Clique-graph Matching by Preserving Global & Local Structure

Wei-Zhi Nie[†], An-An Liu[†*], Zan Gao[‡], Yu-Ting Su[†]
† School of Electronics Information Engineering, Tianjin University, China
‡ School of Computer & Communication Engineering, Tianjin University of Technology, China
⋆Corresponding author: anan0422@gmail.com

Abstract

This paper originally proposes the clique-graph and further presents a clique-graph matching method by preserving global and local structures. Especially, we formulate the objective function of clique-graph matching with respective to two latent variables, the clique information in the original graph and the pairwise clique correspondence constrained by the one-to-one matching. Since the objective function is not jointly convex to both latent variables, we decompose it into two consecutive steps for optimization: 1) clique-to-clique similarity measure by preserving local unary and pairwise correspondences; 2) graph-to-graph similarity measure by preserving global clique-to-clique correspondence. Extensive experiments on the synthetic data and real images show that the proposed method can outperform representative methods especially when both noise and outliers exist.

1. Introduction

Graph matching is very essential in the fields of computer vision and pattern recognition. It has been widely used in many applications such as 3D image retrieval[13], object recognition[1, 6], biomedical identification[5], multiple object tracking[14], etc. Given two sets of feature points lying in different subspaces, each set can be considered as one graph, which consists of two important components, nodes and edges, respectively representing feature points and the relations between pariwise feature points. Graph matching aims to discover each feature point's counterpart in the other subspace while well preserving both global relations with the other nodes in the same graph (global structure) and the local relations with the neighbor nodes (local structure). Although much work has been done for this task, it is still challenging due to the difficulty in discriminative visual feature representation, precise salient point localization, deformation of the correlation between pairwise points due to the viewpoint change, affine transformation, etc.

Previous literatures on this task mainly focus on node-to-node mapping by leveraging the first & second order attributes [8, 16, 12, 11]. Since this node-wise mapping can not well preserve both global and local structure in the feature subspaces, many researchers have been involved into designing the high order attributes for the formulation of the objective function [4, 20]. Consequently, this task can be naturally generalized into the problem of hyper-graph matching [10]. However, hyper-graph matching usually faces several non-trivial problems. On one hand, it is not easy and intuitive to define scale-invariant similarity measure between two hyper-edges. On the other hand, it is mathematically difficult to impose the mapping constraints on the objective function to induce the stable solution. Furthermore, the node order on each hyper-edge is usually ignored [10], which might lead to the variation of the similarity between pairwise hyper-edges and further directly have negative influence on the robustness of graph matching.

In this paper, we originally propose the clique-graph and further propose the method for clique-graph matching by preserving global and local structure. Different from the classic graph or hyper-graph, we replace individual node (the basic unit in the graph/hyper-graph) by one clique, which consists of K nearest neighbors in the specific feature subspace and can convey the local structural attributes in the star model. For clique-graph matching, we formulate the objective function with respective to two latent variables, the clique information in individual clique-graph and the clique correspondence between pairwise clique-graphs. Since the objective function is not jointly convex to both parameters, we further decompose it into two consecutive steps. First, the local graph matching with unary and pairwise correspondences is implemented to align both cliques from two clique-graphs. In this way, we can discover the correspondences of nodes and edges in both cliques and further compute the clique-wise similarity. Second, the clique-graph matching can be realized by clique-to-clique matching only with the unary attribute since the clique-wise similarity already conveys local structure.

The main contributions of the proposed method are summarized as follows:

- The proposed clique-graph can be considered as the generalized form of both classic graph and hyper-graph. Individual clique can convey arbitrary order attributes to represent the local structure;

- The proposed clique-graph matching method can benefit simplifying the hyper-edge correspondence into the formulation only with clique-wise correspondence. Therefore, we can significantly increase the complexity of the local structure with more nodes in one clique to convey more high order attributes of local structure, while only sacrificing little computational complexity. Furthermore, the proposed method can benefit avoiding the difficulty in defining scale-invariant similarity measure between two hyper-edges especially when the order is over 3;

- The proposed method can easily impose the one-to-one constraint for clique-graph matching since the clique-wise similarity measure already preserves the local structure correspondence and the clique-graph matching can be simply formulated only with the unary attribute to preserve the global correspondence;

- The local similarity measure is unique since it is based on the discovery of both unary and pairwise correspondence between pairwise nodes/edges. It can overcome the difficulty in discovering the node order of each hyper-edge for hyper-edge similarity measure.

The rest of the paper is organized as follows. The related work is reviewed in Section 2. Then we detail the clique-graph matching method in Section 3. We further present the experimental results in Section 4. Section 5 concludes this paper.

2. Related Work

Most of the previous methods focus on leveraging unary and pairwise attributes for graph matching. Gori *et al.* [9] proposed the random walk based models, which can enhance the graph topological features at node level, and introduced a polynomial algorithm for the classic graph isomorphism problem, under the restriction of dealing with Markovian spectrally distinguishable graphs. Minsu *et al.* [2] converted the graph matching into the ranking problem and utilized the random walk for solution. Gold *et al.* [8] proposed the graduated assignment algorithm by combining graduated nonconvexity, two-way constraints, and sparsity, which was not restricted to any special class of graph and achieved large improvements in accuracy and speed even in the presence of high noise. Wyk *et al.* [16] introduced a novel projections onto convex sets (POCS) graph matching algorithm, in which two-way assignment constraints were enforced without using elaborate penalty terms, graduated nonconvexity, or sophisticated annealing mechanisms to escape from poor local minima. Leordeanu *et al.* [11] proposed the spectral matching method, which utilized the spectral property to calculate rank-1 approximation of the affinity matrix. Zaslavskiy *et al.* [19] proposed a convex-concave programming approach for the labeled weighted graph matching problem. This method can easily integrate the information on graph label similarities into the optimization problem, and therefore, benefit performing labeled weighted graph matching.

Since the unary and pairwise relations are not discriminative enough to represent the characteristics of local structure, many researchers are involved into formulating this problem in the higher-graph matching [21, 18, 17, 7]. Zass *et al.* [20] proposed a hyper-graph matching method, which introduced a novel view that the matching problem and its corresponding solution were related by the kronecker produce. Duchenne *et al.* [4] proposed the graph matching based on Tensor Matching, which was interpreted as a high order extension of spectral matching. It utilized rank-1 approximation of the affinity tensor as a solution by using high order tensor iteration. Lee *et al.*[10] proposed a hyper-graph matching formulation by reinterpreting the random walk concept on the hyper-graph in a probabilistic manner.

3. Clique-graph Matching

In this section, we will respectively introduce the methods for clique-graph generation and matching.

3.1. Clique-graph Generation

A classic graph $G = (V, E, A)$ consists of the node set $V = \{v_i\}_{i=1}^{I}$, the edge set $E = \{e_j\}_{j=1}^{J}$, and the attribute set $A = \{a_j\}_{j=1}^{J}$ associated with the corresponding edges as shown in Fig.1(a). Given two attributed graphs $G^p = (V^p, E^p, A^p)$ and $G^q = (V^q, E^q, A^q)$, graph matching aims to determine the correct correspondences inbetween. This is usually realized by leveraging the unary attribute with respect to individual node, the pairwise attribute with respect to individual edge, and the high-order attributes with respect to different scales of local structures. Different from previous graph matching methods which consider each node as the basic unit, we propose to regard each clique as the basic unit for matching, which consists of several neighbor nodes in the feature space and can implicitly convey arbitrary order attributes with the specific structure. Consequently, we originally propose the concept of clique-graph and further present the novel method for clique-graph matching. As shown in Fig.1 (b), a clique-graph $\tilde{G} = \{\tilde{V}, \tilde{A}\}$ is composed of two kinds of elements, the clique set $\tilde{V}$ and the attribute set $\tilde{A}$ associated with individual cliques. Each clique $\tilde{V}_i \in \tilde{V}$ can be represented by the star model, $\tilde{V}_i = \{\tilde{c}_i, \{\tilde{l}_{ij}\}_{j=1}^{k}, \{\tilde{e}_{ij}\}_{j=1}^{k}\}$, where

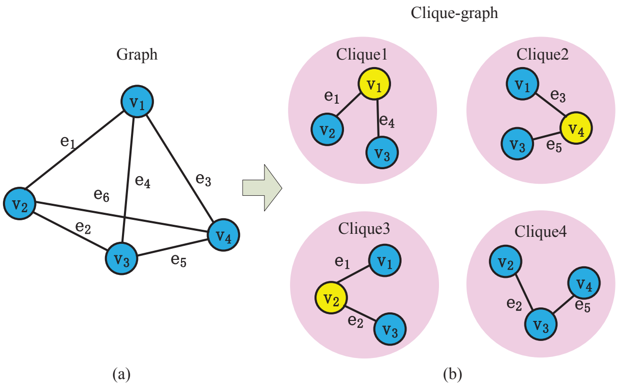


Figure 1. Example of clique-graph generation. (a) graph $G = (V, E, A)$; (b) clique-graph $\tilde{G} = \{\tilde{V}, \tilde{A}\}$. The yellow and blue nodes separately denote the center and leaf nodes of each clique.

$\tilde{c}_i$ denotes the center of the clique, $\{\tilde{l}_{ij}\}_{j=1}^{k}$ denotes k leaf nodes of the clique, $\{\tilde{e}_{ij}\}_{j=1}^{k}$ denotes k edges linking the center node and k leaf nodes. The order of one clique, $\delta(\tilde{V}_i)$, equals to the number of the nodes in it. $\tilde{A}_i \in \tilde{A}$ is the attribute of the i^{th} clique, which represents the importance of this clique in the entire clique-graph and can be calculated based on the term frequency by dividing the total frequency of the nodes in $\tilde{V}_i$ by the produce of the clique number and the node number in each clique. Since each clique implicitly convey the compact local structure information, the edge between two cliques can be ignored by assuming the relation between two cliques is too weak. This simplification will benefit directly imposing the one-to-one constraint for clique-graph matching and effectively avoid the complicated discretization step in previous methods [12, 11, 8, 16].

Given two clique-graphs, $\tilde{G}^p = \{\tilde{V}^p, \tilde{A}^p\}$ and $\tilde{G}^q = \{\tilde{V}^q, \tilde{A}^q\}$, the similarity measure of both can be represented by $J(X, \tilde{G}^p, \tilde{G}^q)$, which means the similarity of both clique-graphs can be computed by considering their structure characteristics ($\tilde{G}^p$ and $\tilde{G}^q$) and the clique-to-clique correspondence X. Therefore, the clique-graph matching can be formulated as maximizing the score function $J(X, \tilde{G}^p, \tilde{G}^q)$:

$$(X, \tilde{G}^p, \tilde{G}^q)^* = \arg \max_{X, \tilde{G}^p, \tilde{G}^q} J(X, \tilde{G}^p, \tilde{G}^q) \tag{1}$$

3.2. Optimization

Clique-graph matching (Fig.2) aims to discover the clique-to-clique correspondence with one-to-one constraint. Thus, the original graph matching can be realized by achieving the correspondences between each pair of centers, which is consistent with the discovered clique-to-clique correspondence.

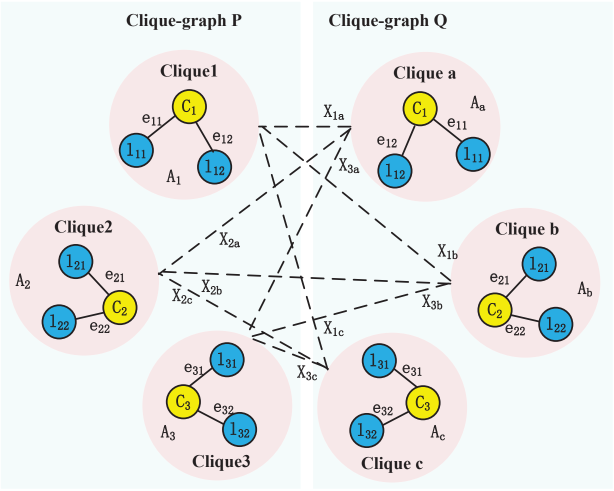


Figure 2. Framework of clique-graph matching.

By denoting the clique-to-clique similarity as $S_{i,m}$, where $\tilde{V}_i \in \tilde{G}^p$ and $\tilde{V}_m \in \tilde{G}^q$, the vector $S = \{S_{i,m}\}_{i=1,\ldots,N^p, m=1,\ldots,N^q} \in R^{N^p N^q \times 1}$, where N^p and N^q respectively denote the clique numbers in $\tilde{G}^p$ and $\tilde{G}^q$, can be used to represent the similarities between all pairwise cliques.

We define the solution of clique-graph matching as a binary indicator matrix $\bar{X} \in \{0, 1\}^{N^p \times N^q}$. If $\tilde{V}_i \in \tilde{G}^p$ is matched to $\tilde{V}_m \in \tilde{G}^q$, $X_{i,m} = 1$. Otherwise, $X_{i,m} = 0$. In the proposed framework for clique-graph matching shown in Fig.2, it is natural to impose one-to-one constraint that makes $\bar{X}$ as a permutation matrix:

$$\bar{X} \cdot 1_{N^q \times 1} \leq 1_{N^p \times 1}, \quad \bar{X}^\top \cdot 1_{N^p \times 1} \leq 1_{N^q \times 1}, \tag{2}$$

where $1_{N^p \times 1}/1_{N^q \times 1}$ denotes an all-ones vector with size N^p/N^q and the inequalities hold for every element. For a convenient representation, we use X to represent the vectorized version of $\bar{X}$.

The problem of clique-graph matching can be formulated as an integer linear programming to discover the indicator vector X^* that maximizes the following objective function:

$$\begin{aligned} &(X, \tilde{G}^p, \tilde{G}^q)^* = \arg \max_{X, \tilde{G}^p, \tilde{G}^q} (S^\top \cdot X), \\ &s.t. \;\; \bar{X} \cdot 1_{N^q \times 1} \leq 1_{N^p \times 1}, \quad \bar{X}^\top \cdot 1_{N^p \times 1} \leq 1_{N^q \times 1} \end{aligned} \tag{3}$$

Till now, we have converted clique-graph matching into the optimization of the objective function above. However, it is non-trivial to optimize Eq.3 directly since both the clique structure information in individual clique-graph and the similarity measure of pairwise cliques from different clique-graphs are latent variables. We will detail the solution of both as follows. After achieving the clique-to-clique similarity, the objective function in Eq.3 can be easily solved by integral linear programming.

1) Clique discovery In our work, individual node of one clique-graph is considered as the center of one clique. We adopt the sparse-subspace clustering method to discover the neighbor nodes of individual center and consequently construct the clique with the center and its neighbors. Supposing individual node can be represented by the feature vector f_i ($f_i \in \mathbb{R}^d$) and all nodes in G can be represented by the feature set, $\{f_i\}_{i=1}^{I}$. Motivated by sparse representation, f_i can be reconstructed by the linear combination of the other nodes $F = \{f_j\}_{j=1,...,I,j\neq i}$:

$$f_i = Fr_i \quad s.t. \quad r_{ii} = 0, \tag{4}$$

where $r_i \triangleq [r_{i1}\ r_{i2} \dots r_{iI}]^\top$; r_{ik}, the k^{th} dimension of coefficient r_i, denotes the weight of the k^{th} node and implicitly means the relatedness between the i^{th} and k^{th} nodes; the constraint $r_{ii} = 0$ avoids the trivial solution of reconstructing f_i by itself.

Motivated by the theory of sparse representation, sparse constraint can be imposed to achieve a sparse solution, whose non-zero entities correspond to the related nodes. Ideally, the L_0-normal of r_i can be intuitively imposed to induce sparsity. However, this will lead to the general NP-hard problem of finding the sparsest representation of the given node. Therefore, we consider minimizing the tightest convex relaxation of the L_0-norm of r_i with L_1-norm. The objective functions can be formulated as:

$$min\|R\|_1 \quad s.t. \quad F = FR,\ diag(R) = 0, \tag{5}$$

where $R \triangleq [r_1\ r_2 \dots r_I]$ is the matrix, whose i^{th} column corresponds to the sparse representation of f_i. Eq.5 can be efficiently solved with convex programming tools [23]. According to the matrix R, we selected the top k nodes which can well reconstruct V_i in the clique $\tilde{V}_i$. Consequently, each compact clique can convey the specific local structure.

2) Clique-wise similarity measure Different from previous work, which usually focuses on measuring the similarity between two hyper-edges by the complicated tensor computation while ignoring the node order on the hyper-edges [10], we convert the problem of clique-wise similarity measure into the problem of graph matching. By achieving the correspondence between the nodes from both graphs, we can effectively discover the node order for similarity computation. Moreover, since the node number of each clique will be far less than the one of the original graph and the structure of each clique is extremely simpler comparing against the structure of the original graph, the clique-wise similarity measure can be formulated only with the unary and pairwise correspondences, which can simplify the complicated high-order correspondence with the low-order correspondence and consequently benefit avoiding the high computational complexity.

We utilize the cosine distance for node-wise similarity measure. It can be calculated by:

$$U_{v_s v_a} = \frac{f_s^p \cdot f_a^q}{||f_s^p||_2 \cdot ||f_a^q||_2}, \tag{6}$$

where f_s^p and f_a^q respectively represent the features of nodes v_s and v_a from $\tilde{V}_i^p$ ($\tilde{V}_i^p \in \tilde{G}^p$) and $\tilde{V}_m^q$ ($\tilde{V}_m^q \in \tilde{G}^q$).

We utilize the distance between two edges, e_{st} and e_{ab} from both cliques, for the pairwise similarity. The similarity can be computed by:

$$E_{e_{st}e_{ab}} = \exp(-|e_{st} - e_{ab}|^2/\epsilon^2), \tag{7}$$

where $\forall e_{st} \in \tilde{V}_i^p$ and $\forall e_{ab} \in \tilde{V}_i^q$, the scaling factor ϵ^2 is set to 0.15 empirically.

With both unary and pairwise similarity measure, the similarity between the $\tilde{V}_i^p \& \tilde{V}_m^q$ cliques, $S_{i,m}$, can be formulated in the quadratic form , which is also constrained by the one-to-one mapping:

$$\begin{aligned} S_{i,m}(Z) = & \tilde{A}_i^p \cdot \tilde{A}_m^q \cdot \{\sum_{v_s v_a} z_{v_s v_a} U_{v_s v_a} + \\ & \sum_{\substack{v_s \neq v_t, \\ v_a \neq v_b, \\ v_s \& v_t \in \tilde{V}_i^p, \\ v_a \& v_b \in \tilde{V}_m^q}} z_{v_s v_t} z_{v_a v_b} E_{e_{st}e_{ab}}\} \\ & s.t. \quad Z \cdot 1_{K_i \times 1} \leq 1_{K_m \times 1}, \quad Z^\top \cdot 1_{K_m \times 1} \leq 1_{K_i \times 1} \end{aligned} \tag{8}$$

where the matrix Z ($Z \in \{0,1\}^{K_i K_m}$) means the node correspondence, i.e., $z_{v_s v_a} = 1$ if the v_s node of the graph $\tilde{G}^p$ corresponds to the v_a node of the graph $\tilde{G}_q$. $U_{v_s v_a}$ and $E_{e_{st}e_{ab}}$ denote the unary and pairwise similarity respectively. $\tilde{A}_i^p \in \tilde{A}^p$ and $\tilde{A}_m^q \in \tilde{A}^q$ are the attributes associated with the corresponding cliques and are imposed as weights; $K_i \& K_m$ respectively denote the node numbers in the $\tilde{V}_i^p \& \tilde{V}_m^q$ cliques.

Since we aim to achieve the similarity of both cliques, the discretization step can be ignored. This objective function can be solved by many state-of-the-art quadratic programming methods. In our work, we implemented the rayleigh quotients maximization method in [15]. By solving Eq.8, we can achieve the optimal clique-wise similarity.

4. Experiment

We term the proposed clique-graph matching method with the clique size H as CGM-H. The proposed method is validated in three scenarios:

- varying the clique size to evaluate its effect on the performance;

- testing CGM-H on the synthetic point set to evaluate the effects of the deformation noise, the outlier number and the edge density on the performances;
- testing CGM-H on the real image datasets to show its performances on the real application.

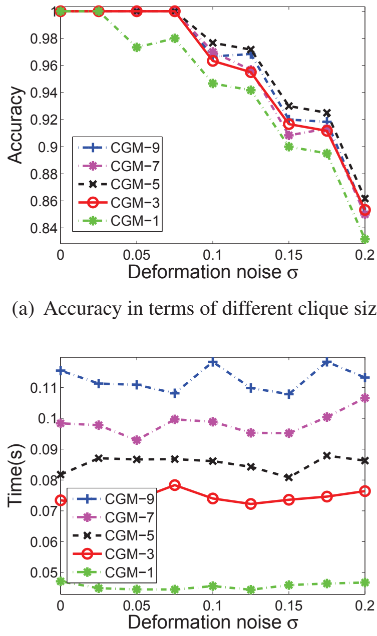
To show the superiority, we compare the proposed method against several representative methods, including SM[11], HGM[20], RRWM[2], RRWHM[10], TM[4], FGM [22], IPFP [12]. For fair comparison, the public codes and the assigned paramters by the authors were utilized. All methods were implemented in Matlab and tested on the PC with single core (CPU: 3.1 GHz; RAM: 8G). For each trial in the experiment on synthetic point data, the same synthetic graphs were shared for all methods. Each quantitative result in the synthetic experiments was achieved by averaging 30 random trials.

4.1. Effect of clique size

Obviously, the clique size have direct influence on the performance of graph matching since different clique size will convey different local structures. In order to select the optimal clique size, we performed cross validation on the synthetic point set. For each trial, we randomly generated two point sets, G^p and G^q with n points respectively. The points of G^p were generated from the uniform distribution within 0 and 1. We created the perturbed point set G^q by adding noise. The deformation noise σ was generated by the Gaussian noise function $N(0, \sigma^2)$. The accuracy is measured by the ratio between the number of the correct matches and the total number of the ground truth. For each trail, we generated 30 points for G^q and G^p respectively. The deformation noise σ was varied from 0 to 0.2 with the step of 0.025. The experimental results are shown in Fig.3.

From Fig.3, we can find that the accuracy will increase by augmenting the clique size before the optimal performance (H=5) is achieved. However, the performance will not monotonically increase. This observation is quite understandable. The proper clique size can well benefit the representation of the local structural information and consequently boost the performance of graph matching. However, either too large or too small clique size can not capture the discriminative characteristics of local structure since too small clique can not well describe the local correlations between the clique centers and their neighbors while too big clique size will increase the chance that different cliques have high overlaps and consequently degrade the discrimination inbetween. By cross validation, the optimal clique size of 5 is achieved and will be empirically set (CGM-5) for the further experiments .

To the best of our knowledge, few literatures show the experiment on hyper-graph matching, in which the order of hyper-graph is over 3, since it is difficult to define scale-invariant similarity measure between two hyper-edges and is extremely time-consuming for computation. In our experiment, we show the speeds of CGM-H with the clique order up to 9 in Fig.3 (b). It is intuitive that the proposed clique-graph matching method only costs acceptable computational complexity even with high order cliques.



(a) Accuracy in terms of different clique sizes

(b) Times in terms of different clique sizes

Figure 3. Experiment on different clique sizes.

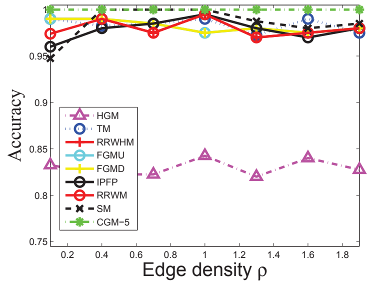


Figure 6. Experiment on different edge densities.

4.2. Evaluation on synthetic dataset

Following the experiment protocol in [10], we performed a comparative evaluation on the synthetic dataset. For each trial, we constructed two graphs, G^p with $n^p = n_{in} + n^p_{out}$ nodes and G^q with $n^q = n_{in} + n^q_{out}$ nodes, each consisting of n_{in} inlier nodes and n^p_{out}/n^q_{out} outlier nodes respectively. The reference graph G^p was generated with random edge density ρ, where each edge e^p_{st} was assigned with a random attribute a^p_{st} uniformly distributing within 0 and 1.

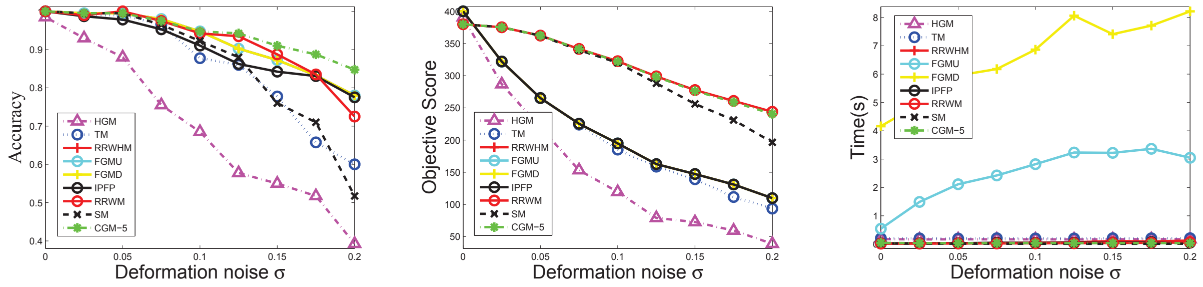


Figure 4. Deformation noise experiment: accuracy, score and speed.

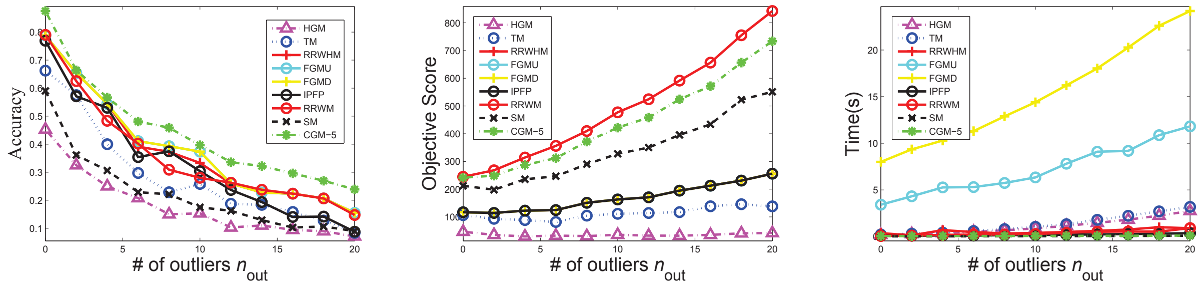


Figure 5. Outlier experiment: accuracy, socre and times.

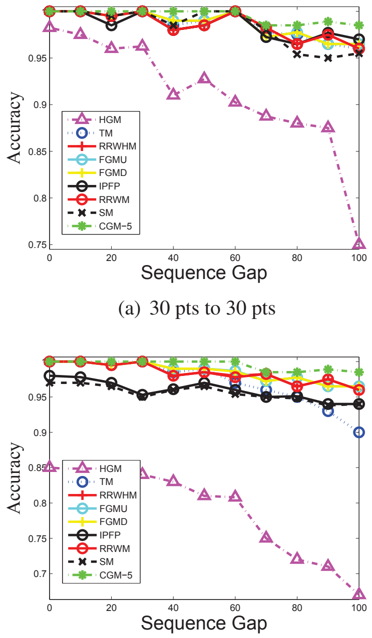
Then we created a perturbed graph G^q by adding noise on the edge attributes between inlier nodes: $a^q_{ab} = a^p_{st} + \sigma$. The deformation noise σ was generated by the Gaussian function $N(0, \sigma^2)$. All the other edges connecting at least one of the outlier nodes were randomly generated in the same way as in G^p. Thus, two graphs G^p and G^q have a common and perturbed sub-graph with n_{in} nodes.

In our experiment, there were three kinds of independent variables: the outlier number n^p_{out}, the deformation noise σ and the edge density ρ. Hence, we conducted three experiments to evaluate their influences on performances respectively. For each parameter setting, we generated 30 different graph pairs and computed the average accuracy and objective score. First, we varied the outlier number from 0 to 20 with the step of 2 while fixing the inlier number $n_{in} = 20$, the deformation noise $\sigma = 0.5$. Second, we varied the deformation noise σ from 0 to 0.2 with the step of 0.025 while fixing the inlier number $n_{in} = 20$, the outlier number $n^p_{out} = n^q_{out} = 0$. Third, we varied the edge density from 0 to 0.1 with the step of 0.02 while fixing the inlier number $n_{in} = 20$ and the outlier number $n^p_{out} = n^q_{out} = 0$.

From Fig.4-6, we can find that the proposed method can outperform all the other state-of-the-art methods in terms of accuracy and time. For graph matching, it is very essential to extract distinct features of individual inlier and outlier nodes for discrimination. Since the inlier and outlier nodes are generated from two different distributions, it is reasonable that the inlier/outlier clique center will have more inlier/outlier neighbors. These cliques can implicitly convey the high-order local structure and benefit the representation of more distinct local attributes for clique graph matching. Therefore the proposed method can be more robust to deal with outliers. Moreover, the proposed method can successfully convert the graph matching problem into a linear programming problem which can greatly reduce the computation complexity.

4.3. Evaluation on CMU House Sequence

We applied the proposed method on the standard CMU House sequence since it has been widely used to evaluate different graph matching algorithms. This sequence consists of 110 frames and 30 feature points are manually labeled consistently across all frames. This allows us to compare the performances of different algorithms over various temporal intervals. The larger temporal interval between adjacent frames will lead to more significant deformation and consequently the graph matching will become more challenging. We matched all possible image pairs with the sequence gap from 10 to 100 frames and the step size of 10 frames. The average matching accuracy per sequence was computed. According to the previous work [10], graph matching for 2 different settings were generated with landmark points as nodes: $(n^p, n^q) = (30, 30), (25, 30)$. For the second setting, we chose 25 points randomly among 30 landmark points. Fig.7 shows the performances by varying the sequence gaps. The proposed approach can produce the competing performances to RRWM and RRWHM and outperform all the others.



(a) 30 pts to 30 pts

(b) 25 pts to 30 pts

Figure 7. Performances on CMU house sequence.

4.4. Evaluation on Real Image Dataset

We first implemented the proposed method on the challenging real image dataset, Caltech+MSRC [10]. The author constructed a dataset of 30 image pairs, which were collected from Caltech-101 and MSRC datasets, and generated candidate correspondences using the MSER detector and the SIFT descriptor. With the distance between pairwise 128-d SIFT descriptors, all the possible candidate matches were collected if the distance of individual pair is less than a loose threshold $\delta = 0.6$, which can allow multiple correspondences for each feature. In [10], the authors manually annotated the groundtruth for all candidate correspondences of each image pair, and the accuracy and relative objective score were computed and compared with several representative methods, including RRWN[2], SM[11], SMAC[3], GAGM[8], IPFP [12]. The results are summarized in Table.1 and some representative examples are shown in Fig.8. We further evaluated the proposed method on the Car and Motorbike image dataset [12], which consists of 20 pairs of cars and 20 pairs of motorbikes selected from VOC PASCAL 2007 with extracted contours, annotated ground truth correspondences and a significant number of outliers. The corresponding experimental results are shown in Table.2. From the comparison experiments in Table.1-2, the proposed method can outperform the others in term of both accuracy. Especially, CGM-5 can significantly augment the speed comparing against RRWM and IPFP.

Table 1. Performance on Caltech+MSRC

Method	Accuracy(%)	Times(s)
CGM-5	74.20	0.07
RRWM[2]	64.01	0.37
SM[11]	52.08	0.05
IPFP[12]	41.2	0.12
SMAC[3]	39.74	-
GAGM[8]	58.74	-

Table 2. Performance on Car+Motor

Car-Method	Accuracy(%)	Time(s)
CGM-5	83.2	0.07
RRWM[2]	82.1	0.29
SM[11]	81.3	0.11
IPFP[12]	81.4	0.46
Motor-Method	Accuracy(%)	Time(s)
CGM-5	87.3	0.08
RRWM[2]	85.6	0.33
SM[11]	80.2	0.17
IPFP[12]	83.1	0.43

5. Conclusion

In this paper, we originally propose the clique-graph and present the clique-graph matching method by preserving global and local structures. We formulate the objective function of clique-graph matching with respective to the clique information in the original graph and the pairwise cliques correspondence. Since the objective function is not jointly convex to both latent variables, we optimize it by decomposing it into two consecutive steps, clique-to-clique similarity measure and the graph-to-graph similarity measure. Consequently, it can preserve both local unary and pairwise correspondences and the global clique-to-clique correspondence. Extensive experiments on the synthetic data and real images show the superiority of the proposed method.

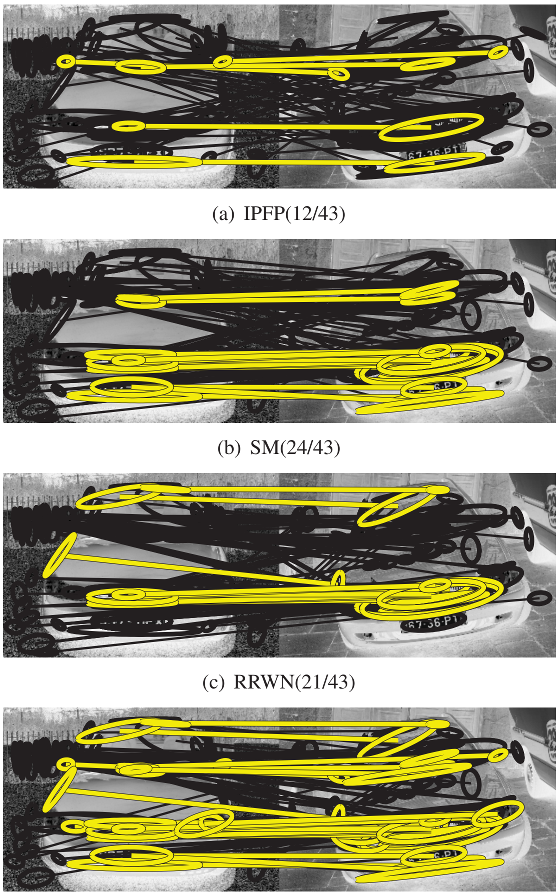
6. Acknowledgment

This work was supported in part by the National Natural Science Foundation of China (61472275, 61170239,61202168), the Tianjin Research Program of Application Foundation and Advanced Technology(15JCYBJC16200), the grant of Elite Scholar Program of Tianjin University (2014XRG-0046).

References

[1] A. Barducci and S. Marinai. Object recognition in floor plans by graphs of white connected components. In *ICPR*, pages 298–301, 2012. 1

[2] M. Cho, J. Lee, and K. M. Lee. Reweighted random walks for graph matching. In *ECCV*, pages 492–505, 2010. 2, 5, 7



(a) IPFP(12/43)

(b) SM(24/43)

(c) RRWN(21/43)

(d) CGM-5(35/43)

Figure 8. Samples of graph matching on Caltech+MSRC. Correct match-yellow line; false match-black line.

[3] T. Cour, P. Srinivasan, and J. Shi. Balanced graph matching. In *NIPS*, pages 313–320, 2006. 7

[4] O. Duchenne, F. Bach, I. Kweon, and J. Ponce. A tensor-based algorithm for high-order graph matching. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(12):2383–2395, 2011. 1, 2, 5

[5] T. S. Gal, T. C. Tucker, A. Gangopadhyay, and Z. Chen. A data recipient centered de-identification method to retain statistical attributes. *Journal of Biomedical Informatics*, 50:32–45, 2014. 1

[6] Y. Gao and Q. Dai. View-based 3d object retrieval: Challenges and approaches. *IEEE MultiMedia*, 21(3):52–57, 2014. 1

[7] Y. Gao, M. Wang, D. Tao, R. Ji, and Q. Dai. 3-d object retrieval and recognition with hypergraph analysis. *TIP*, 21(9):4290–4303, 2012. 2

[8] S. Gold and A. Rangarajan. A graduated assignment algorithm for graph matching. *IEEE Trans. Pattern Anal. Mach. Intell.*, 18(4):377–388, 1996. 1, 2, 3, 7

[9] M. Gori, M. Maggini, and L. Sarti. Exact and approximate graph matching using random walks. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27(7):1100–1111, 2005. 2

[10] J. Lee, M. Cho, and K. M. Lee. Hyper-graph matching via reweighted random walks. In *CVPR*, pages 1633–1640, 2011. 1, 2, 4, 5, 6, 7

[11] M. Leordeanu and M. Hebert. A spectral technique for correspondence problems using pairwise constraints. In *ICCV*, pages 1482–1489, 2005. 1, 2, 3, 5, 7

[12] 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 22: 23rd Annual Conference on Neural Information Processing Systems 2009. Proceedings of a meeting held 7-10 December 2009, Vancouver, British Columbia, Canada.*, pages 1114–1122, 2009. 1, 3, 5, 7

[13] K. Lu, R. Ji, J. Tang, and Y. Gao. Learning-based bipartite graph matching for view-based 3d model retrieval. *IEEE Transactions on Image Processing*, 23(10):4553–4563, 2014. 1

[14] W. Nie, A. Liu, Y. Su, H. Luan, Z. Yang, L. Cao, and R. Ji. Single/cross-camera multiple-person tracking by graph matching. *Neurocomputing*, 139:220–232, 2014. 1

[15] T.Cour, P.Srinivasan, and J.Shi. Balanced graph matching. In *NIPS*, 2006. 4

[16] B. J. van Wyk and M. A. van Wyk. A pocs-based graph matching algorithm. *IEEE Trans. Pattern Anal. Mach. Intell.*, 26(11):1526–1530, 2004. 1, 2, 3

[17] C. Wang, L. Wang, and L. Liu. Improving graph matching via density maximization. In *ICCV*, pages 3424–3431, 2013. 2

[18] J. Yan, Y. Tian, H. Zha, X. Yang, Y. Zhang, and S. M. Chu. Joint optimization for consistent multiple graph matching. In *ICCV*, pages 1649–1656, 2013. 2

[19] M. Zaslavskiy, F. R. Bach, and J. Vert. A path following algorithm for the graph matching problem. *IEEE Trans. Pattern Anal. Mach. Intell.*, 31(12):2227–2242, 2009. 2

[20] R. Zass and A. Shashua. Probabilistic graph and hypergraph matching. In *CVPR*, 2008. 1, 2, 5

[21] Q. Zhang, X. Song, X. Shao, H. Zhao, and R. Shibasaki. Learning graph matching: Oriented to category modeling from cluttered scenes. In *ICCV*, pages 1329–1336, 2013. 2

[22] F. Zhou and F. D. la Torre. Factorized graph matching. In *CVPR*, pages 127–134, 2012. 5

[23] M. Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *ICML*, pages 928–936, 2003. 4