Subgraph Matching using Compactness Prior for Robust Feature Correspondence

Yumin Suh
n12345@snu.ac.kr

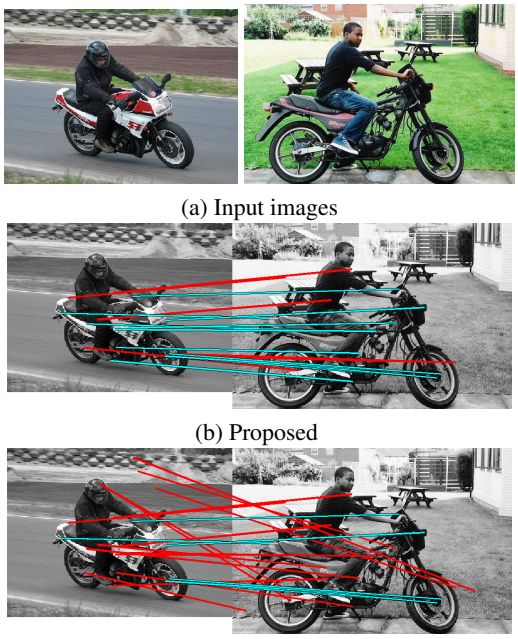
Kamil Adamczewski
kamil.m.adamczewski@gmail.com

Kyoung Mu Lee
kyoungmu@snu.ac.kr

Department of ECE, ASRI, Seoul National University, Seoul, Korea

Abstract

Feature correspondence plays a central role in various computer vision applications. It is widely formulated as a graph matching problem due to its robust performance under challenging conditions, such as background clutter, object deformation and repetitive patterns. A variety of fast and accurate algorithms have been proposed for graph matching. However, most of them focus on improving the recall of the solution while rarely considering its precision, thus inducing a solution with numerous outliers. To address both precision and recall feature correspondence should rather be formulated as a subgraph matching problem. This paper proposes a new subgraph matching formulation which uses a compactness prior, an additional constraint that prefers sparser solutions and effectively eliminates outliers. To solve the new optimization problem, we propose a meta-algorithm based on Markov chain Monte Carlo. By constructing Markov chain on the restricted search space instead of the original solution space, our method approximates the solution effectively. The experiments indicate that our proposed formulation and algorithm significantly improve the baseline performance under challenging conditions when both outliers and deformation noise are present.



(a) Input images

(b) Proposed

(c) Graph matching [30]

Figure 1. An example of feature matching. True positive matches are represented by blue lines and false positive matches are represented by red lines.

1. Introduction

Feature correspondence plays a central role in various computer vision applications such as object detection [28], object recognition [6, 3], and tracking [12]. It is a challenging problem because background clutter, appearance variation, and repetitive patterns frequently occur in real world images, thus reducing the informativeness of the local descriptors. Therefore, the geometric relationship between features is commonly used to compensate for the unreliable match scores obtained from local descriptors. Graph matching provides a powerful tool for feature correspondence by modeling feature appearances and their relationships as a graph. This paper proposes a new formulation and algorithm for subgraph matching that enables robust feature correspondence under clutter and appearance variation.

Graph matching has been widely formulated as an Integer Quadratic Programming (IQP) due to its representation power [11, 15, 16, 27, 8, 30]. Compared with other formulations, that allow only simple dot product or squared error as affinity/distance measures [4], IQP can model more complex affinity measures [5]. Also, it is extendable to higher-order graph matching, which uses

higher-order edges [9, 14]. Nonetheless when applied to feature correspondence problem, IQP formulation has two fundamental problems (Section 3.1). (1) Since the solution of IQP is predetermined to have certain number of matches, independently of the number of outliers in the graph, the solution may contain many outliers. As a result, recall has become a prevalent measure of reporting accuracy [6, 11, 15, 16, 27, 8, 5, 25, 30], whereas precision has not been considered. (2) The outliers contained in the solution make the objective function non-discriminative. When outliers are present, IQP objective not only lowers the precision but also the recall of the solution by preferring to select a solution with large number of outliers, instead of one with a small number of inliers, as shown in Figure 1(c).

In this paper, we propose a new graph matching method to solve the aforementioned problems. First, we augment the original IQP objective with additional compactness prior, in which its parameters are discriminatively trained to maximize both precision and recall. Additional compactness prior eliminates outliers while collecting inliers by preferring a compact solution. Then, we propose a new meta-algorithm based on Markov chain Monte Carlo (MCMC), which builds on existing graph matching algorithms as its *core algorithm*, in order to solve the new optimization problem. Constructing Markov chain on the reduced search space instead of the original solution space effectively approximates the solution. In the experiments, the proposed solution consistently and significantly improves the baseline algorithms in the practical conditions in which both outliers and deformation noise exist.

2. Related Works

Several works on graph matching addressed the aforementioned two problems (Section 1) that occur due to the presence of outliers. The following strategies have been proposed: changing formulation, designing a robust algorithm, and restriction to specific problems such as feature correspondence.

Subgraph matching is a method that directly addresses the first problem by finding shared structure of the two input graphs, where structure includes node and edge attributes when dealing with attributed graphs. Although subgraph matching of basic graphs has been widely studied (e.g. subgraph isomorphism and its extension to labeled graphs [1], which are attributed graphs with small number of discrete labels and costs between them), subgraph matching of attributed graphs with general attributes has been rarely addressed in the literature. Pavan et al. [24] and Liu et al. [18, 19] addressed the attributed subgraph matching by designing the objective that maximizes the average intra-cluster affinity, in contrast to maximizing the total affinity used in the original IQP. Since the affinity between inlier and outlier matches is expected to be lower than that between inliers, outliers usually lower the average affinity of the matches. Therefore, maximizing average affinity effectively eliminates outliers from the solution. However, when there exists deformation noise, small but highly correlated outliers may score higher objective than large but loosely correlated inliers, resulting in low recall.

Most of the graph matching algorithms based on IQP attempt to solve the second problem by proposing robust algorithms, while leaving the objective unchanged. Gold and Rangarajan [11] iteratively estimated the convex approximation around the previous solution and Cho et al. [5] exploited random walk to explore the relaxed solution space effectively. Zhou et al. [30] relaxed the original objective into concave and convex forms and solve them using path following algorithm. Cho et al. [7] reduced the effect of outliers by using max-pooling in graph matching, and in their other work Cho et al. [6] significantly enhanced recall using progressive framework that iteratively constructs graphs and finds matches. These methods robustly work with deformation and outliers, however, because of the inherit constraints of the IQP formulation, they still suffer from a large number of outliers in the solution.

Finally, graph matching research as restricted to solving feature correspondance problem further addressed the issue of outliers. Torresani et al. [26] modeled feature correspondence problem by means of a Markov Random Field (MRF), using potentials related to appearance, geometry, occlusion, and feature coherence. The occlusion term used in their formulation corresponds to our compactness prior, enforcing to increase the number of matches in the solution while minimizing the discrepancy. However, their objective is restricted to feature matching, whereas our formulation can be used for general graph matching problems.

Our paper contributes to the literature in three aspects. Firstly, we propose a new graph matching formulation for subgraph matching which uses compactness prior, thus addressing the precision and recall at the same time. Secondly, an MCMC-based meta-algorithm is proposed to solve the new optimization problem. Finally, our method demonstrates the state-of-the-art performance on graph matching when both deformation and outlier exist. This property enables robust feature matching of real images.

3. Graph Matching Model

Let us consider two attributed graphs, $\mathcal{G}_1 = (\mathcal{V}_1, \mathcal{E}_1, \mathcal{A}_1^{\mathcal{V}}, \mathcal{A}_1^{\mathcal{E}})$ with n_1 nodes and $\mathcal{G}_2 = (\mathcal{V}_2, \mathcal{E}_2, \mathcal{A}_2^{\mathcal{V}}, \mathcal{A}_2^{\mathcal{E}})$ with n_2 nodes, where $\mathcal{V}$, $\mathcal{E}$, $\mathcal{A}^{\mathcal{V}}$ and $\mathcal{A}^{\mathcal{E}}$ denote a set of nodes, edges, node attributes and edge attributes, respectively. In this paper, one-to-one matching constraints are adopted so that every node in $\mathcal{G}_1$ is mapped to at most one node in $\mathcal{G}_2$ and vice versa. *Graph matching* identifies the subset of node correspondences between $\mathcal{G}_1$ and $\mathcal{G}_2$ among all of the possible correspondences, which best preserve the

attribute relations under the matching constraints.

We use binary indicator vector $\mathbf{y} \in \{0,1\}^{n_1 n_2}$ to represent the correspondence between $\mathcal{G}_1$ and $\mathcal{G}_2$; $\mathbf{y}_i$ is 1 if i-th match is selected and 0, otherwise. *Affinity matrix* $\mathbf{M}$ consists of the relational similarity values between edges and nodes; the compatibility of two edge attributes $\mathbf{a}_{ij} \in \mathcal{A}_1^{\mathcal{E}}$ and $\mathbf{a}_{ab} \in \mathcal{A}_2^{\mathcal{E}}$ is encoded in the non-diagonal component $\mathbf{M}_{ia;jb}$, and the compatibility of two node attributes $\mathbf{a}_i \in \mathcal{A}_1^{\mathcal{V}}$ and $\mathbf{a}_a \in \mathcal{A}_2^{\mathcal{V}}$ is encoded in the diagonal component $\mathbf{M}_{ia;ia}$.[1]

3.1. IQP Formulation

In the conventional IQP formulation, the quality of a matching between two graphs is measured as the sum of affinity values between all pairs of nodes and edges given by the correspondence. The graph matching is modeled as follows:

$$\begin{aligned} \mathbf{y}^* &= \arg\max_{\mathbf{y}} \mathbf{y}^T \mathbf{M} \mathbf{y} \\ s.t. \quad & \mathbf{y} \in C_{int} \cap C_{one}, \end{aligned} \tag{1}$$

where $C_{int} = \{\mathbf{y} | \mathbf{y} \in \{0,1\}^{n_1 n_2}\}$ enforces the integer constraint and $C_{one} = \{\mathbf{y} | \forall i, \sum_a \mathbf{y}_{ia} \leq 1\} \cap \{\mathbf{y} | \forall a, \sum_i \mathbf{y}_{ia} \leq 1\}$ enforces the one-to-one constraint.

The affinity measure is typically restricted to be nonnegative [11, 15, 16, 5, 25], therefore the number of matches in the final solution is implicitly determined to be the maximum possible number of matches under one-to-one constraint, that is $min(n_1, n_2)$ because more matches result in higher objective. [2] The underlying assumption is that the solutions that contain true matches would have a higher objective value than those do not. However, this assumption poorly holds even with few outliers as shown in the following test.

Let us consider a simple example of graph matching in the presence of outliers. We construct two graphs, each with five inlier nodes. The details of generating those random graphs are described in Section 6.1. Three sets of tests are performed where the number of outliers is 0, 2 and 4, respectively. The results are shown in Figure 2. Each test is illustrated by two bars which represent the objective score, S, of the match $\mathbf{y}$. The left bar represents the *ground truth+*, which is the set of matches that maximizes the objective (1) while containing the true matches $\mathbf{y}_{GT}$. The right bar represents the results obtained via maximizing the IQP objective (1). Each objective score consists of the two parts, S_{in} and S_{out}, i.e., $S = S_{in} + S_{out} = \mathbf{y}^T \mathbf{M} \mathbf{y}$. The value

[1] In a vector $\mathbf{x}$, we denote its i-th component as $\mathbf{x}_i$, yet in the case of the indicator vector $\mathbf{y}$, we slightly abuse the notation; $\mathbf{y}_{ia}$ denotes the component of indicator vector $\mathbf{y}$ that corresponds to the match between i-th node in $\mathcal{G}_1$ and a-th node in $\mathcal{G}_2$.

[2] Even when affinity is allowed to have negative values, its effect is equivalent to thresholding the original solution, which results in a move along the Pareto curve, not the move of the curve itself.

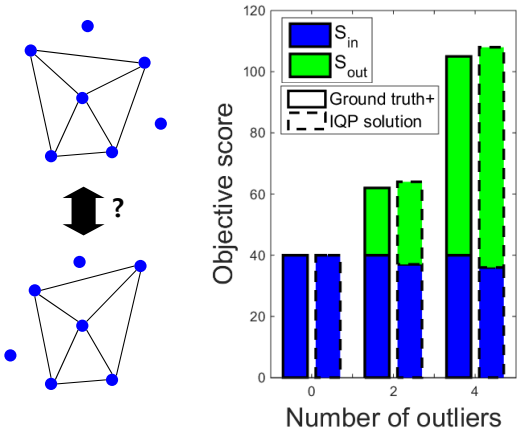


Figure 2. An example of graph matching and the objective scores. Three tests are performed varying the number of ouliers from 0 to 4. The left and right bar represent the objective score of the *ground truth+* and IQP solution, respectively. It shows that maximizing IQP is not necessarily related to the detection of inliers. Refer to the main text for details.

$S_{in} = \hat{\mathbf{y}}^T \mathbf{M} \hat{\mathbf{y}}$, represented by blue color, is the score generated by the matches given by $\hat{\mathbf{y}}$, where $\hat{\mathbf{y}}$ is the indicator vector of the top five matches in the matching vector $\mathbf{y}$. Top matches are measured by the sum of affinity between every possible pairs. The value S_{out}, represented by the green color, is the remaining portion of the score, i.e. $S_{out} = \mathbf{y}^T \mathbf{M} \mathbf{y} - \hat{\mathbf{y}}^T \mathbf{M} \hat{\mathbf{y}}$.

In all three tests, top five matches of *ground truth+* are true, i.e., S_{in} is equal to $\mathbf{y}_{GT}^T \mathbf{M} \mathbf{y}_{GT}$. In contrast, S_{in} of IQP solution is smaller than that of *ground truth+*, which means that IQP solution does not contain all true matches. However , the score of IQP solution S is higher than that of *ground truth+*. It shows that solving IQP is not necessarily related to the detection of the inliers. This observation indicates that IQP formulation may not only lower the precision but also recall.

3.2. Compactness Augmented Model

To overcome the problem of the conventional IQP, we propose to augment the original objective of the IQP with an additional compactness prior, which adjusts number of matches in the solution as a function of the number of matches, $\|\mathbf{y}\|_1$. The augmented optimization problem is as follows:

$$\begin{aligned} \mathbf{y}^* &= \arg\max_{\mathbf{y}} \mathbf{y}^T \mathbf{M} \mathbf{y} - \lambda^T \Omega(\|\mathbf{y}\|) \\ & s.t. \quad \mathbf{y} \in C_{int} \cap C_{one}, \end{aligned} \tag{2}$$

where $\Omega(\|\mathbf{y}\|) = [\omega_1(\|\mathbf{y}\|), \omega_2(\|\mathbf{y}\|), \cdots, \omega_n(\|\mathbf{y}\|)]$ is a vector, and $\lambda = [\lambda_1, \cdots, \lambda_n]^T$ is the parameter vector that adjusts the weight of each component. Though the learning described in Section 4 is applicable to general Ω, in this paper, we used $\Omega(\|\mathbf{y}\|) = [\|\mathbf{y}\|, \|\mathbf{y}\|^2]$ for simplicity. As in

the conventional IQP, the affinity matrix $\mathbf{M}$ is constrained to be element-wise nonnegative.

The objective function enforces the matches to have high total affinity by the first term, while enforcing the proper number of matches by the second term. The level of compactness is determined by the parameter λ, whose value depends both on affinity measure and graph size. Note that our formulation has different meaning from introducing dummy nodes used in [11], whose purpose is to balance the size of two graphs while exhibiting nearly zero effect on enhancing accuracy.

4. Learning Compactness Prior

In this section we determine the most favorable form of the compactness prior for a given problem setting, which is equivalent to finding the optimal values λ^*. We use the training set $S_t = \{(\mathcal{G}_1^{(j)}, \mathcal{G}_2^{(j)}), \mathbf{y}^{(j)}\}_{j=1}^{N_t}$, where j-th sample consists of two graphs and the true correspondence between them. The parameter λ is trained to maximize the margin as follows:

$$\lambda^* = \min_{\lambda} r(\lambda) + \frac{C}{N_t} \sum_{j=1}^{N_t} \Delta(\mathbf{y}^{(j)}, \hat{\mathbf{y}}(\mathcal{G}_1^{(j)}, \mathcal{G}_2^{(j)}, \lambda)), \quad (3)$$

where $r(\cdot)$ is a regularizer and $\hat{\mathbf{y}}(\mathcal{G}_1^{(j)}, \mathcal{G}_2^{(j)}, \lambda)$ is the estimated solution of (2) with respect to the corresponding affinity matrix $\mathbf{M}^{(j)}$. Hamming loss $\Delta(\mathbf{y}, \mathbf{y}') = \|\mathbf{y} - \mathbf{y}'\|_1/(n_1 n_2)$, is used as the loss function to address both the precision and recall at the same time.

We solve the optimization problem (3) using standard structured support vector machine (SSVM) learning technique, as it was used for learning graphs in [2, 3]. Since $\mathbf{y}^*$ is invariant to constant multiplicaition of objective in (2), λ^* can be obtained by solving the SSVM as follows:

$$\min_{\tilde{\lambda}} \frac{1}{2}\|\tilde{\lambda}\|^2 + \frac{C}{N_t} \sum_{n=1}^{N_t} \Delta(\mathbf{y}^{(j)}, \hat{\mathbf{y}}(\mathcal{G}_1^{(j)}, \mathcal{G}_2^{(j)}, \tilde{\lambda})), \quad (4)$$

where $\hat{\mathbf{y}}(\mathcal{G}_1, \mathcal{G}_2, \tilde{\lambda}) = \arg\max_{\mathbf{y}} \tilde{\lambda}^T \Phi(\mathcal{G}_1, \mathcal{G}_2, \mathbf{y})$ and the joint feature map $\Phi(\mathcal{G}_1, \mathcal{G}_2, \mathbf{y}) = [\mathbf{y}^T\mathbf{M}\mathbf{y}, -\Omega(\|\mathbf{y}\|)] = [\mathbf{y}^T\mathbf{M}\mathbf{y}, -\|\mathbf{y}\|, -\|\mathbf{y}\|^2]$.$\tilde{\lambda}$ is a constant multiplication of $[1, \lambda] = [1, \lambda_1, \lambda_2]$.

Note that learning the parameter λ using Eq.(3) ensures the augmented prior in the proposed formulation (2) to be at least not harmful. If there is no non-zero λ ($\lambda \neq \mathbf{0}$) that further reduces the objective Eq.(3) then λ is learned to be $\mathbf{0}$, while making the Eq.(2) equivalent to the original IQP of (1). In our experiments, under the problem setting of Section 6.1, which models general graph matching, the learnt value is $\lambda = [0.01, 0.69]$. Since $\lambda_1 \ll \lambda_2$, we approximate λ as $\lambda = [0, \lambda_2]$ in all the experiments.

5. Algorithm

We propose an algorithm based on Markov chain Monte Carlo (MCMC) sampling technique to solve the optimization problem (2). The basic idea behind MCMC is to construct a Markov chain on the state space $\mathcal{X}$ whose stationary distribution is the target density $P(\mathbf{x})$ of interest [23].

We define a state space as $\mathcal{X} = \{\mathbf{x} | \mathbf{x} \in \{0, 1\}^{n_1}\}$, where n_1 is the number of nodes in the graph $\mathcal{G}_1$. Each state $\mathbf{x} \in \mathcal{X}$ corresponds to the activation of nodes in graph $\mathcal{G}_1$; $\mathbf{x}_i = 1$ if i-th node in $\mathcal{G}_1$ is active and 0, otherwise. We can optimize the objective function (2) by defining the target probability distribution on $\mathcal{X}$ as follows:

$$P(\mathbf{x}) \propto \pi(\mathbf{x}) = \exp(y(\mathbf{x})^T\mathbf{M}y(\mathbf{x}) - \lambda^T\Omega(\|y(\mathbf{x})\|), \quad (5)$$

where function $y(\cdot)$ on $\mathbf{x}$ is defined as

$$y(\mathbf{x}) = \arg\max_{\mathbf{y}} \mathbf{y}^T\mathbf{M}\mathbf{y} \quad (6)$$
$$s.t. \quad \mathbf{y} \in C_{int} \cap C_{one} \cap C_{\mathbf{x}},$$

and $C_{\mathbf{x}} = \{\mathbf{y} | \sum_a \mathbf{y}_{ia} = 1, \forall i \, s.t. \mathbf{x}_i = 1\} \cap \{\mathbf{y} | \sum_a \mathbf{y}_{ia} = 0, \forall i \, s.t. \mathbf{x}_i = 0\}$ restricts $y(\mathbf{x})$ to have matches only for the active nodes of $\mathbf{x}$. In other words, $y(\mathbf{x})$ is the solution of conventional IQP between two graphs, $\mathcal{G}_1'$ and $\mathcal{G}_2$, where $\mathcal{G}_1'$ is a subgraph of $\mathcal{G}_1$ with active nodes of $\mathbf{x}$. Therefore, one transition in the Markov chain corresponds to a selection of active nodes in the graph $\mathcal{G}_1$ with its best graph matching with $\mathcal{G}_2$, according to the original IQP (1).

The overall algorithm is presented as Algorithm 1. It effectively approximates the solution by constructing Markov chain on $\mathcal{X}$, which is much smaller than the original solution space $\mathcal{Y}$. Note that when outliers exist on only one side among two graphs, the solution of IQP gets much more accurate by circumventing the problems mentioned in the Section 3.1. We use an existing graph matching algorithm to solve Eq. (6).

5.1. State Proposal

The Metropolis-Hastings algorithm [10] is used to generate the Markov chain. We attempt two different state proposals, namely, random proposal and data-driven proposal.

In random proposal, one node is randomly selected from $\mathcal{V}_1$ and flipped from active to inactive or vice versa:

$$q_{rand}(\mathbf{x} \to \mathbf{x}') = \begin{cases} 1/n_1 & \text{if} \quad \|\mathbf{x} - \mathbf{x}'\|_1 = 1, \mathbf{x}' \in \mathcal{X} \\ 0 & \text{otherwise} \end{cases} \quad (7)$$

At most n_1 steps are required to convert from one state to another in the worst case, which suggest it actively wanders around the state space.

The data-driven proposal is designed based on the intuition that true feature points are likely to be closer together in the image, because they are extracted from the same object. The proposal consists of two steps: First we select the

mode whether to *add a node to* or *delete a node from* activated nodes of the state $\mathbf{x}$ according to the probability p_{add}, and then sample a particular node according to the probability q_{add} or q_{del}. When *add* mode is selected, we sample a node from within inactive nodes, and activate it according to the following probability distribution,

$$q_{add}(\mathbf{x} \to \mathbf{x}') = \begin{cases} \frac{1}{Z} e^{-d(\mathbf{x},\mathbf{x}')} & \text{if} \quad \|\mathbf{x}'\|_1 = \|\mathbf{x}\|_1 + 1 \\ & \text{and} \quad \mathbf{x}' \in \mathcal{X} \\ 0 & \text{otherwise,} \end{cases} \tag{8}$$

where Z is a normalizing constant and $d(\mathbf{x}, \mathbf{x}')$ is the distance between the i-th node and geometric center of the active nodes of $\mathbf{x}$ in the image, when $\mathbf{x}'_i = \mathbf{x}_i + 1$. When *delete* mode is selected, an active node is selected and deactivated with uniform probability,

$$q_{del}(\mathbf{x} \to \mathbf{x}') = \begin{cases} 1/\|\mathbf{x}\|_1 & \text{if} \quad \|\mathbf{x}'\|_1 = \|\mathbf{x}\|_1 - 1 \\ & \text{and} \quad \mathbf{x}' \in \mathcal{X} \\ 0 & \text{otherwise.} \end{cases} \tag{9}$$

Algorithm 1: Graph matching via MCMC

input : Affinity matrix $\mathbf{M}$
output: Assignment vector $\mathbf{y}^*$

Generate initial state $\mathbf{x}$ randomly
Initialize N, T
while $T > T_f$ **do**
  Calculate proposal distribution $q(\mathbf{x} \to \mathbf{x}')$ (Eq. 7 or 10)
  Sample $\mathbf{x}'$ from q
  Calculate acceptance ratio $a(\mathbf{x} \to \mathbf{x}')$ (Eq. 11)
  if $rand() < a(\mathbf{x} \to \mathbf{x}')$ **then**
    $\mathbf{x} \leftarrow \mathbf{x}'$
    if $\pi(\mathbf{x}') > \pi(\mathbf{x}^*)$ *(Eq. 5)* **then**
      $\mathbf{y}^* \leftarrow y(\mathbf{x}')$ (Eq. 6)
      $\mathbf{x}^* \leftarrow \mathbf{x}$
    end
  end
  $T = \gamma T$
  if $N > N_{max}$ **then**
    break
  end
end
return $\mathbf{y}^*$

Finally, the data driven proposal is designed as follows:

$$q_{data}(\mathbf{x} \to \mathbf{x}') = \begin{cases} p_{add} \cdot q_{add}(\mathbf{x} & \to \mathbf{x}') \\ & \text{if } \|\mathbf{x}'\|_1 = \|\mathbf{x}\|_1 + 1 \\ & \text{and} \quad \mathbf{x}' \in \mathcal{X} \\ (1 - p_{add}) \cdot & q_{del}(\mathbf{x} \to \mathbf{x}') \\ & \text{if } \|\mathbf{x}'\|_1 = \|\mathbf{x}\|_1 - 1 \\ & \text{and} \quad \mathbf{x}' \in \mathcal{X} \\ 0 & \text{otherwise.} \end{cases} \tag{10}$$

To ensure that the Markov chain satisfies the detailed balance condition, the acceptance ratio is obtained as follows:

$$a(\mathbf{x} \to \mathbf{x}') = \min\left(\frac{\pi(\mathbf{x}')}{\pi(\mathbf{x})} \frac{q(\mathbf{x}' \to \mathbf{x})}{q(\mathbf{x} \to \mathbf{x}')}, 1\right), \tag{11}$$

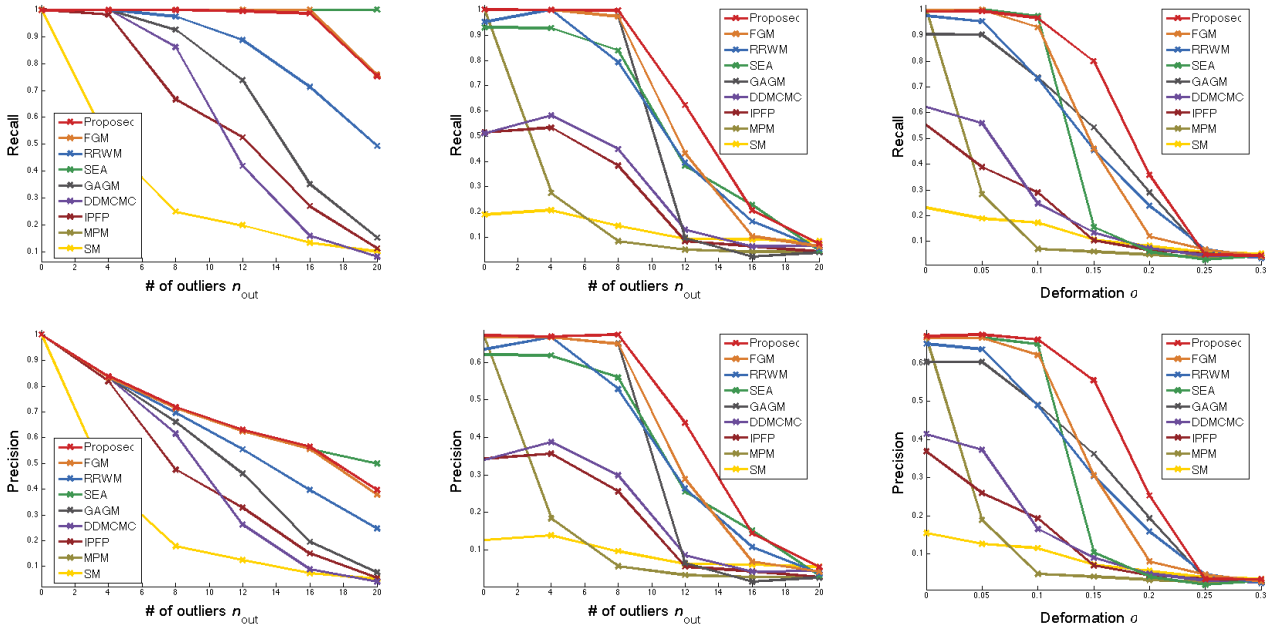
where $q(\cdot)$ is defined as Eq.(7) or Eq.(10), according to the proposal type. We set $p_{add} = 0.5$ in all the experiments.

6. Experiments

This section consists of three parts. In the first part, we evaluate our method on synthetic random graph matching by varying the number of outliers and the amount of deformation noise. We compare the result to eight state-of-the-art graph matching methods, namely, FGM [30], RRWM [5], SEA [18], GAGM [11], DDMCMC [13], IPFP [16], MPM [7] and SM [15]. Since SEA [18] outputs several solutions, we choose the one that maximizes their objective as their output, for fair comparison with other methods. In the second part, we evaluate our method in feature correspondence problems using real images. We compare the result to four graph matching methods, namely, FGM [29], RRWM [5], IPFP [16], and SEA [18]. In the last part, we analyze the effect of changing parameters λ and varying the *core algorithm*.

6.1. Synthetic Random Graph Matching

In this experiment, we construct two graphs, $\mathcal{G}_1$ and $\mathcal{G}_2$, with $n = n_{in} + n_{out}$ nodes at each trial, where n_{in} and n_{out} denote the number of inlier and outlier nodes, respectively. First, the reference graph $\mathcal{G}_1$ is constructed where its edge attributes $\mathbf{a}_{ij} \in \mathcal{A}_1^{\mathcal{E}}$ are randomly assigned from the uniform distribution $[0, 1]$. The target graph $\mathcal{G}_2$ is subsequently constructed by perturbing the edge attributes of inliers in $\mathcal{G}_1$ with Gaussian noise $\epsilon \sim N(0, \sigma_s^2)$, that is, $\mathbf{a}_{i'j'} = \mathbf{a}_{ij} + \epsilon$, where $\mathbf{a}_{i'j'} \in \mathcal{A}_2^{\mathcal{E}}$. The resulting graphs, $\mathcal{G}_1$ and $\mathcal{G}_2$, share a common subgraph with n_{in} nodes. We use the negative exponential of square distance as an affinity measure, thus the affinity matrix M is calculated by assigning $M_{ia,jb} = \exp(-||\mathbf{a}_{ij} - \mathbf{a}_{ab}||/\sigma_s^2)$. The diagonal components of M are set to zero, because no node attributes are used. Scaling factor σ_s^2 is chosen to be 0.1 according to the setting in [5].

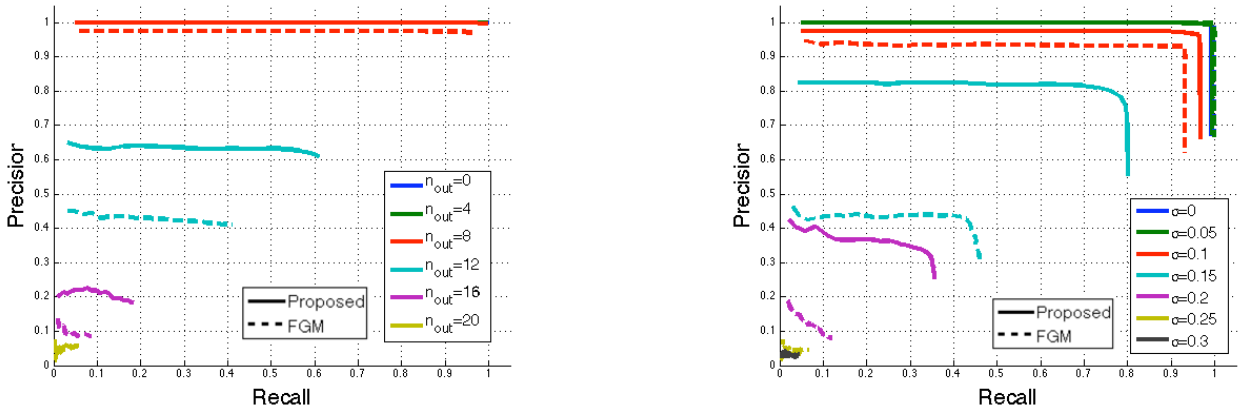


(a) Varying number of outliers without deformation noise

(b) Varying number of outliers with deformation noise

(c) Varying deformation noise with outliers

Figure 3. Synthetic graph matching experiments. The recall (top) and precision (bottom) are plotted for three different conditions: (a) varying the number of outliers with no deformation noise, (b) varying the number of outliers with deformation noise $\sigma = 0.1$, and (c) varying the amount of deformation with 10 outliers.



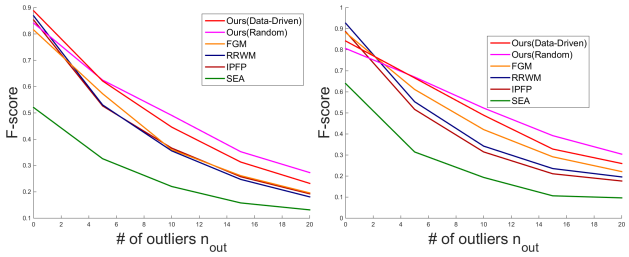
(a) Varying number of outliers

(b) Varying deformation

Figure 4. Precision-recall curves are plotted for two different conditions: (a) varying the number of outliers with fixed deformation noise $\sigma = 0.15$ and (b) varying the amount of deformation noise with fixed number of outliers $n_{out} = 10$.

We evaluate our method under three different experimental settings, namely, (1) varying the number of outliers with no deformation, (2) varying the number of outliers with some deformation, and (3) varying the amount of deformation with some outliers. In all of the experiments, we fix the number of inliers n_{in} to be 20 and use RRWM as a *core algorithm* of the proposed meta-algorithm. We independently generate 40 graph matching problems for each experimental setting and record the average recall and precision. Figure 3(a) shows the results of varying the number of outliers n_{out} from 0 to 20 by increments of 4 while fixing the deformation noise $\sigma = 0$. Figure 3(b) shows the results of varying the number of outliers n_{out} from 0 to 20 by increments of 4 while fixing deformation noise $\sigma = 0.1$. Figure 3(c) shows the results of varying the deformation noise σ from 0 to 0.3 by increments of 0.05 while fixing the number of outliers $n_{out} = 10$.

Our method achieves the best performance in both recall



(a) Winebottle (b) Face

Figure 5. The result of feature matching experiments on (a) winebottle and (b) face class. The F-score is plotted while varying the number of outliers.

and precision when deformation noise exists, as shown in Figure 3 (b) and (c). As shown in Figure 3 (a), our method slightly gives way to MPM and SEA, achieving similar performance as FGM in the setting when only outliers exist with no deformation present. MPM and SEA perform robustly in the presence of outliers, however, their performance rapidly falls as deformation noise increases, whereas the proposed method performs consistently well. As shown in Figure 3 (b) and (c), our method works robustly when both deformation and outliers exist, which is an inseparable condition in the real world images. Using FGM as a *core algorithm* is expected to further improve the performance as our method consistently improves the core method as shown in Figure 7 (a). However, we use RRWM as the core algorithm instead of FGM, because the running time of FGM is more than hundred times longer than that of RRWM given the codes provided by the authors.

Figure 4 (a) and (b) present the precision-recall curve for the experiments in Figure 3 (b) and (c), respectively. The curve is obtained by greedily deleting most unreliable matches one by one from the solution and averaging precision for each recall. Reliability is measured as the sum of affinity values between the corresponding match and the remaining inlier matches. We present the precision-recall curve for the proposed algorithm and FGM. Only the result of FGM, which achieved the most robust performance after ours, is plotted for visibility. The curve of the proposed method is placed up and to the right of FGM curve, which means that our method performs better than FGM, even after the outlier removal by greedy post-processing.

6.2. Feature Correspondence

This part of the experiments focuses on evaluating our method on the real image feature correspondence task. Although the SNU dataset [5] and the Car and Motorbike dataset [17] are frequently used in the graph matching literature, they contain only 30 and 50 pairs of images, respectively. Therefore, we created 1000 pairs of images using Willow object class dataset [3], which provides images for five classes and their key point annotations. We randomly

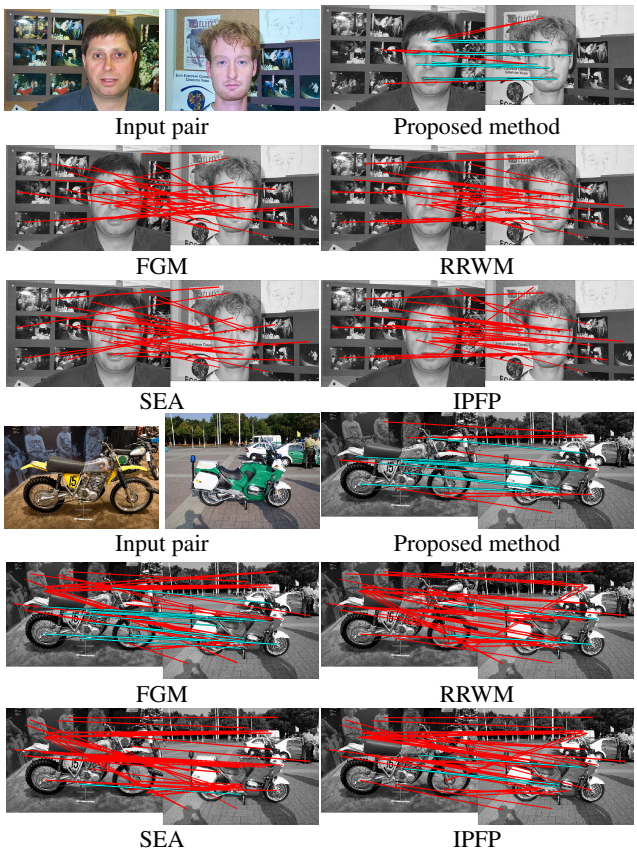
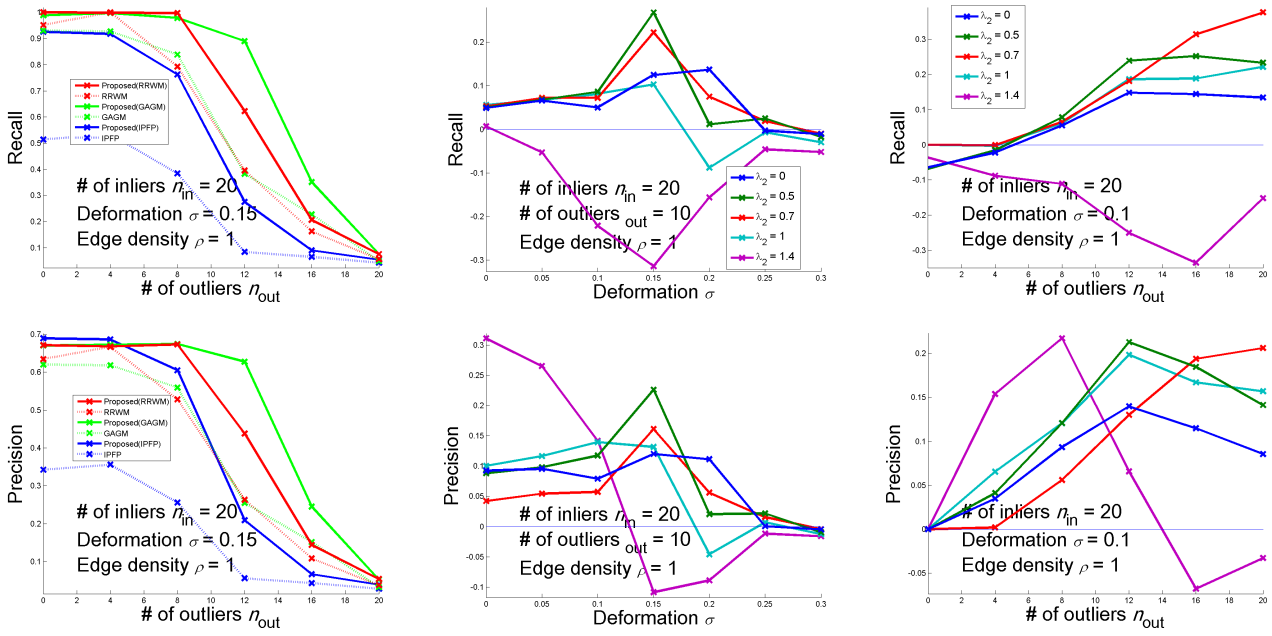


Figure 6. Example results of the real image matching using local features. True positive matches are represented by cyan lines, and false positive matches are represented by red lines.

select 200 pairs of images from each class, respectively. We use MSER detector [21] and Hessian detector [22] to extract interesting points and HOG descriptor [20] and HARG [3] to represent the node and edge attributes, respectively. In all the experiments, the greedy mapping is used for the post-processing of discretization. Our method is compared to four graph matching methods, FGM, RRWM, IPFP and SEA.

To evaluate the effect of increasing the number of outliers, we perform matching while varying the number of outliers. The outliers are randomly sampled from the feature points extracted from the images using the above detectors. The results are shown in Figure 5. Figure 5 (a) and (b) show results of winebottle and face classes, respectively. To evaluate the precision and recall at the same time, the F-score, which is the harmonic mean of the two measures, is plotted while varying the number of number of outliers between 0, 5, 10, 15 and 20. Our method with both random and data-driven proposal consistently outperforms the others. Some example results are shown in Figure 6. Additional experiments on the remaining classes are shown in the supplementary material.



(a) The effect of core algorithm (b) The effect of parameters (c) The effect of parameters

Figure 7. Analysis of the proposed method. (a) The accuracy of proposed method while varying the core algorithm. (b) and (c) Performance gain of the proposed method with respect to the core algorithm.

6.3. Analysis

6.3.1 Effect of baseline graph matching method

To determine the effect of changing the core algorithm of our method, we perform the same synthetic random graph matching experiments as in Section 6.1 while varying the core algorithms. Three core algorithms, RRWM, GAGM, and IPFP are compared in Figure 7 (a). Our method consistently improves the performance of the original core algorithm in both precision and recall.

6.3.2 Effect of parameter selection

To evaluate the robustness of the formulation when λ varies, we perform the same experiments of Section 6.1 for various λ values. For simplicity, we assume $\lambda_i = 0$ for all $i \neq 2$ and vary λ_2 from 0 to 1.4. RRWM is used as a core algorithm. In the first test, deformation σ varies from 0 to 0.3, whereas n_{out} is fixed to 10. In the second test, n_{out} varies from 0 to 20, whereas σ is fixed to 0.1. The resulting precision and recall gain relative to the core algorithm are shown in Figure 7 (b) and (c). In most of the cases, the gain is larger than 0, which means that the additional compactness prior has positive effect. The algorithm works robustly when λ has value between 0.5 and 1.

7. Conclusion

In this paper, we address the feature correspondence problem using subgraph matching. To overcome the limitations of the previous IQP formulation, we augment the original objective with additional compactness prior. The objective in this form is trained to minimize the max margin loss. To solve the new formulation, we propose a novel MCMC-based meta-algorithm. Exploring the reduced search space allows the proposed algorithm to effectively approximate the correspondence. The experimental results indicate that the proposed method outperforms the other methods in both synthetic graph matching and feature correspondence problems performing robustly in the noisy environment and in the presence of outliers.

Acknowledgements

This research was supported in part by the MKE(The Ministry of Knowledge Economy), Korea and Microsoft Research, under IT/SW Creative research program supervised by the NIPA(National IT Industry Promotion Agency) (NIPA-2013-H0503-13-1041), and in part by the Advanced Device Team, DMC R&D, Samsung Electronics.

References

[1] P. Anchuri, M. J. Zaki, O. Barkol, S. Golan, and M. Shamy. Approximate graph mining with label costs. In *Proceedings of the 19th ACM SIGKDD international conference*

on Knowledge discovery and data mining, pages 518–526. ACM, 2013.

[2] T. S. Caetano, J. J. McAuley, L. Cheng, Q. V. Le, and A. J. Smola. Learning graph matching. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31(6):1048–1058, 2009.

[3] M. Cho, K. Alahari, and J. Ponce. Learning graphs to match. In *Proceedings of the IEEE Interational Conference on Computer Vision*, 2013.

[4] M. Cho and J. Lee. Feature correspondence and deformable object matching via agglomerative correspondence clustering. In *2009 IEEE 12th International Conference on Computer Vision*, pages 1280–1287. IEEE, 2009.

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

[6] M. Cho and K. M. Lee. Progressive graph matching: Making a move of graphs via probabilistic voting. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 398–405. IEEE, 2012.

[7] M. Cho, J. Sun, O. Duchenne, and J. Ponce. Finding matches in a haystack: A max-pooling strategy for graph matching in the presence of outliers. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013.

[8] T. Cour, P. Srinivasan, and J. Shi. Balanced graph matching. volume 19, page 313. MIT; 1998, 2007.

[9] O. Duchenne, F. Bach, I.-S. Kweon, and J. Ponce. A tensor-based algorithm for high-order graph matching. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(12):2383–2395, 2011.

[10] W. R. Gilks. *Markov chain monte carlo*. Wiley Online Library, 2005.

[11] S. Gold and A. Rangarajan. A graduated assignment algorithm for graph matching. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 18(4):377–388, 1996.

[12] S. Kim, S. Kwak, J. Feyereisl, and B. Han. Online multi-target tracking by large margin structured learning. In *Computer Vision–ACCV 2012*, pages 98–111. Springer, 2013.

[13] J. Lee, M. Cho, and K. M. Lee. A graph matching algorithm using data-driven markov chain monte carlo sampling. In *Pattern Recognition (ICPR), 2010 20th International Conference on*, pages 2816–2819. IEEE, 2010.

[14] J. Lee, M. Cho, and K. M. Lee. Hyper-graph matching via reweighted random walks. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1633–1640. IEEE, 2011.

[15] M. Leordeanu and M. Hebert. A spectral technique for correspondence problems using pairwise constraints. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, volume 2, pages 1482–1489. IEEE, 2005.

[16] 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*, pages 1114–1122, 2009.

[17] M. Leordeanu, R. Sukthankar, and M. Hebert. Unsupervised learning for graph matching. *International journal of computer vision*, 96(1):28–45, 2012.

[18] H. Liu, L. J. Latecki, and S. Yan. Fast detection of dense subgraphs with iterative shrinking and expansion. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (9):2131–2142, 2013.

[19] H. Liu and S. Yan. Common visual pattern discovery via spatially coherent correspondences. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 1609–1616. IEEE, 2010.

[20] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60:91–110, 2004.

[21] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide-baseline stereo from maximally stable extremal regions. *Image and vision computing*, 22(10):761–767, 2004.

[22] K. Mikolajczyk and C. Schmid. An affine invariant interest point detector. In *Proceedings of the 7th European Conference on Computer Vision-Part I*, pages 128–142, 2002.

[23] K. P. Murphy. *Machine Learning: A Probabilistic Perspective*. The MIT Press, 2012.

[24] M. Pavan and M. Pelillo. Dominant sets and pairwise clustering. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(1):167–172, Jan. 2007.

[25] Y. Suh, M. Cho, and K. M. Lee. Graph matching via sequential monte carlo. In *Computer Vision–ECCV 2012*, pages 624–637. Springer, 2012.

[26] L. Torresani, V. Kolmogorov, and C. Rother. Feature correspondence via graph matching: Models and global optimization. In *Computer Vision–ECCV 2008*, pages 596–609. Springer, 2008.

[27] R. Zass and A. Shashua. Probabilistic graph and hypergraph matching. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.

[28] Q. Zhang, X. Song, X. Shao, H. Zhao, and R. Shibasaki. Attributed graph mining and matching: An attempt to define and extract soft attributed patterns. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 1394–1401. IEEE, 2014.

[29] F. Zhou and F. De la Torre. Factorized graph matching. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 127–134. IEEE, 2012.

[30] F. Zhou and F. De la Torre. Deformable graph matching. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 2922–2929. IEEE, 2013.