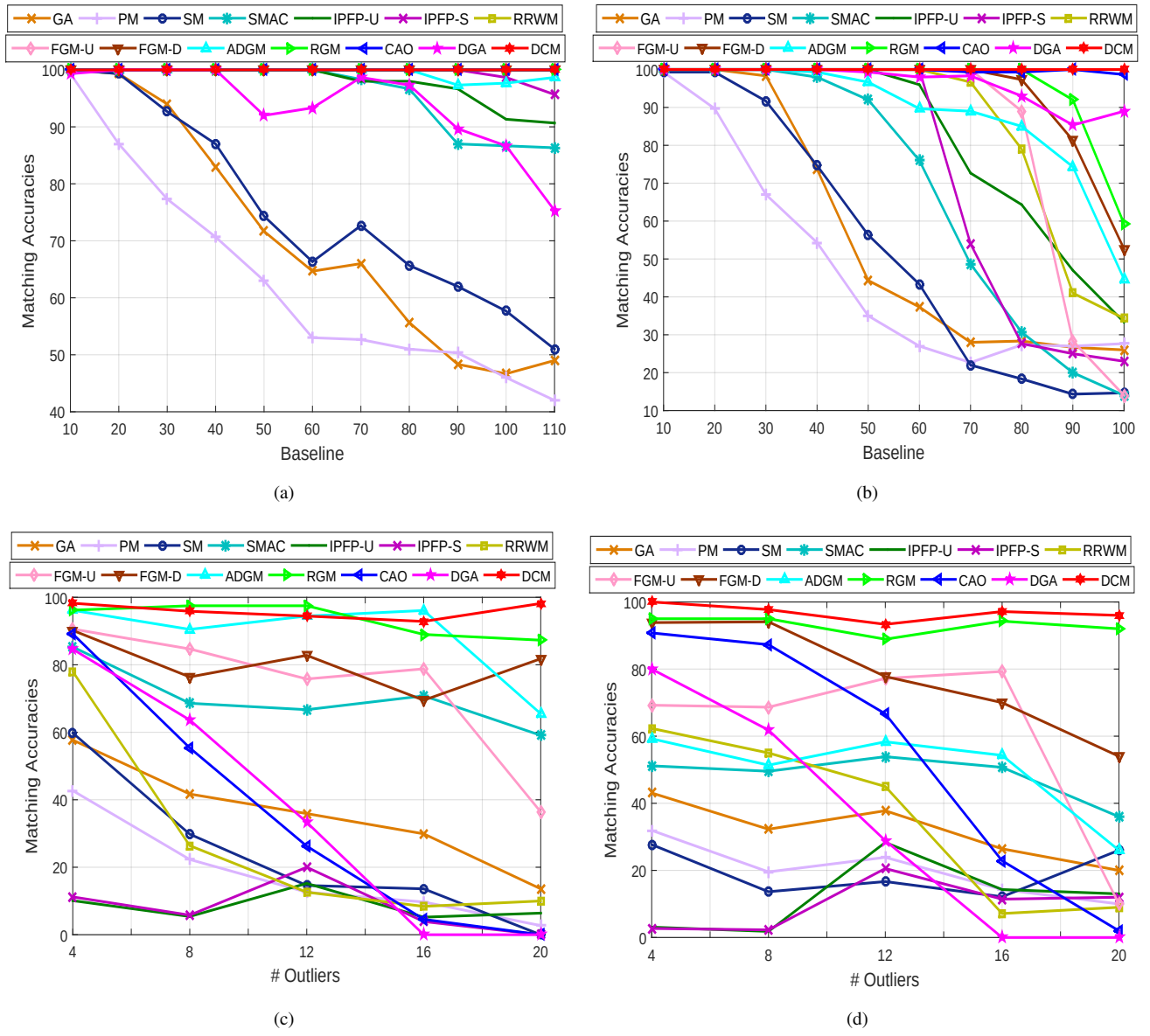
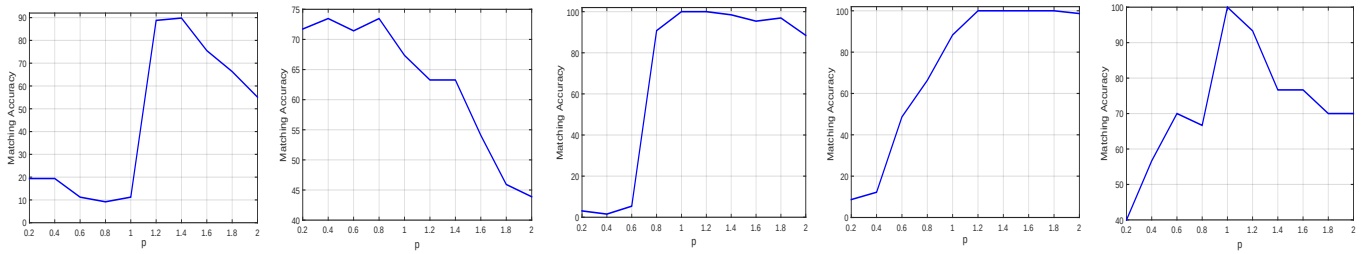
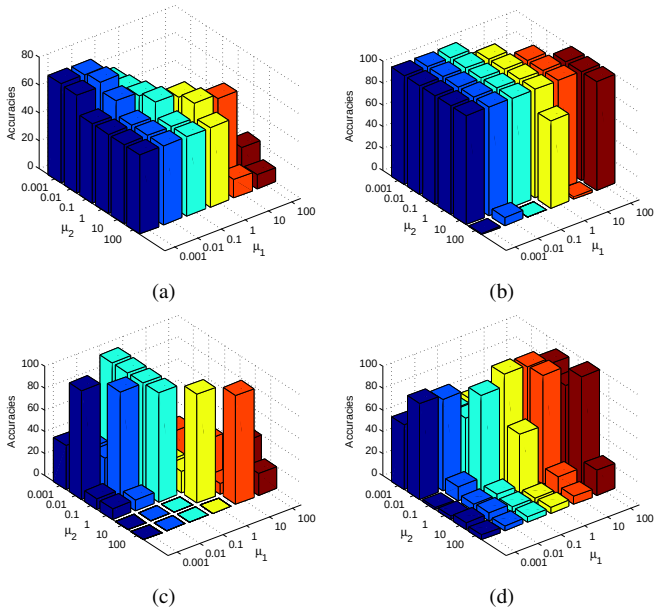


Fig. 5. Matching results of different algorithms on the CMU-House/Hotel database. (a) and (c) display the matching accuracies in the house sequence. (b) and (d) display the matching accuracies in the hotel sequence.



(a) (b) (c) (d) (e)

Fig. 6. Matching results of the proposed DCM with different parameter p on the different databases. (a) Synthetic database with deformation variation. (b) Synthetic database with noise variation. (c) Affine Covariant Regions database. (d) CMU-House/Hotel database. (e) WILLOW-ObjectClass database.



(a) (b) (c) (d)

Fig. 7. Matching results of the proposed DCM with different values of the parameters μ_1 and μ_2 on the different databases. (a) Synthetic database. (b) Affine Covariant Regions database. (c) WILLOW-ObjectClass database. (d) CMU-House/Hotel database.

TABLE VI
RUNNING TIME (SECONDS) OF OUR DCM ON THE DIFFERENT DATABASES.

Database	Solving $\mathbf{X}$	Optimizing $\mathbf{Q}_1$ and $\mathbf{Q}_2$	Total
SYN	85.6438	0.0014	706.9703
WOC	0.7435	0.0003	2.3455
ACR	84.1314	0.0021	435.4436
CMU	4.7817	0.0005	15.0029

TABLE VII
COMPARISON RESULTS OF COMPUTATION TIME (SECONDS) ON THE DIFFERENT DATABASES.

Database	RRWM	IPFP-S	ADGM	FGM-D	DGA	DCM
SYN	31.20	4.80	9.25	117.05	4.46	706.97
WOC	0.21	0.22	1.34	1.89	0.33	2.35
ACR	95.27	7.05	30.98	230.44	9.22	435.44
CMU	0.64	0.37	1.41	5.07	0.54	15.00

F. Running Time

In this paper, our DCM is implemented in MATLAB (R2015a). The computer processor is Intel(R) Core(TM) i7-6700T CPU @ 2.80GHz, and the memory is 8-GB. We report the running time in the optimizing dual calibration matrices $\mathbf{Q}_1$, $\mathbf{Q}_2$ and solving $\mathbf{X}$ at each iteration, and also show the total runtime for the the entire algorithm. The results are listed in Table VI. For the synthetic (SYN) dataset, the tropical fish images with 98 points are used. The proposed algorithm takes about 85.6438s and 0.0014s for updating $\mathbf{X}$ and dual calibration matrices $\mathbf{Q}_1$, $\mathbf{Q}_2$ at each iteration. After performing 8 iterations, the convergence is reached and the total runtime is 706.9703s for the whole algorithm. For the WILLOW-ObjectClass (WOC) and CMU datasets, we use car images with 10 points and house images with 30 points for testing, respectively. Our algorithm is able to converge after performing 3 iterations, and the total runtime is 2.3455s and 15.0029s, respectively. For the Affine Covariant Regions database, the wall images containing 130 points are used. The whole algorithm takes about 435.4436s when the convergence is reached (after performing 5 iterations).

To show the computational complexity straightforwardly, we select some representative methods and report the comparison results on running time on different databases in Table VII. It can be found that the proposed DCM obtains better performance in the variety of graph matching tasks such as deformations, rotations and outliers than the existing methods but has high computational complexity. The main reasons are that: (1) we follow the same experimental setting of FGM-D [44] and adopt Delaunay triangulation [63] for edge generation to build graph structures, the number of edges increases heavily with the number of points. For example, 10 points generate about 40 edges on the WOC dataset, while 98 points generate about 564 edges on the SYN dataset. (2) we use a two steps iteration algorithm to solve the proposed graph matching model and the algorithm can converge after performing 3-8 iterations. At each iteration, the main computational complexity is to solve the matching matrix $\mathbf{X}$ by adopting the path-following algorithm proposed in [47]. For example, updating $\mathbf{X}$ on the SYN dataset (fish sequence) takes 85.6438s at each iteration. It should be noted that the proposed method is time-consuming, but it is within today's computing ability.

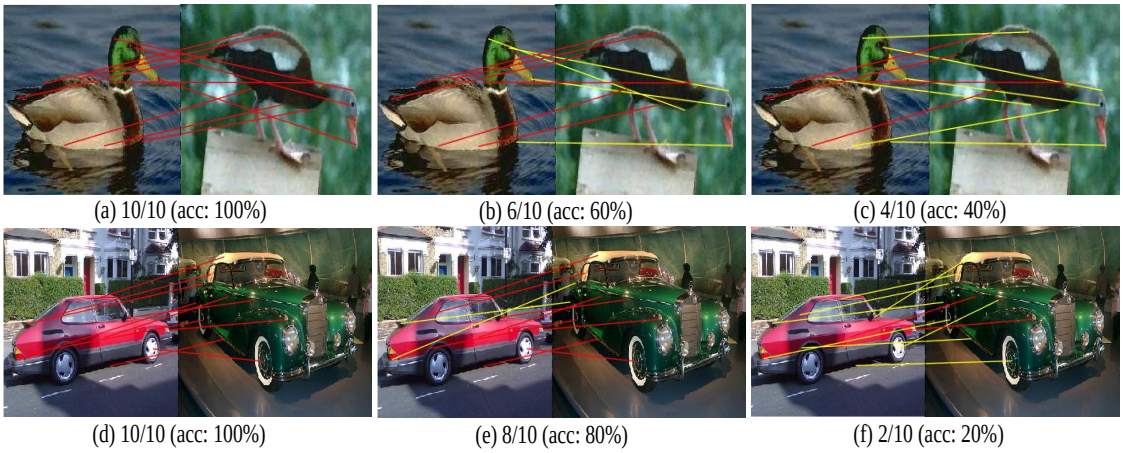


Fig. 8. The visualization results for evaluating the effectiveness of edge alignment. The correct and incorrect matching point pairs are connected in red and yellow lines, respectively. The first column (a and d) represents to the matching results of the proposed method. The second column (b and e) represents to the matching results of the proposed method when $\mu_2 = 0$. The third column (c and f) represents to the matching results of the proposed method when $\mu_1 = \mu_2 = 0$.

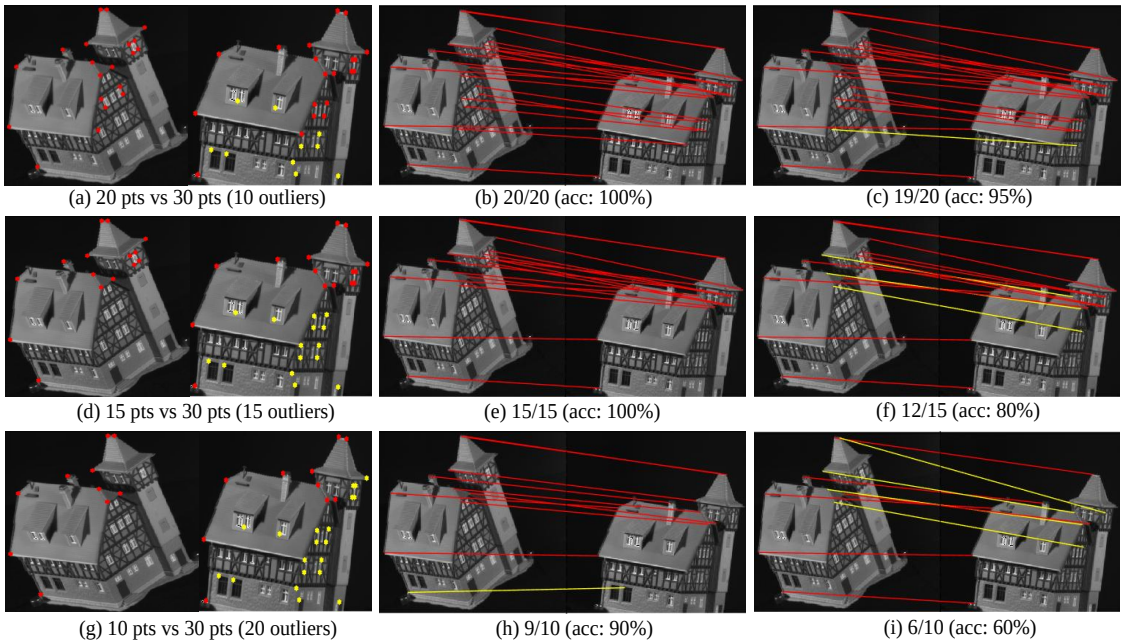


Fig. 9. The visualization results for evaluating the robustness of $L_{2,p}$-norm. The yellow points are outliers. The correct and incorrect matching point pairs are connected in red and yellow lines, respectively. (a) 20 points in the model graph and 30 points in the target graph (10 points are outliers). (d) 15 points in the model graph and 30 points in the target graph (15 points are outliers). (g) 10 points in the model graph and 30 points in the target graph (20 points are outliers). The second column (b, e and h) represents to the matching results of using $L_{2,p}$-norm. The third column (c, f and i) represents to the matching results of using L_2-norm.

G. Discussion

In this section, we analyse the effectiveness of the edge alignment and $L_{2,p}$-norm, and discuss the possible reason for matching failure.

1) Impact analysis of edge alignment: In the proposed model (8), we incorporate the second and third terms to align the edges. The second term is to maximize the edge similarities after performing dual calibration, which is helpful to establish the correspondence relationship between two graphs but could make $\mathbf{Y}$ too sparse in the process of matching. Considering that Delaunay triangulation [63] is used for edge generation to build graph structures, and each pair of points have multiple edges that should be aligned. If $\mathbf{Y}$ is too sparse, it will go against the improvement of matching performance. Thus we empirically add the third term to achieve the balanced effect. The corresponding experiments on the WILLOW-ObjectClass dataset are conducted for investigating the effectiveness of edge alignment. The matching results with ten points are shown in Fig. 8. The correct and incorrect matching point pairs are connected in red and yellow, respectively. The first column (a and d) represents to the matching results of the

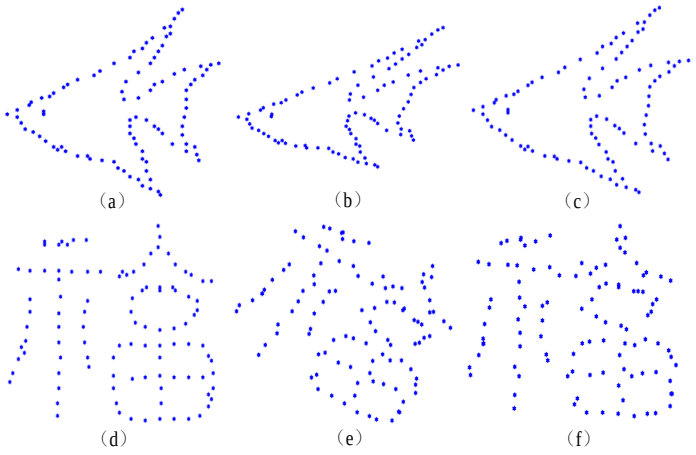


Fig. 10. Examples on the synthetic database (the tropical fish and Chinese character 'blessing'). a and d are original images with 98 and 105 points, respectively. b and e are the deformed/rotated images. c and f are the aligned images of b and e after performing dual calibration mechanism.

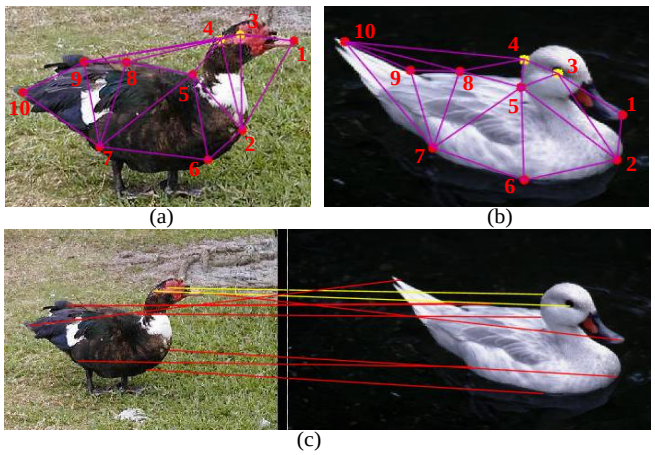


Fig. 11. Matching failure example on the WILLOW-ObjectClass dataset. The yellow points represent the matching failure. The correct and incorrect matching point pairs are connected in red and yellow lines, respectively.

proposed method. The second column (b and e) represents to the matching results of the proposed method when $\mu_2 = 0$ (*i.e.* no third term in model (8)). The third column (c and f) represents to the matching results of the proposed method when $\mu_1 = \mu_2 = 0$. From the results of the first and second column, it can be seen that the third term in our model (eq. 8) is effective to improve the performance. Comparing the results of the first column (a and d) with the third column (c and f), it also indicates the effectiveness of the presented edge alignment (combining the second and third terms).

In addition, we also conduct the experiments on the synthetic database (the tropical fish and Chinese character 'blessing'). As observed in Fig. 10, establishing correspondence between a and b (or d and e) is more difficult than between a and c (or d and f). Actually, the matching accuracies of a vs. b and a vs. c are 94.90% and 96.94%, respectively. The matching accuracies of d vs. e and d vs. f are 9.52% and 77.14%, respectively. The results further demonstrate the effectiveness of the edge alignment.

2) Robustness analysis of $L_{2,p}$-norm vs. L_2-norm: We adopt the $L_{2,p}$-norm to replace L_2-norm as the similarity metric in the proposed graph matching algorithm as $L_{2,p}$-norm can be effectively applied to different types of data sources due to the flexibility of setting p value. As a consequence, it can reduce the influence of outliers to improve the robustness of the algorithm. In order to validate the robustness of using $L_{2,p}$-norm, we use house sequence with 30 points on the CMU dataset to conduct the experiments. We choose N (N =10, 15, 20) points as outliers. That is, for each match, there are 30-N points in the model image and 30 points in the target image. The visual matching results are shown in Fig. 9. The second column (b, e and h) represents to the matching results of using $L_{2,p}$-norm, and the third column (c, f and i) represents to the matching results of using L_2-norm. From the results, it can be observed that $L_{2,p}$-norm is more robust than L_2-norm in our method.

3) Matching failure analysis: In order to analyse the possible reasons for matching failure, we show a visual example of two duck images in Fig. 11. For each image, ten manually labeled points are used for matching. We mark them as 1 to 10, which can be seen in Fig. 11(a) and (b). Furthermore, we follow the experimental setting in [44] [47], and adopt Delaunay triangulation [63] for edge generation to build graph structures. For instance, points 1, 2, 3 make a triangle structure and points 2, 3, 4 make another structure in Fig. 11(a). As observed in Fig. 11(c), points 3 and 4 are two matching failure cases. We think the possible reason of matching failure is that the two points make two different triangle structures as shown in Fig. 11(a) and (b). In Fig. 11(a), points 2, 3, 4 make a triangle structure. In Fig. 11(b), points 3, 4, 5 make a triangle structure. The different triangle structures have different edges, which could lead to lose efficacy of the proposed edge alignment module. In this case it may be lead to match failure if the point alignment module can't have positive effect.

V. Conclusion and Discussion

In this paper, we propose a dual calibration mechanism (DCM) for establishing feature point correspondence in graph matching. Specifically, we first employ a dual calibration strategy to solve two geometric calibration matrices. By performing the dual calibration, the feature points correspondence between two images with deformation and rotation variations can be obtained. Second, we use the $L_{2,p}$-norm as the similarity metric in the presented model to measure the points and edges similarities for reducing the influence of outliers. Finally, we incorporate the dual calibration and $L_{2,p}$-norm based similarity metric into the graph matching model and develop an effective algorithm to solve the model. In addition, we also theoretically prove the convergence of the presented algorithm. Experimental results in the variety of graph matching tasks such as deformations, rotations and outliers evidence the competitive performance of the presented DCM model over the state-of-the-art approaches.

It should be noted that the proposed method can obtain better matching accuracies than the existing methods but it has

high computational complexity due to using the path-following algorithm presented in [47] to solve the matching matrix $\mathbf{X}$. Thus how to develop a more effective algorithm to reduce the computational cost is a challenging and important task. In addition, different from our work, Gai *et al.* [40] use the idea of linking two coordinate systems and 3D reconstruction to propose dual calibration model. Can the idea be used for the graph matching task? If the answer is yes, how to design the new calibration scheme? Recently, cost aggregation methods [64] [65] have attracted much attention and have been successfully used in stereo matching [64]. It would be an interesting problem whether the cost aggregation can be used for improving the graph matching performance or not. Deep neural network methods have good feature extraction ability and also have been widely applied in correspondence matching such as dense correspondence [66] [67] and stereo matching [68]. Adaptively incorporating the deep neural network and graph matching model could potentially yield better performance. Our future work will focus on these topics.

ACKNOWLEDGMENT

This work was supported in part by the National Natural Science Foundation of China (61876104), in part by the Natural Science Foundation of Guangdong Province (2019A1515011266, 2020A1515010702), and in part by the Guangzhou University Research Fund (220030401). The authors thank the literature [43] for providing the compared experimental data of the existing methods. In addition, The authors also thank Dr. Zongyuan Ge for polishing the manuscript and thank the editors and anonymous reviewers for their valuable suggestions to improve the manuscript.

REFERENCES

[1] M. Cho, K. Alahari, and J. Ponce, "Learning graphs to match," in *IEEE International Conference on Computer Vision*, 2013, pp. 25–32.

[2] O. Duchenne, A. Joulin, and J. Ponce, "A graph-matching kernel for object categorization," in *IEEE International Conference on Computer Vision*, 2011, pp. 1792–1799.

[3] A. C. Berg, T. L. Berg, and J. Malik, "Shape matching and object recognition using low distortion correspondences," in *IEEE International Conference on Computer Vision and Pattern Recognition*, 2005, pp. 26–33.

[4] O. Duchenne, F. Bach, I. S. Kweon, and J. Ponce, "A tensor-based algorithm for high-order graph matching," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 12, pp. 2383–2395, 2011.

[5] Q. X. Huang, G. X. Zhang, L. Gao, S. M. Hu, A. Butscher, and L. Guibas, "An optimization approach for extracting and encoding consistent maps in a shape collection," *Acm Transactions on Graphics*, vol. 31, no. 6, pp. 1–11, 2012.

[6] Z. Yun, C. Wang, W. Yang, X. Gu, D. Samaras, and N. Paragios, "Dense non-rigid surface registration using high-order graph matching," in *IEEE International Conference on Computer Vision and Pattern Recognition*, 2010, pp. 382–389.

[7] Q. Zhao, S. Pizer, M. Niethammer, and J. Rosenman, "Geometric-feature-based spectral graph matching in pharyngeal surface registration," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 2014, pp. 259–266.

[8] Z. Cai, L. Wen, Z. Lei, N. Vasconcelos, and S. Z. Li, "Robust deformable and occluded object tracking with dynamic graph," *IEEE Transactions on Image Processing*, vol. 23, no. 12, pp. 5497–5509, 2014.

[9] T. Wang and H. Ling, "Gracker: A graph-based planar object tracker," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 6, pp. 1494–1501, 2018.

[10] H. Xiong, D. Zheng, Q. Zhu, B. Wang, and Y. F. Zheng, "A structured learning-based graph matching method for tracking dynamic multiple objects," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 23, no. 3, pp. 534–548, 2013.

[11] Q. Zhou, H. Fan, S. Zheng, H. Su, X. Li, S. Wu, and H. Ling, "Graph correspondence transfer for person re-identification," *arXiv preprint arXiv:1804.00242*, 2018.

[12] S. Mohammadi, D. F. Gleich, T. G. Kolda, and A. Grama, "Triangular alignment (tame): A tensor-based approach for higher-order network alignment," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 14, no. 6, pp. 1446–1458, 2016.

[13] E. L. Lawler, "The quadratic assignment problem," *European Journal of Operational Research*, vol. 9, no. 4, pp. 586–599, 1963.

[14] B. Luo and E. R. Hancock, "Structural graph matching using the em algorithm and singular value decomposition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 10, pp. 1120–1136, 2001.

[15] X. Bai, E. R. Hancock, and R. C. Wilson, "A generative model for graph matching and embedding," *Computer Vision and Image Understanding*, vol. 113, no. 7, pp. 777–789, 2009.

[16] J. Tang, B. Jiang, A. Zheng, and B. Luo, "Graph matching based on spectral embedding with missing value," *Pattern Recognition*, vol. 45, no. 10, pp. 3768–3779, 2012.

[17] W. Feng, Z. Q. Liu, L. Wan, C. M. Pun, and J. Jiang, "A spectral-multiplicity-tolerant approach to robust graph matching," *Pattern Recognition*, vol. 46, no. 10, pp. 2819–2829, 2013.

[18] E. Bonabeau, "Graph multidimensional scaling with self-organizing maps," *Information Sciences*, vol. 143, no. 1, pp. 159–180, 2002.

[19] S. Jouili and S. Tabbone, "Graph embedding using constant shift embedding," in *International Conference on Recognizing Patterns*, 2010, pp. 83–92.

[20] M. M. Luqman, J. Y. Ramel, J. Lladós, and T. Brouard, "Fuzzy multilevel graph embedding," *Pattern Recognition*, vol. 46, no. 2, pp. 551–565, 2013.

[21] E. Z. Borzeshi, M. Piccardi, K. Riesen, and H. Bunke, "Discriminative prototype selection methods for graph embedding," *Pattern Recognition*, vol. 46, no. 6, pp. 1648–1657, 2013.

[22] M. Leordeanu and M. Hebert, "A spectral technique for correspondence problems using pairwise constraints," in *IEEE International Conference on Computer Vision*, 2005, pp. 1482–1489.

[23] R. Zhang and W. Wang, "Second- and high-order graph matching for correspondence problems," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 28, no. 10, pp. 2978–2992, 2018.

[24] W. Nie, A. Liu, Y. Gao, and Y. Su, "Hyper-clique graph matching and applications," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 29, no. 6, pp. 1619–1630, 2019.

[25] M. Cho, J. Lee, and K. M. Lee, "Reweighted random walks for graph matching," in *European Conference on Computer Vision*, 2010, pp. 492–505.

[26] D. K. Lê-Huu and N. Paragios, "Alternating direction graph matching," in *IEEE International Conference on Computer Vision and Pattern Recognition*, 2017, pp. 6253–6261.

[27] B. Jiang, J. Tang, X. Cao, and B. Luo, "Lagrangian relaxation graph matching ," *Pattern Recognition*, vol. 61, pp. 255–265, 2017.

[28] S. Khan, M. Nawaz, G. Xu, and H. Yan, "Image correspondence with cur decomposition based graph completion and matching," *IEEE Transactions on Circuits and Systems for Video Technology*, DOI: 10.1109/TCSVT.2019.2935838 2019.

[29] A. Egozi, Y. Keller, and H. Guterman, "A probabilistic approach to spectral graph matching." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 1, pp. 18–27, 2013.

[30] J. Lee, M. Cho, and K. M. Lee, "A graph matching algorithm using data-driven markov chain monte carlo sampling," pp. 2816–2819, 2010.

[31] J. Yan, Z. Chao, H. Zha, L. Wei, X. Yang, and S. M. Chu, "Discrete hyper-graph matching," in *IEEE International Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1520–1528.

[32] J. Yan, C. Li, Y. Li, and G. Cao, "Adaptive discrete hypergraph matching," *IEEE Transactions on Cybernetics*, vol. 48, no. 22, pp. 765–779, 2018.

[33] Y. Suh, M. Cho, and K. M. Lee, "Graph matching via sequential monte carlo," in *European Conference on Computer Vision*, 2012, pp. 624–637.

[34] K. Adamczewski, Y. Suh, and K. M. Lee, "Discrete tabu search for graph matching," in *IEEE International Conference on Computer Vision*, 2015, pp. 109–117.

[35] H. Chui and A. Rangarajan, "A new point matching algorithm for non-rigid registration," *Computer Vision and Image Understanding*, vol. 89, no. 2, pp. 114–141, 2003.

[36] A. Myronenko and X. Song, "Point set registration: Coherent point drift," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 12, pp. 2262–2275, 2010.

[37] M. A. Pinheiro, J. Kybic, and P. Fua, "Geometric graph matching using monte carlo tree search," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 11, pp. 2171–2185, 2017.

[38] F. Wang, N. Xue, Y. Zhang, X. Bai, and G.-S. Xia, "Adaptively transforming graph matching," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 625–640.

[39] J. Jiao, Q. Liao, Y. Zhu, T. Liu, Y. Yu, R. Fan, L. Wang, and M. Liu, "A novel dual-lidar calibration algorithm using planar surfaces," in *IEEE Intelligent Vehicles Symposium (IVS)*, 2019, pp. 1499–1504.

[40] S. Gai, F. Da, and X. Dai, "A novel dual-camera calibration method for 3d optical measurement," *Optics and Lasers in Engineering*, vol. 104, pp. 126–134, 2018.

[41] F. Nie, H. Huang, C. Xiao, and C. Ding, "Efficient and robust feature selection via joint $l_{2,1}$-norms minimization," in *International Conference on Neural Information Processing Systems*, 2010, pp. 1813–1821.

[42] H. Zhao, W. Zheng, and F. Nie, "A new formulation of linear discriminant analysis for robust dimensionality reduction," *IEEE Transactions on Knowledge and Data Engineering*, DOI: 10.1109/TKDE.2018.2842023.

[43] Y.-F. Yu, G. Xu, M. Jiang, H. Zhu, D.-Q. Dai, and H. Yan, "Joint transformation learning via the $l_{2,1}$-norm metric for robust graph matching," *IEEE Transactions on Cybernetics*, DOI: 10.1109/TCYB.2019.2912718 2019.

[44] F. Zhou and F. De la Torre, "Factorized graph matching," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 9, pp. 1774–1789, 2016.

[45] E. M. Loiola, N. M. M. de Abreu, P. O. Boaventura-Netto, P. Hahn, and T. Querido, "A survey for the quadratic assignment problem," *European Journal of Operational Research*, vol. 176, no. 2, pp. 657–690, 2007.

[46] H. M. Park and K. J. Yoon, "Multi-attributed graph matching with multi-layer random walks," in *European Conference on Computer Vision*, 2016, pp. 189–204.

[47] F. Zhou and F. D. L. Torre, "Deformable graph matching," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 2922–2929.

[48] Q. Wang, Q. Gao, X. Gao, and F. Nie, "ℓ_2, p-norm based pca for image recognition," *IEEE Transactions on Image Processing*, vol. 27, no. 3, pp. 1336–1346, 2018.

[49] H. Chui and A. Rangarajan, "A new point matching algorithm for non-rigid registration," *Computer Vision and Image Understanding*, vol. 89, no. 23, pp. 114–141, 2003.

[50] "Oxford VGG database," http://www.robots.ox.ac.uk/∼vgg/data/.

[51] "CMU-house/hotel database," http://vasc.ri.cmu.edu/idb/html/motion/house/.

[52] S. Gold and A. Rangarajan, "A graduated assignment algorithm for graph matching," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 4, pp. 377–388, 1996.

[53] M. Leordeanu, M. Hebert, and R. Sukthankar, "An integer projected fixed point method for graph matching and map inference," in *Advances in Neural Information Processing Systems*, 2009, pp. 1114–1122.

[54] M. Leordeanu and M. Hebert, "A spectral technique for correspondence problems using pairwise constraints," in *EEE International Conference on Computer Vision*, 2005, pp. 1482–1489.

[55] R. Zass and A. Shashua, "Probabilistic graph and hypergraph matching," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2008, pp. 1–8.

[56] T. Cour, P. Srinivasan, and J. Shi, "Balanced graph matching," in *Advances in Neural Information Processing Systems*, 2007, pp. 313–320.

[57] J. Yan, M. Cho, H. Zha, X. Yang, and S. Chu, "Multi-graph matching via affinity optimization with graduated consistency regularization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 6, pp. 1228–1242, 2015.

[58] D. K. Lê-Huu and N. Paragios, "Alternating direction graph matching," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 6253–6261.

[59] H. Zhu, C. Cui, L. Deng, R. C. C. Cheung, and H. Yan, "Elastic net constraint-based tensor model for high-order graph matching," *IEEE Transactions on Cybernetics*, DOI: 10.1109/TCYB.2019.2912718 2019.

[60] A. Vedaldi and B. Fulkerson, "Vlfeat: An open and portable library of computer vision algorithms," in *ACM International Conference on Multimedia*, 2010, pp. 1469–1472.

[61] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[62] X. Shi, H. Lin, W. Hu, J. Xing, and Y. Zhang, "Tensor power iteration for multi-graph matching," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 5062–5070.

[63] D. T. Lee and B. Schachter, "Two algorithms for constructing a delaunay triangulation," *International Journal of Parallel Programming*, vol. 9, no. 3, pp. 219–242, 1980.

[64] G.-S. Hong and B.-G. Kim, "A local stereo matching algorithm based on weighted guided image filtering for improving the generation of depth range images," *Displays*, vol. 49, pp. 80–87, 2017.

[65] G.-S. Hong, B.-G. Kim, and K.-K. Kwon, "Efficient depth map estimation method based on gradient weight cost aggregation strategy for distributed video sensor networks," *International Journal of Distributed Sensor Networks*, vol. 10, no. 1, pp. 1–7, 2014.

[66] Y. Liu, J. Pan, and Z. Su, "Deep feature matching for dense correspondence," in *International Conference on Image Processing*, 2017, pp. 795–799.

[67] Y. Liu, J. Pan, Z. Su, and K. Tang, "Robust dense correspondence using deep convolutional features," *The Visual Computer*, vol. 36, no. 4, pp. 827–841, 2020.

[68] J. Zbontar and Y. Lecun, "Stereo matching by training a convolutional neural network to compare image patches," *Journal of Machine Learning Research*, vol. 17, no. 1, pp. 2287–2318, 2016.

Yu-Feng Yu (M'19) received the Ph.D. degree in statistics from Sun Yat-Sen University, Guangzhou, China, in 2017. He is currently an assistant professor in the Department of Statistics, Guangzhou University. From 2016 to 2017, he was a visiting scholar in the Lane Department of Computer Science and Electrical Engineering, West Virginia University, Morgantown, WV, USA. From 2017 to 2018, he was a senior research associate in the Department of Electronic Engineering, City University of Hong Kong. His research interests include image processing, statistical optimization, pattern recognition and machine learning.

Guoxia Xu (M'19) received the B.S. degree in information and computer science from Yancheng Teachers University, Jiangsu Yancheng, China in 2015, and the M.S. degree in computer science and technology from Hohai University, Nanjing, China in 2018. He was a research assistant in City University of Hong Kong and Chinese University of Hong Kong. Now, he is pursing his Ph.D. degree in Department of Computer Science, Norwegian University of Science and Technology, Gjovik Norway. His research interest includes pattern recognition, image processing, and computer vision.

Ke-Kun Huang (M'14) received the B.Sc., M.Sc and Ph.D. degrees in applied mathematics from Sun Yat-sen University, Guangzhou, China, in 2002, 2005 and 2016, respectively. He is currently a Professor with the Department of Mathematics, Jiaying University, Meizhou, China. His research interests include image processing and pattern recognition. He has published over 10 papers on top journals including IEEE TNNLS, IEEE TGRS, IEEE TIP, IEEE TCYB and Pattern Recognition. He was elected as NanYue Excellent Teacher of GuangDong province.

Hu Zhu (M'17) received the B.S. degree in mathematics and applied mathematics from Huaibei Coal Industry Teachers College, Huaibei, China, in 2007, and the M.S. and Ph.D. degrees in computational mathematics and pattern recognition and intelligent systems from Huazhong University of Science and Technology, Wuhan, China, in 2009 and 2013, respectively. In 2013, he joined the Nanjing University of Posts and Telecommunications, Nanjing, China. His research interests include pattern recognition, image processing, and computer vision.

Long Chen (M'11) received the M.S. degree in Computer Engineering from the University of Alberta, Edmonton, AB, Canada, in 2005, and the Ph.D. degree in Electrical Engineering from the University of Texas at San Antonio, San Antonio, TX, USA, in 2010. From 2010 to 2011, he was a Post-Doctoral Fellow with the University of Texas at San Antonio. He is currently an Associate Professor with the Department of Computer and Information Science, University of Macau, Macau, China. His current research interests include computational intelligence techniques and their applications. Dr. Chen has been working in publication matters for several IEEE conferences. He was the Publications Co-Chair of the IEEE International Conference on Systems, Man and Cybernetics in 2009, 2012, and 2014.

Hao Wang (M'07) is an associate professor in the Department of Computer Science in Norwegian University of Science and Technology, Norway. He received his Ph.D. degree and B.Eng. degree, both in computer science and engineering, from South China University of Technology in 2006 and 2000, respectively. His research interests include big data analytics, industrial internet of things, high performance computing, and safety-critical systems. He has published 140+ papers in reputable international journals and conferences. He served as a TPC co-chair for IEEE CPSCom 2020, IEEE CIT 2017, ES 2017, and DataCom 2015, a senior TPC member for CIKM 2019, and reviewers/TPC members for many journals and conferences. He is the Chair for Sub-TC on Healthcare in IEEE IES Technical Committee on Industrial Informatics.