International Journal of Pattern Recognition
and Artificial Intelligence
Vol. 18, No. 3 (2004) 265–298

World Scientific
www.worldscientific.com

THIRTY YEARS OF GRAPH MATCHING IN PATTERN RECOGNITION

D. CONTE*,‡, P. FOGGIA†,§, C. SANSONE†,¶ and M. VENTO*,‖

*Dipartimento di Ingegneria dell'Informazione e di Ingegneria Elettrica,
Università di Salerno – Via P.te Don Melillo,1 I-84084, Fisciano (SA), Italy
†Dipartimento di Informatica e Sistemistica,
Università di Napoli "Federico II" – Via Claudio, 21 I-80125 Napoli, Italy
‡dconte@unisa.it
§foggiapa@unina.it
¶carlosan@unina.it
‖mvento@unisa.it

A recent paper posed the question: "Graph Matching: What are we really talking about?". Far from providing a definite answer to that question, in this paper we will try to characterize the role that graphs play within the Pattern Recognition field. To this aim two taxonomies are presented and discussed. The first includes almost all the graph matching algorithms proposed from the late seventies, and describes the different classes of algorithms. The second taxonomy considers the types of common applications of graph-based techniques in the Pattern Recognition and Machine Vision field.

Keywords: Graph matching algorithms; pattern recognition.

1. Introduction

Starting from the late seventies, graph-based techniques have been proposed as a powerful tool for pattern representation and classification in structural Pattern Recognition (PR). After the initial enthusiasm induced by the apparent "smartness" of this data structure, graphs have been practically left unused for a long period of time. Recently, the use of graphs in PR is obtaining a growing attention from the scientific community (see Table 1). This is perhaps due to the fact that the computational cost of the graph-based algorithms, although still high in most cases, is now becoming compatible with the computational power of new computer generations.

In this scenario it is usual to observe that relatively recent applications make use of graph algorithms that date back to the early eighties. Moreover, the analysis of the state of the art of graph-based techniques is made difficult by the considerable extension of the bibliography sources that must be taken into account, being spread over the last three decades.

Table 1. The number of papers dealing with different applicative areas by means of graph matching techniques as a function of the time periods. In parentheses is the number of papers that are centered around the presentation of an application.

Period	2D & 3D Image Analysis	Document Processing	Biometric Identification	Image Databases	Video Analysis	Biomedical and Biological
up to 1990	3 (0)	1 (1)				
1991–1994	3 (0)	2 (2)	1 (1)			1 (1)
1995–1998	8 (3)	9 (5)	6 (6)	5 (5)		1 (0)
1999–2002	19 (6)	8 (4)	8 (8)	8 (7)	6 (6)	2 (2)

Here we attempt to catalogue the literature on basic techniques for graph matching and related problems. Further, we report on the Pattern Recognition and Machine Vision applications where graphs are used sometimes in the recognition process.

To this end, our review is centered around two different taxonomies: a taxonomy of matching algorithms, that is presented discussing the different problems and solution strategies involved. The second is the taxonomy of the most common applications of graph-based techniques in the PR field.

2. Algorithms Taxonomy

In many applications a crucial operation is the comparison between two objects or between an object and a model to which the object could be related. When structured information is represented by graphs this comparison is performed using some form of graph matching. Graph matching is the process of finding a correspondence between the nodes and the edges of two graphs that satisfies some (more or less stringent) constraints ensuring that similar substructures in one graph are mapped to similar substructures in the other.

In this section we will present a review of the algorithms that have been proposed and used in the PR field for the graph matching problem (in the several forms in which this problem can be posed), and associated problems, such as graph prototyping and graph clustering. Figure 1 presents a synoptic picture of the works that will be reviewed, organized according to the kind of problem tackled by each algorithm and the solution technique.

We have divided the matching methods into two broad categories: the first contains exact matching methods that require a strict correspondence among the two objects being matched or at least among their subparts. The second category defines inexact matching methods, where a matching can occur even if the two graphs being compared are structurally different to some extent. Sections 2.1 and 2.2 will be dedicated to exact and inexact matching, respectively; in Sec. 2.3, we will discuss algorithms dealing with other (i.e. non graph-matching) problems involving graphs in PR.

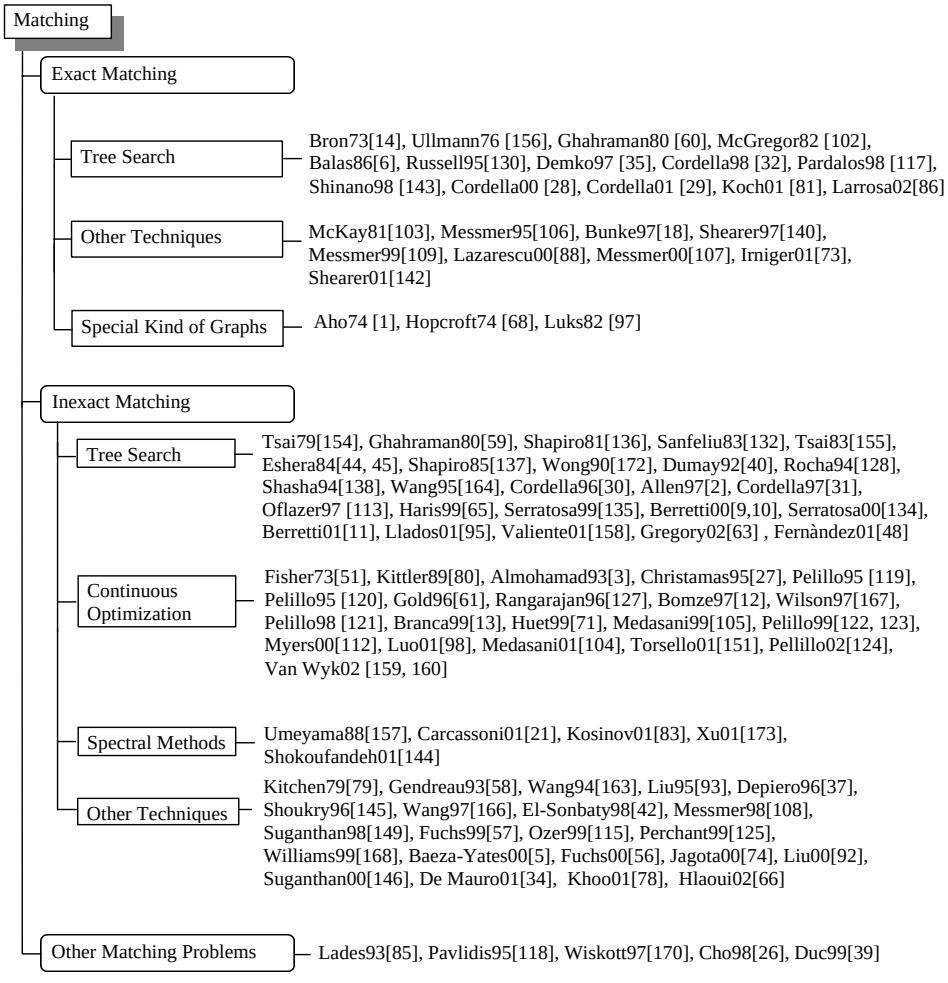

Fig. 1. The taxonomy of the reviewed algorithms. For each algorithm the taxonomy reports the first author, the year and the corresponding bibliographic reference.

2.1. *Exact matching algorithms*

Exact graph matching is characterized by the fact that the mapping between the nodes of the two graphs must be *edge-preserving* in the sense that if two nodes in the first graph are linked by an edge, they are mapped to two nodes in the second graph that are linked by an edge as well. In the most stringent form of exact matching, *graph isomorphism*, this condition must hold in both directions, and the mapping must be *bijective*. That is, a one-to-one correspondence must be found between each node of the first graph and each node of the second graph. A weaker form of matching is *subgraph isomorphism*, that requires that an isomorphism holds between one of the two graphs and a node-induced subgraph of the other. Actually, some authors (e.g. Ref. 172) use the term *subgraph isomorphism* in a slightly weaker

sense, dropping also the condition that the mapping should be edge-preserving in both directions. The resulting matching type, that other authors (e.g. Ref. 60) call *monomorphism*, requires that each node of the first graph is mapped to a distinct node of the second one, and each edge of the first graph has a corresponding edge in the second one; the second graph, however, may have both extra nodes and extra edges. A still weaker form of matching is *homomorphism*, that drops the condition that nodes in the first graph are to be mapped to distinct nodes of the other; hence, the correspondence can be many-to-one. Finally, another interesting matching type maps a subgraph of the first graph to an isomorphic subgraph of the second one; since such a mapping is not uniquely defined, usually the goal of the algorithm is to find the largest subgraph for which such a mapping exists. Hence, this problem is known in literature as finding the *maximum common subgraph* (*MCS*) of the two graphs. Actually, there are two possible definitions of the problem, depending on whether node-induced subgraphs or plain subgraphs are used. In the first case, the maximality of the common subgraph is referred to the number of nodes, while in the second it is the number of edges that is maximized. It is widely known that the problem of finding the MCS of two graphs can be reduced to the problem of finding the maximum *clique* (i.e. a fully connected subgraph) in a suitably defined *association graph*.[4] Hence we will also cite some maximum clique detection methods that have been used in PR.

The matching problems mentioned above are all *NP-complete* except for graph isomorphism, for which it has not yet been demonstrated if it belongs to NP or not. Polynomial isomorphism algorithms have been developed for special kinds of graphs (e.g. for trees by Aho *et al.*[1] in 1974, for planar graphs by Hopcroft and Wong[68] in 1974, for bounded valence graphs by Luks[97] in 1982) but no polynomial algorithms are known for the general case. Hence, exact graph matching has exponential time complexity in the worst case. However, in many PR applications the actual computation time can be still acceptable, because of two factors: first, the kinds of graphs encountered in practice are usually different from the worst cases for the algorithms. Second, node and edge attributes can be used very often to reduce dramatically the search time.

Of the above-mentioned matching problems, exact isomorphism is very seldom used in PR, since more often than not the graphs being compared are obtained as the result of a (sometimes very complex) description process that is inevitably subject to some form of noise, and so missing or extra nodes and edges can appear, hampering the isomorphism. Subgraph isomorphism and monomorphism, instead, albeit more demanding from a computational viewpoint, can be effectively used in many contexts, and several algorithms for these problems have been proposed. Finally, the MCS problem is receiving much attention, albeit exact methods known up to now are only able to deal with graphs with a small number of nodes.

2.1.1. *Techniques based on tree search*

Most of the algorithms for exact graph matching are based on some form of tree search with backtracking. The basic idea is that a partial match (initially empty) is iteratively expanded by adding to it new pairs of matched nodes; the pair is chosen using some necessary conditions that ensure its compatibility with the constraints imposed by the matching type with respect to the nodes mapped so far, and usually using also some heuristic condition to prune as early as possible unfruitful search paths. Eventually, either the algorithm finds a complete matching, or it reaches a point where the current partial mapping cannot be further expanded because of the matching constraints. In this latter case the algorithm backtracks, i.e. undoes the last additions until it finds a partial matching for which an alternative extension is possible. If all the possible mappings that satisfy the constraints have already been tried, the algorithm halts. Several different implementation strategies of this kind of algorithm have been employed, differing in the order the partial matches are visited. Probably the simplest is *depth-first search* that requires less memory than others and lends itself very well to a recursive formulation; it is also known as *branch and bound*. A nice property of such algorithms is that they can be very easily adapted to take into account the attributes of nodes and edges in constraining the desired matching, with no limitations on the kind of attributes that can be used. This is very important for PR applications where often attributes play a key role in reducing the computational time of the matching.

The first important algorithm of this family is due to Ullmann[156] in 1976. Ullmann's algorithm is widely known and, despite its age, it is still widely used and is probably the most popular graph matching algorithm. The problems addressed by the algorithm are graph isomorphism, subgraph isomorphism and monomorphism, but the author also suggests a way to employ it for maximum clique detection and hence for the MCS problem (although the nature of the algorithm makes it less suited to this problem). To prune unfruitful matches, Ullmann proposes a so-called *refinement procedure*, that works on a matrix of possible future matched node pairs to remove, on the basis of a suitably defined necessary condition, the ones that are not consistent with the current partial matching.

Another interesting monomorphism algorithm based on backtracking has been proposed by Ghahraman *et al.*[60] in 1980. In this paper the authors, in order to prune the search space, use a technique that is somewhat resembling the association graph cited before. Namely, they work on the so-called *netgraph* obtained from the Cartesian product of the nodes of two graphs being matched. Monomorphisms between these two graphs correspond to particular subgraphs of the netgraph. The authors devise two necessary conditions that must be satisfied by the netgraph if the current partial mapping is going to lead to a complete monomorphism: a *strong necessary condition* and a *weak necessary condition*, less stringent than the former but also fairly easier to verify. From these conditions stem two versions of the algorithm in which the conditions are used to quickly detect unfruitful partial

solutions. A major drawback of these algorithms is that, at least in the implementation suggested by the authors, the netgraph is represented using a matrix of size $N^2 \times N^2$, where N is the number of nodes of the largest graph. Consequently, only small graphs can be reasonably dealt with.

A more recent algorithm for both isomorphism and subgraph isomorphism is the VF algorithm, due to Cordella *et al.*[28,32] The authors define a heuristic that is based on the analysis of the sets of nodes adjacent to the ones already considered in the partial mapping. This heuristic is fast to compute leading in many cases to a significant improvement over Ullmann's and other algorithms, as shown in Refs. 33 and 38. In a 2001 paper,[29] the authors propose a modification of the algorithm (called VF2) that reduces the memory requirement from $O(N^2)$ (that compares favorably with other algorithms) to O(N) with respect to the number of nodes in the graphs, thus making the algorithm particularly interesting for working with large graphs.

One of the most recent tree search methods for isomorphism has been proposed by Larrosa and Valiente[86] in 2002; the authors reformulate graph isomorphism as a *Constraint Satisfaction Problem* (CSP), a problem that has been studied very deeply in the framework of discrete optimization and operational research. Thus the authors apply to graph matching some heuristics derived from the CSP literature.

The tree search approach has also been used for the clique detection and the MCS problem. In particular, the most famous clique detection algorithm, developed by Bron and Kerbosh[14] in 1973, falls into this category. This algorithm, also cited as ACM Algorithm #457, has been among the first algorithms using backtracking for a problem related to graph matching. It is based on the use of a simple but effective heuristic for pruning the search tree that requires a relatively small data structure to be computed (an array of counters). Its simplicity, together with an acceptable performance in most cases (compared to more recent algorithms), make it still widely used.

A more recent, effective algorithm for clique detection is due to Balas and Yu[6]; like Bron–Kerbosh, this algorithm also searches for maximum cliques using tree search. The difference lies in the heuristic, that, for Balas–Yu, is based on graph coloring techniques.

Since maximum clique detection is inherently an expensive problem, some recent work (Refs. 117 and 143, both in 1998) has also investigated the possibility of obtaining a significant speed-up by a parallel algorithm. In particular, Shinano *et al.* applied to clique detection a publicly available tool (PUBB)[a] for developing parallel branch and bound algorithms. Pardalos *et al.*, instead, presented an original parallel algorithm developed using the MPI[b] message passing library.

Another algorithm that applies backtracking to the MCS problem is due to McGregor[102] in 1982. Differently from the methods outlined above, McGregor's

[a]http://al.ei.tuat.ac.jp/~yshinano/pubb/
[b]http://www-unix.mcs.anl.gov/mpi/

algorithm faces the MCS problem without converting it into a maximum clique problem. An experimental comparison between MCS algorithms performed by Bunke *et al.*[16] shows that under some conditions (namely, sparse graphs) this algorithm may outperform clique based methods.

Finally, we can also cite two papers presenting algorithms for problems that are strongly related to MCS. In the paper by Koch[81] in 2001, a slightly simplified version of the MCS problem is faced: the search of the *connected* MCS. The author introduces a technique to reduce this problem to maximum clique detection, and then suggests the use of the Bron–Kerbosh algorithm to find the maximum clique. In the paper by Demko[35] in 1997, a generalization of MCS to hypergraphs is investigated. The proposed method is not based on depth-first search but explores the search tree by using the widely known A* algorithm.[130]

2.1.2. *Other techniques*

Probably the most interesting matching algorithm that is not based on tree search is *Nauty*, developed by McKay[103] in 1981. The algorithm deals only with the isomorphism problem, and is regarded by many authors as the fastest isomorphism algorithm available today. It is based on group theory. In particular, it uses some results coming from this theoretical framework to construct in an efficient way the *automorphism group* of each of the input graphs. From the automorphism group, a *canonical labeling* is derived, that introduces a node ordering that is uniquely defined for each equivalence class of isomorphic graphs. So, two graphs can be checked for isomorphism by simply verifying the equality of the adjacency matrices of their canonical forms. The equality verification can be done in $O(N^2)$ time, but the construction of the canonical labeling can require an exponential time in the worst case (Miyazaki[111] in 1997 showed some classes of graphs that exhibit this exponential behavior). Anyway, in the average case this algorithm has quite impressive performance, although in Refs. 38 and 53, it has been verified that under some conditions it can be outperformed by other algorithms like the above mentioned VF2. Furthermore, it does not lend itself very well to exploit node and edge attributes of the graphs, that in many PR applications can provide an invaluable contribution to reduce the matching time.

The fact that the canonical labeling needs to be computed separately for each graph, independently of the other graph being matched, can make the Nauty algorithm really effective for matching a single graph against a large, fixed database of graphs, for which the canonical labeling has been pre-computed.

This property is shared by some other proposed algorithms that are specifically aimed at reducing the cost of matching one input graph against a large library of graphs, suitably preprocessed. Amongst the first algorithms of this kind is the one introduced in Bruno Messmer's Ph.D. thesis[106] in 1995, and successively presented in a paper by Messmer and Bunke[107] in 2000. The proposed approach, that is inspired by the RETE algorithm (used for rule matching in expert system engines),

is based on a recursive decomposition of each graph of the library into smaller subgraphs, until trivial, one-node graphs are reached. The matching process, then, exploits the fact that some of the parts are common to several graphs in the library, to avoid repeating their comparison against the input graph. In this way, the total matching time has a sublinear dependency on the number of graphs in the library. The initial version of the algorithm addresses the isomorphism and subgraph isomorphism problems.

Messmer and Bunke proposed a more impressive algorithm[18,109] in 1997. Their new algorithm, that deals with isomorphism and subgraph isomorphism, in a preprocessing phase builds a decision tree from the graph library. Using this decision tree, an input graph can be matched against the whole library in a time that is $O(N^2)$ with respect to the input graph size, and completely independent of the number of graphs in the library. An extension to MCS is presented in a paper by Shearer *et al.*[140] in 1997. More recently, Shearer *et al.*[142] also proposed an extension of this method that is further optimized for the case of a sequence of input graphs that are changing slowly over time. There is of course a price that must be paid for the excellent performance of this algorithm: the preprocessing phase requires a time that is always exponential with respect to the number of nodes in the graphs. A still more important problem is that the space required to store the decision tree is also exponential with respect to the number of nodes. For these reasons, the algorithm is practically applicable only for very small graphs (no more than a dozen nodes).

Other two recent papers, by Lazarescu *et al.*[88] in 2000 and by Irniger and Bunke[73] in 2001, proposed the use of decision trees for speeding up the matching against a large library of graphs. In these cases, the decision tree is not used to perform the matching process, but only for quickly filtering out as many library graphs as possible, applying then a complete matching algorithm only to the remaining ones.

2.2. *Inexact matching algorithms*

The stringent constraints imposed by exact matching are in some circumstances too rigid for the comparison of two graphs. In many applications, the observed graphs are subject to deformations due to several causes: intrinsic variability of the patterns, noise in the acquisition process, presence of nondeterministic elements (e.g. neural networks) in the processing steps leading to the graph representation, are among the possible reasons for having actual graphs that differ somewhat from their ideal models. So the matching process must be tolerant: it must accommodate the differences by relaxing, to some extent, the constraints that define the matching type. Even when no deformation is expected, this can be useful. As we have seen, exact graph matching algorithms (except for special kinds of graphs) require exponential time in the worst case. If this is too costly, it may be wiser to turn to algorithms that do not guarantee to find the best solution, but that, at least, give

a good approximate solution in reasonable time.

These two different needs (which may actually both be present) have led to the development of *inexact* graph matching algorithms. Usually, in these algorithms the matching between two nodes that do not satisfy the edge-preservation requirements of the matching type is not forbidden. Instead, it is penalized by assigning to it a cost that may take into account other differences (e.g. among the corresponding node/edge attributes). So the algorithm must find a mapping that minimizes the matching cost.

Optimal inexact matching algorithms always find a solution that is the global minimum of the matching cost. This implies that if an exact solution exists, it will be found by such algorithms. Hence they can be seen as a generalization of exact matching algorithms. Optimal algorithms face the problem of graph variability and they do not necessarily provide an improvement of the computation time. On the contrary, they are usually fairly more expensive than their exact counterparts.

Approximate or *suboptimal* matching algorithms, instead, only ensure to find a local minimum of the matching cost. Usually this minimum is not very far from the global one, but there are no guarantees. Even if an exact solution exists, they may not be able to find it and for some applications this may not be acceptable. If it is acceptable, then the suboptimality of the solution is abundantly repaid by a shorter, usually polynomial, matching time.

A significant number of inexact graph matching algorithms base the definition of the matching cost on an explicit model of the errors (deformations) that may occur (i.e. missing nodes, etc.), assigning a possibly different cost to each kind of error. These algorithms are often denoted as *error-correcting* or *error-tolerant*. Another way of defining a matching cost is to introduce a set of *graph edit operations* (e.g. node insertion, node deletion, etc.); once each operation is assigned a cost, the cheapest sequence of operations needed to trasform one of the two graphs into the other is computed. The cost of this sequence is called the *graph edit cost*.

Some of the inexact matching methods also propose the use of the matching cost as a measure of dissimilarity of the graphs, e.g. for selecting the most similar in a set of graphs, or for clustering. In some cases, the cost formulation verifies the mathematical properties of a distance function (e.g. the triangular inequality); then we have a *graph distance* that can be used to extend to graphs some of the algorithms defined in metric spaces. Of particular interest is the *graph edit distance*, obtained if the graph edit costs satisfy some constraints (e.g. the cost of node insertion must be equal to the cost of node deletion). Bunke demonstrated in a paper[17] of 1997 that by a suitable assignment of costs to edit operations, the MCS problem can be considered a special case of graph edit distance computation. In Ref. 19, a demonstration of the metric properties of the resulting distance is provided, while in a 1999 paper,[15] the same author shows that the graph isomorphism and subgraph isomorphism problems can be reduced to graph edit distance. Two recent papers by Wallis *et al.*[162] and by Fernandez and Valiente[48] suggested further improvements to the distance proposed by Bunke.

In the following we will review the most important inexact graph matching methods, grouped on the basis of the kind of algorithm employed for matching.

2.2.1. *Techniques based on tree search*

Tree search with backtracking can also be used for inexact matching. In this case the search is usually directed by the cost of the partial matching obtained so far, and by a heuristic estimate of the matching cost for the remaining nodes. This information can be used either to prune unfruitful paths in a branch and bound algorithm, or also to determine the order in which the search tree must be traversed, as in the A* algorithm. In this latter case, if the heuristic provides a close estimate of the future matching cost, the algorithm finds the solution quite rapidly; but if this is not the case, the memory requirement is considerably larger than for the branch and bound algorithm.

The first tree based inexact algorithm proposed in PR literature is due to Tsai and Fu[154] in 1979. The paper introduces a formal definition of error-correcting graph matching of *Attributed Relational Graphs* (ARG), based on the introduction of a graph edit cost. While the formalism is general, the proposed algorithm takes into account only the operations of node and edge substitution, omitting insertion and deletion. Hence, the graphs being matched are required to be structurally isomorphic. The proposed heuristic is based on the computation of the future node matching cost by neglecting the constraint that the mapping has to be injective; the search method ensures to find the optimal solution. In a 1983 paper,[155] the same authors propose an extension of the method that also considers insertion and deletion of nodes and edges, for an error-correcting subgraph isomorphism. A more recent paper by Wong *et al.*[172] in 1990 proposes an improvement of the heuristic of Tsai and Fu for error-correcting monomorphism, taking into account also the future cost of edge matching.

A similar approach is used in a paper by Sanfeliu and Fu[132] in 1983 where the definition of a true graph edit distance is attempted. In this paper the authors consider as basic edit operations the node and edge substitution together with node split and node merging.

Two successive papers by Eshera and Fu[44,45] in 1984, proposed a suboptimal method for the distance computation. This method is based on the decomposition of the two ARG's into the *Basic ARG's* (BARG's), that are subgraphs made by a node together with the edges starting from that node and their other endpoints. The graph matching is approximated by the simpler problem of finding an optimal match between the sets of BARG's of the two graphs, that can be computed in a polynomial time using dynamic programming.

In a paper of 1980, Gharaman *et al.*,[59] proposed an optimal inexact graph monomorphism algorithm that is based on the use of branch and bound together with a heuristic derived from the netgraph (see Sec. 2.1.1).

Another interesting early paper is due to Shapiro and Haralick[136] in 1981, the

authors propose an algorithm for finding the optimal error-correcting homomorphism between two hypergraphs. The matching algorithm is based on branch and bound with heuristics. In a later paper (1985),[137] the same authors showed that the distance proposed by Sanfeliu and Fu[132] in 1983 does not fulfil all the metric properties and propose a distance between hypergraphs that is based on the number of unmatched relations.

Among the more recent proposals based on tree search we can cite the optimal algorithm by Dumay *et al.*[40] in 1992, where a graph distance is computed using A*. The use of A* has been proposed more recently by Berretti *et al.*[9–11] in their 2000 and 2001 papers. The proposed algorithm uses a heuristic that is based on the estimate of the future cost using a *bipartite matching problem*. This problem consists in finding the largest matching between two sets of nodes forming a *bipartite graph*, with the constraint that each node must be used at most once. This is a considerably simpler problem than graph matching and can be solved in polynomial time. Another interesting aspect of the method proposed by Berretti *et al.*[11] is that their heuristic is defined incrementally, in such a way to avoid recomputing most of its terms when passing from a search state to its successors. A* search appears also in a recent paper by Gregory and Kittler[63] in 2002, where a fast, simple heuristic is used that takes into account only the future cost of unmatched nodes. The authors assume that at least for small graphs the less accurate estimate of the future cost is abundantly repaid by the time savings obtained in computing a less complicated heuristic.

Another recent inexact algorithm has been proposed by Cordella *et al.* in two papers[30,31] in 1996 and 1997. This algorithm deals with deformations by defining a transformation model in which under appropriate conditions a subgraph can be collapsed into a single node. The transformation model is contextual, in the sense that a given transformation may be selectively allowed depending on the attributes of neighboring nodes and edges.

Along the same lines, Serratosa *et al.*[135] in 1999 presented an inexact matching method that also exploits some form of contextual information. The authors define a distance between *Function Described Graphs* (FDG) that are ARG's enriched with additional information relative to the joint probability of the nodes in order to model with one FDG a set of observed ARG's. The proposed method finds the optimal distance using tree search. In a successive paper by the same authors[134] a more efficient, suboptimal algorithm is presented. This algorithm is based on the distance between the *expanded vertices* that are analogous to the BARG's cited above.

As with exact matching, parallelization has also been investigated for the inexact case. In particular, a parallel algorithm has been presented in 1997 by Allen *et al.*[2] The algorithm uses a parallelized branch and bound to compute a graph distance between two graphs having the same number of nodes.

Let us now consider some inexact matching algorithms that deal with special restricted classes of graphs: trees, planar graphs and Region Adjacency Graphs

(RAG's). The following are only a small sample, with no pretension of completeness, of the many algorithms that have been proposed for matching special kinds of graphs. Shasha *et al.* in a 1994 paper,[138] proposed a tree edit distance and compare two algorithms for its computation: an optimal algorithm, based on tree search, and a suboptimal one, that employs simulated annealing. Oflazer[113] in 1997 defined an error-correcting tree matching algorithm where only graph edit operations applied to leaves are considered; the algorithm is based on branch and bound. In 1999, Haris and Efstradiatis[65] proposed a method for computing the error correcting MCS between a tree and a *directed acyclic graph* (DAG), by performing a clique detection on an association graph. Valiente, in a 2001 paper,[158] proposed a tree distance definition showing that it can be computed in linear time with respect to the total number of nodes in the two trees being matched.

For planar graphs, Rocha and Pavlidis[128] presented an optimal algorithm for error-correcting homomorphism.

In a paper by Wang and Abe (1995),[164] a distance between RAG's is proposed, and is computed using a suboptimal algorithm. More recently, Llados *et al.* in a 2001 paper[95] defined a graph edit distance for RAGs using edit operations that are devised to model common distortions in image segmentation; the distance is computed using an optimal algorithm based on branch and bound.

2.2.2. *Continuous optimization*

The matching methods examined so far rely on a formulation of the matching problems directly in terms of graphs, or of mathematical structures with the same expressive power. A radically different approach is to cast graph matching, that is inherently a discrete optimization problem, into a continuous, nonlinear optimization problem. Then, there are many optimization algorithms that can be used to find a solution to this problem. These algorithms do not ensure the optimality of the solution (although the most sophisticated of them include techniques to avoid trivial local optima). Furthermore, the found solution needs to be converted back from the continuous domain into the initial discrete problem by a process that may introduce an additional level of approximation. Nevertheless, in many application contexts this approach is very appealing because of its extremely reduced computational cost that is usually polynomially dependent (and with a low exponent) on the size of the graphs. Moreover, the solution is, in many cases, built by successive improvements of an initial tentative mapping, allowing the system designer to choose between a quick and inaccurate solution and a more expensive one that is possibly more precise, by tuning the parameters of the algorithms.

The first family of methods based on this approach uses *relaxation labeling*. One of the pioneering works for this approach is due to Fischler and Elschlager[51] in 1973. The basic idea is that each node of one of the graphs can be assigned one label out of a discrete set of possible labels, that determines which node of the other graph it corresponds to. During the matching process, for each node there is a vector of

the probabilities of each candidate label. Initially, these probabilities are computed (heuristically) on the basis of node attributes, of node connectivity and possibly of other available information. Then, in successive iterations, each probability is modified taking into account the label probabilities of the neighboring nodes, until the process converges to a fixed point, or a maximum number of iterations is reached. At this point, for each node the label having the maximum probability is chosen.

Among the drawbacks of the initial formulations of this technique, is the fact that node/edge attributes are used only in the initialization of the matching process; moreover, the design of iteration scheme lacked a theoretical foundation. These problems have been solved by more recent papers adopting this technique. In particular, in 1989 Kittler and Hancock[80] provided a probabilistic framework for relaxation labeling, in which the update rules previously used for the probabilities are given a theoretical motivation. In 1995, Christmas *et al.*[27] proposed a method, based on the theoretical framework of Kittler and Hancock, that is able to take into account during the iteration process (and not only during initialization) both node and edge attributes. Wilson and Hancock[167] in 1997 extended the probabilistic framework by introducing a Bayesian consistency measure, that can be used as a graph distance. The authors also compare three different relaxation schemes on the basis of this measure. An extension of this method has been proposed by Huet and Hancock[71] in 1999. This method also takes into account edge attributes in the evaluation of the consistency measure. Myers *et al.*[112] in 2000 proposed a new matching algorithm based on the Wilson and Hancock probabilistic relaxation framework that introduces the definition of a Bayesian graph edit distance. This distance is then approximated by considering independently the BARG's of the graphs (that the authors denote as *supercliques*), so as to perform the computation in polynomial time. Finally, in a recent paper (2001), Torsello and Hancock[151] proposed the use of relaxation labeling also for computing an edit distance between trees.

A recent method by Luo and Hancock[98] is based, like the ones mentioned above, on a probabilistic model of matching. In this case the nodes of the input graph play the role of observed data while the nodes of the model graph act as hidden random variables. The matching is then found by using the *Expectation-Maximization* (EM) algorithm.[36] It should be noted that EM, like the other algorithms described in this section, is not guaranteed to determine a global minimum and moreover is critically dependent on initial model estimates.

Given that these recent relaxation labeling algorithms do not suffer from the problems of the early formulations, relaxation labeling only deals with a one-way correspondence: at the completion of the algorithm each node gets a label, but there is no guarantee that each label is assigned to only one node. Whether this is required or not depends on the particular application.

A different family of methods is based on a formulation of the problem as a *Weighted Graph Matching Problem* (WGM) that permits the enforcement of two-way constraints on the correspondence.

Weighted Graph Matching can be seen as a generalization of the MCS problem if edge induced subgraphs are considered. It consists in finding a matching, usually expressed by means of a matching matrix M, between a subset of the nodes of the first graph and a subset of the nodes of the second graph. The edges of the graphs are labeled with weights that are real numbers, usually between 0 and 1. The desired matching must optimize a function depending on the weights of the edges preserved by the match. The elements of M are constrained to assume only the discrete values 0 and 1 and the sum of each row and of each column must be not greater than 1 (it is the symmetry between the constraints on the rows and on the columns of M that gives the two-way nature to the solutions of WGM). Usually the problem is transformed into a continuous one by allowing M elements to have continuous values between 0 and 1. In this case, the WGM problem becomes a quadratic optimization problem. An important limitation of this approach, from the perspective of PR applications, is that nodes cannot have attributes and edges cannot have other attributes than their weight. This restriction imposes a severe limit on the use of the semantic information often available in real applications.

Among the first papers based on this formulation is the work by Almohamad and Duffuaa[3] in 1993. In this paper the quadratic problem is linearized and solved using the simplex algorithm.[87] The approximate, continuous solution found this way is then converted back into discrete form using the so-called Hungarian method[87] for the assignment problem.

Rangarajan and Mjolsness[127] in 1996, proposed a method based on Lagrangian relaxation networks in which the constraints on the rows and on the columns of the matching matrix are satisfied separately and then equated through a Lagrange multiplier. The authors add to the function to be optimized a so-called *self-amplification term* to break the symmetry in the solution space that could be an obstacle to the convergence of the algorithm if multiple global optima exist.

Also in a 1996 paper, Gold and Rangarajan[61] presented the *Graduated Assignment Graph Matching* (GAGM) algorithm. In this algorithm a technique known as *graduated nonconvexity* is employed to avoid poor local optima. With this method the constraints on the matching matrix are enforced gradually through a control parameter that is increased at each iteration of the algorithm. In this way, during the initial iterations the algorithm will be free to converge to a good value of the objective function that may not satisfy all the constraints. Then, successively, with a larger value of the control parameter that imposes more stringent constraints, the algorithm will move gradually towards a consistent solution, that of course is not guaranteed to be optimal.

Another approach that uses continuous optimization to solve graph matching problems is based on a theorem by Motzkin and Straus that establishes a close relation between the clique problem and continuous optimization. Namely, the Motzkin–Straus theorem proves that all the maximum cliques of a graph correspond to maxima of a well-defined quadratic functional. The functional

proposed by Motzkin and Straus does not satisfy the converse property: there may be maxima of this functional that do not correspond to maximum cliques. More recently, in 1997, Bomze[12] proposed a modified functional for which the correspondence holds in both senses.

Among the first papers to suggest the use of Motzkin–Straus theorem for graph matching was that of Pelillo and Jagota[120] in 1995 where the theoretical aspects of the problem are discussed, together with a method to avoid spurious maxima of the Motzkin–Straus functional (this paper appeared before Bomze theorem). In the same year Pelillo[119] proposed an implementation of the method where the quadratic problem is solved by means of relaxation networks,[129] an iterative local optimization technique.

In 1998, Pelillo[121] presented a unified framework for relational matching based on the Bomze functional and on a family of *replicator equations*, derived from evolutionary game theory, that can be used to solve the corresponding quadratic problem; the author shows that the relaxation networks used previously can be seen as a special case of replicator equations. The same author showed in a 1999 paper[123] how this framework can be used to develop a neural architecture for graph matching.

In 1999, Pelillo *et al.*[122] introduced a technique to reduce the MCS problem between trees to a clique problem and then solved it using replicator equations. A generalization of the method is presented in a 2002 paper by Pelillo[124] where the algorithm is extended to *free* trees (i.e. trees without a single root), and also a generalization of replicator equations: *monotone game dynamics* is used, showing that, suitably chosen, nonlinear monotone game dynamics may exhibit a faster convergence than linear replication equations.

Branca *et al.*[13] proposed in 1999 an extension of the framework defined by Pelillo[121] that is able to deal with a weighted version of the clique problem: the solution to be found is the one that maximizes the sum of the weights attributed to the edges of the graph. Moreover, their method also works with hypergraphs that they call *high-order graphs*.

Several other inexact matching methods based on continuous optimization have been proposed in the recent years. Among them we can cite the *Fuzzy Graph Matching* (FGM) by Medasani *et al.*,[104,105] that is a simplified version of WGM based on fuzzy logic. In FGM the objective function is considerably simpler than in WGM since the cost of matching two nodes does not depend on the matching found for the other nodes of the graphs. Hence the authors are able to derive in closed form an optimal update equation for their iterative algorithm. Another recent approach, proposed by van Wyk *et al.*[159,160] in 2002 is based on the theory of the so-called *Reproducing Kernel Hilbert Spaces* (RHKS) for casting the matching problem into a system identification problem; this latter is then solved by constructing a *RKHS interpolator* to approximate the unknown mapping function.

2.2.3. *Spectral methods*

Spectral methods are based on the following observation: the eigenvalues and the eigenvectors of the adjacency matrix of a graph are invariant with respect to node permutations. Hence, if two graphs are isomorphic, their adjacency matrices will have the same eigenvalues and eigenvectors. Unfortunately, the converse is not true: we cannot deduce from the equality of eigenvalues/eigenvectors that two graphs are isomorphic. However, since the computation of eigenvalues/eigenvectors is a well studied problem, that can be solved in polynomial time, there is a great interest in their use for graph matching. An important limitation of these methods is that they are purely structural, in the sense that they are not able to exploit node or edge attributes, that often, in PR applications, convey information very relevant for the matching process. Further, some of the spectral methods are actually able to deal only with real weights assigned to edges by using an adjacency matrix with real valued elements.

Among the pioneering works on spectral methods there is the paper by Umeyama[157] in 1998. This work proposed an algorithm for the weighted isomorphism between two graphs. Although the author used the term Weighted Graph Matching, it is a slightly more restricted problem than the WGM described above: the graphs must have the same number of nodes, and the matching matrix must be a permutation matrix (so all the nodes must participate to the matching). Umeyama used the eigendecomposition of adjacency matrices of the graphs to derive (in closed form) a simple expression of the orthogonal matrix that optimizes the objective function, under the assumption that the graphs are isomorphic. From this expression he derived a method for computing the optimal permutation matrix when the two graphs are isomorphic, and a suboptimal permutation matrix if the graphs are nearly isomorphic. Unfortunately, if it is not known in advance that the graphs are nearly isomorphic, this method can produce a very poor result.

A more recent paper of 2001, by Xu and King,[173] proposed a solution to the weighted isomorphism problem that combines the use of eigenvalues/eigenvectors with continuous optimization techniques. In particular, the method approximates the permutation matrix with a generic orthogonal matrix. An objective function is defined using Principal Component Analysis and then gradient descent is used to find the optimum of this function. The authors reported that this method is both faster and more accurate than Umeyama's.

In 2001, Carcassoni and Hancock[21] proposed a spectral method that is based on the use of spectral features to define clusters of nodes that are likely to be matched together in the optimal correspondence; the method uses hierarchical matching by first finding a correspondence between clusters and then between the nodes in the clusters. This method does not suffer from the limitation that the graphs must have the same number of nodes.

Another method that combines a spectral approach with the idea of clustering has been presented by Kosinov and Caelli[83] in 2002. In this method, a vector space,

called the *graph eigenspace*, is defined using the eigenvectors of the adjacency matrices, and the nodes are projected onto points in this space. Then, a clustering algorithm is used to find nodes of the two graphs that are to be put in correspondence. The authors show that this method is very robust to graph distortions, in the sense that corresponding nodes are always not very far in the graph eigenspace. On the other hand there is no guarantee that the converse hold, since completely unrelated nodes can have very close projections.

A method that is partly related to spectral techniques has been proposed in 2001 by Shokoufandeh and Dickinson.[144] The authors use the eigenvalues to associate to each node of a Directed Acyclic Graph a "Topological Signature Vector" (TSV) that is related to the structure of the subgraph made of the descendants of the node. These TSV are used both for a quick indexing in a graph database and for the actual graph matching algorithm. This latter is based on the combination of a greedy search procedure and of bipartite graph matching. As pointed out by the authors, the algorithm does not provide any guarantee of optimality, but should perform well on graphs with a rich structure in terms of depth and branching factor.

2.2.4. *Other techniques*

In this subsection, we will briefly present approaches to inexact matching that do not fall within the previously mentioned categories. These are: decomposition methods, neural networks, genetic algorithms, methods based on bipartite matching and methods based on local properties.

The decomposition approach introduced by Messmer and Bunke for exact graph matching has been extended to the inexact case in a 1998 paper by the same authors.[108] The proposed algorithm finds an optimal error-tolerant subgraph isomorphism between an input graph and a library of preprocessed model graphs, in sublinearly time dependent on the number of model graphs. In 1999, Fuchs and Le Men[57] proposed an improvement of this algorithm, performing first a suboptimal stochastic search to find a reasonable upper bound to the matching cost that is then used to prune the search space while searching for the optimal solution. The same authors, in a 2000 paper,[56] further extended the method to exploit prior knowledge possibly available from application-specific constraints.

Neural graph matching algorithms are usually based on an energy minimization framework, and use some kind of Hopfield network like in the clique detection method proposed by Shoukry and Aboutabl[145] in 1996 or the method by Suganthan and Yan[149] in 1998. A different approach is followed by Suganthan[146] in 2000. It is based on the idea of an unsupervised training of a neural network that must learn the correspondence between the nodes of a sample graph and the ones of a model graph. The kind of network used is a *neural gas* that is derived from Kohonen's self-organizing maps (SOM). A recent interesting paper by De Mauro *et al.*[34] in 2001 proposed the use of a recurrent neural network to compute the distance between directed acyclic graphs by projecting the graphs on a vector space and then using

Euclidean distance.

Among the applications of genetic algorithms to graph matching we can cite the paper by Liu *et al.*[93] in 1995, where a *microgenetic algorithm* is applied to the WGM problem; the paper by Wang *et al.*[196] in 1997, where a genetic algorithm is used for error-correcting isomorphism; and the paper by Perchant *et al.*[125] in 1999, where a genetic algorithm is employed to find a fuzzy homomorphism. Another interesting paper in this area is due to Khoo and Suganthan[78] in 2001. Their method uses a genetic algorithm to find the MCS between two graphs.

As already mentioned, bipartite graph matching is a simpler problem than graph matching, for which polynomial algorithms exist. Hence, some methods have been proposed that find an approximate graph matching by converting it into a bipartite matching problem. For this approach we can cite the papers by Wang *et al.*[163] in 1994, by El-Sonbaty and Ismail[42] in 1998, by Baeza and Valiente[5] in 2000 and by Liu *et al.*[92] in the same year.

Methods based on local properties perform the matching by considering only features that can be computed directly from a node or from its immediate neighbors to find a correspondence; hence the matching found may fail to preserve the overall structure of the graphs. In this category we can cite the papers by Depiero *et al.*[37] in 1996 and by Ozer *et al.*[115] in 1999. An improvement over this technique is the definition of an iterative algorithm in which the local constraints are propagated to neighboring nodes at each iteration step. Although this scheme does not ensure to find the optimal matching, it can provide in many cases good results. Examples of this approach are the *discrete relaxation algorithm* proposed in 1979 by Kitchen and Rosenfeld[79] for hypergraph monomorphism, and the error-correcting isomorphism algorithm proposed in 2002 by Hlaoui *et al.*[66]

Finally, we must say that other heuristic approaches to inexact graph matching have been proposed: at least in principle, any of the heuristic techniques that have been used for combinatorial problems or for continuous global optimization problems can be adapted to some approximate form of graph matching. With no presumption of completeness, we can cite here, as examples, simulated annealing[74] (Jagota *et al.*, 2000) and tabu search[58,168] (Gendreau *et al.*, 1993).

2.3. *Other matching problems*

In this section we briefly present other matching problems based on graphs, that have been used in the context of Pattern Recognition and Machine Vision, but do not fall strictly in the category of graph matching.

Among them, the most important is the so-called *Elastic Graph Matching* problem (EGM). Despite its name, it is not really a graph matching problem but, rather, an image matching problem that is based on a graph structure. More precisely, a regular or irregular grid is superimposed on the model image; some image features are computed at the intersections of the grid lines and are used as attributes. Successively, an isomorphic grid is superimposed on the sample image,

and is then deformed in order to have the best matching between the features computed at the sample grid points and the ones recorded previously for the model. This deformation process uses the graph structure of the two grids to define a deformation cost that constrains the entity of the permissible deformations. The best placement of the grid on the sample is usually looked for using simulated annealing (but genetic algorithms also have been used).

Probably the first paper proposing EGM is the work by Lades *et al.*[85] in 1993, where the problem is formulated in a neural framework. A 1997 paper by Wiskott *et al.*[170] extended the method by introducing a *bunch graph* for the model, that is a graph in which multiple alternative feature vectors are assigned to each node. In 1999, Duc *et al.*[39] improved the matching error definition by allowing nodes with different weights.

Other matching techniques have been proposed for Pattern Recognition applications. We will cite here just as examples a method employed for pattern classification, and a method developed to solve an indexing problem. Pavlidis *et al.*[118] in 1995 proposed an algorithm for matching *graph embeddings*: graphs whose nodes corresponds to distinct points on the plane and whose edges represent strokes connecting these points. The matching algorithm is strongly dependent upon the geometric information attached to the graphs. Chou and Shapiro[26] in 1998 proposed a pattern matching technique that is called *probabilistic relational indexing*. In this method the patterns are represented by graphs. The matching is performed by decomposing the graphs into 2-*graphs* (subgraphs made exactly of two nodes), and then computing a probabilistic similarity measure between the two sets of 2-graphs.

3. Application Taxonomy

Over the past 30 years several applications of graph-based techniques in Pattern Recognition and Machine Vision have been reported in the literature. Many of these applications are used to evaluate the performance of given graph matching techniques.

It is possible to identify at least six application areas where graph matching techniques have been successfully used. They are:

- 2D and 3D image analysis;
- Document processing;
- Biometric identification;
- Image databases;
- Video analysis;
- Biological and biomedical applications.

A taxonomy reflecting these areas is shown in Fig. 2. Here, the aim is to draw a closer link between applications and the basic graph matching techniques.

Many of the papers cited in the previous taxonomy within the image analysis field use this kind of applications for testing the performance of a

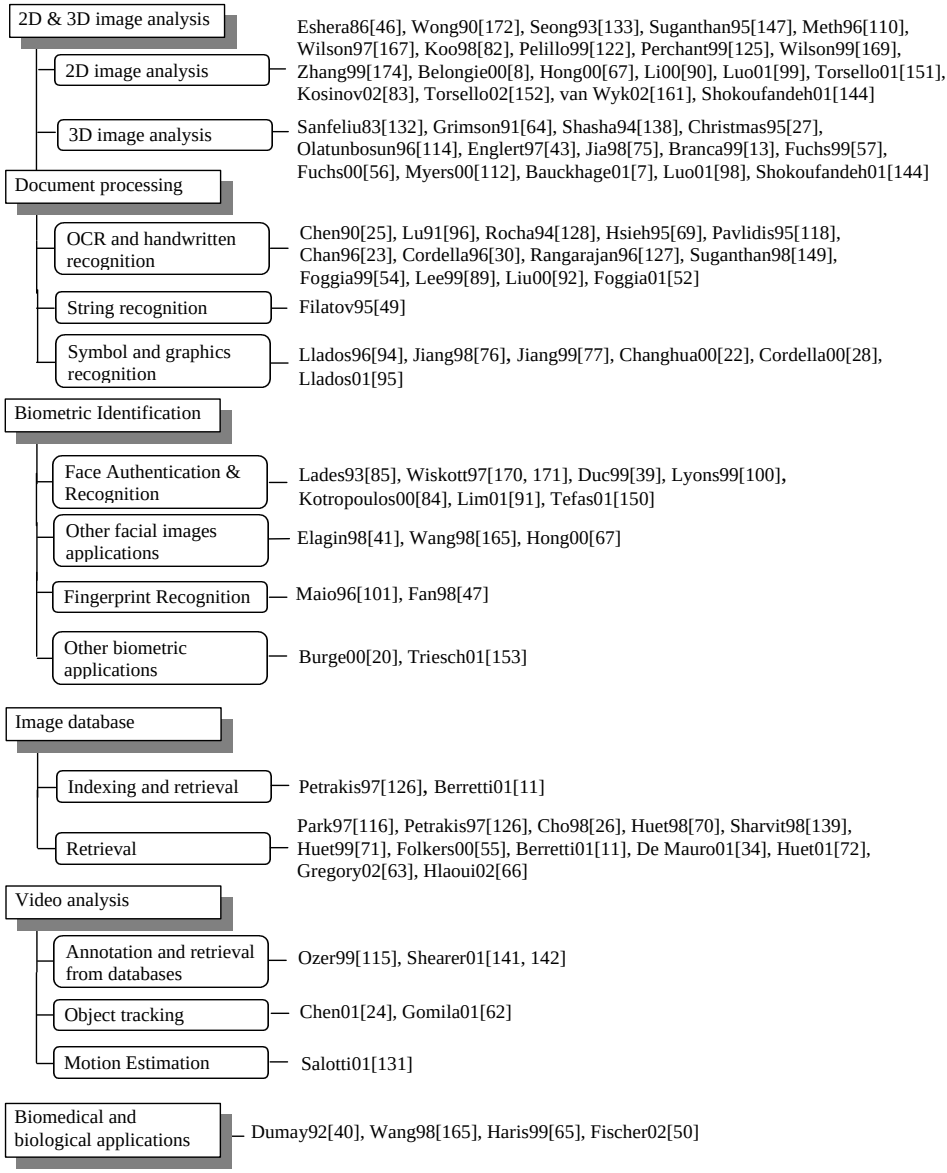

Fig. 2. Taxonomy of the reviewed graph matching applications. For each paper, the first author, the year and the corresponding bibliographic reference are reported.

given graph matching technique. For 2D image analysis, this is the case of papers that report results in the areas of *pattern recognition*,[46,67,133,144,161,172,174] *shape recognition*[83,99,122,147,151,152] *scene recognition*[125] and processing of *SAR images*.[167,169] Among the papers dealing with 3D image analysis, we can recognize the areas of *robotic vision*,[132] *stereo matching*,[27,112] *object matching*,[98] *object*

recognition[64,75,138,144] and *object reconstruction.*[43]

On the other hand, among the papers that are focused on the applicative context in the field of 2D image analysis, *object recognition* is afforded by Meth and Chellappa,[110] Li and Lee[90] and Belongie and Malik,[8] while Koo and Yoo[82] presented an application in the field of *visual inspection.* As regards 3D image analysis applications, Branca *et al.*[13] addressed the problem of *automatic navigation*, while Bauckhage *et al.*[7] and Olatunbosun *et al.*[114] the 3*D object recognition*, and Fuchs and Le Men[56,57] the 3*D object reconstruction.*

While the above mentioned 2D image analysis applications use quite different matching techniques (tree isomorphism,[82] error-correcting subgraph isomorphism with a similarity measure,[110] inexact graph matching with a neural approach,[90] weighted bipartite matching[8]), the 3D applications use only two different matching techniques: maximal clique search on the association graph[13,114] or error-correcting subgraph isomorphism algorithms.[7,56,57]

Graph matching have been used in document processing applications such as *OCR*, *handwritten recognition*, *string recognition*, *symbol* and *graphics recognition.*

OCR and handwritten character recognition have been widely used as test-bed applications for demonstrating the validity of graph-based techniques on real-world problems — as illustrated in papers.[30,52,54,23,127]

In other cases the main focus of the work is on the application: for example, Refs. 25, 69, 89, 92, 96, 118, 128 and 149.

While in Ref. 89, elastic graph matching is used in the recognition phase, the other authors cited above use inexact graph matching for dealing with the high variability of handwritten characters.

As regards handwritten recognition, some papers specifically deal with offline[149] and online[25,69,92,96] handwritten Chinese characters. They used different inexact matching techniques: Hopfield networks (presented in Ref. 148) is proposed in Ref. 149 while a tree search is performed in Ref. 96 and a relaxation labeling approach is adopted in Ref. 25. In Ref. 92, another suboptimal approach is proposed; the graph matching problem is transformed into a two-layer assignment problem and solved with the Hungarian method, while in Ref. 69, a bipartite weighted matching is used. Within the OCR field, in Ref. 128, an error-correcting matching algorithm based on tree search is used, while in Ref. 118, an *ad hoc* matching is defined between the so-called *graph embeddings.* The handwritten digit string recognition problem was addressed by Filatov *et al.* in Ref. 49 where an error-correcting graph-subgraph isomorphism algorithm is used.

To the field of symbol and graphics recognition can be ascribed the papers by Lladòs *et al.*[94,95] and Changhua *et al.*[22] The first two papers both use an inexact subgraph matching procedure that in Ref. 94 is based on discrete relaxation. On the other hand, in Ref. 22, the recognition of graphical hand-sketched symbols is realized through a similarity measure and the A* algorithm. Furthermore, technical drawings and graphic symbols are used for testing graph-based techniques

in Refs. 28, 76 and 77, respectively.

Graph-based techniques have been widely used within the context of biometric applications mainly with reference to identification problems implemented by means of elastic graph matching procedures. Among all the biometric identification problems, a key role is played by *face authentication*, *face recognition* and *fingerprint recognition*. Moreover, there are other applications based on facial images, such as *facial expression recognition* and *face pose estimation*, as well as other less known applications, such as *hand posture recognition* and *ear recognition*. All papers dealing with these problems by means of graph-based techniques have their main focus on the application, *per se*.

In the areas of face authentication and face recognition graph matching has been used in the systems proposed by Van Der Malsburg, Wiskott *et al.*,[85,170,171] by Lim and Reinders,[91] by Kotropoulos *et al.*,[84,150] by Duc *et al.*[39] and by Lyons *et al.*[100] In all these papers the face identification process is typically carried out by elastic graph matching algorithms. Among the other applications dealing with face images, papers by Wang *et al.*[165] and Hong *et al.*[67] made use of graph matching techniques in the context of *facial expression recognition* while Elagin *et al.*[41] use graph matching for *pose estimation*. They all used elastic graph matching procedures.

The use of graph matching in the context of hand posture recognition is described in the paper of Triesch and von der Malsburg.[153] Once again, the authors proposed elastic graph matching for recognition.

Another biometric system is the one proposed by Burge and Burger[20] based on the ear recognition. A subgraph error-correcting graph matching technique is proposed by the authors. Finally, fingerprint recognition by graph matching has been addressed in the papers by Maio and Maltoni[101] and by Fan *et al.*[47] They used two different approaches for recognizing fingerprints. In Ref. 101, an inexact graph matching based on a branch and bound search is proposed, while in Ref. 47, a fuzzy bipartite graph matching technique is used.

Image databases are another field in which graph-based techniques have been successfully employed. In this framework, typical applications are *indexing* and *retrieval*: few papers[11,126] addressed both aspects, while, for the most part,[26,34,55,63,66,70–72,116,139] the retrieval problem is of most interest. Among these papers only in Ref. 66, an image database is simply used for testing the performance of an error-correcting isomorphism algorithm.

As regards the matching phase, error-correcting subgraph isomorphism algorithms are mainly used.

Among the papers that address both the indexing and the retrieval problem, Berretti *et al* in Ref. 11 proposed an error-correcting algorithm, combining the A* search with an original look-ahead estimate. In the paper by Petrakis and Faloutsos,[126] a subgraph isomorphism matching algorithm with a distance measure is used.

Among the papers that mainly address the retrieval problem, Folkers *et al.*[55] and Sharvit *et al.*[139] use exact algorithms. In particular, in Ref. 55, an exact subgraph

isomorphism is proposed that makes use of a suitably defined similarity measure for pruning some isomorphism checks, while in Ref. 139, a weighted graph matching that is a variant of the method presented in Ref. 61, is employed. All the other papers use error-correcting matching algorithms. Gregory and Kittler[63] utilize an error-correcting subgraph isomorphism based on tree search. Cho and Yoo[26] proposed a subgraph isomorphism algorithm that makes use of a similarity measure, while Park *et al.*[116] define a similarity measure directly obtainable by the adjacency matrices of the graphs. Finally, a learning technique, based on a recurrent neural network, is proposed by de Mauro *et al.*[34] In the papers by Hancock and Huet,[70–72] the aim is to retrieve 2D images from large databases. They also make use of inexact graph matching algorithms. In Ref. 70, a fuzzy variant of the Hausdorff distance that uses only the values of the edge attributes is proposed for comparing graphs. In Ref. 71, the matching process is realized by means of a Bayesian graph matching algorithm that uses an extension of the relaxation technique reported by Wilson and Hancock.[167] Huet *et al.*[72] presented an application of the image retrieval for verifying similarities among different technical drawings representing patents; the matching is realized by means of the distance presented in Ref. 70.

Among the video analysis problems, *retrieval from video databases*,[141,142] *annotation of video databases*,[115] *object tracking*[24,62] and *motion estimation*[131] have been addressed by using graph-based techniques. In all these papers the application, *per se*, is of central focus. In this case, since the above mentioned applications are very different, the matching techniques employed are quite unlike each other.

In the framework of retrieval from video databases, Shearer *et al.*[141] used an exact decision tree-based algorithm applied to the detection of the largest common subgraph, while in Ref. 142, the same authors proposed an extension of the decision tree-based isomorphism algorithm presented by Bunke and Messmer[18] in order to cope with dynamically changing graphs.

A quite peculiar approach to the problem of retrieval from databases is the one presented by Ozer *et al.*[115] The aim of this work was to annotate images or videos where a particular object is present, so that a simple textual query can be performed for retrieving images from a preprocessed database. They proposed a cost-based inexact subgraph matching procedure in conjunction with a depth-first search that uses a brute force approach.

Both the papers by Chen *et al.*[24] and Gomila and Mayer[62] exploit the use of graph matching for object tracking in video sequences. They used different matching techniques: in Ref. 24, a bipartite matching algorithm is applied, while in Ref. 62, an error-correcting matching algorithm using relaxation labeling is proposed.

Salotti and Laachfoubi[131] presented an application of motion estimation in aerial videos. In this context, in order to collect information for preventing fires, their aim is to estimate the shift of smoke clouds within a video. The shift estimation is performed by means of an inexact matching procedure based on a cost function for matching nodes relative to successive video frames.

Finally, graph matching techniques have been used within biomedical[40,65,165]

and biological[50] applications. While in Ref. 165, the problem of finding motifs in multiple RNA secondary structures is used only for testing a graph-based approach, the other biomedical applications have their main focus on the application context. Both address the problem of the correct identification of coronary arteries (artery labeling) starting from medical images, but the labeling is carried out by using different graph matching techniques. In the paper by Dumay *et al.*,[40] an inexact graph matching procedure employing the A* algorithm to perform the tree search is used, while Haris *et al.*[65] reformulate the labeling problem in terms of the maximal clique detection in the association graph.

As regards biological applications, the identification of diatoms described by Fischer *et al.*[50] can be cited. Diatoms are unicellular algae found in water and in other places where there is humidity and enough light for allowing photosynthesis. The matching procedure presented here can be seen as a simple form of error-correcting graph matching.

4. Conclusions

In this paper we have reviewed, discussed and categorized more than 160 papers reporting graph matching algorithms in the context of the Pattern Recognition and Machine Vision. Among them, more than 100 papers discussing applications have been cited.

The links between the different application areas and the graph-based techniques employed have also been highlighted in order to provide useful hints to researchers when considering the use of graph matching in a particular domain.

References

1. A. V. Aho, J. E. Hopcroft and J. D. Ullman, *The Design and Analysis of Computer Algorithms* (Addison Wesley, 1974).
2. R. Allen, L. Cinque, S. Tanimoto, L. G. Shapiro and D. Yasuda, A parallel algorithm for graph matching and its MasPar implementation, *IEEE Trans. Parall. Distrib. Syst.* **8** (1997) 490–501.
3. H. A. Almohamad and S. O. Duffuaa, A linear programming approach for the weighted graph matching problem, *IEEE Trans. Patt. Anal. Mach. Intell.* **15** (1993) 522–525.
4. A. P. Ambler, H. G. Barrow, C. M. Brown, R. M. Burstall and R. J. Popplestone, A versatile computer-controlled assembly system, in *Proc. 3rd Int. Joint Conf. Artificial Intelligence*, 1973, pp. 298–307.
5. R. Baeza-Yates and G. Valiente, An image similarity measure based on graph matching, in *Proc. Seventh Int. Symp. String Processing and Information Retrieval*, 2000, pp. 28–38.
6. E. Balas and C. S. Yu, Finding a maximum clique in an arbitrary graph, *SIAM J. Comput.* **15** (1986) 1054–1068
7. C. Bauckhage, S. Wachsmuth and G. Sagerer, 3D assembly recognition by matching functional subparts, in *Proc. 3rd IAPR-TC15 Workshop on Graph-Based Representations in Pattern Recognition*, 2001, pp. 95–104.

8. S. Belongie and J. Malik, Matching with shape contexts, in *Proc. IEEE Workshop on Content-Based Access of Image and Video Libraries*, 2000, pp. 20–26.
9. S. Berretti, A. Del Bimbo and E. Vicario, A look-ahead strategy for graph matching in retrieval by spatial arrangement, *Int. Conf. Multimedia and Expo*, 2000, pp. 1721–1724.
10. S. Berretti, A. Del Bimbo and E. Vicario, The computational aspect of retrieval by spatial arrangement, in *Proc. 15th Int. Conf. Pattern Recognition*, 2000, pp. 1047–1051.
11. S. Berretti, A. Del Bimbo and E. Vicario, Efficient matching and indexing of graph models in content-based retrieval, *IEEE Trans. Patt. Anal. Mach. Intell.* **23** (2001) 1089–1105.
12. I. M. Bomze, Evolution towards the maximum clique, *J. Global Optim.* **10** (1997) 143–164.
13. A. Branca, E. Stella and A. Distante, Feature matching by searching maximum clique on high order association graph, in *Proc. 10th Int. Conf. Image Analysis and Processing*, 1999, pp. 642–658.
14. C. Bron and J. Kerbosch, Finding all cliques of an undirected graph, *Commun. ACM* **16** (1973) 575–577.
15. H. Bunke, Error correcting graph matching: on the influence of the underlying cost function, *IEEE Trans. Patt. Anal. Mach. Intell.* **21** (1999) 917–922.
16. H. Bunke, P. Foggia, M. Vento, C. Sansone and C. Guidobaldi, A comparison of algorithms for maximum common subgraph on randomly connected graphs, in *Proc. Joint IAPR Int. Workshops SSPR and SPR*, 2002, pp. 123–132.
17. H. Bunke, On a relation between graph edit distance and maximum common subgraph, *Patt. Recogn. Lett.* **18** (1997) 689–694.
18. H. Bunke and B. T. Messmer, Recent advances in graph matching, *Int. J. Patt. Recogn. Artif. Intell.* **11** (1997) 169–203.
19. H. Bunke and K. Shearer, A graph distance metric based on the maximal common subgraph, *Patt. Recogn. Lett.* **19** (1998) 255–259.
20. M. Burge and W. Burger, Ear biometrics in computer vision, in *Proc. 15th Int. Conf. Pattern Recognition*, 2000, pp. 822–826.
21. M. Carcassoni and E. R. Hancock, Weighted graph-matching using modal clusters, in *Proc. 3rd IAPR-TC15 Workshop Graph-Based Representations in Pattern Recognition*, 2001, pp. 260–269.
22. L. Changhua, Y. Bing and X. Weixin, Online hand-sketched graphics recognition based on attributed relational graph matching, in *Proc. 3rd World Congress Intelligent Control and Automation*, 2000, pp. 2549–2553.
23. K. P. Chan, Learning templates from fuzzy examples in structural pattern recognition, *IEEE Trans. Syst. Man Cybern.* **B26** (1996) 118–123..
24. H. T. Chen, H. Lin and T. L. Liu, Multi-object tracking using dynamical graph matching, in *Proc. 2001 IEEE Computer Soc. Conf. Computer Vision and Pattern Recognition*, 2001, pp. 210–217.
25. L. H. Chen and J. R. Lieh, Handwritten character recognition using a 2-layer random graph model by relaxation matching, *Patt. Recogn.* **23** (1990) 1189–1205.
26. S. J. Cho and S. I. Yoo, Image retrieval using topological structure of user sketch, *IEEE Int. Conf. Syst. Man Cybern.*, 1998, pp. 4584–4588.
27. W. J. Christmas, J. Kittler and M. Petrou, Structural matching in computer vision using probabilistic relaxation, *IEEE Trans. Patt. Anal. Mach. Intell.* **17** (1995) 749–764.

28. L. P. Cordella, P. Foggia, C. Sansone and M. Vento, Fast graph matching for detecting CAD image components, in *Proc. 15th Int. Conf. Pattern Recognition*, 2000, pp. 1034–1037.
29. L. P. Cordella, P. Foggia, C. Sansone and M. Vento, An improved algorithm for matching large graphs, in *Proc. 3rd IAPR-TC15 Workshop Graph-Based Representations in Pattern Recognition*, 2001, pp. 149–159.
30. L. P. Cordella, P. Foggia, C. Sansone and M. Vento, An efficient algorithm for the inexact matching of ARG graphs using a contextual transformational model, in *Proc. 13th Int. Conf. Pattern Recognition*, 1996, pp. 180–184.
31. L. P. Cordella, P. Foggia, C. Sansone and M. Vento, Subgraph transformations for inexact matching of attributed relational graphs, *Computing* **12**(Suppl.) (1997) 43–52.
32. L. P. Cordella, P. Foggia, C. Sansone, F. Tortorella and M. Vento, Graph matching: a fast algorithm and its evaluation, in *Proc. 14th Int. Conf. Pattern Recognition*, 1998, pp. 1582–1584.
33. L. P. Cordella, P. Foggia, C. Sansone and M. Vento, Performance evaluation of the VF graph matching algorithm, in *Proc. Int. Conf. Image Analysis and Processing*, 1999, pp. 1172–1177.
34. C. De Mauro, M. Diligenti, M. Gori and M. Maggini, Similarity learning for graph-based image representations, in *Proc. 3rd IAPR-TC15 Workshop Graph-Based Representations in Pattern Recognition*, 2001, pp. 250–259.
35. Ch. Demko, Generalization of two hypergraphs. Algorithm of calculation of the greatest sub-hypergraph common to two hypergraphs annotated by semantic information, *Computing* **12**(Suppl.) (1997) 1–9.
36. A. P. Dempster, N. M. Laird and D. B. Rubin, Maximum-likelihood from incomplete data via the EM algorithm, *J. Roy. Stat. Soc. Series* **B39** (1977) 1–38.
37. F. Depiero, M. Trivedi and S. Serbin, Graph matching using a direct classification of node attendance, *Patt. Recogn.* **29** (1996) 1031–1048.
38. M. De Santo, P. Foggia, C. Sansone and M. Vento, A large database of graphs and its use for benchmarking graph isomorphism algorithms, *Patt. Recogn. Lett.* **24** (2003) 1067–1079.
39. B. Duc, S. Fischer and J. Bigun, Face authentication with Gabor information on deformable graphs, *IEEE Trans. Imag. Process.* **8** (1999) 504–516.
40. A. C. M. Dumay, R. J. van der Geest, J. J. Gerbrands, E. Jansen and J. H. C. Reiber, Consistent inexact graph matching applied to labelling coronary segments in arteriograms, in *Proc. Int. Conf. Pattern Recognition, Conf. C* (1992) 439–442.
41. E. Elagin, J. Steffens and H. Neven, Automatic pose estimation system for human faces based on bunch graph matching technology, in *Proc. Third IEEE Int. Conf. Automatic Face and Gesture Recognition*, 1998, pp. 136–141.
42. Y. El-Sonbaty and M. A. Ismail, A new algorithm for subgraph optimal isomorphism, *Patt. Recogn.* **31** (1998) 205–218.
43. R. Englert, A. B. Cremers and J. Seelmann-Eggebert, Recognition of polymorphic pattern in parameterized graphs for 3d building reconstruction, *Computing* **12**(Suppl.) (1997) 11–20.
44. M. A. Eshera and K. S. Fu, A similarity measure between attributed relational graphs for image analysis, in *Proc. 7th Int. Conf. Pattern Recognition*, 1984, pp. 75–77.
45. M. A. Eshera and K. S. Fu, A graph distance measure for image analysis, *IEEE Trans. Syst. Man Cybern.* **14** (1984) 398–408.

46. M. A. Eshera and K. S. Fu, An image understanding system using attributed symbolic representation and inexact graph-matching, *IEEE Trans. Patt. Anal. Mach. Intell.* **8** (1986) 604–618.
47. K. C. Fan, C. W. Liu and Y. K. Wang, A fuzzy bipartite weighted graph matching approach to fingerprint verification, in *Proc. IEEE Int. Conf. Systems, Man and Cybernetics*, 1998, pp. 4363–4368.
48. M. L. Fernàndez and G. Valiente, A graph distance metric combining maximum common subgraph and minimum common supergraph, *Patt. Recogn. Lett.* **22** (2001) 753–758.
49. A. Filatov, A. Gitis and I. Kil, Graph-based handwritten digit string recognition, in *Proc. 3rd Int. Conf. Document Analysis and Recognition*, 1995, pp. 845–848.
50. S. Fischer, K. Gilomen and H. Bunke, Identification of diatoms by grid graph matching, in *Proc. Joint IAPR Int. Workshops SSPR and SPR*, 2002, pp. 94–103.
51. M. Fischler and R. Elschlager, The representation and matching of pictorial structures, *IEEE Trans. Comput.* **22** (1973) 67–92.
52. P. Foggia, R. Genna and M. Vento, Symbolic versus connectionist learning: an experimental comparison in a structured domain, *IEEE Trans. Knowl. Data Engin.* **13** (2001) 176–195.
53. P. Foggia, C. Sansone and M. Vento, A performance comparison of five algorithms for graph isomorphism, in *Proc. 3rd IAPR-TC15 Workshop Graph-Based Representations in Pattern Recognition*, 2001, pp. 188–199.
54. P. Foggia, C. Sansone, F. Tortorella and M. Vento, Definition and validation of a distance measure between structural primitives, *Patt. Anal. Appl.* **2** (1999) 215–227.
55. A. Folkers, H. Samet and A. Soffer, Processing pictorial queries with multiple instances using isomorphic subgraphs, in *Proc. 15th Int. Conf. Pattern Recognition*, 2000, pp. 51–54.
56. F. Fuchs and H. Le Men, Efficient subgraph isomorphism with *a priori* knowledge, in *Proc. Joint IAPR Int. Workshops SSPR and SPR*, 2000, pp. 427–436.
57. F. Fuchs and H. Le Men, Building reconstruction on aerial images through multi-primitive graph matching, in *Proc. 2nd IAPR-TC15 Workshop Graph-Based Representations in Pattern Recognition*, 1999, pp. 21–30.
58. M. Gendreau, L. Salvail and P. Soriano, Solving the maximum clique problem using a tabu search approach, *Ann. Operat. Res.* **41** (1993) 385–403.
59. D. E. Ghahraman, A. K. C. Wong and T. Au, Graph optimal monomorphism algorithms, *IEEE Trans. Syst. Man Cybern.* **10** (1980) 181–188.
60. D. E. Ghahraman, A. K. C. Wong and T. Au, Graph monomorphism algorithms, *Trans. Syst. Man Cybern.* **10** (1980) 189–197.
61. S. Gold and A. Rangarajan, A graduated assignment algorithm for graph matching, *IEEE Trans. Patt. Anal. Mach. Intell.* **18** (1996) 377–388.
62. C. Gomila and F. Meyer, Tracking objects by graph matching of image partition sequences, in *Proc. 3rd IAPR-TC15 Workshop Graph-Based Representations in Pattern Recognition*, 2001, pp. 1–11.
63. L. Gregory and J. Kittler, Using graph search techniques for contextual colour retrieval, in *Proc. Joint IAPR Int. Workshops SSPR and SPR*, 2002, pp. 186–194.
64. W. E. Grimson, *Object Recognition by Computer: The Role of Geometric Constraints* (MIT Press, 1991).
65. K. Haris, S. N. Efstradiatis, N. Maglaveras, C. Pappas, J. Gourassas and G. Louridas, Model-based morphological segmentation and labeling of coronary angiograms, *IEEE Trans. Med. Imag.* **18** (1999) 1003–1015.

66. A. Hlaoui and S. Wang, A new algorithm for graph matching with application to content-based image retrieval, in *Proc. Joint IAPR Int. Workshops SSPR and SPR*, 2002, pp. 291–300.
67. P. Hong, R. Wang and T. Huang, Learning patterns from images by combining soft decisions and hard decisions, in *Proc. 2000 IEEE Computer Society Conf. Computer Vision and Pattern Recognition*, 2000, pp. 79–83.
68. J. E. Hopcroft and J. Wong, Linear time algorithm for isomorphism of planar graphs, in *Proc. 6th Annual ACM Symp. Theory of Computing*, 1974, pp. 172–184.
69. A. J. Hsieh, K. C Fan and T. I. Fan, Bipartite weighted matching for on-line handwritten Chinese character recognition, *Patt. Recogn.* **28** (1995) 143–151.
70. B. Huet and E. R. Hancock, Fuzzy relational distance for large-scale object recognition, in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 1998, pp. 138–143.
71. B. Huet and E. R. Hancock, Shape recognition from large image libraries by inexact graph matching, *Patt. Recogn. Lett.* **20** (1999) 1259–1269.
72. B. Huet, N. J. Kern, G. Guarascio and B. Merialdo, Relational skeletons for retrieval in patent drawings, in *Proc. Int. Conf. Image Processing*, 2001, pp. 737–740.
73. C. Irniger and H. Bunke, Graph matching: filtering large databases of graphs using decision trees, in *Proc. 3rd IAPR-TC15 Workshop Graph-Based Representations in Pattern Recognition*, 2001, pp. 239–249.
74. A. Jagota, M. Pelillo and A. Rangarajan, A new deterministic annealing algorithm for maximum clique, in *Proc. Int. Joint Conf. Neural Networks* **6** (2000) 505–508.
75. J. Jia and K. Abe, Automatic generation of prototypes in 3D structural object recognition, in *Proc. Fourteenth Int. Conf. Pattern Recognition*, 1998, pp. 697–700.
76. X. Jiang and H. Bunke, Marked subgraph isomorphism of ordered graphs, in *Proc. Joint IAPR Int. Workshops SSPR and SPR*, 1998, pp. 122–131.
77. X. Jiang, A. Munger and H. Bunke, Synthesis of representative symbols by computing generalized median graphs, in *Proc. Int. Workshop Graphics Recognition GREC '99*, 1999, pp. 187–194.
78. K. G. Khoo and P. N. Suganthan, Multiple relational graphs mapping using genetic algorithms, in *Proc. 2001 Congress on Evolutionary Computation*, 2001, pp. 727–733.
79. L. Kitchen and A. Rosenfeld, Discrete relaxation for matching relational structures, *IEEE Trans. Syst. Man Cybern.* **9** (1979) 869–874.
80. J. Kittler and E. R. Hancock, Combining evidence in probabilistic relaxation, *Int. J. Patt. Recogn. Artif. Intell.* **3** (1989) 29–51.
81. I. Koch, Enumerating all connected maximal common subgraphs in two graphs, *Theoret. Comput. Sci.* **250** (2001) 1–30.
82. J. H. Koo and S. I. Yoo, A structural matching for two-dimensional visual pattern inspection, in *Proc. IEEE Int. Conf. Systems, Man and Cybernetics*, 1998, pp. 4429–4434.
83. S. Kosinov and T. Caelli, Inexact multisubgraph matching using graph eigenspace and clustering models, in *Proc. Joint IAPR Int. Workshops SSPR and SPR*, 2002, pp. 133–142.
84. C. Kotropoulos, A. Tefas and I. Pitas, Frontal face authentication using morphological elastic graph matching, *IEEE Trans. Imag. Process.* **9** (2000) 555–560.
85. M. Lades, J. C. Vorbruggen, J. Buhmann, J. Lange, C. von der Malsburg, R. P. Wurz and W. Konen, Distortion invariant object recognition in the dynamic link architecture, *IEEE Trans. Comput.* **42** (1993) 300–311.
86. J. Larrosa and G. Valiente, Constraint satisfaction algorithms for graph pattern matching, *Math. Struct. Comput. Sci.* **12** (2002) 403–422.

87. E. S. Lawler, *Combinatorial Optimization: Networks and Matroids* (Dover Pubns Publisher, 2001).
88. M. Lazarescu, H. Bunke and S. Venkatesh, Graph matching: fast candidate elimination using machine learning techniques, in *Proc. Joint IAPR Int. Workshops SSPR and SPR*, 2000, pp. 236–245.
89. R. S. T. Lee and J. N. K. Liu, An oscillatory elastic graph matching model for recognition of offline handwritten Chinese characters, *Third Int. Conf. Knowledge-Based Intelligent Information Engineering Systems*, 1999, pp. 284–287.
90. W. J. Li and T. Lee, Object recognition by sub-scene graph matching, *IEEE Int. Conf. Robotics and Automation*, 2000, pp. 1459–1464.
91. R. Lim and M. J. T. Reinders, Facial landmarks localization based on fuzzy and gabor wavelet graph matching, in *Proc. 10th IEEE Int. Conf. Fuzzy Systems*, 2001, pp. 683–686.
92. J. Z. Liu, K. Ma, W. K. Cham and M. M. Y. Chang, Two-layer assignment method for online Chinese character recognition, *IEEE Proc. Vis. Imag. Sign. Process.* **147** (2000) 47–54.
93. C. W. Liu, K. C. Fan, J. T. Horng and Y. K. Wang, Solving weighted graph matching problem by modified microgenetic algorithm, *IEEE Int. Conf. Syst. Man Cybern.*, 1995, pp. 638–643.
94. J. Llados, J. Lopez-Krahe and E. Marti, Hand drawn document understanding using the straight line Hough transform and graph matching, in *Proc. 13th Int. Conf. Pattern Recognition*, 1996, pp. 497–501.
95. J. Llados, E. Marti and J. J. Villanueva, Symbol recognition by error-tolerant subgraph matching between region adjacency graphs, *IEEE Trans. Patt. Anal. Mach. Intell.* **23** (2001) 1137–1143.
96. S. W. Lu, Y. Ren and C. Y. Suen, Hierarchical attributed graph representation and recognition of handwritten Chinese characters, *Patt. Recogn.* **24** (1991) 617–632.
97. E. M. Luks, Isomorphism of graphs of bounded valence can be tested in polynomial time, *J. Comput. Syst. Sci.* **25** (1982) 42–65.
98. B. Luo and E. R. Hancock, Structural graph matching using the EM algorithm and singular value decomposition, *IEEE Trans. Patt. Anal. Mach. Intell.* **23** (2001) 1120–1136.
99. B. Luo, A. Robles-Kelly, A. Torsello, R. C. Wilson and E. R. Hancock, Clustering shock trees, in *Proc. 3rd IAPR-TC15 Workshop Graph-Based Representations in Pattern Recognition*, 2001, pp. 217–228.
100. M. J. Lyons, J. Budynek and S. Akamatsu, Automatic classification of single facial images, *IEEE Trans. Patt. Anal. Mach. Intell.* **21** (1999) 1357–1362.
101. D. Maio and D. Maltoni, A structural approach to fingerprint classification, in *Proc. 13th Int. Conf. Pattern Recognition*, 1996, pp. 578–585.
102. J. J. McGregor, Backtrack search algorithm and the maximal common subgraph problem, *Software — Practice and Experience* **12** (1982) 23–34.
103. B. D. McKay, Practical graph isomorphism, *Congressus Numerantium* **30** (1981) 45–87.
104. S. Medasani, R. Krishnapuram and Y. S. Choi, Graph matching by relaxation of fuzzy assignments, *IEEE Trans. Fuzzy Syst.* **9** (2001) 173–182.
105. S. Medasani and R. Krishnapuram, A fuzzy approach to content-based image retrieval, in *Proc. IEEE Int. Conf. Fuzzy Systems*, 1999, pp. 1251–1260.
106. B. Messmer, *Efficient Graph Matching Algorithm for Preprocessed Model Graphs*, Ph.D. thesis, Institut fur Informatik und Angewandte Mathematik, University of Bern (1995).

107. B. T. Messmer and H. Bunke, Efficient subgraph isomorphism detection: a decomposition approach, *IEEE Trans. Knowl. Data Eng.* **12** (2000) 307–323.
108. B. T. Messmer and H. Bunke, A new algorithm for error-tolerant subgraph isomorphism detection, *IEEE Trans. Patt. Anal. Mach. Intell.* **20** (1998) 493–504.
109. B. T. Messmer and H. Bunke, A decision tree approach to graph and subgraph isomorphism detection, *Patt. Recogn.* **32** (1998) 1979–1998.
110. R. Meth and R. Chellappa, Target indexing in synthetic aperture radar imagery using topographic features, in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing*, 1996, pp. 2152–2155.
111. T. Miyazaki, The complexity of McKay's canonical labeling algorithm, *Groups and Computation II*, DIMACS Series Discrete Mathematics Theoretical Computer Science **28** (1997) 239–256.
112. R. Myers, R. C. Wilson and E. R. Hancock, Bayesian graph edit distance, *IEEE Trans. Patt. Anal. Mach. Intell.* **22** (2000) 628–635.
113. K. Oflazer, Error-tolerant retrieval of trees, *IEEE Trans. Patt. Anal. Mach. Intell.* **19** (1997) 1376–1380.
114. S. Olatunbosun, G. R. Dowling and T. J. Ellis, Topological representation for matching coloured surfaces, in *Proc. Int. Conf. Image Processing*, 1996, pp. 1019–1022.
115. B. Ozer, W. Wolf and A. N. Akansu, A graph based object description for information retrieval in digital image and video libraries, in *Proc. IEEE Workshop Content-Based Access of Image and Video Libraries*, 1999, pp. 79–83.
116. I. K. Park, I. D. Yun and S. U. Lee, Models and algorithms for efficient color image indexing, in *Proc. IEEE Workshop Content-Based Access of Image and Video Libraries*, 1997, pp. 36–41.
117. P. Pardalos, J. Rappe and M. G. C. Resende, An exact parallel algorithm for the maximum clique problem, in *High Performance Algorithms and Software in Nonlinear Optimization* (Kluwer Academic Publishers, 1998).
118. T. Pavlidis, W. J. Sakoda and H. Shi, Matching graph embeddings for shape analysis, in *Proc. Third Int. Conf. Document Analysis and Recognition*, 1995, pp. 729–733.
119. M. Pelillo, Relaxation labeling networks for the maximum clique problem, *J. Artif. Neural Networks* **2** (1995) 313–328.
120. M. Pelillo and A. Jagota, Feasible and infeasible maxima in a quadratic program for maximum clique, *J. Artif. Neural Networks* **2** (1995) 411–420.
121. M. Pelillo, A unifying framework for relational structure matching, in *Proc. Fourteenth Int. Conf. Pattern Recognition*, 1998, pp. 1316–1319.
122. M. Pelillo, K. Siddiqi and S. W. Zucker, Matching hierarchical structures using association graphs, *IEEE Trans. Patt. Anal. Mach. Intell.* **21** (1999) 1105–1120.
123. M. Pelillo, Replicator equations, maximal cliques and graph isomorphism, *Neural Comput.* **11** (1999) 1933–1955.
124. M. Pelillo, Matching free trees, maximal cliques and monotone game dynamics, *IEEE Trans. Patt. Anal. Mach. Intell.* **24** (2002) 1535–1541.
125. A. Perchant, C. Boeres, I. Bloch, M. Roux and C. Ribeiro, Model-based scene recognition using graph fuzzy homomorphism solved by genetic algorithm, in *Proc. 2nd IAPR-TC15 Workshop Graph-Based Representations in Pattern Recognition*, 1999, pp. 61–70.
126. G. M. Petrakis and C. Faloutsos, Similarity searching in medical image databases, *IEEE Trans. Knowl. Data Eng.* **9** (1997) 435–447.
127. A. Rangarajan and E. D. Mjolsness, A Lagrangian relaxation network for graph matching, *IEEE Trans. Neural Networks* **7** (1996) 1365–1381.

128. J. Rocha and T. Pavlidis, A shape analysis model with applications to a character recognition system, *IEEE Trans. Patt. Anal. Mach. Intell.* **16** (1994) 393–404.
129. A. Rosenfeld, R. A. Hummel and S. W. Zucker, Scene labelling by relaxation operations, *IEEE Trans. Syst. Man Cybern.* **6** (1976) 420–433.
130. S. J. Russel and P. Norvig, in *Artificial Intelligence: A Modern Approach*, Series in Artificial Intelligence (Prentice Hall, 1995).
131. M. Salotti and N. Laachfoubi, Topographic graph matching for shift estimation, in *Proc. 3rd IAPR-TC15 Workshop Graph-Based Representations in Pattern Recognition*, 2001, pp. 54–63.
132. A. Sanfeliu and K. S. Fu, A distance measure between attributed relational graphs for pattern recognition, *IEEE Trans. Syst. Man Cybern.* **13** (1983) 353–363.
133. D. Seong, H. S. Kim and K. H. Park, Incremental clustering of attributed graphs, *IEEE Trans. Syst. Man Cybern.* **23** (1993) 1399–1411.
134. F. Serratosa, R. Alquezar and A. Sanfeliu, Efficient algorithms for matching attributed graphs and function-described graphs, in *Proc. 15th Int. Conf. Pattern Recognition*, 2000, pp. 867–872.
135. F. Serratosa, R. Alquezar and A. Sanfeliu, Function-described graphs: a fast algorithm to compute a sub-optimal matching measure, in *Proc. 2nd IAPR-TC15 Workshop Graph-Based Representations in Pattern Recognition*, 1999, pp. 71–77.
136. L. G. Shapiro and R. M. Haralick, Structural descriptions and inexact matching, *IEEE Trans. Patt. Anal. Mach. Intell.* **3** (1981) 504–519.
137. L. G. Shapiro and R. M. Haralick, A metric for comparing relational descriptions, *IEEE Trans. Patt. Anal. Mach. Intell.* **7** (1985) 90–94.
138. D. Shasha, J. T. L. Wang, K. Zhang and F. Y. Shih, Exact and approximate algorithms for unordered tree matching, *IEEE Trans. Syst. Man Cybern.* **24** (1994) 668–678.
139. D. Sharvit, J. Chan, H. Tek and B. B. Kimia, Symmetry-based indexing of image databases, in *Proc. IEEE Workshop Content-Based Access of Image and Video Libraries*, 1998, pp. 56–62.
140. K. Shearer, H. Bunke, S. Venkatesh and S. Kieronska, Efficient graph matching for video indexing, *Computing* **12**(Suppl.) (1997) 53–62.
141. K. Shearer H. Bunke and S. Venkatesh, Video indexing and similarity retrieval by largest common subgraph detection using decision trees, *Patt. Recogn.* **34** (2001) 1075–1091.
142. K. Shearer, S. Venkatesh and H. Bunke, Video sequence matching via decision tree path following, *Patt. Recogn. Lett.* **22** (2001) 479–492.
143. Y. Shinano, T. Fujie, Y. Ikebe and R. Hirabayashi, Solving the maximum clique problem using PUBB, in *Proc. First Merged Int. Parallel Processing Symposium*, 1998, pp. 326–332.
144. A. Shokoufandeh and S. Dickinson, A unified framework for indexing and matching hierarchical shape structures, Lecture Notes in Computer Science, Vol. 2059 (2001) 67–84.
145. A. Shoukry and M. Aboutabl, Neural network approach for solving the maximal common subgraph problem, *IEEE Trans. Syst. Man Cybern.* **26** (1996) 785–790.
146. P. N. Suganthan, Attributed relational graph matching by neural-gas networks, in *Proc. 2000 IEEE Signal Processing Society Workshop on Neural Networks for Signal Processing X*, 2000, pp. 366–374.
147. P. N. Suganthan, E. K. Teoh and D. Mital, Pattern recognition by graph matching using the potts MFT neural networks, *Patt. Recogn.* **28** (1995) 997–1009.
148. P. N. Suganthan, E. K. Teoh and D. Mital, A self organizing Hopfield network for

attributed relational graph matching, *Imag. Vis. Comput.* **13** (1995) 61–73.
149. P. N. Suganthan and H. Yan, Recognition of handprinted Chinese characters by constrained graph matching, *Imag. Vis. Comput.* **16** (1998) 191–201.
150. A. Tefas, C. Kotropoulos and I. Pitas, Using support vector machines to enhance the performance of elastic graph matching for frontal face authentication, *IEEE Trans. Patt. Anal. Mach. Intell.* **23** (2001) 735–746.
151. A. Torsello and E. R. Hancock, Computing approximate tree edit-distance using relaxation labeling, in *Proc. 3rd IAPR-TC15 Workshop Graph-Based Representations in Pattern Recognition*, 2001, pp. 125–136.
152. A. Torsello and E. R. Hancock, Learning structural variations in shock trees, in *Proc. Joint IAPR Int. Workshops SSPR and SPR*, 2002, pp. 113–122.
153. J. Triesch and C. von der Malsburg, A system for person-independent hand posture recognition against complex backgrounds, *IEEE Trans. Patt. Anal. Mach. Intell.* **23** (2001) 1449–1453.
154. W. H. Tsai and K. S. Fu, Error-correcting isomorphisms of attributed relational graphs for pattern analysis, *IEEE Trans. Syst. Man Cybern.* **9** (1979) 757–768.
155. W. H. Tsai and K. S. Fu, Subgraph error-correcting isomorphisms for syntactic pattern recognition, *IEEE Trans. Syst. Man Cybern.* **13** (1983) 48–61.
156. J. R. Ullman, An algorithm for subgraph isomorphism, *J. Assoc. Comput. Mach.* **23** (1976) 31–42.
157. S. Umeyama, An eigendecomposition approach to weighted graph matching problems, *IEEE Trans. Patt. Anal. Mach. Intell.* **10** (1988) 695–703.
158. G. Valiente, An efficient bottom-up distance between trees, in *Proc. Eighth Int. Symp. String Processing and Information Retrieval*, 2001, pp. 212–219.
159. B. J. van Wyk and M. A. van Wyk, Non-Bayesian graph matching without explicit compatibility calculations, in *Proc. Joint IAPR Int. Workshops SSPR and SPR*, 2002, pp. 74–82.
160. B. J. van Wyk, M. A. van Wyk and H. E. Hanrahan, Successive projection graph matching, in *Proc. Joint IAPR Int. Workshops SSPR and SPR*, 2002, pp. 263–271.
161. M. A. van Wyk, T. S. Durrani and B. J. van Wyk, A RKHS interpolator-based graph matching algorithm, *IEEE Trans. Patt. Anal. Mach. Intell.* **24** (2002) 988–995.
162. W. D. Wallis, P. Shoubridge, M. Kraetz and D. Ray, Graph distances using graph union, *Patt. Recogn. Lett.* **22** (2001) 701–704.
163. J. T. L. Wang, K. Zhang and G. W. Chirn, The approximate graph matching problem, in *Proc. 12th IAPR Int. Pattern Recognition, Conf. B*, 1994, pp. 284–288.
164. C. Wang and K. Abe, Region correspondence by inexact attributed planar graph matching, in *Proc. Fifth Int. Conf. Computer Vision*, 1995, pp. 440–447.
165. M. Wang, Y. Iwai and M. Yachida, Expression recognition from time-sequential facial images by use of expression change model, in *Proc. Third IEEE Int. Conf. Automatic Face and Gesture Recognition*, 1998, pp. 354–359.
166. Y. K. Wang, K. C. Fan and J. T. Horng, Genetic-based search for error-correcting graph isomorphism, *IEEE Trans. Syst. Man Cybern.* **B27** (1997) 589–597.
167. R. C. Wilson and E. R. Hancock, Structural matching by discrete relaxation, *IEEE Trans. Patt. Anal. Mach. Intell.* **19** (1997) 634–648.
168. M. L. Williams, R. C. Wilson and E. R. Hancock, Deterministic search for relational graph matching, *Patt. Recogn.* **32** (1999) 1255–1271.
169. R. C. Wilson and E. R. Hancock, Graph matching with hierarchical discrete relaxation, *Patt. Recogn. Lett.* **20** (1999) 1041– 1052.
170. L. Wiskott, J. M. Fellous, N. Kruger and C. von der Malsburg, Face recognition by elastic bunch graph matching, *IEEE Trans. Patt. Anal. Mach. Intell.* **19** (1997) 775–779.

171. L. Wiskott, Phantom faces for face analysis, in *Proc. Int. Conf. Image Processing*, 1997, pp. 308–311.
172. A. K. C. Wong, M. You and S. C. Chan, An algorithm for graph optimal monomorphism, *IEEE Trans. Syst. Man Cybern.* **20** (1990) 628–638.
173. L. Xu and I. King, A PCA approach for fast retrieval of structural patterns in attributed graphs, *IEEE Trans. Syst. Man Cybern.* **B31** (2001) 812–817.
174. H. Zhang and H. Yan, Graphic matching based on constrained Voronoi diagrams, in *Proc. Fifth Int. Symp. Signal Processing and Its Applications*, 1999, pp. 431–434.

Donatello Conte graduated in computer engineering from the University "Federico II" of Naples, Italy. He is currently a Ph.D. student from the University of Salerno, Italy. He is a member of the IAPR Technical Committee on Graph-Based Representations (TC15).

His research interests concern graph matching, real-time video analysis and intelligent video surveillance.

Carlo Sansone received a Laurea degree in electronic engineering in 1993, and a Ph.D. in electronic and computer engineering in 1997, both from the University of Naples "Federico II", Italy. From November 2002, he is an Associate Professor of computer science at the Dipartimento di Informatica e Sistemistica of the University of Naples "Federico II".

Professor Sansone has authored over 50 research papers in international journals and conference proceedings. He is also co-editor of a special issue on *Graph-Based Representations in Pattern Recognition* published in the *Pattern Recognition Letters* journal.

He is active in the areas of pattern recognition, graph matching, and video and audio analysis. In particular, his research activities focus on classification techniques, both statistical and structural, exact and inexact graph matching, multi-expert classification and video segmentation.

Pasquale Foggia received a Laurea degree in computer engineering in 1995, and a Ph.D. in electronic and computer engineering in 1999, both from the University "Federico II" of Naples, Italy. Currently, he is an Associate Professor of computer science at the Dipartimento di Informatica e Sistemistica of the University of Naples "Federico II".

Professor Foggia is author of more than 35 research papers published in international journals and conference proceedings. He has been also co-editor of a special issue of the *Pattern Recognition Letters* journal on *Graph-Based Representations in Pattern Recognition*. Since 2002 he has been the secretary of the IAPR Technical Committee on Graph-Based Representations (TC15).

His interests cover the areas of pattern recognition, graph-based representations, graph matching, machine learning, real-time video analysis, intelligent video-surveillance and traffic monitoring.

Mario Vento received the Laurea degree in electronic engineering cum laude in 1984, and a Ph.D. degree in electronic and computer engineering in 1988, both from University of Naples "Federico II". Currently he is a Full Professor of computer science and artificial intelligence at the University of Salerno.

He has authored over 100 research papers in international journals and conference proceedings. Dr. Vento is a member of IAPR, chairman of IAPR Technical Committee on *Graph-Based Representations* (TC15) and secretary of the Italian Association on Pattern Recognition (GIRPR), affiliated to IAPR.

His research interests are in artificial intelligence, pattern recognition and machine learning. He is especially dedicated to classification techniques, contributing to neural network theory, statistical learning, graph matching and multi-expert classification.