Graph Matching by Relaxation Technique

Seong Hak Cheong and Sang Uk Lee
School of Electrical Engineering,
Seoul National University, Seoul, 151-742, Korea
email : shcheong@phoenix.dwe.co.kr

ABSTRACT
In this paper, we describe a hybrid relaxation approach to a graph matching problem, by combining both the discrete and continuous relaxation techniques. Compatibility coefficient, critical factor for both relaxation techniques, is defined in terms of nodes and arcs attributes, and the distance measure between graphs is defined as the inner product of the probability vector and the compatibility vector. The discrete relaxation is used as a preprocessing step to determine the initial matching probabilities, and in the continuous relaxation stage, the final matching probabilities are computed by the gradient projection method. Experimental results show that the proposed algorithm is robust to the corruption of the topologies of the graphs, and the matching probabilities converge very rapidly, alleviating an enormous computational load required for the relaxation process.

1 Introduction
Relational matching is a task that pervades computer vision at both high and intermediate level. It is also considered as a relational isomorphism problem. When we only deal with binary relation or structural constraints, the relational matching is actually a graph matching or graph isomorphism problem. It is invariably poor initial image segmentation that limits the effectiveness of classical graph matching methods[3]. Since these segmentation errors corrupt the topologies of the relational graphs, isomorphism or exact matching may not be possible. Therefore, the matching process returns with incorrect or false results[3, 4].

Shapiro and Haralick[5] proposed a general version of inexact matching, in which images are described by weighted relational data structures, and inexactness of a matching is expressed by various errors for primitives and constraints. Solutions to the relational matching can be found by so-called blind tree search methods[1]. Although the tree search methods, combined with some heuristics, can improve the performance to a certain extent[4], these methods can only be applied to sparse data sets, like high level descriptions[1], due to the exponential complexity of tree searches.

Because of its intrinsically parallel nature and the capability to provide consistent interpretations from incomplete or uncertain input, the relaxation process provides an alternative approach for the inexact matching. Two distinct types of the relaxation algorithms have been proposed in [7]. The discrete relaxation is carried out by the iterative application of label discarding rule, while in the continuous relaxation, the matching probabilities are updated in the continuous probability domain.

In this paper, we propose a hybrid, combining the discrete and continuous relaxation, approach to the problem of graph matching.

We describe the matching process in terms of the attributed relational graph( ARG )[2], relational structure composed of attributed nodes and arcs. The graph representation of images offers several powerful capabilities that are very useful for image analysis, such as the proper handling of hierarchy of the images and topological invariances.

The outline of this paper is as follows. In section II, we describe basic definitions and terminology in ARG, and the notion behind the relaxation process. In section III, the proposed matching method is presented in detail, and section IV presents the results of the matching experiments on the synthetic and real images. Finally, section V gives the conclusions.

2 ARG and Relaxation Technique
2.1 Attributed Relational Graph ( ARG )
The ARG is a relational structure which consists of a set of nodes and a set of arcs, representing the relations between the nodes.

The detailed definition of ARG is given in [2], which we review briefly for the sake of completeness.

Definition : The ARG is defined as $G = \{ N, B, A, \ R, \ G_N, \ G_B \}$ whose component are as follows: $N = \{ n_1, n_2, \ldots, n_M \}$ is a finite set of nodes, where $|N|$ is the number of nodes in $N$. $B = \{ b_1, b_2, \ldots, b_M \}$ is a finite set of ordered node pairs (or directed branches( arcs )), i.e., $b = (n_i, n_j)$ for some $1 \leq i, j \leq |N|$. denoting
the arc from node \( n_i \) to node \( n_j \), where \( |B| \) is the number of arcs in \( B \). \( A \equiv \{ E_m(N) : (l_k(a_1), r_k(a_1)) \} \) is an alphabet of node attributes. \( E_m \) is the \( m \)th node entity, and \( l_k \) and \( r_k \) are the name and the value of the \( k \)th attribute, respectively. \( R \equiv \{ R_n(B) : (l_k(r_j), r_k(r_j)) \} \) is an alphabet of branch attributes. \( R_n \) is the \( n \)th relation, and \( l_k \) and \( r_k \) are the name and value of the \( k \)th attribute respectively. \( G_B \) is a function (or a set of functions) for generating the branch attributes. \( G_B \) is a function (or a set of functions) for generating the branching attributes.

2.2 Discrete and Continuous Relaxation Technique

Based on definitions of previous section, we discuss some of the notion behind the relaxation process. The relaxation process involves a set of units and a set of labels (names). Let \( N_1 = \{a_1, \ldots, a_N \} \) be a set of \( N \) units and \( N_2 = \{\lambda_1, \ldots, \lambda_M \} \) be a set of \( M \) labels, where \( N_1 \) and \( N_2 \) are sets of the nodes in the ARG \( G_1 \) and \( G_2 \), respectively.

The discrete relaxation is carried out by the iterative application of label discarding rule\[7\].

In the continuous relaxation, the labeling state for each unit \( a_i \in N_1 \) is represented by an vector :

\[
\vec{p}_i = [p_i(\lambda_j) | \lambda_j \in N_2],
\]

where \( p_i(\lambda_j) \geq 0 \), and the sum for all \( \lambda_j \) is equal to 1.

The real value \( p_i(\lambda_j) \in [0, 1] \) reflects the strength with which \( a_i \) is matched with \( \lambda_j \).

The compatibility vector \( \vec{q}_i = [q_i(\lambda_j) | \lambda_j \in N_2] \) is defined in terms of \( p_i(\lambda_j) \) and compatibility coefficient \( r(i, \lambda_j, k, \lambda_l) \), given by

\[
q_i(\lambda_j) = \Phi(p_i(\lambda_j), r(i, \lambda_j, k, \lambda_l)),
\]

and the \( q_i(\lambda_j) \) represents the strength that unit \( a_i \) matches with label \( \lambda_j \), considering the compatibility coefficient \( r(i, \lambda_j, k, \lambda_l) \) and the present probability \( p_i(\lambda_j) \).

3 Matching Algorithms

First, we define the compatibility coefficient as

\[
r(i, \lambda_j, k, \lambda_l) = \beta \gamma \left( \sum_n \frac{|\gamma_p(i, k) - \gamma_p(\lambda_j, \lambda_l)|}{\delta_n} \right) + (1 - \beta) \gamma \left( \sum_n \frac{|\gamma_p(k) - \gamma_p(\lambda_l)|}{\alpha_n} \right)
\]

where \( \beta \) is the relative weight between node and edge attributes of ARG, and \( \alpha, \delta \) are the weights to each attribute, and \( \gamma \) is any function satisfying the monotonically decreasing property, respectively.

In (3), \( |\cdot| \) can be defined in many different ways, according to the attribute type, i.e., whether it is symbolic or numerical. A maximum value of the compatibility coefficient is obtained, if the arc attributes and node attributes are equal for units and labels. Large differences between attributes yield a null value of the compatibility coefficient.

Next to the definition of compatibilities, in order to find best mapping from \( N_1 \) to \( N_2 \), it is necessary to define the match evaluation function\[1\]. We use a simple criterion based on the inner product of probability vector \( \vec{p}_i \) and compatibility vector \( \vec{q}_i \), given by

\[
C(\vec{p}_1, \vec{p}_2, \cdots, \vec{p}_{|N_1|}) = \sum_{i=1}^{|N_1|} \vec{p}_i \cdot \vec{q}_i
\]

where \( |N_1| \) is the number of nodes in \( G_1 \). \( \vec{q}_i \) is a function of \( \vec{p}_i \)'s and compatibility coefficients as discussed in the next section, respectively. The criterion \( C \) in (5), which is also the distance measure between ARGs, is used for both estimating the initial matching probabilities and the optimization process for computing the final matching probabilities with different \( \vec{q}_i \).

The relaxation process converges to a local maximum in the vicinity of the initial probabilities. Accordingly, the initial probabilities affect the convergence rate and the performance of relaxation algorithm significantly. We employ the discrete relaxation to estimate the initial matching probabilities, which we shall present in the next section.

3.1 Estimation of the Initial Matching Probabilities

The discrete relaxation can be formally abstracted as follows\[7\]. If the constraint \( \Lambda \) is the set of all pairs \( \lambda_i, \lambda_l \), such that label \( \lambda_j \) at unit \( a_i \) is compatible with label \( \lambda_k \) at unit \( a_k \), the discrete relaxation is accomplished by means of the label discarding rule: discard a label \( \lambda_j \) at a node \( i \), if there exists a neighbor \( k \) of \( i \) such that every label \( \lambda_l \) currently assigned to \( k \) is incompatible with \( \lambda_j \) at \( i \), i.e., \( (\lambda_j, \lambda_l) \notin \Lambda \). In the discrete relaxation, we should determine the set \( \Lambda \), i.e., whether \( r(i, \lambda_j, k, \lambda_l) \) is 1 or 0.

We first define the discrete constraints set loosely as

\[
R(i, \lambda_j, k, \lambda_l) = \begin{cases} 1, & r(i, \lambda_j, k, \lambda_l) \geq t \\ 0, & \text{otherwise} \end{cases}
\]

where \( t \) is a given threshold. But, the threshold is adjusted after considering the criterion \( C \) and compatibility vector \( \vec{q}_i \), which is defined as

\[
q_i(\lambda_j) = \sum_k (R(i, \lambda_j, k, \lambda_l)).
\]

Then, the criterion (5) is of first order, with respect to the probability \( p_i(\lambda_j) \). So the final probability in the first stage \( \varphi_i(\lambda_j) \) can be computed directly from \( q_i(\lambda_j) \), given by

\[
\varphi_i(\lambda_j) = \begin{cases} 1 & \text{if } q_i(\lambda_j) = \max_k \{q_i(\lambda_j)\} \\ 0 & \text{otherwise} \end{cases}
\]
For each \( \phi_i(\lambda_j) = 1 \), there exist \( k \), \( \lambda_i \) pairs that support the maximum compatibility function \( q_k(\lambda_j) \). All \( k \), \( \lambda_i \) pairs can be stored in the form of a table, called the maximum-compatibility-pair table: there exist \( N \) tables, one for each maximum compatibility function. Table 1 shows the general form of the maximum-compatibility-pair table for \( i, \lambda_j \) pair.

Let us denote the maximum-compatibility-pair table by \( T_{ij} \). For each \( i, \lambda_j \) pair, \( k \), \( \lambda_i \) pairs in the maximum-compatibility-pair table denote the sets of pairs supporting the maximum \( q_k(\lambda_j) \).

If we adjust the discrete constraints set, according to

\[
R(i, \lambda_j, k, \lambda_i) = \begin{cases} 
1, & \text{if } (\lambda_j, \lambda_i) \in T_{ik} \\
0, & \text{otherwise.}
\end{cases}
\]

Then, we can count the number of units \( a_{ki} \)'s of an unit \( a_i \), which has labels compatible to a given label \( \lambda_j \) at \( a_i \) by

\[
p_k(\lambda_j)^{(0)} = \frac{1}{N} \sum_k \left( \max_{\lambda_i} \{ R(i, \lambda_j, k, \lambda_i) \phi_k(\lambda_i) \} \right).
\]

It can be easily seen that if \( p_k(\lambda_j)^{(0)} \) in (10) is equal to 1, then \( T = \Lambda \), and the unit \( a_i \) matches with the label \( \lambda_j \). Hence the next continuous relaxation stage is unnecessary. Although \( p_k(\lambda_j)^{(0)} \) is not 1, it can be used as the initial probability in the next continuous relaxation stage, since it reflects the strength for unit \( a_i \) to match with label \( \lambda_j \).

### 3.2 Computation of the Final Matching Probabilities

The final probabilities are computed from the initial probabilities using the continuous relaxation technique. Mathematically, the continuous relaxation technique can be viewed as an iterative process for minimizing certain criterion[8].

If we define the compatibility function as [6]

\[
q_k(\lambda_j) = \sum_{\lambda_i} \phi_k(\lambda_i) p_k(\lambda_i),
\]

then the criterion (5) is of the second order, with respect to the probability \( p_k(\lambda_j) \). Hence solution can be found by many different optimization techniques. Rather than adopting one of more elaborate schemes, we employ the gradient projection method, since it is known to locate an optimum point quickly when, as in our algorithm, the initial point is close to an optimum point. The gradient of the criterion \( C \) in (5) is \( \frac{\partial C}{\partial p_k(\lambda_j)} = 2q_k(\lambda_j) \), hence the iteration of the \( \tilde{p}_k \)'s is carried out, according to

\[
\tilde{p}_k^{(n+1)} = \tilde{p}_k^{(n)} + \alpha (\tilde{q}_k^{(n)} - \tilde{q}_k^{(n)}),
\]

where \( f \) and \( s \) are identity matrix and step size, respectively, and \( \tilde{p}_k \) is a vector for the satisfaction of constraints. The \( \tilde{p}_k \) can be either \( \tilde{p}_k \) or normalized \( \tilde{q}_k \). The step size \( s \) in (12) can be determined optimally by the method in [9]. If the obtained final probability \( p_k(\lambda_j) \) is equal or close to 1, then \( (i, \lambda_j) \in f \), that is node \( i \) in \( N_1 \) matches with node \( \lambda_j \) in \( N_2 \). Otherwise \( (i, \lambda_j) \notin f \), implying the null match \( \phi \).

### 4 Experimental Results

The proposed graph matching method is evaluated on the synthetic and real images.

We choose the node alphabet as the line segments( L ) with the length as their attributes, and arcs of the ARG representation represent the relationships between line segments. The arcs are chosen to correspond to following relations: Joint ( J ) with joint angle ( a ), and Others ( O ) with attribute ( d, \( \mu, \theta \) ), where d is the disance, \( \mu \) is the direction of vector between center points, and \( \theta \) is the angle between the two line segments, respectively.

Experimental results with synthetic line segments are presented in Figure 1. Figure 1 (a) contains the various linear structures composed of line segments, where uniform noise is added to distort the original structures. In graph’s viewpoint, the noise generates the corruption of topologies in the original graph. But, Figure 1 (c) shows that the proposed matching algorithm can locate the most similar linear structures with the model in the input image.

![Figure 1: location of the desired structures](image)

Figure 2 - 3 shows the matching ARGs extracted from real aerial images. First we extract the line segments from input image and model, as shown in Figure 2(b), (d). Model correspond to the upper left part of the input image, but rotated counter-clockwise by 90 degrees.

Through matching between two ARGs, matched nodes( line segments ) in the input image can be identified as shown in Figure 3.

The initial probabilities for the \( i \)th node in the Figure 2 (b) are shown in Figure 4(a). Figure 4(a) shows the difference between initial probabilities estimated from the proposed method and those from comparing only node attributes. Those computed from the proposed method is shown to be superior, since fewer unit-label
pairs yield a higher probability (the final $p_i(\lambda_k)^* = 1$). This adjustment of the initial probability makes the iteration process converge very rapidly as shown in Figure 4(b). However, the iteration process, starting from the initial probability estimated from only comparing node attributes, has not yet converged.

Figure 4: Adjust : proposed method. Node_Comparison : initial probabilities are assigned by only comparing node attributes

5 Concluding Remarks

In this paper, we have proposed an hybrid graph matching algorithm based on the discrete and continuous relaxation techniques. The discrete relaxation is used as a preprocessing step for the estimation of the initial matching probabilities, and the continuous relaxation is used for the computation of the final matching probabilities.

The proposed matching algorithm, evaluated on both synthetic and real images, was shown to be robust to the corruption of topologies of the ARGs, and provides additional advantage of converging very rapidly. This fast convergence can be achieved by the nearly accurate estimation of the initial matching probabilities and gradient projection method with optimal step size selection.

References


