

Reweighted Random Walks for Graph Matching

Minsu Cho, Jungmin Lee, and Kyoung Mu Lee

Department of EECS, ASRI, Seoul National University, 151-742, Seoul, Korea
<http://cv.snu.ac.kr>

Abstract. Graph matching is an essential problem in computer vision and machine learning. In this paper, we introduce a random walk view on the problem and propose a robust graph matching algorithm against outliers and deformation. Matching between two graphs is formulated as node selection on an association graph whose nodes represent candidate correspondences between the two graphs. The solution is obtained by simulating random walks with reweighting jumps enforcing the matching constraints on the association graph. Our algorithm achieves noise-robust graph matching by iteratively updating and exploiting the confidences of candidate correspondences. In a practical sense, our work is of particular importance since the real-world matching problem is made difficult by the presence of noise and outliers. Extensive and comparative experiments demonstrate that it outperforms the state-of-the-art graph matching algorithms especially in the presence of outliers and deformation.

Key words: graph matching, random walks, feature correspondence

1 Introduction

Graph matching is an essential problem in theoretical computer science; it is related to various research areas in computer vision, pattern recognition, and machine learning [1]. The problem of graph matching is to determine a mapping between the nodes of the two graphs that preserves the relationships between the nodes as much as possible. In computer vision, it is widely known that the fundamental problem of establishing correspondences between two sets of visual features can be effectively solved by graph matching. Thus, graph matching is used in various tasks, such as feature tracking, image retrieval, object recognition, and shape matching. Many graph matching algorithms proposed in the 1980s and 1990s focused on exploiting relatively weak unary and pair-wise attributes and did not specifically aim at optimizing a well-defined objective function [1]. Recent resurgence of combinatorial optimization approaches to feature matching [2–8] has changed the situation and firmly settled graph matching formulations based on Integer Quadratic Programming (IQP), which is a generalization of the classical graph matching problems. IQP explicitly takes into consideration both unary and pair-wise terms reflecting the compatibilities in local appearance as well as the pair-wise geometric relationships between the

matching features. Since IQP is known to be NP-hard, approximate solutions are required. Our work provides a novel interpretation of graph matching in a random walk view and relates it to the IQP formulation. **Introducing an association graph constructed with nodes as candidate correspondences and edges as pair-wise compatibilities between candidate correspondences**, we show that the search for correspondences between the given two graphs can be cast as a node ranking [9, 10] and selection problem in the association graph. For this ranking, we introduce an *affinity-preserving random walk* and derive a ranking based on its quasi-stationary distribution, and prove its equivalence to the spectral relaxation [3] for the IQP formulation. Then, in this random walk view, we adopt the personalization strategy of Web ranking algorithms [11] and propose the *reweighted random walk* algorithm by reweighting jumps for the graph matching constraints. It achieves noise-robust graph matching by simultaneously updating and exploiting the confidences of candidate correspondences. In a practical sense, our work is of particular importance since the real-world matching problem is made difficult by the presence of deformation and outliers.

A myriad of algorithms have been proposed for graph matching, and those closely related to ours are follows. Maciel and Costeira [12] formulated graph matching as a constrained integer optimization problem with a concave optimization scheme, but the complexity of its minimization was still non-polynomial. Gold and Rangarajan [13] proposed the Graduated Assignment (GAGM) algorithm to solve the IQP by relaxing the integer constraint. In their deterministic annealing approach, GAGM gradually updates the derivative of the relaxed IQP in the soft assignment step driven by an annealing schedule. The SPGM algorithm proposed by van Wyk and van Wyk [14] iteratively updates the objective function of IQP by projecting an approximation of the current matching matrix onto the convex space of the matching constraints. Leordeanu and Hebert [3] proposed a simple and efficient approximation to the IQP using spectral relaxation, which computes the leading eigenvector of symmetric nonnegative affinity matrix. Their Spectral Matching (SM) ignored the integer constraints in the relaxation step and induced them during the discretization step by a **greedy approach**. They also recently proposed an **iterative matching method** (IPFP) [8] with climbing and convergence properties which optimizes the IQP in the discrete domain. Cour et al. [4] extended SM[3] to **Spectral Matching** with Affine Constraint (SMAC) by introducing affine constraints into the spectral decomposition that encodes the one-to-one matching constraints. Lee et al. [15] presented a Markov chain Monte Carlo algorithm to solve the IQP based on the spectral relaxation. Zass and Shashua [6] showed that matching problems can be represented by a matrix constructed by Kronecker products, and also introduced a probabilistic framework for hypergraph matching. Duchenne et al. [7] extended the method of [3] to high-order graph matching which was formulated as a tensor eigendecomposition problem. Our problem formulation is related to the previous IQP formulations of [13, 3, 4, 15], but we approached it from a random walk view. Note that the previous random walk-based approaches [16, 17] use the random

walk theory to find a signature for each node in a graph, and their problem formulations are different from ours.

This paper presents three main contributions. First, it establishes a novel random walk view for graph matching and provides a basis for random walk interpretations of recent spectral matching [3, 4, 7] and other iterative algorithms [13, 6]. Second, in this view, we propose a powerful matching algorithm inspired by the personalization strategy of Web ranking algorithms [11] and the Sinkhorn method [18]. Third, it is extensively demonstrated against several state-of-the-art graph matching algorithms. The comparison not only reveals the superior performance of our algorithm but also facilitates a comprehensive study of recent graph matching algorithms.

2 Problem Formulation

The objective of graph matching is to determine the correct correspondences between two attributed graphs $G^P = (V^P, E^P, A^P)$ and $G^Q = (V^Q, E^Q, A^Q)$, where V represents a set of nodes, E , edges, and A , attributes. Each node $v_i^P \in V^P$ or edge $e_{ij}^P \in E^P$ has an associated attribute vector $\mathbf{a}_i^P \in A^P$ or $\mathbf{a}_{ij}^P \in A^P$. In feature correspondence problems, a node attribute \mathbf{a}_i^P usually describes a local appearance of feature i in an image P , and an edge attribute \mathbf{a}_{ij}^P represents the geometric relationship between features i and j in the image P . For each pair of edges $e_{ij}^P \in E^P$ and $e_{ab}^Q \in E^Q$, there is an affinity or compatibility $\mathbf{W}_{ia;jb} = f(\mathbf{a}_i^P, \mathbf{a}_j^P, \mathbf{a}_{ij}^P, \mathbf{a}_a^Q, \mathbf{a}_b^Q, \mathbf{a}_{ab}^Q)$ that measures the mutual consistency of attributes between the pairs of candidate correspondences (v_i^P, v_a^Q) and (v_j^P, v_b^Q) . Thus, using a matrix form \mathbf{W} , a non-diagonal element $\mathbf{W}_{ia;jb}$ contains a pair-wise affinity between two correspondences (v_i^P, v_a^Q) and (v_j^P, v_b^Q) , and a diagonal term $\mathbf{W}_{ia;ia}$ represents a unary affinity of a correspondence (v_i^P, v_a^Q) . Representing the correspondence with an assignment or permutation matrix $\mathbf{X} \in \{0, 1\}^{n^P \times n^Q}$ is common, such that $\mathbf{X}_{ia} = 1$ implies that node v_i^P corresponds to node v_a^Q , e.g., feature i in the image P is matched to feature a in the image Q , and $\mathbf{X}_{ia} = 0$ otherwise. In this paper, we denote $\mathbf{x} \in \{0, 1\}^{n^P n^Q}$ as a column-wise vectorized replica of \mathbf{X} . The graph matching problem can be formulated as an integer quadratic program (IQP), that is, finding the indicator vector \mathbf{x}^* that maximizes the quadratic score function as follows.

$$\begin{aligned} \mathbf{x}^* &= \arg \max(\mathbf{x}^T \mathbf{W} \mathbf{x}) \\ \text{s.t. } \mathbf{x} &\in \{0, 1\}^{n^P n^Q}, \forall i \sum_{a=1}^{n^Q} \mathbf{x}_{ia} \leq 1, \forall a \sum_{i=1}^{n^P} \mathbf{x}_{ia} \leq 1, \end{aligned} \quad (1)$$

where the two-way constraints refer to the one-to-one matching from G^P to G^Q . In general, no efficient algorithm exists that can guarantee the optimality bounds since the IQP is NP-hard, thus approximate solutions are required.

3 Random Walks for Graph Matching

Basically, the problem of graph matching between the two graphs G^P and G^Q can be interpreted in a random walk view by constructing an association graph $G^{rw} = (V^{rw}, E^{rw}, A^{rw})$ as follows. Given the pair-wise affinity matrix \mathbf{W} , we consider each candidate correspondence $(v_i^P, v_a^Q) \in V^P \times V^Q$ as a node $v_{ia} \in V^{rw}$, its associated weight $\mathbf{W}_{ia;jb}$ as the attribute $a_{ia;jb} \in A^{rw}$ of the edge $e_{ia;jb} \in E^{rw}$. This is illustrated by an example in Fig.1(a). The original graph matching problem between G^P and G^Q is equivalent to selecting reliable nodes in the graph G^{rw} since the selected nodes in G^{rw} corresponds to graph or subgraph matching between G^P and G^Q . To select the nodes in G^{rw} , we adopt the statistics of the Markov random walks which has been used to compute the ranking or relevance of graphs in the Web environments [9, 10]. Thus, graph matching between G^P and G^Q can be transformed into the node ranking and selection problem by random walks on G^{rw} . In this view, we introduce an *affinity-preserving random walk* algorithm in Sec.3.1, which paves the way for *reweighted random walk* algorithm in Sec.3.2.

3.1 Affinity-Preserving Random Walks

The standard way to define a random walk on a graph is to allow a random walker to take off on an arbitrary node and then successively visit new nodes by randomly selecting one of the outgoing edges according to a Markov transition kernel of the graph. In general, in order to define the transition matrix on weighted graphs, traditional random walk approaches convert affinity or weight matrix \mathbf{W} to the row stochastic matrix by $\mathbf{P} = \mathbf{D}^{-1}\mathbf{W}$, where \mathbf{D} is a diagonal matrix with entries $D_{ii} = d_i = \sum_j W_{ij}$. This normalization is required not only for transforming \mathbf{W} into a stochastic matrix, that is “stochasticizing”, but also for other particular reasons in the applications. For example, in PageRank [10], each out-going hyperlink from a node i is row-normalized by $1/d_i$ so that every webpage has the same total out-going weights. We may state this idea as

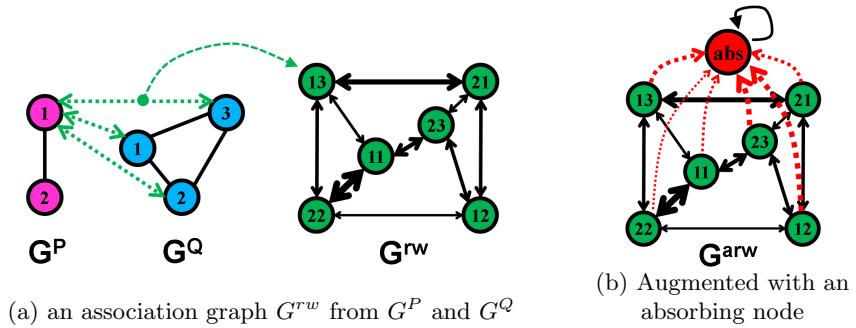


Fig. 1. Association Graphs for Graph Matching by Random Walks.

Internet Democracy: each webpage has a total of one vote [19]. However, this democratic normalization is problematic in our approach for graph matching since in G^{rw} some nodes correspond to false candidate correspondences (outlier nodes) or more distorted ones than others. In such cases, the normalization can strengthen the adverse effect of outliers and weak correspondences, and pervert random walkers. For an example, consider Fig.1(a) where v_1^P and v_2^P correspond to v_1^Q and v_2^Q , respectively. The democratic normalization on G^{rw} scales up the affinities of outgoing edges of outlier nodes such as v_{12} , v_{13} , v_{21} , and v_{23} compared with the affinities of two inlier nodes v_{11} and v_{22} because the affinity sum of an outlier node is usually smaller than that of an inlier node.

How then can we preserve the original affinity relations while transforming the affinity matrix into the stochastic transition matrix for random walks? We define the maximum degree $d_{\max} = \max_i d_i$, and construct an augmented graph G^{arw} with an absorbing node v_{abs} which soaks affinity $d_{\max} - d_i$ out of all the nodes $v_i \in V^{rw}$ as shown in Fig.1(b). We treat this graph as a special Markov chain which has an absorbing node, i.e., a state which, once reached, cannot be transitioned out of. Since each node in the affinity matrix of G^{arw} has the same degree of d_{\max} , its normalized affinity matrix by $1/d_{\max}$ results in a stochastic matrix and corresponds to an absorbing Markov chain[20] which preserves the relative affinity relations of the original graph G^{rw} . We call this approach an “affinity-preserving random walk”, and formulate its transition matrix \mathbf{P} and absorbing Markov chain as follows.

$$\mathbf{P} = \begin{pmatrix} \mathbf{W}/d_{\max} & \mathbf{1} - \mathbf{d}/d_{\max} \\ \mathbf{0}^T & 1 \end{pmatrix}, \quad \left(\mathbf{x}^{(n+1)T} \ x_{\text{abs}}^{(n+1)} \right) = \left(\mathbf{x}^{(n)T} \ x_{\text{abs}}^{(n)} \right) \mathbf{P}, \quad (2)$$

where \mathbf{W}/d_{\max} is the $n^P n^Q \times n^P n^Q$ substochastic matrix, and $\mathbf{1}$ is a $n^P n^Q \times 1$ vector with all elements 1, and $\mathbf{0}$ with all elements 0. This absorbing Markov chain has transient nodes of V^{rw} from which its random walker is certain to be absorbed into an absorbing node v_{abs} . Its steady state distribution is always $(\mathbf{0}^T \ 1)$, thus cannot be used for node ranking in the same way as PageRank [10]. For ranking on the absorbing Markov chain, we denote $X^{(n)}$ as the node where a random walker in the absorbing Markov chain of Eq.(2) stays at time n , and define the conditional distribution $(\bar{\mathbf{x}}^{(n)})$ as

$$\bar{\mathbf{x}}_{ia}^{(n)} = P(X^{(n)} = v_{ia} \mid X^{(n)} \neq v_{\text{abs}}) = \frac{\mathbf{x}_{ia}^{(n)}}{1 - x_{\text{abs}}^{(n)}}, \quad (3)$$

which refers to the distribution of unabsorbed random walkers at time n . If $\bar{\mathbf{x}}^{(n+1)} = \bar{\mathbf{x}}^{(n)} = \bar{\mathbf{x}}$, we call $\bar{\mathbf{x}}$ a *quasi-stationary distribution* of the absorbing Markov chain. This corresponds to a steady-state distribution in the Markov chain without absorbing nodes. Following the approach of PageRank [10] based on the steady-state distribution in ergodic Markov chains, we define the affinity-preserving PageRank as follows.

Definition 1. The affinity-preserving PageRank of a graph with affinity matrix \mathbf{W} is the quasi-stationary probability $\bar{\mathbf{x}}$ of Eq.(3) in affinity-preserving random walks of Eq.(2).

Theorem 1. *The quasi-stationary distribution of the affinity-preserving random walks of Eq.(2) is proportional to the left principal eigenvector of \mathbf{W} .*

Proof. According to Eq.(2),

$$\begin{pmatrix} \mathbf{x}^{(n+1)T} & x_{\text{abs}}^{(n+1)} \end{pmatrix} = \begin{pmatrix} \mathbf{x}^{(n)T} \mathbf{W} / d_{\max} & 1 - \mathbf{x}^{(n)T} \mathbf{d} / d_{\max} \end{pmatrix}$$

Thus, by Eq.(3),

$$\bar{\mathbf{x}}^{(n+1)} = \frac{\mathbf{x}^{(n)T} \mathbf{W}}{\mathbf{x}^{(n)T} \mathbf{d}} = \frac{\bar{\mathbf{x}}^{(n)T} \mathbf{W}}{\bar{\mathbf{x}}^{(n)T} \mathbf{d}}$$

Then, since $\bar{\mathbf{x}}^{(n+1)} = \bar{\mathbf{x}}^{(n)} = \bar{\mathbf{x}}$ for quasi-stationary distribution, $\bar{\mathbf{x}}$ must satisfy $\lambda \bar{\mathbf{x}}^T = \bar{\mathbf{x}}^T \mathbf{W}$. If \mathbf{W} is irreducible, and the elements of $\bar{\mathbf{x}}$ are non-negative, then it follows from the extended Perron Frobenius theorem that λ is the real maximal eigenvalue of \mathbf{W} , and $\bar{\mathbf{x}}$ is a normalized non-negative left eigenvector of \mathbf{W} corresponding to the maximal eigenvalue. Therefore, the quasi-stationary distribution $\bar{\mathbf{x}}$ is equivalent to the left eigenvector of \mathbf{W} with non-negative components.

Interestingly, this affinity-preserving PageRank is the solution of a relaxed version of the original IQP problem, and is equivalent to the spectral relaxation of [3]. By dropping two-way matching constraints and relaxing integer constraints from Eq.(1), the original IQP is approximated to a continuous problem as

$$\mathbf{x}^* = \arg \max(\mathbf{x}^T \mathbf{W} \mathbf{x}) \text{ s.t. } \mathbf{x} \in [0, 1]^{n^P n^Q}, \quad (4)$$

which is interpreted as a classical Rayleigh quotient problem in [3], whose solution \mathbf{x}^* is obtained by the eigenvector associated with the largest eigenvalue of \mathbf{W} . The result is the same as the affinity-preserving PageRank in our random walk view, and can also be computed efficiently by the power iteration method. This view provides a basis for random walk interpretations of recent spectral methods [3, 4, 7] and iterative algorithms [13, 6] on graph matching problem. Assuming that the solution of the relaxed problem is close to the optimal discrete solution, the final solution is obtained by incorporating the matching constraints on it. A greedy mapping [3] or the Hungarian algorithm [21] can be adopted for the final discretization.

As demonstrated in our experiment in Sec.4, the affinity-preserving random walk matching (equivalent to SM in Sec.4) consistently outperforms conventional random walk matching (denoted by NRWM in Sec.4) which uses row-normalized affinity matrix.

3.2 Reweighted Random Walks

In the previous affinity-preserving random walks, the matching constraints of Eq.(1) are ignored and not reflected in the random walk process. Inducing the matching constraints only as a post-processing discretization step like [3] leads to a weak local optimum. How then can we reflect the two-way matching constraints in the affinity-preserving random walk? We adopts the personalization

approach widely used in Web ranking methods [11, 19], which strengthens the effects of reliable nodes in random walks. This is achieved by adopting a jump or teleport in the random walk; the random walker moves by traversing an edge with probability α or by performing a jump to some constrained nodes with probability $1 - \alpha$. α represents the bias between the two possible actions, i.e., following an edge or jumping. To address the lack of personalization or user's focus, Web ranking algorithms adopted this approach in topic-sensitive or query-dependent variants [19]. In our formulation, adopting the personalized jump, the probability distribution is updated using the following equation:

$$\begin{pmatrix} \mathbf{x}^{(n+1)T} & x_{\text{abs}}^{(n+1)} \end{pmatrix} = \alpha \begin{pmatrix} \mathbf{x}^{(n)T} & x_{\text{abs}}^{(n)} \end{pmatrix} \mathbf{P} + (1 - \alpha) \mathbf{r}^T, \quad (5)$$

where a *reweighting jump* vector \mathbf{r} is added to the affinity-preserving random walk of Eq.(2). In this approach, we use the jumps for generating a biased random walk to the matching constraints. One possible way is to use the result of the discrete assignment mapping of current \mathbf{x} as the jump vector \mathbf{r} at each iteration. However, this scheme is vulnerable to the discretization of the wrong solution in early steps. Thus, we propose a robust reweighting scheme as described in Algorithm 1. The reweighting procedure consists of two steps: inflation and bistochastic normalization [18]. The inflation step of $\exp(\beta \mathbf{x} / \max \mathbf{x})$ attenuates small values of \mathbf{x} and amplifies large values of \mathbf{x} . In this way, unreliable correspondences contribute insignificantly through the individual exponentials over the components of \mathbf{x} . Then, for the two-way constraint that a node in the graph G^P must correspond to only one node in the graph G^Q and vice versa, the bistochastic normalization scheme of Sinkhorn [18] is applied as in [13], which alternatively normalizes the rows and columns of \mathbf{X} (matrix form of \mathbf{x}). Any square matrix whose elements are all positive is proven to converge to a bistochastic matrix¹ just by the iterative process [18]. At each iteration, the reweighting jumps are introduced on the transient part of the current affinity-preserving random walking $\mathbf{x}^{(n)T} \mathbf{W} / d_{\max}$. Thus, the reweighted random walk is formulated by

$$\begin{pmatrix} \mathbf{x}^{(n+1)T} & x_{\text{abs}}^{(n+1)} \end{pmatrix} = \alpha \begin{pmatrix} \mathbf{x}^{(n)T} & x_{\text{abs}}^{(n)} \end{pmatrix} \mathbf{P} + (1 - \alpha) \begin{pmatrix} f_C(\mathbf{x}^{(n)T} \mathbf{W})^T & 0 \end{pmatrix}, \quad (6)$$

where $f_C(\cdot)$ denotes the reweighting function incorporating two-way constraints. Note that this is a dynamic Markov chain whose jump distribution is dynamically varying and dependent on the present distribution of \mathbf{x} unlike conventional jumps in random walks [11, 19]. As the $f_C(\cdot)$ generates a jump distribution close to a good solution, the subsequent random walks strengthen the distribution and move toward a integer solution. Its fast convergence is observed empirically in all our experiments. To further tighten the random walks by the matching constraints, we enforce *conflicting walk prevention* which entails that random walks to conflicting nodes are prevented according to the matching constraints. In the

¹ A bistochastic matrix is a matrix whose elements are all positive and whose rows and columns all add up to one: it may roughly be thought of as the continuous analog of a permutation matrix allowing $x_i \in [0, 1]$.

Algorithm 1 Reweighted Random Walk Graph Matching

```

1: Given the weight matrix  $\mathbf{W}$ , the reweight factor  $\alpha$ , and the inflation factor  $\beta$ 
2: Prevent conflicting walks by setting  $\mathbf{W}_{ia;jb} = 0$  for all conflicting match pairs
3: Set the maximum degree  $d_{\max} = \max_{ia} \sum_{jb} \mathbf{W}_{ia;jb}$ 
4: Initialize the transition matrix  $\mathbf{P} = \mathbf{W}/d_{\max}$ , the starting probability  $\mathbf{x}$  as uniform
5: repeat
6:   ( Affinity-preserving random walking by edges )
7:    $\bar{\mathbf{x}}^T = \mathbf{x}^T \mathbf{P}$ 
8:   ( Reweighting with two-way constraints )
9:    $\mathbf{y}^T = \exp(\beta \bar{\mathbf{x}} / \max \bar{\mathbf{x}})$ 
10:  repeat
11:    normalize across rows by  $\mathbf{y}_{ai} = \mathbf{y}_{ai} / \sum_{i=1}^I \mathbf{y}_{ai}$ 
12:    normalize across columns by  $\mathbf{y}_{ai} = \mathbf{y}_{ai} / \sum_{a=1}^A \mathbf{y}_{ai}$ 
13:  until  $\mathbf{y}$  converges
14:   $\mathbf{y} = \mathbf{y} / \sum \mathbf{y}_{ai}$ 
15:  ( Affinity-preserving random walking with reweighted jumps)
16:   $\mathbf{x}^T = \alpha \bar{\mathbf{x}}^T + (1 - \alpha) \mathbf{y}^T$ 
17:   $\mathbf{x} = \mathbf{x} / \sum \mathbf{x}_{ai}$ 
18: until  $\mathbf{x}$  converges
19: Discretize  $\mathbf{x}$  by the matching constraints

```

case of two-way constraints of Eq.(1), a random walker in node v_{ia} is prohibited to move to nodes $\forall b \neq a, v_{ib}$ and $\forall j \neq i, v_{ja}$. This is easily implemented by initially eliminating such conflicting elements in the affinity matrix \mathbf{W} .

The quasi-stationary distribution of this reweighted random walk is efficiently computed using the power iteration method as summarized in Algorithm.1. Its computational complexity is $O(|E^P||E^Q|)$ per iteration, where $|E^P|$ and $|E^Q|$ are the numbers of edges in the two graphs, respectively. In the final discretization step, any linear assignment algorithm can be adopted, such as a greedy algorithm in [3] or the Hungarian algorithm [21].

From an algorithmic point of view, our method has some resemblance to SM[3] and GAGM[13]. Without the reweighting jumps, our affinity-preserving random walking can be considered as the power iteration version of [3] as explained in Sec.3.1. The Sinkhorn method [18] introduced in our reweighing jumps is also adopted in the softassign step of GAGM [13]. However, our reweighting step does not require a deterministic annealing schedule as GAGM and is designed to effectively select reliable nodes for reweighted jumps. Our method provides faster convergence to a better optimum by the balance of walks and jumps as demonstrated in the experiments.

4 Experiments

We performed intensive experiments for the proposed method in three tasks: (1) synthetically generated random graphs, (2) point matching task using the CMU

House image sequence², and (3) feature matching using real images. These three experiments are designed to evaluate the performance of our algorithm (RRWM) on various graph matching tasks and to compare it with other state-of-the-art methods: SM[3], SMAC[4], HGM[6], IPFP[8], GAGM[13], and SPGM[14]. We additionally tested the performance of the random walk matching with conventional row-wise normalization denoted by NRWM. For SMAC³ and HGM⁴, the publicly available codes by authors were used, and SM, IPFP, and SPGM were implemented by us. For GAGM, we revised and tuned the code based on the implementation provided by Cour [4]. All methods were implemented using MATLAB and tested on 2.40 GHz Core2 Quad desktop PC. For each trial in all experiments, the same affinity matrix was shared as the input⁵ and the Hungarian algorithm⁶ was commonly used at final discretization step for all methods. Control parameters of GAGM and SPGM were based on the authors' papers and tuned for better performance. For our RRWM, we fixed $\alpha = 0.2$, $\beta = 30$ in all experiments. These settings allow us to quantify the accuracy and robustness of all algorithms, and fairly compare them with one another.

4.1 Synthetic Random Graph Matching

In this experiment, following the experimental protocol of [13, 4], we performed a comparative evaluation on random graph matching problems. For each trial, we constructed two graphs, G^P with $n^P = n_{in} + n_{out}^P$ nodes and G^Q with $n^Q = n_{in} + n_{out}^Q$ nodes, each consisting of n_{in} inlier nodes and the other outlier nodes. The reference graph G^P is generated with random edges of edge density ρ , where each edge $e_{ij}^P \in E^P$ was assigned a random attribute \mathbf{a}_{ij}^P distributed uniformly in $[0, 1]$. We then created a perturbed graph G^Q by adding noise on the edge attributes between inlier nodes: $\mathbf{a}_{ab}^Q = \mathbf{a}_{p(i)p(j)}^P + \varepsilon$, where $p(\cdot)$ is a random permutation function for inlier nodes. The deformation noise ε was distributed using the Gaussian noise function $N(0, \sigma^2)$. All the other edges connecting at least one of the outlier nodes are randomly generated as the same way in G^P . Thus, two graphs G^P and G^Q have a common and perturbed subgraph with size n_{in} . The affinity matrix \mathbf{W} was computed by $W_{ia,jb} = \exp(-|\mathbf{a}_{ij}^P - \mathbf{a}_{ab}^Q|^2 / \sigma_s^2)$, $\forall e_{ij}^P \in E^P, \forall e_{ab}^Q \in E^Q$. The scaling factor σ_s^2 is set to 0.15 empirically to show the best average performance of all the methods. The accuracy is measured by the number of detected true matches divided by the total number of ground truths, and the objective score by computing $\mathbf{x}^T \mathbf{W} \mathbf{x}$ of the IQP objective.

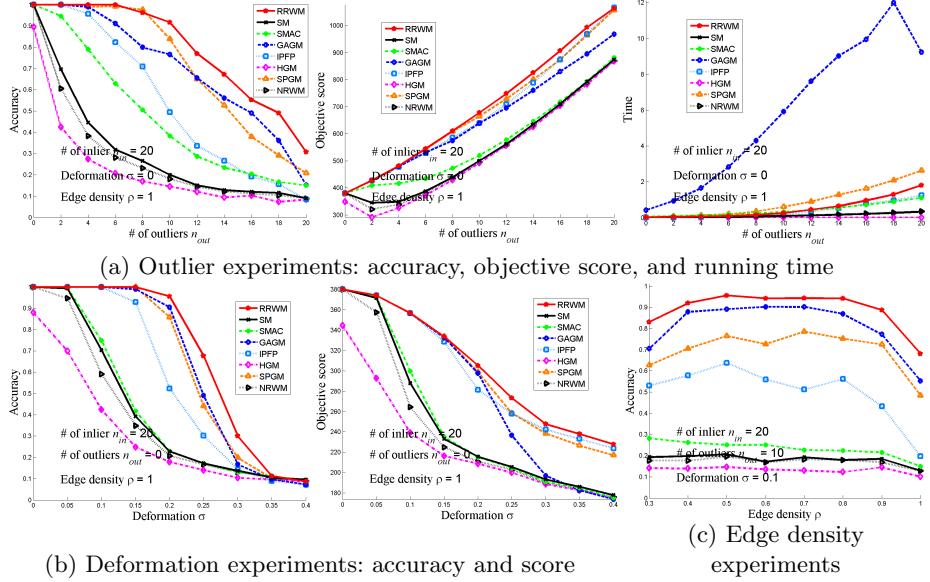
² <http://vasc.ri.cmu.edu/idb/html/motion/>

³ <http://www.seas.upenn.edu/~timothée/>

⁴ <http://www.cs.huji.ac.il/~zass/>

⁵ The original code of HGM [6] uses a high-order affinity, but we edited it to compare with the other methods given the same affinity matrix.

⁶ In the comparative experiments in [4], GAGM was commonly used as a post-processing step for the final discretization, but we found that GAGM alone is a strong IQP solver in our experiments. Thus, for observing the performance of each algorithm, we used the Hungarian algorithm [21], the classic linear assignment solver, for the discretization.

**Fig. 2.** Synthetic Graph Matching Experiments

The results are shown in Fig.2. In our experimental setup, there were three kinds of independent variables: outliers n_{out}^P and n_{out}^Q , deformation noise σ , and edge density ρ . Hence, we conducted three sub-experiments to show their influences on performance. For each parameter setting, we generated 100 different matching problems and evaluated the average accuracy and objective score. First, for the outlier experiment in Fig.2(a), the number of outliers $n_{out}^P = n_{out}^Q$ were varied from 0 to 20 by increments of 2, while fixing inlier number $n_{in} = 20$, deformation noise $\sigma = 0$, and edge density $\rho = 1$. Second, we experimented on deformation by varying the deformation noise σ from 0 to 0.4 with increments of 0.05 as in Fig.2(b), while fixing the number of inliers $n_{in} = 20$, outliers $n_{out}^P = n_{out}^Q = 0$, edge density $\rho = 1$. Third, in Fig.2(c), we varied the edge density ρ from 0.3 to 1 by increments of 0.1, while the number of inliers $n_{in} = 20$, outliers $n_{out}^P = n_{out}^Q = 10$, deformation noise $\sigma = 0.1$. From all the plots in Fig.2 with outlier, deformation and edge density variation, we can see that the proposed RRWM outperforms all the other state-of-the-arts methods in both accuracy and objective score. GAGM and SPGM are comparable to RRWM, but with increasing outliers and deformation, RRWM shows consistently better performance. Recent methods based on spectral relaxation and probabilistic interpretation are still less robust to outlier and deformation than GAGM, SPGM, and ours which incorporate the matching constraints in iterative optimizing process. It indicates that tightening relaxation with the matching constraints is an important factor for robust graph matching. As shown in the right plot of Fig.2(a), RRWM achieves faster convergence than GAGM and SPGM with in-

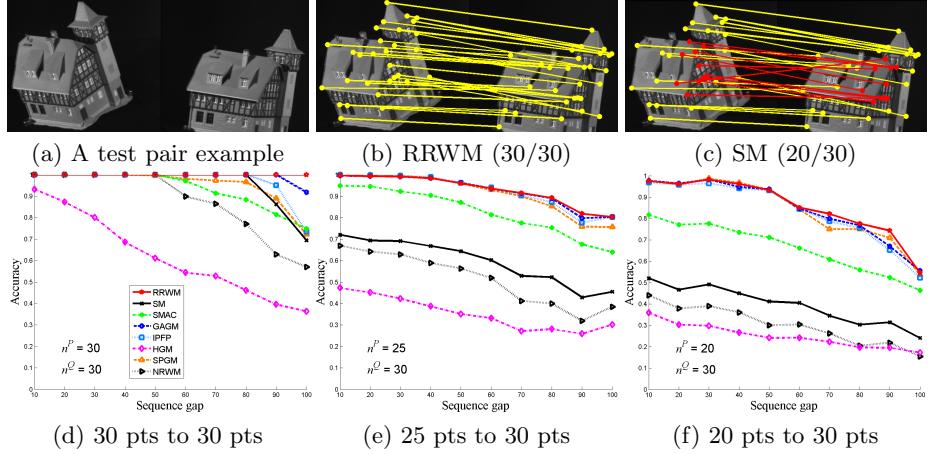


Fig. 3. The CMU House sequence experiments

creasing graph sizes and deformation although all the methods have the same theoretical complexity. We did the same experiments adding outliers only to G^Q with $n_{out}^Q = 0$, and the results also show similar trends as in Fig.2 in all aspects.

Comparing the performance of SM with NRWM, we can observe the effect of affinity-preserving since SM is equivalent to affinity-preserving random walk matching as proved in Sec.3.1. As shown in all the experiments, affinity-preserving provides more robustness to outlier and deformation than conventional row-normalization. It is consistently demonstrated also in the following experiments.

4.2 Feature Point Matching across Image Sequences

In this section, we performed feature point matching on the CMU House sequence which has been widely used in previous works [7, 5] and compared with other methods. In order to assess the matching accuracy, 30 landmark feature points were manually tracked and labeled across all frames. This allows us to compare the performance of the different algorithms over a varying temporal baseline: the larger the temporal baseline between the frames, the larger the relative deformation, and the more difficult the matching. We matched all possible image pairs, total 560 pairs, spaced by 10, 20, 30, 40, 50, 60, 70, 80, 90, and 100 frames and computed the average matching accuracy per sequence gap. Graph matching problems for 3 different settings were generated with landmark points as nodes: $(n^P, n^Q) = (30, 30), (25, 30)$, and $(20, 30)$. In the settings for subgraph matching where $n^P < 30$, we chose n^P points randomly among 30 landmark points. The affinity matrix is conducted by $\mathbf{W}_{ia,jb} = \exp(-|\mathbf{a}_{ij}^P - \mathbf{a}_{ab}^Q|^2/\sigma_s^2)$, where \mathbf{a}_{ij}^P was assigned Euclidean distance between two points. We fixed the scaling factor $\sigma_s^2 = 2500$ and the edge density $\rho = 1$. In this experiment, as n^P decreases,

relative outlier nodes increases. As the sequence gap increases, deformation noise increases. Figure.3 shows the performance curves for $n^P = 30, 25$, and 20 with respect to the sequence gap. RRWM, GAGM, and SPGM give best performances in this experiment, and RRWM generated perfect matching in the 30 to 30 problem. Note that as outliers or deformation increases, RRWM, GAGM, SPGM shows larger performance gap from other methods, and RRWM converges faster than both GAGM and SPGM as shown in Fig.2(a).

We also did extensive point matching experiments on random synthetic point sets, and RRWM showed the best performance as similar to this experiments.

4.3 Real Image Matching

In this experiment we applied our method to challenging real image matching problems using local feature detectors. We constructed a dataset of 30 image pairs containing various images most of which are collected from Caltech-101⁷ and MSRC⁸ datasets, and generated candidate correspondences using the MSER detector [22] and the SIFT descriptor [23]. Using the distance of 128 -dim SIFT descriptor, all the possible candidate matches were collected if the feature pair has closer distance in SIFT feature space than a loose threshold $\delta = 0.6$, allowing multiple correspondences for each feature.

To measure the dissimilarity between two candidate region correspondences (i, a) and (j, b) , we adopted the mutual projection error function $d_{ia;jb}$ used in [24], and set $\mathbf{W}_{ia;jb} = \max(50 - d_{ia;jb}, 0)$. The ground truths were manually labeled for all candidate correspondences of each image pair, and the accuracy and relative objective score were computed and compared with SM, SMAC, and GAGM. The results are summarized in Table.1 and some representative examples are shown in Fig.4. In Fig.4(c)-(f), the algorithm, true matches per ground truths, and objective scores are captioned. As shown in the examples, this experiment and the dataset are designed for producing the challenging feature matching problems where unary local features are very ambiguous. Our RRWM clearly outperforms other methods both in accuracy and objective score as summarized in Table.1. Note that the second best, GAGM was about ten times slower than RRWM in this experiment as similar in the previous experiments.

For the full results of our comparative experiments and more information, refer to our project site: <http://cv.snu.ac.kr/research/~RRWM/>

Methods	RRWM	SM	SMAC	GAGM
Avg. of accuracy (%)	64.01	52.08	39.74	58.74
Avg. of relative score (%)	100	82.41	59.35	91.13

Table 1. Matching performance on the real image dataset (30 pairs)

⁷ http://www.vision.caltech.edu/Image_Datasets/Caltech101/

⁸ <http://research.microsoft.com/en-us/projects/objectclassrecognition/>



Fig. 4. Some results of real image matching on our dataset. True matches are represented by cyan lines, and false matches by black lines.

5 Conclusion

In this paper, we introduced a graph matching framework based on random walks and proposed a novel graph matching algorithm inspired by the personalized random walks [10, 11] and the Sinkhorn method [18]. The experiments demonstrated that it outperforms the state-of-the-art methods [3, 4, 6, 8, 13, 14] in the presence of outliers and deformation. The comparison reveals that the matching accuracy in the challenging situations largely depends on the effective exploitation of the matching constraints. Our random walk framework is extendable to high-order graph matching adopting the tensor representation as in [7, 6]. In our future work, we will improve our framework and method for this direction.

Acknowledgements

This work was supported by the Korea Research Foundation Grant funded by the Korean Government(MOEHRD) (KRF-2008-314-D00377).

References

1. Conte, D., Foggia, P., Sansone, C., Vento, M.: Thirty years of graph matching in pattern recognition. *IJPRAI* (2004)
2. Berg, A.C., Berg, T.L., Malik, J.: Shape matching and object recognition using low distortion correspondences. *CVPR* (2005)
3. Leordeanu, M., Hebert, M.: A spectral technique for correspondence problems using pairwise constraints. *ICCV* (2005)
4. Cour, T., Srinivasan, P., Shi, J.: Balanced graph matching. *NIPS* (2006)
5. Torresani, L., Kolmogorov, V., Rother, C.: Feature correspondence via graph matching: Models and global optimization. *ECCV* (2008)
6. Zass, R., Shashua, A.: Probabilistic graph and hypergraph matching. *CVPR* (2008)
7. Duchenne, O., Bach, F., Kweon, I., Ponce, J.: A tensor-based algorithm for high-order graph matching. *CVPR* (2009)
8. Leordeanu, M., Herbert, M.: An integer projected fixed point method for graph matching and map inference. *NIPS* (2009)
9. Kleinberg, J.: Authoritative sources in a hyperlinked environment. *Journal of the ACM* (1999)
10. Page, L., Brin, S., Motwani, R., Winograd, T.: The pagerank citation ranking: Bringing order to the web. Technical Report, Stanford University (1998)
11. Haveliwala, T.H.: Topic-sensitive pagerank. *WWW* (2002)
12. Maciel, J., Costeira, J.P.: A global solution to sparse correspondence problems. *PAMI* (2003)
13. Gold, S., Rangarajan, A.: A graduated assignment algorithm for graph matching. *PAMI* (1996)
14. van Wyk, B.J., van Wyk, M.A.: A pocs-based graph matching algorithm. *PAMI* (2004)
15. Lee, J., Cho, M., Lee, K.M.: Graph matching algorithm using data-driven markov chain monte carlo sampling. *ICPR* (2010)
16. Gori, M., Maggini, M., Sarti, L.: Exact and approximate graph matching using random walks. *PAMI* (2005)
17. Robles-Kelly, A., Hancock, E.R.: String edit distance, random walks and graph matching. *IJPRAI* (2004)
18. Sinkhorn, R.: A relationship between arbitrary positive matrices and doubly stochastic matrices. *Ann. Math. Statistics* (1964)
19. Langville, A.N., Meyer, C.D.: Deeper inside pagerank. *Internet Mathematics* (2003)
20. Seneta, E.: Non negative matrices and markov chains. Springer (2006)
21. Munkres, J.: Algorithms for the assignment and transportation problems. *SIAM* (1957)
22. Matas, J., Chum, O., Urban, M., Pajdla, T.: Robust wide baseline stereo from maximally stable extremal regions. *BMVC* (2002)
23. Lowe, D.G.: Object recognition from local scale-invariant features. *ICCV* (1999) 1150–1157
24. Cho, M., Lee, J., Lee, K.M.: Feature correspondence and deformable object matching via agglomerative correspondence clustering. *ICCV* (2009)