Available online at www.sciencedirect.com

ScienceDirect

Pattern Recognition Letters 28 (2007) 1428–1437

Pattern Recognition Letters

www.elsevier.com/locate/patrec

Inexact graph matching using a genetic algorithm for image recognition

Surapong Auwatanamongkol *

Department of Computer Science, School of Applied Statistics, National Institute of Development Administration, Bangkok 10240, Thailand

Received 30 April 2005; received in revised form 11 January 2007
Available online 28 February 2007

Communicated by T.K. Ho

Abstract

Exact graph matching using a genetic algorithm for image recognition has been introduced in previously published work. The algorithm was based on angle matching between two given graphs. It has proven to be quite effective in exact graph matching. However, the algorithm needs some modifications in order to handle cases where the number of nodes, shapes and rotations of the two graphs are different. This paper presents modifications such as the introduction of node exemption, inexact matching between straight lines and curves in the graphs and consideration of rotational degrees of the graphs. Each angle in a graph is also given a weight to indicate the significant degree of identifying the graph. A multi-objective function is used to reflect the similarity between two graphs. The experiments designed to evaluate the algorithm have shown very promising results. It is highly accurate in matching graphs with dissimilarities in shape, number of nodes and degrees of rotation.

Keywords: Genetic algorithm; Inexact graph matching; Image recognition and classification

1. Introduction

Patterns or images can be effectively represented as graphs and it is possible to identify a given image by matching the graphic representation of the image against those of other known images to look for isomorphism. The graph matching problem has been proven to be NP-complete. When the numbers of nodes in the two graphs are different, the graph matching problem becomes inexact and more difficult. Several techniques have been proposed to solve this problem, e.g. probabilistic relaxation (Bengoetxea et al., 2001; Cesar et al., 2002; Christmas et al., 1995; Coughlan and Shen, 2004; Skomorowski, 1999; Williams et al., 1997; Wilson and Hancock, 1999), EM algorithm (Cross and Hancock, 1998; Finch et al., 1998), neural networks (Lee and Liu, 2000; Lyul and Hong, 2002; Shams et al., 2001), decision trees (Messmer and Bunke, 1999) and a genetic algorithm (Cross et al., 1997, 2000; Khoo and Suganthan, 2002; Singh et al., 1997).

Previously, an exact graph matching technique for image recognition using a Genetic Algorithm was introduced (Auwatanamongkol, 2000). The basic idea of the algorithm is to find the best match between the nodes of an unknown input graph and the nodes of a known graph in order to find the highest degree of matching. This can be measured in terms of how well each angle between two adjacent links of one graph is matched with its counterpart on the other graph. The algorithm showed promise when matching two similar graphs with the same number of nodes.

However, some optimization and several modifications still needed to be carried out on the algorithm in order to improve its performance and accuracy for recognizing similarities between two graphs. Specifically, it should be capable of handling cases in which the number of nodes on the two graphs are not equal. In such cases, some nodes are left

* Tel.: +662 727 3032; fax: +662 374 4061.
E-mail address: surapong@as.nida.ac.th

0167-8655/$ - see front matter © 2007 Elsevier B.V. All rights reserved.
doi:10.1016/j.patrec.2007.02.013

unmatched and some need special exemption, otherwise they can cause a considerable reduction in the overall degree of matching. Criteria need to be set up in order to determine which of the nodes can be exempted. The disparity between the lengths of any two matched links on the two graphs can also be considered when evaluating the degree of matching. Besides structural matching, rotational matching can play a significant role in evaluating the two graphs. The rotational similarity between the two graphs should be incorporated into the degree of matching to enhance its accuracy.

This paper proposes these modifications to the previously reported algorithm. The modified algorithm was tested and shown to increase the accuracy and discriminatory power of the original algorithm.

2. Image pattern representation

An image pattern is assumed to have been segmented and is represented as a graph. Therefore, lines in an image pattern can be subdivided into straight line segments. This segmentation process is referred to as "pattern quantization" and its goal is to extract structural information from the pattern. The known image patterns must also be segmented in such a way that the numbers of nodes in all of the known image patterns are roughly the same. This will help to equalize the chance that each node in the input image pattern can match with its counterpart in any known image pattern.

In addition, and to make the representation of the image pattern independent of rotational and scaling factors which can complicate the graph matching, an image pattern transformation is proposed that eliminates the rotation and scaling dependencies and produces a new representation which suits the graph matching algorithm. The pattern transformations consist of the following steps:

1. Between any pair of nodes that are not directly connected, a new link is introduced which does not cross over any original links. These new links are referred to as "indirect links" and the original ones "direct links". For instance, the indirect links AC and BD (dotted lines) are introduced into the pattern shown in Fig. 1. Notice that some angles on the graph are divided by indirect links into sub-angles, e.g. α and β.
2. Each direct or indirect link incident to a particular node is represented by the degree of the angle (or sub-angle) at the node, which is formed by the link and its succeeding link in a clockwise direction, e.g. the link AC, AD and AB are represented by the degrees of the angles α, β and γ, respectively.

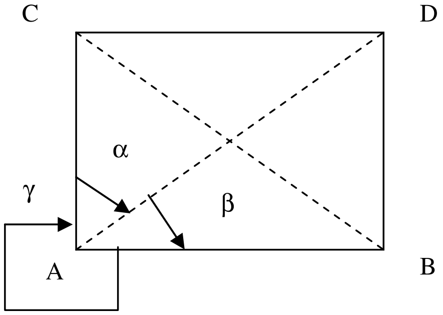

Fig. 1. A graph with augmented indirect links (dotted lines).

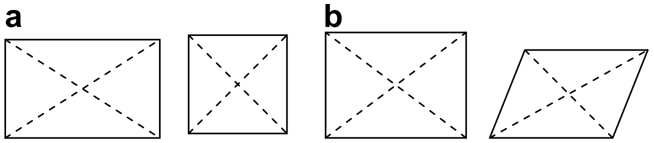

Fig. 2. Similar graphs distinguished by the introduction of indirect links.

The new representation uses angles which have not been subjected to scaling and rotation factors. The introduction of indirect links also gives some extra structural information, e.g. line of sight between two nodes. This extra information is needed as it can help distinguish one graph from another. For instance, the two graphs, a square and a rectangle, shown in Fig. 2(a), can be distinguished by the degrees of angles associated with the indirect links. Two similar graphs shown in Fig. 2(b), which cannot be distinguished by the lengths of matched links since they are the same, can now be distinguished by matching angles associated with direct and indirect links.

Each angle in the original graph (without augmented indirect links) is assigned with an angle weight that represents the significant level of the angle that helps identify the graph. Accordingly, the sum of all angle weights in a graph must be equal to 1. Once the graph is augmented with indirect links, the weight of a sub-angle should be determined from the weight of its respective angle. The sum of weights for all sub-angles of an angle must be equal to the weight of the angle itself. For instance, the sum of the weights of sub-angles α and β must be equal to the weight of the angle BAC. Using this constraint, it is possible to calculate the weights of sub-angles from the weight of the angle by distributing the weight to the sub-angles as follows:

$$W_i = W_j * p/n_j + W_j * (1-p) * d_i/d_j,$$

where W_i and W_j are the weight of sub-angle i and its associated angle j, respectively, n_j is the number of sub-angles in j, and d_i and d_j are the degrees of angle i and j, respectively. According to the above equation, a part of W_j, W_j*p, is divided equally for each of the sub-angles of j while the other part of W_j, W_j*$(1-p)$, is divided according to the proportions of the sub-angle and of j in degrees. The first part divided equally among the sub-angles would guarantee that any sub-angles with small degrees will get a share of its respective angle weight and will not be dominated by the larger sub-angles. A reasonable value of p could be 0.3.

3. Inexact graph matching

The objective of inexact graph matching is to find the best 1-to-1 match between the nodes in a given input graph and the nodes in a known graph, which yields the maximum degree of matching. In addition, it would be beneficial to match as many straight line segments between the two graphs as possible. The candidate matching solution can simply be encoded as a vector of integer numbers. The vector length is n where n is the number of nodes in the input graph. Each element in the vector has a value ranging between 0 and m, where m is the number of nodes in the known graph. The ith element of the vector represents the node id of the known graph that the ith node of the input graph is matched with. Since the number of nodes in the two graphs, i.e. m and n, can be different, some nodes in both graphs must be left unmatched. The zero value on a particular element designates a null matched or unmatched condition for the corresponding node in the input graph.

However, if these unmatched nodes are part of a curve connected between two matched nodes, they should be exempted from the evaluation of the degree of matching. For instance, consider nodes B and C in Fig. 3. These two nodes are part of a curve connected between nodes A and D in an input graph. If the curve ABCD is matched with the straight line PQ in a known graph, i.e. node A matches with node P, node D matches with node Q and nodes B and C are unmatched, nodes B and C should be excluded from the node matching process.

Let us define the straightness degree of a line as μ. μ can be quantified using a simple ratio, ε, between the length of the straight line connecting the two end points and the length of the line itself. For instance, the ratio ε for the line ABCD can be computed as

$$\varepsilon_{\mathrm{ABCD}} = \text{Length of AD}/\text{Length of ABCD}.$$

The ratio ranges between 0 and 1, where 1 is the most favorable (a straight line) and 0 is the least favorable (a curve). Consequently, the ratio can be used as a simple means to quantify the degree of straightness of the line as follows:

$$\mu_{\mathrm{ABCD}} = (\varepsilon_{\mathrm{ABCD}}) ** 2.$$

The square function corresponds to the concentration operation in fuzzy logic which acts as a mapping function between the degree of straightness μ and the ratio ε. The mapping function is shown in Fig. 4. Note that the degree of straightness of an indirect link is always equal to 1.

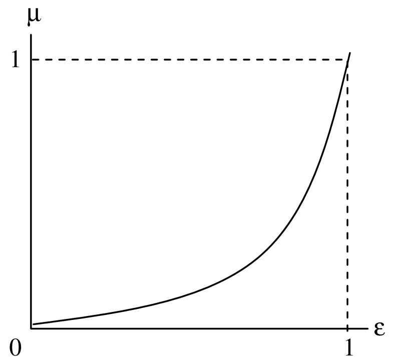

Fig. 4. The mapping function S between μ and ε.

Let L_{ABCD} and L_{PQ} be the normalized lengths of the lines ABCD and PQ, respectively. The normalized length of a line is defined as the ratio of the line length and the total length of all line segments in its respective graph. The line length disparity measure between the lines ABCD and PQ, $\sigma(\mathrm{ABCD},\mathrm{PQ})$, can now be computed as follows:

$$\sigma(\mathrm{ABCD},\mathrm{PQ}) = (\min(L_{\mathrm{ABCD}},L_{\mathrm{PQ}})/\max(L_{\mathrm{ABCD}},L_{\mathrm{PQ}})) ** 2.$$

The normalized line length is used instead of the absolute line length in order to eliminate the effect of the scaling factor and the square function is used as a concentration mapping function between the normalized line length ratio and the line length disparity measure. The disparity measure value lies between 0 and 1, and approaches 1 as the two normalized line lengths approach parity.

We can now define the degree of exemption E for an exempted node as a product of the degree of straightness of the line that the node is in and the line length disparity measure, since the two factors will dictate the degree to which the node should be exempted.

Consider the two graphs in Fig. 5. If nodes A, D and E are matched with nodes P, Q and R, respectively, while nodes B and C are exempted, the indirect link AD and the curvature line ABCD must be replaced by the direct link AD. All indirect links involved with the exempted nodes, i.e. AC, AD, BE and CE, must now be discarded.

From Fig. 5 we can see that the curve ABCD, which is replaced by the indirect link AD, can be matched with the direct link PQ, the direct link AE can be matched with the direct link PR and the direct link DE can be matched with

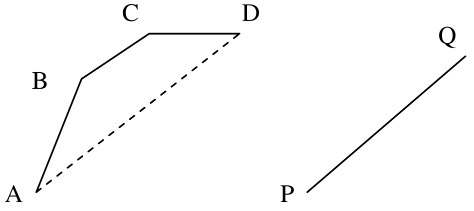

Fig. 3. Inexact matching between two lines, ABCD and PQ.

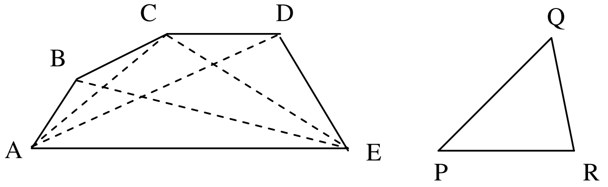

Fig. 5. An input graph with exempted nodes (B and C) and its known counterpart.

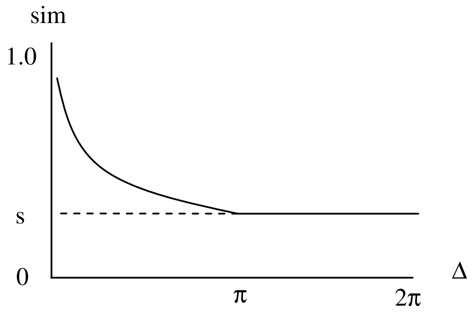

Fig. 6. Similarity function used for angle matching.

the direct link QR. In the first case where the original graphs are considered, the degree of matching for the angle BAE, ϕ_{BAE}, can be defined in terms of the absolute difference in value between the magnitudes of the angles DAE and QPR, Δ, the degrees of straightness of ABCD and AE and the weight of the angle BAE, W_{BAE}, as follows:

$$\phi_{\mathrm{BAE}} = W_{\mathrm{BAE}} * \mu_{\mathrm{ABCD}} * \mu_{\mathrm{PR}} * \mathrm{sim}(\Delta),$$

where sim is a similarity function defined over the difference Δ. The value of the similarity function should decrease as the difference Δ increases. The function value has a maximum of 1.0 when Δ is equal to 0 and decreases to a constant s when Δ approaches 2π or 360°. Fig. 6 illustrates an example of the similarity function sim where the value of s has been set at 0.3. The function can be defined as follows:

$$\begin{aligned} \mathrm{Sim} &= s && \text{if } \Delta > \pi, \\ &= s + (1 - s) * (1 - \Delta/\pi) * * 2 && \text{otherwise.} \end{aligned}$$

The degree of matching for the angle QPR can be defined in the same way as follows:

$$\phi_{\mathrm{QPR}} = W_{\mathrm{QPR}} * \mu_{\mathrm{PQ}} * \mu_{\mathrm{PR}} * \mathrm{sim}(\Delta).$$

Notice that the maximum value for a degree of matching of an angle is equal to the weight on the angle itself. In addition, since maximizing the degree of matching is desirable, the two lines that form the angle must be as straight as possible. In the second case where indirect links are added, the angle BAE is subdivided into two sub-angles. The degree of matching for the angle can be computed in the same way as in the first case. However, the weight of any sub-angle which has its end nodes exempted must be given a penalty because of the existence of the exempted end nodes. For instance, since the sub-angle CAD has its end node C exempted, its weight must be reduced by a penalty factor as follows:

$$W'_{\mathrm{CAD}} = W_{\mathrm{CAD}} * \mathrm{minimum}(E_{\mathrm{C}}, E_{\mathrm{D}}).$$

The penalty factor is basically the smaller of the degrees of exemption of the two end nodes of the sub-angle, bearing in mind that the degree of exemption for a non-exempted node, e.g. node D, is equal to one. Finally, the penalized weight of an angle which consists of sub-angles can be computed as the sum of the penalized weights of its sub-angles. For instance, the penalized weight of the angle BAE can be computed as follows:

$$W'_{\mathrm{BAE}} = W'_{\mathrm{BAC}} + W'_{\mathrm{CAD}} + W'_{\mathrm{DAE}}.$$

The degree of matching for an angle incident at an exempted node, e.g. angles ABC and BCD, should be approximated otherwise the angle could be accounted for as an unmatched angle. Approximations must be considered in the following two cases.

In the first case where no indirect links are added, the approximated value could be quantified as a product of the weight of the angle and the degree of exemption of the node that the angle is incident to. For instance, the approximated degree of matching of angle ABC in Fig. 5 can be computed as a product of the weight of angle ABC and the degree of exemption of node B.

In the second case when indirect links are added, the degree of matching of the angle ABC can be approximated using the degrees of matching of the angles BAC and CDE since the angle ABC could have indirect links connected to nodes similar to those of the angles BAC and CDE. The matching degree of the angle ABC can be computed using an averaged degree of matching of the two angles BAC and CDE as follows:

$$\phi_{\mathrm{ABC}} = W_{\mathrm{ABC}} * (\phi_{\mathrm{BAC}}/W_{\mathrm{BAC}} + \phi_{\mathrm{CDE}}/W_{\mathrm{CDE}})/2.$$

In the case where angles BAC or CDE is larger than 180°, only sub-angles of BAC or CDE that are visible from B or C, respectively, i.e. the sub-angles within 180° from the line BA and CD will be used instead of BAC or CDE.

With reference to particular node matching between two graphs, a degree of matching that one graph matches with the other can be computed using degrees of angle matching for all angles in the former graph as follows.

Let S_{a} be the sum of the degrees of angle matching ϕ for all angles in the graph.

Let S_{t} be the sum of the maximum values of the degrees of angle matching for all angles in the graph. S_{t} is actually the total of the angle weights in the graph, which is equal to one.

Therefore, the degree of matching for the graph matching with the other can be computed as

$$\phi(\mathrm{graph}) = S_{\mathrm{a}}/S_{\mathrm{t}} = S_{\mathrm{a}}.$$

Note that the degree of matching for one graph matching with the other may not be equal when applied the other way around since the numbers of unmatched and exempted nodes, as well as the individual angle matching degrees for both graphs, are not necessarily the same.

4. Structural fitness function

The degree of matching for a graph can also consist of the degree of matching for the original graph without indirect links and the degree of matching for the augmented graphs with indirect links. The latter's degree of matching is incorporated so that similar graphs can be distinguished by exploiting the extra structural information inferred by the indirect links. Hence, the total degree of matching between the two graphs can be defined as a multi-objective function that consists of four degrees of matching which need to be maximized at the same time.

Let G_1 and A_1 be the input graph and its corresponding augmented graph and G_2 and A_2 be the known graph and its corresponding augmented graph, respectively. Consequently, $\phi(G_1)$, $\phi(G_2)$, $\phi(A_1)$ and $\phi(A_2)$ are the degrees of matching for the graphs G_1, G_2, A_1 and A_2, respectively. We can use a simple weighting approach (Fonseca and Fleming, 1995) to combine all the matching degrees into one fitness value that represents the structural degree of matching between the two graphs. The structural fitness function can be expressed as follows:

$$\begin{aligned}\text{Structural fitness function}\\ &= w_1 * (w_2 * \phi(G_1) + (1 - w_2) * \phi(A_1))\\ &\quad + (1 - w_1) * (w_2 * \phi(G_2) + (1 - w_2) * \phi(A_2)),\end{aligned}$$

where w_1 is the weight for combining the degrees of matching for the input graph and the known graph, while w_2 is the weight for combining the degrees of matching for the original graph and its augmented graph. The weights w_1 and $1 - w_1$ reflect the preferences given to the input graph and known graph, respectively, whereas the weights w_2 and $1 - w_2$ reflect the preferences given to the original graph and its augmented graph respectively. If no preference is given to the input or known graphs and to the original or augmented graphs, the values of the weights w_1 and w_2 can both be set to 0.5.

5. Rotational fitness function

Besides structural similarity, rotational similarity needs to be considered since this can help distinguish one graph from the other. For instance, graphs of part of the alphabet N to Z can have a very high structural similarity but the rotational similarity is quite low since the degrees of rotation difference between the two is about 90°. Accordingly rotational similarity can be used to distinguish between the two graphs. The rotational similarity between each direct link and its counterpart in the other graph must be computed and then be used to compute the average for all of the link similarities, which represents the rotational similarity for the whole graph.

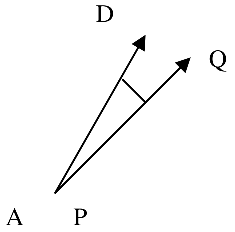

Fig. 7. Degree of rotation difference between two matched links, AB(CD) and PQ.

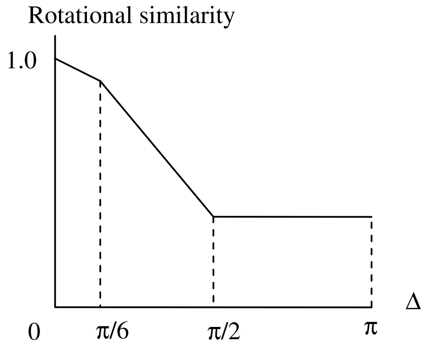

Fig. 8. A rotational similarity function defined over degree of rotation difference Δ.

When considering Fig. 7, let vectors AD and PQ represent the links AB(CD) and PQ from the graphs in Fig. 5, respectively. Node A can be superimposed onto node P since they are a match. Consequently, the difference in the degrees of rotation Δ between the two links can be measured in terms of the degree of angle DAQ. The difference can then be transformed into terms of rotational similarity using the function shown in Fig. 8. Hence, the rotational fitness for links AB(CD) and PQ can be defined in terms of their degree of straightness and rotational similarity as follows:

$$\begin{aligned}&\text{Rotational fitness of AB(CD) and PQ}\\ &\quad = \mu_{\text{ABCD}} * \mu_{\text{PQ}} * \text{Rotational similarity of ABCD and PQ.}\end{aligned}$$

Note that the rotational fit of links BC and CD are approximately equal to the rotational fitness of link AB(CD).

Finally, the rotational fitness function can then be computed using the previously described weighting scheme as follows:

$$\begin{aligned}&\text{Rotational fitness function}\\ &\quad = w_1 * \text{Rotational fitness of } G_1\\ &\qquad + (1 - w_1) * \text{Rotational fitness of } G_2.\end{aligned}$$

The two fitness functions, structural and rotational, can now be combined to form the total fitness function as follows:

$$\begin{aligned}\text{Fitness function} &= w_3 * \text{Structural fitness function}\\ &\quad + (1 - w_3) * \text{Rotational fitness function,}\end{aligned}$$

where w_3 is the weight that represents the preference on structural fitness. A reasonable value for w_3 is 0.7.

6. Population of the potential solutions

Two important issues concerning the population of the potential solutions are the population size and the algorithm used to initialize the population. The population size and initialization algorithm play significant roles in improving the performance of the genetic algorithm. Too small a population size can lead to local optima or premature convergences while too large a population size will exhaust processing time and memory space. A good initial population can help direct the search to the global optima and so shorten the search time. In this section, an initialization algorithm for the graph matching problem is proposed, which can help improve convergence whilst not requiring a very large population size.

Consider a node in an input graph and define an adjacent line of the node as a line starting at the node and ending at a node with an adjacent degree not equal to two or the node itself (forming a cycle in this case). In order to construct an initial population, allow each node of the input graph to be matched to each node on the known pattern graph. Once the input node is matched to a particular node of the known pattern graph, allow each adjacent line of the input node to be matched to each adjacent line of the known pattern node. Subsequently, each matching pair of adjacent lines will generate one population. When a line on the input graph is matched to a line on the known pattern graph, the corresponding starting and ending nodes of the two lines are matched with each other. The intermediate nodes inside the two lines are matched in sequential order. If the numbers of intermediate nodes on the two lines are not equal, some of the nodes on the longer line are left unmatched by random selection. The rest nodes in both patterns will be left unmatched, i.e. assigned with null mappings. By using this matching scheme, it is possible to generate mappings between nodes of the two patterns based on sequences of nodes in adjacent line segments, which form the building blocks for an optimal solution. The whole process can be repeated for many rounds until the desired number of populations is achieved.

7. The mutation operator

The mutation performed on a candidate solution is as follows. Firstly, the position of the element to be mutated is selected based on the degree of matching between a node and its counterpart. The idea is based upon a heuristic process where the elements or nodes with less degrees of matching should have a higher probability of mutation in order to improve the fitness value of the new solution. Suppose that an input graph node X is matched to a pattern graph node Y and the number of matched angles at nodes X and Y is ψ, the degree of matching between node X and node Y can be defined as

$$\min(\psi/\deg(\mathrm{X}), \psi/\deg(\mathrm{Y})),$$

where deg(X) and deg(Y) are the numbers of angles (number of edges) at nodes X and Y. The degree of matching between nodes X and Y is classified as the minimum value of the ratio between the number of matched angles and the number of angles at the two nodes. For an unmatched node, the degree of matching for the node is set to zero. A fitness proportionate selection algorithm can then be used to select the mutation point where the fitness value of each node or element can be determined as follows:

$$\begin{aligned}\text{Fitness} = {} & \text{Base fitness} + (1 - \text{Base fitness}) \\ & * (1 - \text{degree of matching between the node} \\ & \quad \text{and its counterpart}).\end{aligned}$$

The base fitness is the minimum value of the fitness for any node and is between 0 and 1. It is used to give the chance of any node with a perfect match being selected. Once the mutation node has been selected, the node will either be matched with any node on the pattern graph or else none. Since the number of nodes in the input pattern and known pattern may not be equal, the possibility that a node in an input graph will be matched to null should be equal to the reciprocal of the number of known pattern nodes if this number is less than or equal to the number of input pattern nodes, otherwise it should be equal to the ratio between the difference in the number of nodes between the two patterns and the number of known pattern nodes. In the event that the mutation node is chosen to match with null, the matching of the node is simply changed to null. On the other hand, a node in the known pattern needs to be selected to match with the mutation node.

Using the same heuristic approach as with the selection of mutation nodes, the selection of pattern graph nodes can be carried out based on the degree of matching between the pattern graph nodes and their counterparts. So that a 1-to-1 mapping relationship for the mutated solution can be maintained, if there is already an input graph node that matches with a selected pattern node, the input graph node will be matched with null instead. Once an input graph node has been reassigned to match with a new known graph node, a part of one adjacent line of the input graph node should be matched with a part of one adjacent line of the known graph node. This is to create a building block for the new solution. The part of the adjacent line in either graph can be randomly picked, i.e. the length of the part can be uniformly picked between 0 and the total length of the adjacent line. The intermediate nodes in the two parts can be matched in the same way as the adjacent lines of the two graphs during the population initialization process. The new solution generated using the mutation must be matched on a 1-to-1 basis; if there are input graph nodes that match with the intermediate nodes on the known graph part, they must be matched as null before the intermediate nodes on the input graph part are reassigned to match the intermediate nodes of the known graph part.

8. The crossover operator

The crossover operator applied to the two candidate solutions is based upon a uniform crossover operation. In this case, elements in the two candidate solutions can not be interchanged freely since the offspring can have duplicate matches which violate the 1-to-1 relationship. Elements in the two solutions can form disjointed sets where all elements in each set need to be interchanged at once,

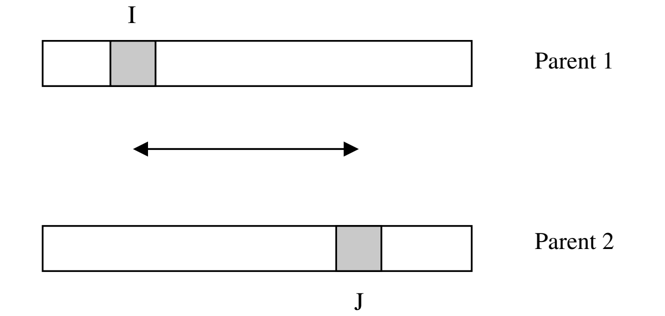

Fig. 9. The *I*th and *J*th elements belonging to the same interchange set.

otherwise duplicate mappings can exist in the offspring. Each of the disjoint sets is referred to as an interchange set.

Consider the two candidate solutions shown in Fig. 9. Suppose that the *I*th element of one candidate solution contains the same matching value as the *J*th element of the other candidate solution and the matching value is not null, then the two elements must belong to the same interchange set. All interchange sets can be built by scanning all of the elements and performing a disjoint set union operation. The crossover can then be carried out by interchanging all of the elements in each set. A predefined interchange probability can be used to control whether all elements in a set are to be interchanged or not. A probability of 0.7 was used in these experiments.

9. Selection schemes and terminating conditions

A selection scheme plays a major role in achieving a successful convergence to the global optima (Back,

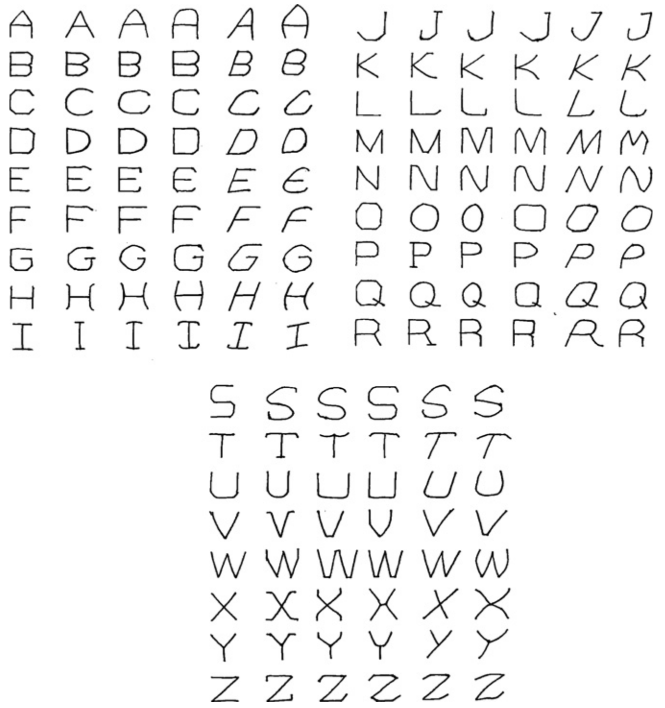

Fig. 10. Synthetic patterns of A–Z where the first character in any row is a known pattern and the following are the five test patterns.

Table 1
Recognition results from graph matching for the first experiment using the rotational fitness function

Graph	Suit 1	Suit 2	Suit 3	Suit 4	Suit 5
A	A	A	A	A	A
B	B	B	B	B	B
C	C	C	C	C	C
D	D	D	D	D	D
E	E	E	E	E	E
F	F	F	F	F	F
G	G	G	G	G	G
H	H	H	H	H	H
I	I	I	I	I	I
J	J	J	J	J	J
K	K	K	K	K	K
L	L	L	L	L	L
M	M	M	M	M	M
N	N	N	N	N	N
O	O	O	O	O	O
P	P	P	P	P	P
Q	Q	Q	Q	Q	Q
R	R	R	R	R	R
S	S	S	S	S	S
T	T	T	T	T	T
U	U	U	U	U	U
V	V	V	U	V	V
W	W	W	W	W	W
X	X	X	X	X	X
Y	Y	Y	Y	Y	Y
Z	Z	Z	Z	Z	Z

Accuracy of graph matching = 99.23%.

Table 2
Recognition results from graph matching for the first experiment not using the rotational fitness function

Graph	Suit 1	Suit 2	Suit 3	Suit 4	Suit 5
A	A	A	A	H	A
B	B	B	B	B	B
C	C	C	C	C	C
D	O	D	D	D	D
E	E	E	E	E	E
F	F	F	F	F	F
G	G	G	G	G	G
H	H	H	H	H	H
I	I	I	I	I	I
J	J	J	J	C	J
K	K	K	K	K	K
L	L	L	L	L	L
M	M	M	M	M	C
N	Z	N	Z	N	N
O	O	O	D	O	O
P	P	P	P	P	P
Q	Q	Q	Q	Q	Q
R	R	R	R	R	R
S	S	S	S	S	S
T	T	T	T	T	T
U	U	U	U	C	C
V	V	V	U	V	J
W	W	M	M	W	M
X	X	X	X	X	X
Y	Y	Y	Y	Y	Y
Z	Z	N	N	Z	Z

Accuracy of graph matching = 87.69%.

1994; Goldberg and Deb, 1991) and a poor selection scheme can cause premature convergences. A tournament selection scheme was used since it gave the best performance in previous work. The selection scheme is as follows:

(1) Choose two individuals at random from a population.
(2) Generate a random number r between 0 and 1.
(3) If $r < K$ (K is a parameter, e.g. 0.7 as used in these experiments), the fitter of the two individuals is

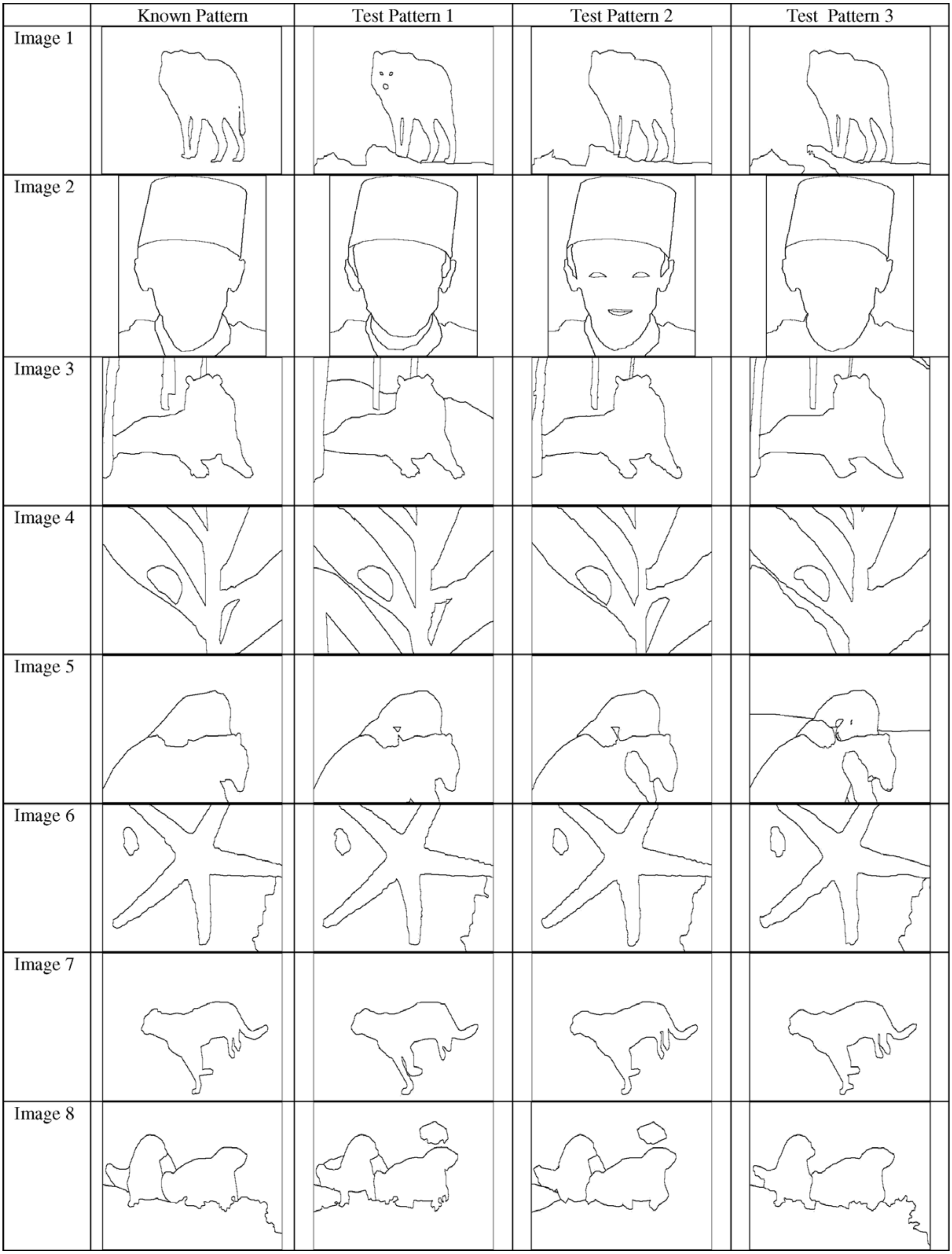

Fig. 11. Eight known patterns and their counterparts.

selected as the parent, otherwise the less fit individual is selected.

(4) Return the two individuals to the original population so they can be selected again.

The iteration process is continued until there is no improvement on the maximum fitness for a number of iterations. According to these experiments, the appropriate value of the number was 100.

10. Experiments with the genetic algorithm

To evaluate the effectiveness of the proposed algorithm, two experiments were conducted. One was performed on synthetic patterns and the other on benchmark images available at the web site of the University of California, Berkeley, USA, for image segmentation (http://www.eecs. berkeley.edu/Research/Projects/CS/vision/grouping/segbench/).

10.1. First experiment – synthetic patterns

Graphs of alphabet character patterns, A–Z, were chosen as the known graphs in these experiments designed to evaluate the effectiveness of the proposed algorithm. Each graph was plotted on to a 18×24 matrix, quantized and then converted into angular representations as mentioned in Section 2. The number of nodes in each graph was no more than 15. The testing sets consisted of five suits of input graphs of characters A–Z. Hence, in total 130 input graphs were tested. Each of the five suits contains twenty six character graphs of A–Z. The shapes and number of nodes of each test graph in one suit are different from those of the known graphs and those from the other suits. They were input as XY coordinates, quantized and then converted into angular representations. Fig. 10 shows the patterns of A–Z used in the experiment.

One round of the population initialization process was performed to create a reasonable population size. The test graph in each suit was tried twice and the best match between the trials was selected. The best results from the experiments are shown in Table 1. The best results were achieved with a crossover rate of 0.6 and a mutation rate of 0.1. Table 2 shows the results of recognizing the graphs when the rotational fitness function was not used. By applying the rotational fitness function, an accuracy of 99.23% in recognizing the graphs was achieved. The rotation fitness function can help distinguish similar graphs such as N and Z, M and W, D and O, etc.

10.2. Second experiment – benchmark images

Eight images from the web site were selected as the test patterns. Each image was segmented manually by several people so that the result images after segmentation differed in some instances. For each image, one of its segmented images was used as the known pattern and the other three as test patterns. Fig. 11 shows the eight known patterns and twenty four test patterns. Each pattern was quantized into a graph of between 50 and 100 nodes. One round of the population initialization process was performed so that a reasonable sized population was created. Each test pattern was twice matched against the eight known patterns and the best match of the two runs was selected. A crossover rate of 0.6 and a mutation rate of 0.1 were found to yield the best results. The algorithm achieved an accuracy of 87.5% in recognizing the patterns but three out of twenty four test patterns were misclassified. The misclassifications occurred on test patterns 2 and 3 of image 5 and test pattern 2 of image 8. These known and their corresponding test patterns may have had too many differences in their line segments for this test. Some extra line segments can cause angle mismatches and different indirect links at some nodes which can decrease the value of the similarity function to some extent.

11. Conclusions

Inexact graph matching using a genetic algorithm was designed to take into account the dissimilarities between two graphs in terms of shape, number of nodes and the rotational degrees of the two graphs. The algorithm can tolerate a number of differences between graphs whilst achieving a very high accuracy on graph matching. The algorithm can be extended to handle multi-dimensional graph matching and is well suited to parallel implementation, a prerequisite for large image recognition. Future investigations are recommended regarding the assignment scheme for angle weights, the handling of extra line segments and parallel implementations of the algorithm.

References

Auwatanamongkol, Surapong, 2000. Pattern Recognition using Genetic Algorithm. In: Congress of Evolutionary Computation. IEEE Press, pp. 822–828.

Back, T., 1994. Selective pressure in evolutionary algorithms: A characterization of selection mechanism, In: Proc. 1st IEEE Conf. on Evolutionary Computation, pp. 57–62.

Bengoetxea E., Larranaga P., Bloch I., Perchant A., 2001. Image recognition with graph matching using estimation of distribution algorithms, In: Proc. Medical Image Understanding and Analysis – MIUA, pp. 89–92.

Cesar R., Bengoetxea E., Bloch I., 2002. Inexact graph matching using stochastic optimization techniques for facial feature recognition, In: Internat. Conf. on Pattern Recognition.

Christmas, W.J., Kittler, J., Petrou, M., 1995. Structural matching in computer vision using probabilistic relaxation. IEEE Trans. Pattern Anal. Machine Intell. 17 (8), 749–764.

Coughlan James, Shen Huiying, 2004. Shape matching with belief propagation: Using dynamic quantization to accommodate occlusion and clutter, In: Proc. 2004 IEEE Conf. on Computer Vision and Pattern Recognition Workshop (CVPRW'04).

Cross, A.D., Hancock, E.R., 1998. Graph matching with a dual-step EM algorithm. IEEE Trans. Pattern Anal. Machine Intell. 20 (11), 1236–1253.

Cross, A.D.J., Wilson, R.C., Hancock, E.R., 1997. Inexact graph matching using genetic search. Pattern Recognit. 30 (6), 953–970.

Cross, A.D.J., Myers, R., Hancock, E.R., 2000. Convergence of a hill-climbing genetic algorithm for graph matching. Pattern Recognit. 33 (11), 1863–1880.

Finch, A.W., Wilson, R.C., Hancock, E.R., 1998. Symbolic graph matching with the EM algorithm. Pattern Recognit. 31 (11), 1777–1790.

Fonseca, C.M., Fleming, P.J., 1995. An overview of evolutionary algorithms in multiobjective optimizations. Evolution. Comput. 3, 1–16.

Goldberg, D.E., Deb, K., 1991. A comparative analysis of selection schemes used in genetic algorithms. In: Foundations of Genetic Algorithms. Morgan Kaufmann.

Khoo, K., Suganthan, P., 2002. Evaluation of genetic operators and solution representations for shape recognition by genetic algorithms. Pattern Recognition Lett. 23 (13), 1589–1597.

Lee, R., Liu, J., 2000. Tropical cyclone identification and tracking system using integrated neural oscillatory elastic graph matching and hybrid RBF network track mining techniques. IEEE Trans. Neural Networks 11 (3), 680–689.

Messmer, B.T., Bunke, H., 1999. A decision tree approach to graph and subgraph isomorphism detection. Pattern Recognit. 32, 1979–1998.

Shams, L., Brady, M., Schaal, S., 2001. Graph matching vs information maximization for object detection. Neural Networks 14 (3), 345–354.

Singh, M., Chaudhury, A., Chatterjeeand, S., 1997. Matching structural shape descriptions using genetic algorithms. Pattern Recognit. 30 (9), 1451–1462.

Skomorowski, M., 1999. Use of random graph parsing for scene labeling by probabilistic relaxation. Pattern Recognition Lett. 60, 649–956.

Williams, M., Wilson, R.C., Hancock, E.R., 1997. Multiple graph matching with Bayesian inference. Pattern Recognition Lett. 18 (11–13), 1275–1281.

Wilson, R.C., Hancock, E.R., 1999. Graph matching with hierarchical discrete relaxation. Pattern Recognition Lett. 20 (10), 1041–1052.

Yang, Lyul, Rae, Hong P., 2002. A surface-based approach to 3D object recognition using mean field annealing neural network. Pattern Recognit. 35 (2), 299–316.