Pattern Recognition, *IEEE Transactions on Systems, Man, and Cybernetics*, **SMC-9**, no. 12, 1979, pp. 757-768.

14. W.H. Tsai and K.S. Fu, Subgraph Error-Correcting Isomorphisms for Syntactic Pattern Recognition, *IEEE Transactions on Systems, Man, and Cybernetics*, **SMC-13**, no. 1, 1983, pp. 48-62.
15. J.R. Ullmann, An Algorithm for Subgraph Isomorphism, *Journal of Association for Computing Machinery*, **Vol. 23**, no. 1, 1976, pp. 31-42.
16. O. Ventä and T. Kohonen, A Content-Addressing Software Method for the Emulation of Neural Networks, *IEEE Internal Conference on Neural Networks*, San Diego, California July 24-27 1988, vol. I, SOS Printing, San Diego, 1988, pp. 191-198.
17. T. Westman, *Semantic Net Based Approach for Defect Analysis in Visual Inspection* (in Finnish), Lic. Tech. Thesis, University of Oulu, Department of Electrical Engineering, Finland, 1987.
18. A.K.C. Wong, Structural Pattern Recognition: A Random Graph Approch. *Pattern Recognition Theory and Applications* (edited by P.A. Devijver and J. Kittler), NATO ASI Series, vol. F30, Springer-Verlag, Berlin Heidelberg, 1987, pp. 323-345.
19. A.K.C. Wong and M.L. You, Entropy and Distance of Random Graphs with Application to Structural Pattern Recogniton, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **PAMI-7**, no. 5, 1985, pp. 599-609.
20. L. Xu and E. Oja, Improved Simulated Annealing, Boltzmann Machine, and Attributed Graph Matching, in L. B. Almeida and C. J. Wellekens (Eds.), *Neural Networks*, Lecture Notes in Computer Science 412, Springer, Berlin, 1990.

not able to look for graph isomorphisms. The Ullmann algorithm is quick and simple, but only suitable for nonattributed graphs.

In the future, graph matching algorithms could be developed using parallel computing. Applications of neural networks and VLSI implementation techniques can also open new chances.

REFERENCES

1. A.V. Aho, R. Sethi and J.D. Ullmann, *Compilers: Principles, Techiques and Tools*, Addison-Wesley, Reading, Massachusetts, 1986.
2. E. Backer and J.J. Gerbrands, Inexact Graph Matching used in Machine Vision, in P. A. Devijver and J. Kittler (Eds.), *Pattern Recognition Theory and Applications*, Springer, Berlin, 1987.
3. P.A. Devijver and J. Kittler, *Pattern Recognition: A Statistical Approach*, Prentice-Hall, Englewood Cliffs, New Jersey, 1982.
4. M.A. Eshera and K.S. Fu, A Graph Distance Measure for Image Analysis, *IEEE Transactions on Systems, Man, and Cybernetics*, **SMC-14**, no. 3, 1984, pp. 398-408.
5. T. Kohonen and E. Reuhkala, A Very Fast Associative Method for the Recognition and Correction of Misspelt Words, Based on Redundant Hash Addressing, *Proceedings the 4th International Joint Conference on Pattern Recognition*, Kyoto Japan, November 7-10 1978, pp. 807-809.
6. H. Kälviäinen, Master's thesis (in Finnish), Lappeenranta University of Technology, Department of Information Technology, Finland, 1989.
7. K. Mäkisara and E. Oja, Recognition of Images Using Relational Structures and Content-Addressable Memory, *Proceedings of the First Scandinavian Conference on Image Analysis*, Linköping 14-16, January 1980, Studentlitteratur, Lund, 1980, pp. 112-117.
8. T. Pavlidis, *Structural Pattern Recognition*, Springer, New York, 1977.
9. E. Reuhkala, *Recognition of Strings of Discrete Symbols with Special Application to Isolated Word Recognition*, Dr. Techn. Thesis, Helsinki University of Technology, Department of Technical Physics , Finland, 1983.
10. L.G. Shapiro and R.M. Haralick, Structural Descriptions and Inexact Matching, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **PAMI-4**, no. 6, 1982, pp. 595-602.
11. A.C. Shaw, Parsing of Graph-Representable Pictures, *Journal of Association for Computing Machinery*, **Vol. 17**, 1970, pp. 453-481.
12. O. Silven, Dr. Tech. Thesis, University of Oulu, Department of Electrical Engineering, Finland, 1989.
13. W.H. Tsai and K.S. Fu, Error-Correcting Isomorphism of Attributed Relational Graphs for

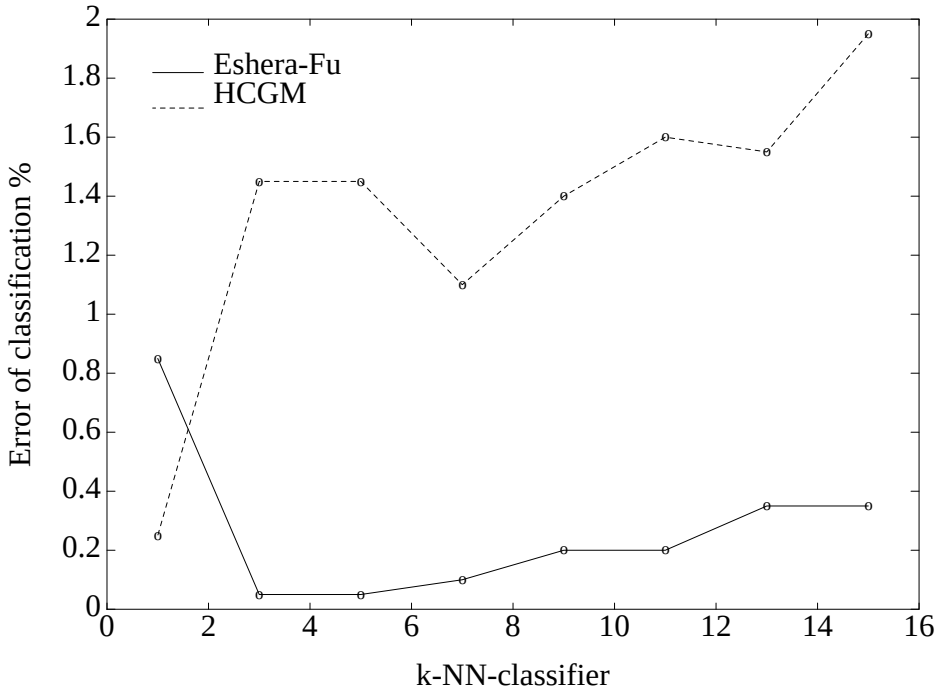


FIG. 7. An error of classification of the Eshera-Fu algorithm and the HCGM algorithm, when a training set has 20+20 graphs and a test set 1000+1000 graphs.

suitable method is highly problem-dependent, and thus it is hard to find an application which would test these algorithms impartially.

5. CONCLUSIONS

Five graph matching algorithms and their variations have been studied. Four of them were well-known graph matching methods in structural pattern recognition: The Ullmann, the Eshera-Fu, the Tsai-Fu, and the Wong algorithm. In addition to this the Hash Coded Graph Matching (HCGM) algorithm was developed and applied. Speed of algorithms, capacity for classification and suitability for different graph types were studied. Algorithms were tested in two classification problems: recognition of turbine blades and characters.

When comparing graph matching algorithms it turns out that an exact, general comparison is difficult, because the problem is highly application dependent. Every method had its advantages and disadvantages. Thus the Eshera-Fu algorithm is quite slow but adaptive for many cases and accurate, while the Tsai-Fu and Wong algorithms are fast and fairly accurate, but need prior information. The graph matching algorithm with hash coding is easily applied and versatile but its error of classification is more sensitive compared to the other methods and it is

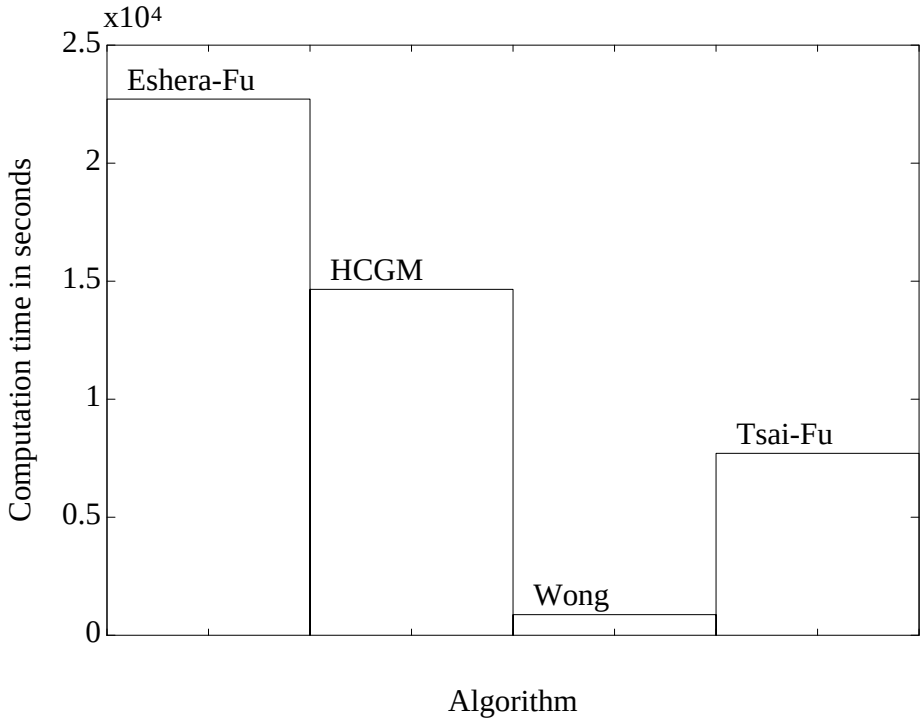


FIG. 6. Computation times of 2000 test graphs.

4.3.2. *Error of Classification*

Due to the nature of the Wong and Tsai-Fu methods, the recognition of letters H and F will always succeed, and the error of classification of these methods could not be estimated. On the other hand, a comparison of the Eshera-Fu algorithm and the HCGM algorithm is valid and is shown in Fig. 7. The Eshera-Fu method is more exact.

4.4. *Conclusions of Test Results*

Based on these tests, the Eshera-Fu algorithm turned out to be the slowest method. Because the complexity of the algorithm is exponential in the number of graph nodes, this method seems to be applicable to real world problems only using parallel programming. The HCGM algorithm was faster than the Eshera-Fu method, but it was not so exact.

The Tsai-Fu and especially the Wong algorithm were fast methods and their computation time increased linearly. The Wong algorithm was the most precise method in classification of turbine blades. However, because these algorithm need the estimation of probability distributions and the Wong algorithm has some other demands, too, they are best suitable for some special applications.

An exact general comparison of graph matching algorithms is difficult. Generally the most

describing the good and the faulty turbine blades, respectively. Test graphs were classified in terms of the entropies between them and the supergraphs. In addition classification using only one supergraph having some decision surface was tried, but it did not lower the error of classification for this method. The figures of error are high in all cases, which is due to the difficulty of the problem and also perhaps a suboptimal choice for the features describing the graph attributes.

4.3. *Character Recognition*

Wong suggested how to make graphs from handwritten letters [19]. He defined attributes of nodes and links of characters and calculated probability distributions of attributes of different characters from samples. In our simulation using these probability distributions, a proper amount of graphs of letter F and H were generated for classification. Every graph had three nodes. Then two training sets were formed describing the two letters, and test graphs were classified using the different graph matching algorithms either to letter class F or to letter class H.

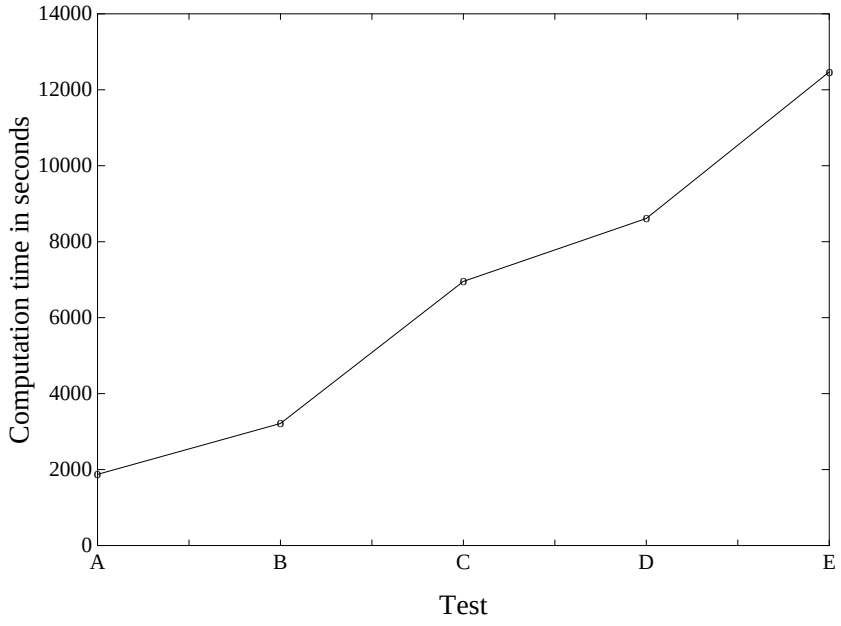
4.3.1. *Speed of Classification*

The speed of classification is shown in Fig. 5. Two training set were used, having 10 and 20 graphs, respectively, both having an equal number of letters H and F. There were equal numbers of letters H and F in the test set of 40, 200, and 2000 graphs.

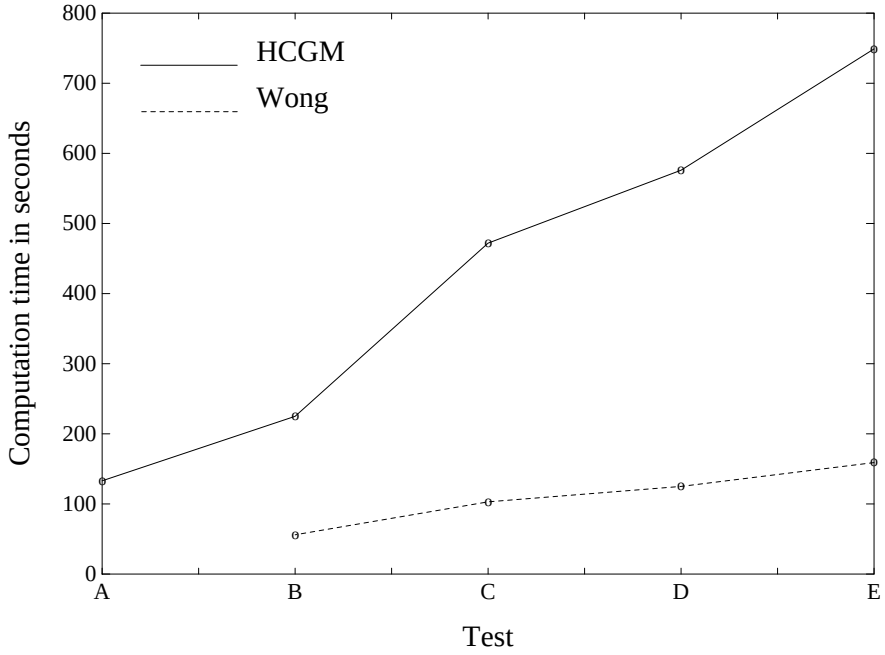
		20+40	40+40	20+200	40+200	40+2000
1.	The Wong algorithm	-	26	-	83	873
2.	The Tsai-Fu algorithm	70	139	379	700	7707
3.	The HCGM algorithm	74	111	423	642	14654
4.	The Eshera-Fu algorithm	178	378	1082	2307	22715

FIG. 5. Classification of letter H and F. Computing times are in seconds.

The results on classification of 2000 test graphs can be seen in Fig. 6. The Eshera-Fu algorithm seems to be the slowest and the Wong algorithm the fastest method. The computation time of the HCGM algorithm grows faster than that of the other methods as function of the number of test graphs, because of the garbage collection of the automatic hash function in the Common Lisp version that was used.

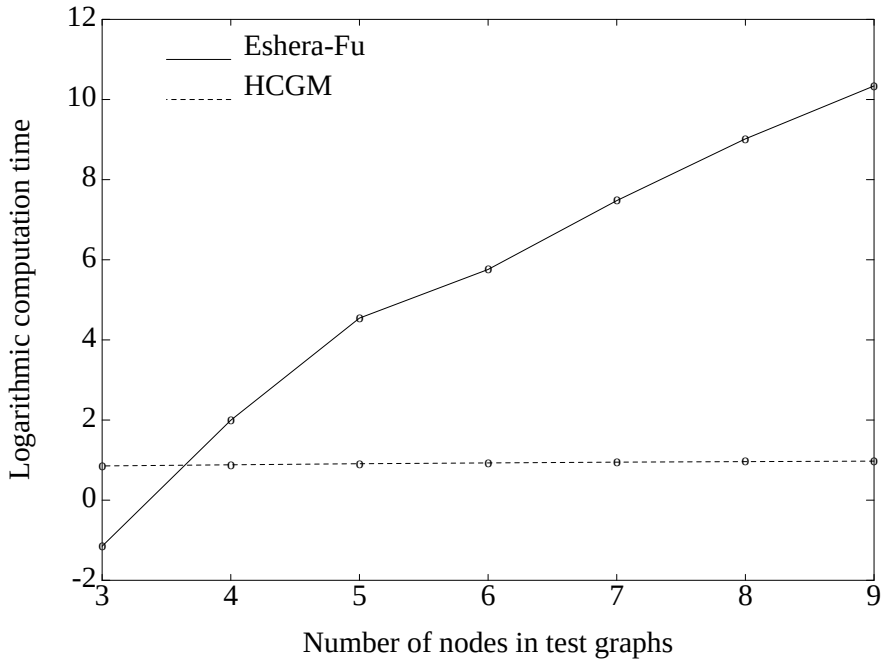


(a)

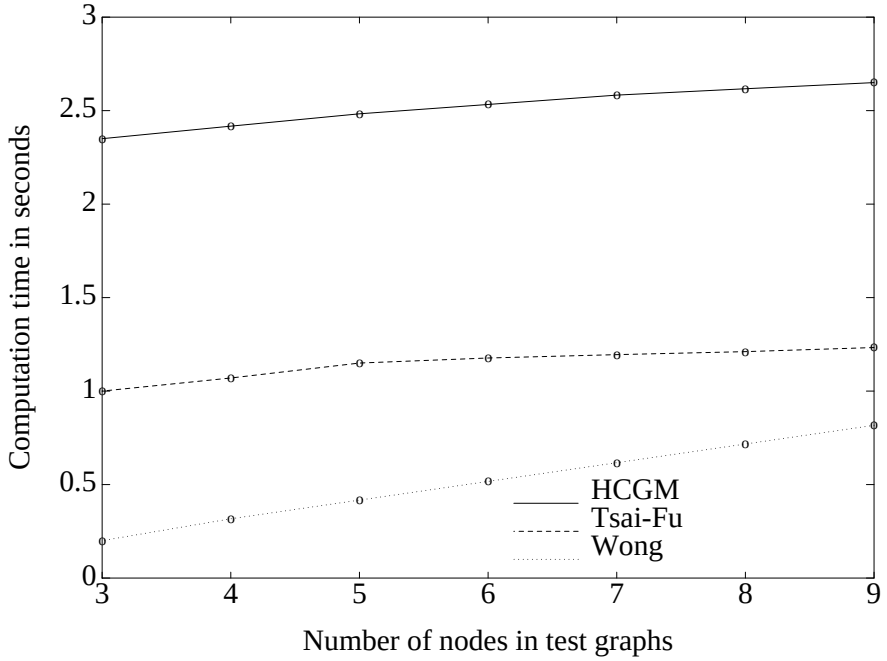


(b)

FIG. 4. Classification of turbine blades: (a) Computation time of the Eshera-Fu algorithm; (b) Computation times of the HCGM and the Wong algorithms.



(a)



(b)

FIG. 3. Computation times of algorithms: (a) Computation times of the Eshera-Fu and the HCGM algorithm (logarithmic scale); (b) Computation times of the HCGM, the Wong and Tsai-Fu algorithm (linear scale).

The HCGM algorithm is much faster than the Eshera-Fu algorithm, because the computational complexity of the Eshera-Fu method is exponential, while the HCGM method has linear complexity. The same is true of the Tsai-Fu and Wong algorithms. The exact comparison of these methods having linear computation time is difficult due to the different nature of algorithms, but in practice the Wong and Tsai-Fu algorithms are faster than the HCGM method (Fig. 3b).

4.2. *Classification of Turbine Blades*

The algorithms were tested for classifying graphs obtained from gray-level images of turbine blades [17]. Every turbine graph had seven nodes. The turbine blades were either good or faulty, there being a crack in the blade in the latter case. This problem is important in automated visual inspection of the blades, but due to the poor quality and variability of the images the task is not easy.

Graphs containing good and faulty turbine blades were formed into two separate training sets. Test graphs were classified into the two sets using the different methods. After this it is possible to estimate the error of classification, because it was known in advance to which class each test graph really belonged.

4.2.1. *Speed of Classification*

To examine the speed of classification, the turbine blades were classified in five different tests. As shown by Fig. 4, the Eshera-Fu algorithm needs considerably more computation time than the HCGM and the Wong algorithms. The smallest test with the Eshera-Fu algorithm took more time than the largest test with the other two methods together.

4.2.2. *Error of Classification*

A test graph was classified to one of the two classes using the k-NN classifier. For every test graph, the distances to graphs of both training sets were computed and test graphs were classified based on the nearest neighbour decision rule [3].

The error of classification and the corresponding computing time in our largest test were:

1. the Wong algorithm 23.2 % 159 s,
2. the Eshera-Fu algorithm 31.7 % 12470 s,
3. the HCGM algorithm 41.5 % 749 s.

The Wong algorithm performed uniformly the best: it had the smallest error of classification and it was also fastest. In applying the Wong method, two supergraphs were synthesized,

	E	H	W	U	T
- finding one possible graph isomorphism	x		x	x	x
- looking all possible graph isomorphisms	x			x	
- finding one possible subgraph isomorphism	x		x	x	
- looking for all possible subgraph isomorphisms	x			x	
- looking for matches which are not isomorphic	x	x			
- suitable for directed graphs	x	x	x	x	x
- suitable for undirected graphs		x		x	
- suitable for attributed relational graphs	x	x	x		x
- suitable for nonattributed relational graphs	x	x		x	
- one node can have several attributes	x	x			
- one link can have several attributes	x	x			
- attributes can be symbolic	x	x	x		x
- a distance measure between two graphs will be obtained as a result	x	x	x		x
- need of preinformation (probability distributions)			x		x
- available for classification tasks	x	x	x	x	x

E = the Eshera-Fu algorithm, H = the HCGM algorithm, W = the Wong algorithm, U = the Ullmann algorithm, and T = the Tsai-Fu algorithm.

FIG. 2. Properties of graph matching algorithms.

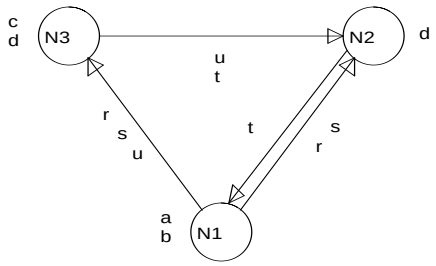
4. TEST RESULTS

In this section, graph matching algorithms are compared in terms of speed and matching performance. First the practical computation time is studied as a function of the number of graph nodes and then the algorithms are applied on two classification problems: the detection of faulty turbine blades from gray-level images and the classification of simple optical characters. All the algorithms have been implemented in Common Lisp on a Sun 3/60 workstation.

4.1. *Speed of the Algorithms*

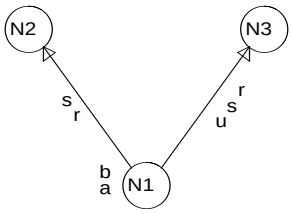
When a graph matching algorithm is used for real data, the computational complexity is an important factor. It is essential to know how the computation time will grow as a function of the number of graph nodes. Although the results here are experimental, they show the qualitative behavior of this function for the different algorithms.

Fig. 3a shows the computation time of the Eshera-Fu algorithm and the HCGM algorithm. Test graphs have 3 to 9 nodes. The output degree of nodes in each test graph is three, i.e., every node has three output links.

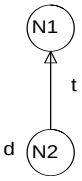


[((n_1 (b) (a)) ((s) (r) (n_2)))
((n_1 (b) (a)) ((r) (s) (u) (n_3)))
((n_2 (d)) ((t) (n_1)))
((n_3 (c) (d)) ((u) (t) (n_2)))]

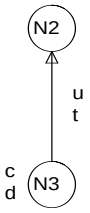
(a)



[(a b) ((r s) (r s u))]

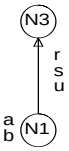


[(d) ((t))]

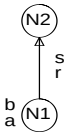


[(c d) ((t u))]

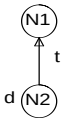
(b)



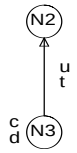
[(a b) (r s u)]



[(a b) (r s)]

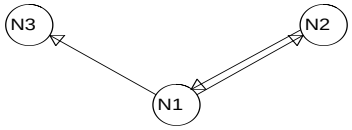


[(d) (t)]

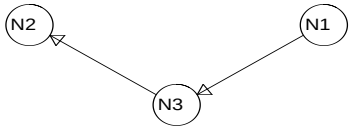


[(c d) (t u)]

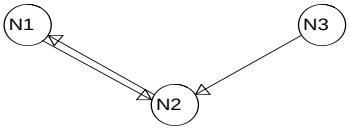
(c)



(1 0 0)



(1 0)



(1 1 0)

(d)

FIG. 1. Decomposition of an attributed relational graph into different basic elements: (a) An attributed relational graph; (b) Decomposition into BARGs; (c) Decomposition into links; (d) Decomposition into non-attributed structures.

The method can be expanded to the case of several graphs, with which one test graph is to be compared. In this case similarity measures between the test graph and other graphs are obtained very quickly.

When the HCGM algorithm is implemented, we have to decide which kind of basic elements will be used. In our simulations, three types were used: the BARGs of a graph, the links of a graph, and non-attributed structures of a graph. An example how to make basic elements is shown in Fig. 1. In updating the similarity measure R in the above algorithm, different kinds of basic structures can be weighted differently.

3. THE COMPARISON OF GENERAL PROPERTIES OF GRAPH MATCHING ALGORITHMS

Some general properties of the studied graph matching algorithms have been given in Fig. 2. All methods can be used for classification, and all except the Ullmann algorithm will give a graph distance measure which can be a similarity or distance measure, depending on an algorithm. All algorithms except the Ullmann algorithm can be applied to attributed relational graphs. The Ullmann method can be applied only to nonattributed graphs but it is a good preclassifier for other graph matching algorithms, giving all isomorphisms between two graphs. With the Ullmann algorithm it is possible to use undirected graphs, too.

The Eshera-Fu algorithm is the most versatile method. It will find all isomorphisms and will define the most precise distance measure without any preinformation. It allows links and nodes with many attributes unlike the Wong and Tsai-Fu algorithm and is suitable for nonattributed relational graphs, too. The big disadvantage of this method is exponentially growing computational complexity.

Contrary to the other algorithms, the Hash Coded Graph Matching algorithm does not look for or require genuine graph isomorphism. It will count quickly and simply a similarity measure of two graphs. The real advantage of the method is that it is applicable to many kinds of graphs. It can compare both nonattributed and attributed relational graphs and both undirected and directed graphs.

The Wong and Tsai-Fu algorithm demand preinformation. Before using the Tsai-Fu algorithm, pattern deformation probabilities must be known, and before using the Wong algorithm, a random graph including probabilities of attributes has to be defined. In addition to this, the methods are not directly applicable for nodes and links which have many attributes. However, these two algorithms can be sometimes very useful because they are fairly fast.

image analysis [13]. They extented their pattern deformation model so that numerical attributes and probability distributions could be introduced into primitives and relations in non-hierarchical relational graphs. An ordered-search algorithm was proposed for determining error-correcting isomorphisms. Finally using probability densities of the attributes of nodes and links in two ARGs, a pattern deformation distance and the best match of the two graphs will be obtained. Tsai and Fu also expanded their algorithm to apply for subgraph error-correcting isomorphisms [14].

Wong defined the entropy and distance of random graphs [19], [18], having probability distributions of attributes of nodes and links. The synthesis of random graphs was defined, in which random elements of two graphs are synthesized and a new random graph is produced. Using such synthesis of random graphs, Wong proposed a distance measure between random graphs based on the minimum change of entropy before and after their merging.

Here we also consider an algorithm called Hash Coded Graph Matching (HCGM) [6], which applies the well-known searching technique known as hashing [1]. This is a very rapid content-addressable way to search for information (a hash value) directly from a hash table with the help of a hash key. Aside of conventional database applications, hash coding has been earlier applied to some problems of pattern recognition and speech recognition and correction [5], [7], [9], [16].

While in the Eshera-Fu algorithm graphs are decomposed into specific basic elements and attributes of these elements are compared, in the HCGM algorithm elements themselves are compared directly. Besides this the HCGM does not look for or require subgraph isomorphisms. It simply compares every structure of one graph to all structures of another graph. Now the basic elements are hash keys and the labels or names of entire graphs are hash values. Because structures wider than attributes are compared in the HCGM algorithm and no state-space to show subgraph isomorphisms is used, it is faster, but not so exact a method as the Eshera-Fu algorithm.

The comparison of two graphs U and V using the HCGM algorithm is the following:

Step 1. Decompose V into basic elements $V_1', V_2', \cdots, V_m'$ and make a hash table in which all elements are represented so that the hash keys are the basic elements V_i' and the hash value is the name of graph V.

Step 2. Decompose U into basic elements $U_1', U_2', \cdots, U_m'$.

Step 3. Set the number of suitable structures, R to zero.

Step 4. For each U_i', search if a corresponding structure is found in the hash table. If a corresponding structure is in the hash table, increment R by one.

Step 5. As a result, the number of suitable structures R defines an approximate similarity measure between graph U and graph V.

relevant properties and the types of problems they are best suited for. The starting-point was an attempt to build modules for a hybrid Computer Vision system in which the high-level matching problem cannot be isolated from low-level statistical processing stages and thus error-tolerant matching techniques are needed. Also Neural Network approaches and Simulated Annealing optimization to the same problem were studied, with some preliminary results given elsewhere [20].

Section 2 introduces the methods used, Section 3 compares them with regard to the application characteristics, Section 4 gives results on speed and matching accuracy for both artificial and real-world problems, and Section 5 presents some conclusions.

2. ATTRIBUTED GRAPH MATCHING ALGORITHMS

The main aim was to study graph matching algorithms which could match attributed relational graphs (ARG) [13], [4] because attributed relational graphs have shown superior adequacy when used for representation and analysis in computer vision. Five graph matching algorithms were selected: The Ullmann [15], the Eshera-Fu [4], the Tsai-Fu [13], and the Wong algorithm [19], and another graph matching algorithm based on hash coding [6]. The Ullmann method can be applied only to non-attributed relational graphs, but it was included as a basic reference method. The graph matching algorithm based on hash coding was developed by the authors and will be explained later in this section.

Ullmann introduced a basic algorithm [15] that could find subgraph isomorphisms. In the method, adjacency matrices of the two graphs are formed and subgraph isomorphisms are found with an algorithm employing a refinement procedure which eliminates successor nodes in the tree search instead of a brute-force tree-search enumeration procedure. As a result, all possible subgraph isomorphisms are obtained. The graphs must be non-attributed.

Eshera and Fu developed a graph distance measure for image analysis applications [4]. In their method, a distance measure between two ARGs is defined as the minimum total cost of a sequence of weighted error transformations that will produce the other ARG when performed on one of the ARGs (or a subgraph). The two ARGs are first decomposed into two sets of Basic Attributed Relational Graphs (BARG) that are graphs having the simple form of a one-level tree. Then a state space representation of the problem is generated, in which each state corresponds to the reconstruction of a pair of subgraphs from the two studied graphs using BARGs. The third and last step of the approach is to find the shortest path over the direct acyclic branch-weighted lattice from the initial state to one of the final states. This gives the distance of the two graphs.

Tsai and Fu represented error-correcting isomorphisms of attributed relational graphs for

Comparisons of Attributed Graph Matching Algorithms for Computer Vision

Heikki Kälviäinen and Erkki Oja

Lappeenranta University Of Technology
Department of Information Technology
P.O. Box 20
53851 Lappeenranta
Finland

ABSTRACT. Attributed relational graphs have shown good adequacy for representation and analysis in computer vision. In some applications, pattern recognition requires that an attributed graph representation of an object is matched with several similar graphs stored in a database. However, there is no single algorithm that could be recommended for this problem. In this study, some well-known attributed graph matching algorithms were compared to find out the relevant properties and the types of problems they are best suited for. In addition, an algorithm based on hash coding, developed by the authors, was considered. Some test results are given on both the speed and classification accuracy of the algorithms for artificial and real images.

1. INTRODUCTION

Attributed graphs have turned out to be very useful data structures for image representation and understanding in Computer Vision systems [10]. Examples of graph-based representations of patterns include the picture languages of Shaw [11], the Region Adjancency Graphs of Pavlidis [8] for the scene analysis and understanding problem, and the representation of 3-D objects in the stereo vision problem [2]. Fu and his co-workers emhasized the importance of Attributed Relational Graphs (ARG) for the structural matching problem [13], [4], and they can be successfully used e.g. in industrial vision based quality inspection [12].

However, there does not seem to be any single standard algorithm that could be recommended for the attributed graph matching problem. Often, the problem differs from the graph or subgraph isomorphism problem since in cases with noise and distortions, it cannot even be assumed that the two graphs to be compared have structural (non-attributed) isomorphisms. Thus the comparison should rather be based on graph distance measures. In some cases statistics can be collected on the variations occurring in a graph representation of, say, a hand-written character to be recognized, in which case random graphs may prove to be necessary.

Some well-known graph matching methods are compared in this study to find out the