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 converges 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: $N = \{a_1, a_2, \cdots, a_{|N|}\}$ is a finite set of nodes, where $|N|$ is the number of nodes in N. $B = \{r_1, r_2, \cdots, r_{|B|}\}$ is a finite set of ordered node pairs(or directed branches(arcs)), *i.e.*, $b = (n_i, n_j)$ for some $1 \le i, j \le |N|$,denoting

[0] S. H. Cheong is currently a research engineer in DAEWOO Ele., Co., Ltd., Seoul 100 - 714, Korea

the arc from node n_i to node n_j, where $|B|$ is the number of arcs in B. $A = \{E_m(N) : (l_k(a_i), v_k(a_i))\}$ is an alphabet of node attributes. E_m is the mth node entity ,and l_k and v_k are the name and the value of the kth attribute, respectively. $R = \{R_n(B) : (\zeta_k(r_i), \eta_k(r_i))\}$ is an alphabet of branch attributes. R_n is the nth relation, and l_k and v_k are the name and value of the kth attribute respectively. G_N is a function(or a set of functions) for generating the node attributes. G_B is a function(or a set of functions) for generating the branch 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 $N_1 = \{a_1, \cdots, a_N\}$ be a set of N units and $N_2 = \{\lambda_1, \cdots, \lambda_M\}$ 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 units $a_i \in N_1$ is represented by an vector :

$$\vec{p}_i = [p_i(\lambda_j) | \lambda_j \in N_2], \tag{1}$$

where $p_i(\lambda_j) \geq 0$, and the sum for all λ_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 λ_j.

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

$$q_i(\lambda_j) = \Phi(p_i(\lambda_j), r(i, \lambda_j, k, \lambda_l)), \tag{2}$$

and the $q_i(\lambda_j)$ represents the strength that unit a_i matches with label λ_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

$$\begin{aligned} & r(i, \lambda_j, k, \lambda_l) & (3) \\ = \ & \beta\ \gamma\left(\sum_n \left(\frac{\|\eta_n(i,k) - \eta_n(\lambda_j, \lambda_l)\|}{\delta_n}\right)\right) & (4) \\ + \ & (1-\beta)\ \gamma\left(\sum_n \left(\frac{\|v_n(k) - v_n(\lambda_l)\|}{\alpha_n}\right)\right) & \end{aligned}$$

where β is the relative weight between node and edge attributes of ARG, and α, δ are the weights to each attribute, and γ is an 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 a simple criterion based on the inner product of probability vector $\vec{p}_i$ and compatibility vector $\vec{q}_i$, given by

$$C(\vec{p}_1, \vec{p}_2, \cdots, \vec{p}_{|N_1|}) = \sum_{i=1}^{|N_1|} \vec{p}_i \cdot \vec{q}_i \tag{5}$$

where $|N_1|$ is the number of nodes in $G1$, $\vec{q}_i$ 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 Λ_{ik} is the set of all pairs λ_j, λ_l, such that label λ_j at unit a_i is compatible with label λ_k at unit a_k, the discrete relaxation is accomplished by means of the label discarding rule: discard a label λ_j at a node i, if there exists a neighbor k of i such that every label λ_l currently assigned to k is incompatible with λ_j at i, *i.e.*, $(\lambda_j, \lambda_l) \notin \Lambda_{ik}$. In the discrete relaxation , we should determine the set Λ_{ik}, *i.e.*, whether $r(i, \lambda_j, k, \lambda_l)$ is 1 or 0.

We first define the discrete constraints set loosely as

$$R(i, \lambda_j, k, \lambda_l) = \begin{cases} 1, & r(i, \lambda_j, k, \lambda_l) \geq t \\ 0, & \text{otherwise,} \end{cases} \tag{6}$$

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

$$q_i(\lambda_j) = \sum_k (R(i, \lambda_j, k, \lambda_l)). \tag{7}$$

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

$$\varphi_i(\lambda_j) = \begin{cases} 1 & \text{if } q_i(\lambda_j) = \max_{\lambda_j}\{q_i(\lambda_j)\} \\ 0 & \text{otherwise.} \end{cases} \tag{8}$$

k	1	2	$\cdots$	N
λ_l	k_1's	k_2's	$\cdots$	k_N's

Table 1: maximum-compatibility-pair table for (i, λ_j)

For each $\varphi_i(\lambda_j) = 1$, there exist k, λ_l pairs that support the maximum compatibility function $q_i(\lambda_j)$. All k, λ_l 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, λ_j pair.

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

If we adjust the discrete constraints set, according to

$$R(i, \lambda_j, k, \lambda_l) = \begin{cases} 1, & \text{if } (\lambda_j, \lambda_l) \in T_{i,k} \\ 0, & \text{otherwise.} \end{cases} \qquad (9)$$

Then, we can count the number of units a_k's of an unit a_i, which has labels compatible to a given label λ_j at a_i by

$$p_i(\lambda_j)^{(0)} = \frac{1}{N}\sum_k(\max_{\lambda_l}\{R(i, \lambda_j, k, \lambda_l)\}\varphi_k(\lambda_l)). \qquad (10)$$

It can be easily seen that if $p_i(\lambda_j)^{(0)}$ in (10) is equal to 1, then $T = \Lambda$, and the unit a_i matches with the label λ_j. Hence the next continuous relaxation stage is unnecessary. Although $p_i(\lambda_j)^{(0)}$ 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 λ_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]

$$q_i(\lambda_j) = \sum_k(\sum_{\lambda_l} r(i, \lambda_j, k, \lambda_l)p_k(\lambda_l)), \qquad (11)$$

then the criterion (5) is of the second order, with respect to the probability $p_i(\lambda_j)$. 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 $\frac{\partial C}{\partial p_i(\lambda)} = 2q_i(\lambda_j)$, hence the iteration of the $\vec{p_i}$'s is carried out, according to

$$\vec{p_i}^{(n+1)} = I\vec{p_i}^{(n)} + s(\vec{q_i}^{(n)} - \vec{n_i}^{(n)}), \qquad (12)$$

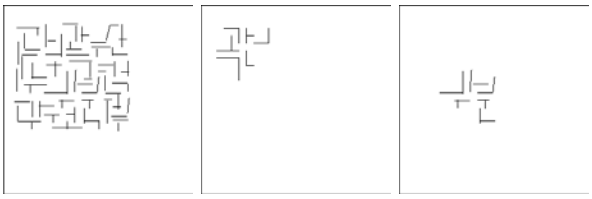
where I and s are identity matrix and step size , respectively, and $\vec{n}_i$ is a vector for the satisfaction of constraints. The $\vec{n}_i$ can be either $\vec{p}_i$ or normalized $\vec{q}_i$. The step size s in (12) can be determined optimally by the method in [9]. If the obtained final probability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

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 (α), and Others (O) with attribute (d, μ, θ), where d is the disance, μ is the direction of vector between center points, and θ 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.



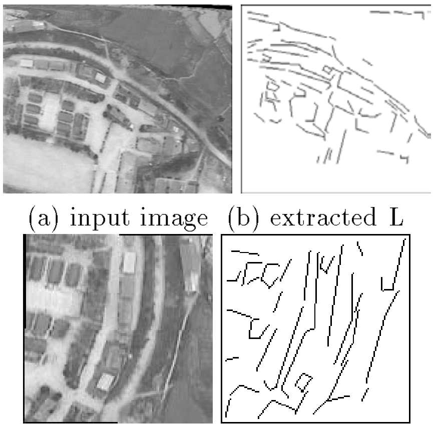
(a) input image (b) model (c) matching result

Figure 1: location of the desired structures

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.

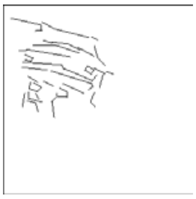
The initial probabilities for the ith node in the Figure 2 (b) are shown in Figure 4(a). Figure 4(a) shows the difference between initial probabilities estimated from the proposed method and those from comparing only node attributes. Those computed from the proposed method is shown to be superior, since fewer unit-label



(a) input image (b) extracted L

(c) model (d) extracted L

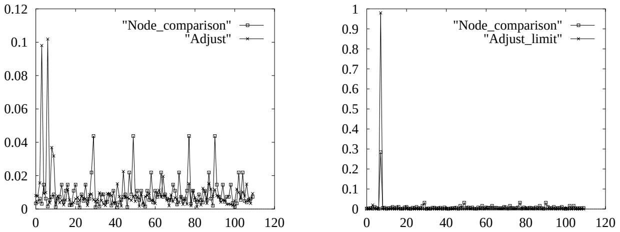
Figure 2: line segments extracted from the model and input image



(a) matched line segments

Figure 3: matching result

pairs yield a higher probability(the final $p_i(\lambda_8)^* = 1$). This adjustment of the initial probability makes the iteration process converges very rapidly as shown in Figure 4(b). However, iteration process, starting from the initial probability estimated from only comparing node attributes, has not yet converged.



(a) initial prob. (b) 20 th iteration

Figure 4: Adjust : proposed method, Node_Compatison : 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

[1] G. Vosselman, *Relational matching.* Berlin : Springer-Verlag,1992.

[2] M. A. Eshera, “ A graph distance measure for image analysis, ” *IEEE Trans. Systems , Man and Cybernetics,* vol. 14, no. 3, pp.398-408, June 1984.

[3] L Shapiro and R. M. Haralick, “ A metric for comparing relational description, ” *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 7, no. 1, pp.90-94, Jan. 1985.

[4] M. A. Eshra and K. S. Fu, “ An image understanding system using attributed symbolic representation and inexact graph matching, ” *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 8, no. 5, pp. 604-617, Sep. 1986.

[5] L. G. Shapiro and R. M. Haralick, “ Structural descriptions and inexact matching,” *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 3, no. 5, pp. 504-519, Sep. 1981.

[6] A. Rosenfeld, R. A. Hummel, and S.W. Zucker, “ Scene labeling by relaxation operations, ” *IEEE Trans. Systems , Man and Cybernetics,* vol. 6, no. 6, pp. 420-453, June 1976.

[7] R. A. Hummel and S. W. Zucker, “ On the foundation of relaxation labeling process, ”*IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 5, no. 3, pp. 267-286, May 1983. Jan. 1979. vol. 4, no. 5, pp. 476-484, Sep. 1982.

[8] O. D. Faugeras and M. Berthod, “ Improving consistency and reducing ambiguity in stochastic labeling : An optimization approach, ” *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 3, no. 4, pp. 412-424, July 1981.

[9] M. Aoki, *Introduction to optmization technique,* New York : The Macmilan Company, 1971.