Chapter 2

The graph matching problem

> *'Imagination is more important than knowledge. Knowledge is limited. Imagination encircles the world.'*
>
> *Albert Einstein*

This chapter explains the graph matching problem in detail. We first introduce some notation and terminology. Next, a classification of the different graph matching types is presented: this PhD thesis concentrates on inexact graph matching problems, but this chapter summarizes other types of graph matching too. The complexity of the different graph matching problems is also analyzed. Finally, the state of the art is presented. The amount of this type of work shows that the interest on the field is increasing with the years.

2.1 Basic notation and terminology

A graph $G = (V, E)$ in its basic form is composed of vertices and edges. V is the set of vertices (also called *nodes* or *points*) and $E \subset V \times V$ (also defined as $E \subset [V]^2$ in the literature) is the set of edges (also known as *arcs* or *lines*) of graph G. The difference between a graph G and its set of vertices V is not always made strictly, and commonly a vertex u is said to be in G when it should be said to be in V.

The *order* (or *size*) of a graph G is defined as the number of vertices of G and it is represented as $|V|$ and the number of edges as $|E|$[1].

If two vertices in G, say $u, v \in V$, are connected by an edge $e \in E$, this is denoted by $e = (u, v)$ and the two vertices are said to be *adjacent* or *neighbors*. Edges are said to be undirected when they have no direction, and a graph G containing only such types of graphs is called *undirected*. When all edges have directions and therefore (u, v) and (v, u) can be distinguished, the graph is said to be *directed*. Usually, the term *arc* is used when the graph is directed, and the term *edge* is used when it is undirected. In this dissertation we will mainly use directed graphs, but graph matching can also be applied to undirected ones[2]. In addition, a directed graph $G = (V, E)$ is called *complete* when there is always an edge $(u, u') \in E = V \times V$ between any two vertices u, u' in the graph.

[1] In some references in the literature the number of vertices and edges are also represented by $|G|$ and $||G||$ respectively.

[2] In this thesis we will use the terms *vertex* and *edge* for graphs representing knowledge. Later in Chapter 4 probabilistic graphical models such as Bayesian and Gaussian networks are introduced, and the terms *node* and *arc* will be used for these in order to distinguish them.

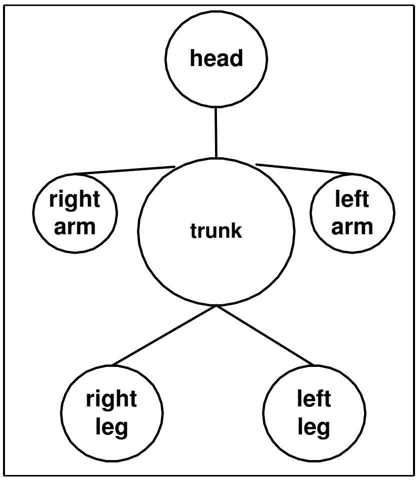

Figure 2.1: Illustration of how the physical parts of a human body can be represented using a graph.

Graph vertices and edges can also contain information. When this information is a simple label (i.e. a name or number) the graph is called *labelled graph.* Other times, vertices and edges contain some more information. These are called vertex and edge *attributes*, and the graph is called *attributed graph.* More usually, this concept is further specified by distinguishing between *vertex-attributed* (or *weighted graphs*) and *edge-attributed* graphs[3].

A *path* between any two vertices $u, u' \in V$ is a non-empty sequence of k different vertices $< v_0, v_1, \ldots, v_k >$ where $u = v_0, u' = v_k$ and $(v_{i-1}, v_i) \in E,\ i = 1, 2, \ldots, k.$ Finally, a graph G is said to be *acyclic* when there are no cycles between its edges, independently of whether the graph G is directed or not.

2.2 Definition and classification of graph matching problems

Many fields such as computer vision, scene analysis, chemistry and molecular biology have applications in which images have to be processed and some regions have to be searched for and identified. When this processing is to be performed by a computer automatically without the assistance of a human expert, a useful way of representing the knowledge is by using graphs. Graphs have been proved as an effective way of representing objects [Eshera and Fu, 1986].

When using graphs to represent objects or images, vertices usually represent regions (or features) of the object or images, and edges between them represent the relations between regions. As an example, we can use a graph to represent a person using the graph shown in Figure 2.1: here all the main physical parts that one expects in a photograph or drawing of a person are shown in the form of vertices in a graph, while edges represent adjacency between the vertices. In this work, we will consider model-based pattern recognition problems, where the model is represented as a graph (the model graph, G_M), and another graph (the data graph, G_D) represents the image where recognition has to be performed. The latter graph is built from a segmentation of the image into regions. The graph in Figure 2.1 could serve as the model graph in a graph matching problem.

Similar graphs can be used for representing objects or general knowledge, and they can be

[3] Attributed graphs are also called *labelled graphs* in some references, and therefore these definitions are also known as *vertex-labelled* and *edge-labelled* graphs.

either directed or undirected. When edges are undirected, they simply indicate the existence of a relation between two vertices. On the other hand, directed edges are used when relations between vertices are considered in a not symmetric way. Note that the graph in Figure 2.1 is undirected, and therefore the attributes on each edge (u, u') are not specified to be from u to u' or vice-versa.

2.2.1 Exact and inexact graph matching

In model-based pattern recognition problems, given two graphs –the model graph G_M and the data graph G_D– the procedure of comparing them involves to check whether they are similar or not. Generally speaking, we can state the graph matching problem as follows: Given two graphs $G_M = (V_M, E_M)$ and $G_D = (V_D, E_D)$, with $|V_M| = |V_D|$, the problem is to find a one-to-one mapping $f : V_D \to V_M$ such that $(u, v) \in E_D$ iff $(f(u), f(v)) \in E_M$. When such a mapping f exists, this is called an isomorphism, and G_D is said to be isomorphic to G_M. This type of problems is said to be *exact graph matching.*

The term *inexact* applied to some graph matching problems means that it is not possible to find an isomorphism between the two graphs to be matched. This is the case when the number of vertices is different in both the model and data graphs[4]. This may be due to the schematic aspect of the model and the difficulty to segment accurately the image into meaningful entities. Therefore, in these cases no isomorphism can be expected between both graphs, and the graph matching problem does not consist in searching for the *exact* way of matching vertices of a graph with vertices of the other, but in finding the *best* matching between them. This leads to a class of problems known as inexact graph matching. In that case, the matching aims at finding a non-bijective correspondence between a data graph and a model graph. In the following we will assume $|V_M| < |V_D|$.

The interest of inexact graph matching has been recently increased in the last years due to the application of computer vision to areas such as cartography, character recognition, and medicine. In these areas, automatic segmentation of images results in an over-segmentation and therefore in the data graph containing more vertices than the model graph. That is why applications on these areas do usually require inexact graph matching techniques. In cartography, the typical example is when a graph is used to represent the knowledge extracted from of a map storing all the features. The matching with an image consists in identifying structures in the image with the help of the map. In character recognition, a model in the form of a graph is generated for each character and the objective is to find which is the model that best suits the analyzed image of a character. In medical images, graphs can be used to represent an anatomical atlas. As a concrete case of the latter, in brain imaging internal brain structures can be recognized with the help of a graph where each vertex represents a brain structure in the atlas, and edges represent spatial relationships between these structures. In the case of the recognition of human facial features from images the regions in the model represent each of the features to be recognized such as mouth, eyes and eyebrows. Experiments carried out in this thesis and presented in Chapter 7 are focused on these two real problems.

[4] A graph matching problem can be considered to be inexact when both graphs to be matched do not contain the same number of vertices and edges. However, it is important to note that in the case of some attributed graph matching problems, the fact of even having the same number of vertices and edges does not imply the existence of an isomorphism, and in the latter case that would also be an inexact graph matching problem. In this PhD thesis we will not consider the latter type of problems when referring to *inexact graph matching.*

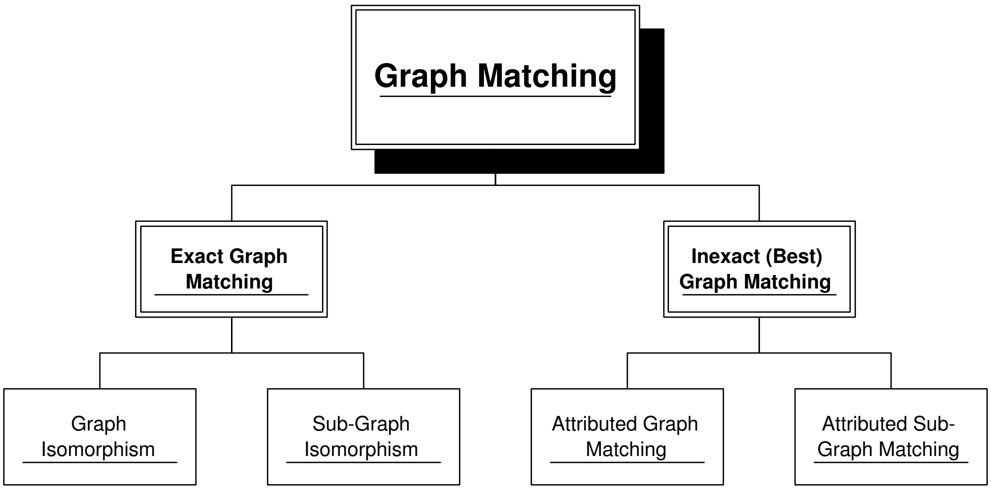
Graph Matching
Exact Graph Matching
Inexact (Best) Graph Matching
Graph Isomorphism
Sub-Graph Isomorphism
Attributed Graph Matching
Attributed Sub-Graph Matching

Figure 2.2: Classification of all the graph matching types into two main classes: exact graph matching and inexact graph matching (in which the best among all the possible non necessarily bijective matchings has to be found).

The best correspondence of a graph matching problem is defined as the optimum of some objective function which measures the similarity between matched vertices and edges. This objective function is also called *fitness function*[5].

In an *inexact graph matching problem* such as the ones described as examples, since we have $|V_M| < |V_D|$, the goal is to find a mapping $f' : V_D \rightarrow V_M$ such that $(u, v) \in E_D$ iff $(f(u), f(v)) \in E_M$. This corresponds to the search for a small graph within a big one. An important sub-type of these problems are *sub-graph matching problems*, in which we have two graphs $G = (V, E)$ and $G' = (V', E')$, where $V' \subseteq V$ and $E' \subseteq E$, and in this case the aim if to find a mapping $f' : V' \rightarrow V$ such that $(u, v) \in E'$ iff $(f(u), f(v)) \in E$. When such a mapping exists, this is called a subgraph matching or subgraph isomorphism.

Exact and inexact graph matching are the terms that we will use in this thesis to differentiate these two basic types of graph matching problems. However, in the literature this type of graph matching problems are also called *isomorphic* and *homomorphic* graph matching problems respectively.

2.2.2 Graph matching using dummy vertices

In some inexact graph matching problems, the problem is still to find a one-to-one, but with the exception of some vertices in the data graph which have no correspondence at all. Real graph matching problem examples of the latter case can be found in [Finch et al., 1998b]. Similar examples can be found for medical applications, as in case of pathologies for instance. This case also happens when the over-segmentation procedure used for the data graph construction has been performed automatically and many regions appear to be components not present in the model or they are simply part of the background.

Examples can be found in the case of recognizing human structures in brain images. The anatomic atlas that the model graph represents is designed according to a healthy brain. However, if the data image that has to be recognized contains pathologies such as a tumor,

[5] In the literature fitness functions also receive the name of *energy functions*, and due to this graph matching algorithms are also regarded as *energy minimization algorithms*.

the graph matching procedure will not be able to match the vertices corresponding to these objects satisfactorily to any of the brain regions to be recognized. Similar examples can be found in satellite images for instance, when the model graph has been extracted from an image obtained some years before the actual photograph to be analyzed and new roads and buildings are present on the place.

More formally, given two graphs $G_M = (V_M, E_M)$ and $G_D = (V_D, E_D)$, the problem consists in searching for a homomorphism $h : V_D \rightarrow V_M \bigcup\{\emptyset\}$, where $\emptyset$ represents the null value, meaning that when for a vertex $a \in V_D$ we have $h(a) = \emptyset$ there is no correspondence in the model graph for vertex a in the data graph. This value $\emptyset$ is known in the literature as the *null vertex* or the *dummy vertex*.

Note that in these cases the fitness function that measures the fitness of the homomorphism h has to be designed taking into account that it should encourage the graph matching algorithm to reduce the number of vertices $a_1, a_2, \ldots, a_n \in V_D$ satisfying the condition $h(a_i) = \emptyset, i = 1, 2, \ldots, n$. In other words, nothing avoids that the homomorphism h for which $\forall a_i \in V_D$ $h(a_i) = \emptyset$ is valid, but this solution would not represent any satisfactory result.

It is important to note that the use of the dummy vertex can be regarded as an additional vertex only for the model graph. However, in some problems the equivalent opposite case is also happening: using the same example of the human brain, the graph matching algorithm could face the case of having to recognize the data image of a patient that has undergone a lobotomy, and therefore a lobe is missing in the image. In this case, the graph matching problem has the particularity that there would not be a correspondence in the optimal solution for all the model vertices, but it does require the use of dummy vertices. This aspect is also common in other image recognition problems such as the aerial image recognition one, in which ancien buildings and factories can leave space in a newer photograph to new parks or roads.

2.2.3 Graph matching allowing more than one correspondence per vertex

Some other graph matching problems allow many-to-many matches, that is, given two graphs $G_M = (V_M, E_M)$ and $G_D = (V_D, E_D)$, the problem consists on searching for a homomorphism $f : V_D \rightarrow W$ where $W \in \mathcal{P}(V_M)\backslash\{\emptyset\}$ and $W \subseteq V_M$. In case of using also dummy vertices, W can take the value $\emptyset$, and therefore $W \in \mathcal{P}(V_M)$.

This type of graph matching problems are more difficult to solve, as the complexity of the search for the best homomorphism has much more combinations and therefore the search space of the graph matching algorithm is much bigger. These types of graph matching problems make sense when the segmentation step in the image does not satisfy the condition of having segmented all the regions completely, and therefore some of the automatically segmented regions are at the same time part of two or more model regions.

An important difficulty in graph matching problems that allow more than a single matching per data vertex is the design of a fitness function to measure the quality of each of the possible homomorphisms. Again, the number of model vertices matched to each single data vertex needs to be kept as low as possible so that the graph matching algorithm is forced to return more concrete results. These aspects are very dependent on each particular problem.

2.3 Complexity of graph matching

Graph matching is considered to be one of the most complex problems in object recognition in computer vision [Bienenstock and von der Malsburg, 1987]. Its complexity is due to its combinatorial nature. Following the classification of graph matching problems explained in Section 2.2, as the nature of each of them is different, we will analyze their complexity separately.

2.3.1 Exact graph matching: graph isomorphism

This whole category of graph matching problems has not yet been classified within a particular type of complexity such as P or NP-complete. Some papers in the literature tried to prove its NP-completeness when the two graphs to be matched are of particular types or satisfy some particular constraints [Basin, 1994, Garey and Johnson, 1979], but it still remains to be proved that the complexity of the whole type remains within the NP-completeness at most.

On the other hand, for some types of graphs the complexity of the graph isomorphism problem has been proved to be of polynomial type. An example is the graph isomorphism of planar graphs, which has been proven in [Hopcroft and Wong, 1974] to be of polynomial complexity, although the cost of the leading constant also appears to be quite large.

As a result, it can be said that this issue remains as an interesting open theoretical problem, although it also encourages researchers to try to find polynomial time solutions for this type of graph matching problems.

2.3.2 Exact sub-graph matching: sub-graph isomorphism

This particular type of graph matching problems has been proven to be NP-complete [Garey and Johnson, 1979]. However, some specific types of graphs can also have a lower complexity. For instance, the particular case in which the big graph is a forest and the small one to be matched is a tree has been shown to be of a polynomial complexity [Garey and Johnson, 1979, Reyner, 1977].

2.3.3 Inexact graph matching: graph and sub-graph homomorphisms

In inexact graph matching, where we have $|V_M| \leq |V_D|$, the complexity is proved in [Abdulkader, 1998] to be NP-complete. Similarly, the complexity of the inexact sub-graph problem is equivalent in complexity to the largest common sub-graph problem, which is known to be also NP-complete.

2.4 State of the art in the literature

This section is a review of the literature on graph matching. All the references commented correspond to graph matching problems and methods, although some of them do not correspond with the concrete types of problems addressed in this dissertation and are included for the interested reader so that he can have an idea on the different subjects and groups working nowadays on the field. The interested reader can additionally find in [Jolion and Kropatsch, 1998, Kropatsch and Jolion, 1999, Jolion et al., 2001] works on practically all aspects discussed on this chapter: subgraph transformations for inexact matching, isomorphism between strong fuzzy relational graphs, a framework for region segmentation algorithms, image sequence segmentation, image analysis with a graph-based parallel computing model, and so on.

Even if this chapter has been focused on the graph matching types that can be found in the literature, this section focuses also in other related aspects such as the image processing techniques, the way of building the attributes, the different algorithms and formalizations of these problems, and their applications. The outline of this section is as follows: Section 2.4.1 reviews image processing techniques for graph matching purposes, as well as the generation of the graphs from images. Section 2.4.2 refers to the different ways of representing the information in graphs, and records references on geometrical properties of graph edit distances, and graph metrics. Finally, Sections 2.4.3, 2.4.4, 2.4.5, 2.4.6, and 2.4.7 explain the different graph matching problems that have been proposed based on genetic algorithms, probability theory-based approaches, decision trees, neural networks, and clustering techniques respectively. Each section comments examples of applications in which these methods have been applied. Finally, Section 2.4.8 discusses the advantages and disadvantages of the main approaches analyzed in the previous sections.

2.4.1 Image processing and graph construction techniques for graph matching

In most of the graph matching problems for image recognition, vertices in graphs represent regions of images, and the division in regions is the result of a segmentation procedure. This segmentation procedure is sometimes performed with the aid of an expert, but other times automatic segmentation and graph construction techniques are applied to create the graphs that are to be matched.

Attributed graph representation using fuzzy set theory

Fuzzy set theory has been used in the literature as a means to create vertex and edge attributes to be applied to graph matching. At a theoretical level there are many key works for these type of attributes, such as the representation of distances in images [Bloch, 1999b] and the representation of relative positions between objects [Bloch, 1999a]. Based on this idea, we can find many references in the literature using this type of attributes for inexact graph matching [Perchant et al., 1999, Perchant and Bloch, 1999], and for sub-graph matching [Hwan, 2001].

Fuzzy attributed graph models are proposed for very different image type representations, such as fingerprint verification [Fan et al., 2000] as a mechanism to structure the knowledge captured in conceptual models, and face detection from color images using a fuzzy pattern matching method [Wu et al., 1999].

Morphological graph matching and elastic graph matching

Elastic graph matching is a graph representation and matching technique that takes into account the possible deformation of objects to be recognized. Elastic graph matching usually consists of two consecutive steps, namely a matching with a rigid grid, followed by a deformation of the grid, which is actually the elastic part. The deformation step is introduced in order to allow for some deformation, rotation, and scaling of the object to be matched.

An example of the application of this type of representation is the identification and tracking of cyclones [Lee and Liu, 1999, 2000]. In the past decades, satellite interpretation was one of the vital methods for the determination of weather patterns all over the world, especially for the identification of severe weather patterns such as tropical cyclones as well as their intensity. Due to the high variation and complexity of cloud activities for the tropical cyclone patterns, meteorological analysts all over the world so far are still relying on subjective human justification for cyclone identification purposes. Elastic graph matching techniques have been proposed for automatic detection of this problem.

Another usual application of this technique is the authentication of human faces, where the deformation patterns of the face are represented as graph attributes [Duc et al., 1999, Kotropoulos et al., 2000a,b, Tefas et al., 2002]. The different facial expression and the rotation patterns of the human head are intended to be represented in the form of graphs. Wavelet transforms are also used for creating the elastic face graph model, such as in [Ma and Xiaoou, 2001] where the face graph model is constructed using the discrete wavelet transform, and in [Duc et al., 1997, Lyons et al., 1999, Wiskott, 1999], where different 2D Gabor wavelet representation are used. Another related problem is the recognition of facial regions such as mouth and nose, which is also solved using these methods [Herpers and Sommer, 1998].

Morphological graph matching applies hyperplanes or deformable spline-based models to the skeleton of non-rigid discrete objects [di Ruberto and Dempster, 2001, Ifrah, 1997, Kumar et al., 2001, Rangarajan et al., 2001, Tefas et al., 2001], the graph being built from skeleton characteristic points. An approach to interactively define features and subsequently recognize parts regarding shape features using a sub-graph matching algorithm is proposed in [Sonthi, 1997]. An illustrative example of mathematical morphology applied to curve fitting and not to skeleton points can be found in [Bakircioglu et al., 1998], where curves on brain surfaces are matched by defining distances between curves regarding their speed, curvature, and torsion. Also, in [Dorai, 1996] another framework for representation and recognition of 3D free-form objects is shown, where the surface representation scheme describes an object concisely in terms of maximal surface patches of constant shape index. Finally, the application of planar surfaces to the modelling the data graph can be solved with probabilistic relaxation as shown in [Branca et al., 2000].

Another geometrical approach is introduced in [Shams, 1999], using generalized cylinders called *geons* as visual primitives to represent object models. Geons constitute a structural level intermediate between local features and whole objects, used here as a basis for powerful generalization between different view points. This work makes use of graph matching techniques and cross-correlation methods to find 2D projections of geons as partial matches between several images.

Examples of other applications of these techniques are: symmetry-based indexing of image databases [Sharvit et al., 1998], shape recognition from large image libraries [Huet and Hancock, 1999], recognition of shape features via multiple geometric abstractions [Sonthi, 1997], shape recognition using probability theory-based methods [Khoo and Suganthan, 2002, Shao and Kittler, 1999], structural matching by discrete relaxation [Wilson and Hancock, 1997], representation and recognition of 3D free-form objects [Cheng et al., 2001, Dorai, 1996], and hand posture recognition [Triesch and von der Malsburg, 2001].

Multiple graph matching

Some graph matching problems are based on the idea of having more than one model, and on performing graph matching to a database of models so that the model that best approaches the characteristics of the data graph is selected. Therefore, the aim here is to recognize a model rather than going deeply to recognize each of the segments of the data image. An illustrative example is found in [Messmer and Bunke, 1999] in which decision trees are used for graph and subgraph isomorphism detection in order to match a graph to the best of a dictionary of graphs.

Due to the computation cost of using databases and to perform image processing on each, several references can be found about the subject of how to perform efficiently graph matching to all the models. For instance, in [Berretti et al., 2001] the usage of an indexing metric to organize large archives of graph models and for (sub)graph error correcting isomorphism problem is proposed. [Sharvit et al., 1998] proposes an approach for indexing pictorial databases based on a characterization of the symmetry in edge maps.

From a more theoretical point of view, [Williams et al., 1997] describes the development of a Bayesian framework for multiple graph matching. The starting point of this proposal is the Bayesian consistency measure developed by Wilson and Hancock [Wilson and Hancock, 1996] which is generalized from matching graph pairs to multiple graphs. In [Huet and Hancock, 1999] a graph-matching technique for recognizing line-pattern shapes in large image databases is described. The methodological contribution of the paper is to develop a Bayesian matching algorithm that uses edge-consistency and vertex attribute similarity. Recognition is performed by selecting from the database the image with the largest a posteriori probability.

One of the problems faced most commonly as multiple graph matching is the human face recognition [Kotropoulos et al., 2000a,b]. The particularity of [Mariani, 2000] is that the images of the database are all taken from the same person viewed from different angles, which aim is to estimate the orientation of the head. Following the same problem, [Hancock et al., 1998] compares the performance of two computer-based face database identification systems based on human ratings of similarity and distinctiveness, and human memory performance. Multiple graph matching has also been applied to many other problems such as the comparison of saliency map graphs [Shokoufandeh et al., 1999].

2.4.2 Distance measures, conceptual graphs, and graph edit distances and metrics

The graph edit distance between two graphs is defined as the number of modifications that one has to undertake to arrive from one graph to be the other. The distance between two graphs is defined as the weighted sum of the costs of edit operations (insert, delete, and relabel the vertices and edges) to transform one graph to the other. The fact of applying these concepts and removing vertices or edges in graphs is analyzed in many works, as removal will lead to smaller graphs and therefore the graph matching problem can be reduced in complexity.

[Fernandez and Valiente, 2001] proposes a way of representing attributed relational graphs, the maximum common subgraph and the minimum common supergraph of two graphs by means of simple constructions, which allow to obtain the maximum common subgraph from the minimum common supergraph, and vice versa. A distance measure between pairs of circular edges and relations among them is introduced in [Foggia et al., 1999]. This measure is to be applied in domains with high variability in the shape of the visual patterns (i.e. where a structural approach is particularly useful). In [Bunke, 1997] the relation between graph edit distance and the maximum common subgraph is analyzed, showing that under a fitness function introduced in this paper, graph edit distance computation is equivalent to solving the maximum common subgraph problem. Other related theoretical subjects are addressed in [Wang et al., 1995], in which two variants of the approximate graph matching problem are considered and analyzed: the computation of the distance between the graphs G_M and G_D, and the definition of the minimum distance between G_M and G_D

when subgraphs can be freely removed from G_D.

Other interesting approach of spectral type to solve graph matching translate this problem to the search of the maximal clique in the association graph. Examples on this are [Doob et al., 1980, Massaro and Pelillo, 2001].

As steps forward in this field, in [Myers et al., 2000] a framework for comparing and matching corrupted relational graphs is introduced. The authors develop the idea of the *edit-distance* and show how this can be used to model the probability distribution for structural errors in the graph-matching problem. This probability distribution is aimed at locating matches using maximum a posteriori label updates. The resulting graph-matching algorithm is compared with the one of [Wilson and Hancock, 1996], where the use of edit-distances is presented as an alternative to the exhaustive compilation of label dictionaries. In addition, [Ing-Sheen and Kuo-Chin, 2001] presents a region-based color image retrieval system using geometric properties, in which relational distance graph matching between two spatial relational graphs is performed to find the best matches with the minimum relational distance. Then, shape matching is applied to obtain the best match with the minimum geometric distance. Finally, in [Jolion, 2003] a new model for graph decimation is proposed with the aim of reducing the computational complexity of algorithms used for clustering, matching, feature extraction and so on.

Many useful applications of these techniques can be found in the literature. An illustrative example is shown in [Haris et al., 1999], where a method for extraction and labelling of the coronary arterial tree is proposed. In [Geusebroek et al., 1999], distance graph matching is applied to the segmentation of tissues, by basing the characterization of tissues on the topographical relationship between the cells. The neighborhood of each cell in the tissue is modelled by the distances to the surrounding cells, and comparison with an example or prototype neighborhood reveals topographical similarity between tissue and model. Optimal video clip ordering on a network in order to reduce computer network traffic in multimedia equipments is solved in [Ng and Shum, 1998] by representing the network as a graph and using graph edit distances and graph matching. The recognition of handwritten digits coming from a standard character database is also solved using this method [Foggia et al., 1999]. Chinese character recognition regarding stroke order and character recognition by analyzing graph distances have been proposed [Liu, 1997a]. Other work applying this approach involving clustering with distance measures is shown in [Gold et al., 1999].

Error correction graph matching

In error-correcting graph matching (or *error-tolerant graph matching* as it is also called) one considers a set of graph edit operations, and defines the edit distance of two graphs G_1 and G_2 as the shortest (or least cost) sequence of edit operations that transform G_1 into G_2. Error-correcting graph matching is a powerful concept that has various applications in pattern recognition and machine vision, and its application is focused on distorted inputs. It constitutes a different approach very similar to other graph matching techniques. In [Bunke and Shearer, 1998] this topic is addressed and a new distance measure on graphs that does not require any particular edit operations is proposed. This measure is based on the maximal common subgraph of two graphs. A general formulation for error-correcting subgraph isomorphism algorithms is presented in [Llados et al., 2001] in terms of adjacency graphs, and [Bunke, 1999] presents a study on the influence of the definition of fitness functions for error correcting graph matching.which reveals guidelines for defining fitness functions for optimization algorithms in error correcting graph matching. In addition, in [Messmer and

Bunke, 1998b] an algorithm for error-correcting subgraph isomorphism detection from a set of model graphs to an unknown input graph is introduced.

Applications of this technique are manyfold. [Fuchs and Men, 2000] present a procedure to compute error-correcting subgraph isomorphism in order to encode a priory knowledge for application to 3D reconstruction of buildings for cartography. Other examples are: relating error-correction and decision trees [Messmer and Bunke, 1998a], related to indexing of graph models [Berretti et al., 2001], crystallographic map interpretation [Oldfield, 2002], and construction of specific frameworks for error-tolerant graph matching [Bunke, 1998].

Conceptual graphs

Conceptual graphs have been used to model knowledge representations since their introduction in the early 80's. The formalism of conceptual graphs introduced in [Sowa, 1984] is a flexible and consistent knowledge representation with a well-defined theoretical basis. Moreover, simple conceptual graphs are considered as the kernel of most knowledge representation formalisms built upon Sowa's model. This formalism can capture semantics in the representation of data, and it offers some useful constructs which makes it a likely platform for a knowledge-based system. An extension of this concept to graph matching is introduced in [Baget and Mugnier, 2002], where reasoning in Sowa's model can be expressed by a graph homomorphism called projection. This paper presents a family of extensions of this mode, based on rules and constraints, keeping graph homomorphism as the basic operation. The use of conceptual graphs is also extended to some other works [Chen, 1996, Emami, 1997, Finch et al., 1997, Lai et al., 1999].

Apart from this type of graphs, graph matching has also been proposed, from both a theoretical or a practical view point, in combination with: matching graphs [Eroh and Schultz, 1998], minimal condition subgraphs [Gao and Shah, 1998], finite graphs [Bacik, 2001], weighted mean of a pair of graphs [Bunke and Gunter, 2001], median graphs [Jiang et al., 2001], and decomposition approaches [Messmer and Bunke, 2000]. Furthermore, different ways of representing patterns are analyzed in terms of symbolic data structures such as strings, trees, and graphs in [Bunke, 2001].

2.4.3 Graph matching using genetic algorithms

The fact of formulating complex graph matching problems as combinatorial optimization ones is not novel, and many references applying different techniques in this field can be found in the literature. Genetic algorithms are just an example of this.

Some of the works in the literature concentrate firstly on the type of crossover and mutation operators that are most suitable for graph matching problems. [Khoo and Suganthan, 2002] present a comparison between different genetic operators, and compares the performance of genetic algorithms when using two different types of individual representation. Furthermore, in [Singh and Chaudhury, 1997] an evolutive algorithm without crossover operators suitable for genetic algorithms is presented in order to obtain faster convergence to the solution. The authors also illustrate methods to parallelize their algorithm.

Another important aspect concerns the use of Bayesian measures. [Cross et al., 1997] describe a framework for performing relational graph matching using genetic search. The authors cast the optimization process into a Bayesian framework using the fitness measure previously introduced in [Wilson and Hancock, 1996]. This study shows that such Bayesian consistency measure could be efficiently optimized using a hybrid genetic search procedure that incorporates a local search strategy using a hill-climbing step. The authors demonstrate analytically that this hill-climbing step accelerates convergence significantly. This idea is extended in [Cross et al., 2000], which is also a convergence analysis for the problem of attributed graph matching using genetic search. [Myers and Hancock, 2001] constitutes an extension of the Bayesian matching framework introduced in [Wilson and Hancock, 1997] by taking into account the cases with ambiguities in feature measurements. This paper also develops an evolutionary optimization framework that is applied on a genetic algorithm adapted for this search. Finally, the main idea in [Myers and Hancock, 2000] is that attributed graph matching problems (or *consistent labelling problems* as called by the authors) usually have more than one valid and satisfactory solution, and the aim of the authors is to propose a method to obtain different solutions at the same time during the genetic search, using an appropriately modified genetic algorithm.

As an example of real applications, works on the field on the human brain images recognition example can be found in [Boeres et al., 1999, Boeres, 2002, Perchant et al., 1999, Perchant and Bloch, 1999].

2.4.4 Graph matching using techniques based on probability theory

We can find in the literature many techniques applying probability theory to graph matching problems. A review on general purpose probabilistic graph matching can be found in [Farmer, 1999], where different types of probabilistic graphs, different techniques for their manipulation, and fitness functions appropriated to use for these problems are presented.

The first works applying probability theory to graph matching [Hancock and Kittler, 1990, Kittler et al., 1993] use an iterative approach using a method called probabilistic relaxation, and only take into account binary relations and assuming a Gaussian error. In these papers the use of binary relations is justified to be enough for defining the whole structure fully. Later a Bayesian perspective is used to account for both unary and binary attributes [Christmas et al., 1995, Gold and Rangarajan, 1996, Wilson and Hancock, 1996, 1997]. Later in [Williams et al., 1999] a comparative study of various deterministic discrete search-strategies for graph-matching is presented, which is based on the previous Bayesian consistency measure reported in [Wilson and Hancock, 1996, 1997]. Tabu search is proposed in this article as a graph matching algorithm. Finally, in [Finch et al., 1998a] a new fitness function is developed for graph matching based on a mixture model that gauges relational consistency using a series of exponential functions of the Hamming distances between graph neighborhoods. The effective neighborhood potentials are associated with the mixture model by identifying the single probability function of zero Kullback-Leibler divergence. This fitness function is simply a weighted sum of graph Hamming distances. Unfortunately, all these works do not consider dependencies between more than two vertices, although an attempt to consider more complex dependencies can be found in [Farmer, 1999] based on a new representation using association probabilistic graphs.

Other papers that can be classified under this section are mentioned next. [Shams et al., 2001] is a comparison between some conventional algorithms for solving attributed graph matching to the mutual information maximization method and an adaptation of multidimensional Gabor wavelet features. [Osman et al., 2000] constitutes an approach to based on modelling human performance for mental representation of objects by a machine learning and matching system based on inductive logic programming and graph matching principles. Finally, in [Liu et al., 2000] a method for recognition of Chinese characters is illustrated where both model and input characters are represented as complete relational graphs. The graph-matching problem is solved with the Hungarian method.

Applying probabilistic relaxation to graph matching

Probabilistic relaxation is also used for solving the graph matching problem when formulated in the Bayesian framework for contextual label assignment [Christmas et al., 1995], and the same idea is applied in [Wilson and Hancock, 1997] combining several popular relational distance measures and an active process of graph-editing. More recently, [Wilson and Hancock, 1999] improves this Bayesian framework for matching hierarchical relational models.

Other articles related to this field are the application of planar surfaces to generate the data graph, and recognition performing graph matching with probabilistic relaxation in [Branca et al., 2000], application to representation and recognition of shapes in objects [Shao and Kittler, 1999], a generalization of the probabilistic relaxation labelling model by applying the relaxation to a graph of labels rather than to a pair of labels combined with random graphs to model scenes [Skomorowski, 1999], and a new relaxation scheme for graph matching in computer vision formulated as a process of eliminating unlikely candidates rather than finding the best match directly [Turner and Austin, 1998]. In the latter, a Bayesian development is used leading to an algorithm which can be implemented on a neural network architecture.

Applying the EM algorithm to graph matching

Another important approach is the EM algorithm. [Cross and Hancock, 1998] and [Finch et al., 1998b] are examples of this, in which two similar EM frameworks are proposed for two different graph matching problems. An algorithm for inexact graph matching that concentrates only on the connectivity structure of the graph and does not draw on vertex or edge attributes (structural graph matching) is proposed in [Luo and Hancock, 2001a]. An extension of the latter work is given in [Luo and Hancock, 2001b]. Also, [Kim and Kim, 2001] present a hierarchical random graph representation for handwritten character modelling in which model parameters of the hierarchical graph are estimated automatically from the training data by EM algorithm and embedded training.

Applying other probability theory methods to graph matching

Other examples of probability theory-based techniques are found in references commented on this thesis, on the following subjects: discrimination of facial regions using simulated annealing [Herpers and Sommer, 1998], multi modal person authentication using probabilistic relaxation [Duc et al., 1997, Mariani, 2000], Bayes theory applied to multiple graph matching [Huet and Hancock, 1999], multiple graph matching using Bayesian inference [Williams et al., 1997], and complex problems in proteonics using simulated annealing [Fariselli and Casadio, 2001].

2.4.5 Applying decision trees to graph matching

Decision trees have also been applied to graph matching. An example of this is [Shearer et al., 2001], in which decision trees are used for solving the largest common subgraph problem instead of applying queries to a database of models. Another example can be found in [Messmer and Bunke, 1998a] where decision trees are applied as a fast algorithm for the computation of error-correcting graph isomorphisms. Decision trees are also been applied to multiple graph matching [Messmer and Bunke, 1999]. The decision tree is created using a set of a priori known model graphs generated from exact subgraph isomorphism detection.

2.4.6 Graph matching using neural networks

Neural networks have also extensively been applied to many graph matching problems. Very different types of neural networks have been tested trying to find the most suitable for each particular graph matching problem. Examples of this are the use of a mean field annealing neural network as a constraint satisfaction network for 3D object recognition [Lyul and Hong, 2002], a congregation of neural networks for the automatic recognition of cortical sulci of human brains in MRi images [Riviere et al., 2000, 2002], hybrid RBF network track mining techniques combined with integrated neural oscillatory elastic graph matching for tropical cyclone identification and tracking [Lee and Liu, 2000], dynamic link matching –i.e. a neural network with dynamically evolving links between a reference model and a data image– for face authentication with Gabor information on deformable graphs [Duc et al., 1999], and a 3D Hopfield neural network for sign language recognition [Huang and Huang, 1998]. On the other hand, in [Rangarajan et al., 1999b] a theoretical study of the convergence properties of a particular neural network is presented.

Other examples of the use of neural networks are: frontal face authentication problems [Kotropoulos et al., 2000a], overlapped shape recognition [Suganthan et al., 1999], recognition of hand printed Chinese characters [Suganthan and Yan, 1998], and a Bayesian development implemented as a neural network architecture [Turner and Austin, 1998].

2.4.7 Graph matching using clustering techniques

The use of clustering techniques has also been applied in graph matching, and many examples can be found in the literature such as [Carcassoni and Hancock, 2001] where clustering techniques are applied for attributed graph-matching. In this paper authors demonstrate how to improve method robustness to structural differences by adopting a hierarchical approach. Another example is [Sanfeliu et al., 2000] where a new type of graphs called *function-described graphs* are defined to represent an ensemble of attributed graphs for structural pattern recognition. The unsupervised synthesis of function-described graphs is studied here in the context of clustering a set of attributed graphs and obtaining a function-described graph model for each cluster. The idea is therefore to take into account the matching for groups of vertices instead of individually.

Other clustering-based approaches have also been applied to automatically recognize form documents [Fan et al., 1998], clustering with distance measures [Gold et al., 1999], automatic satellite interpretation of tropical cyclone patterns [Lee and Liu, 1999], and an eigen decomposition approach [Umeyama, 1988].

2.4.8 Discussion

In this Section, we discuss the pros and cons of the main classes of approaches found in the literature.

The complexity of the graph matching approaches make it very difficult to take into account all types of dependencies between the vertices and edges in the model and data graphs. That is why in most existing frameworks defined for this problem (e.g. Bayesian, morphological and EM frameworks described in the previous sections) only unary and binary relations between vertices are taken into account. The advantage of such simplifications is that the complexity of graph matching is reduced when tackling the original problem, and this allows the use of techniques that require evaluating a lot of individuals through the search for the best solution. On the other hand, these approaches are ignoring many relationships that can be decisive when searching for a satisfactory matching. Due to this, and depending on the type of problem, these simplifications could lead to non satisfactory results.

Regarding the application of genetic algorithms in graph matching problems, the performance that they present is remarkable, specially when they are combined with dedicated frameworks such as Bayesian ones. In addition, the number of evaluations that they require to reach a solution is very low compared to other stochastic heuristic searches. However, one of their drawbacks is that their performance is very dependent on the high number of input parameters that have to be set. This dependence is specially dependent on the type of cross-over and mutation operators selected. The literature presents plenty of different GAs which are more appropriated for particular types of problems, and the user who wants to apply these methods needs to have a lot of experience in order to tune properly the algorithm and obtain satisfactory results. Another important drawback of GAs is the fact that when the fitness function contains many local maxima these algorithms get easily stuck on them. Moreover, in the definitions of similarities proposed for building the fitness functions for GAs, again only unary and binary relationships are taken into account. This simplification is understandable since the number of evaluations that GAs require to reach the optimum solution in such complex problems is high, but important relationships can be missed out.

Regarding the use of neural networks, they have shown in the last years a good performance in solving problems as complex as graph matching ones. The ability of neural networks to take into account particular restrictions of problems has also been proved experimentally. However, regarding the way of representing the knowledge, the main drawback of this approach is that this information is encoded in a kind of *black box* that does not allow easily to infer how the network reasons. Other approaches based on probabilistic structures (such as Bayesian networks) have the ability to express in a format directly comprehensible for researchers how the knowledge is represented.

2.5 Graph matching problem types selected for this thesis

In this thesis, we address the problem of inexact graph matching. This choice has been done taking into account the NP-complete complexity of this problem, which is more difficult to solve than the exact graph matching problem. One of the objectives of this PhD thesis is to show the validity of the EDA paradigm when applied to graph matching problems, and therefore the inexact sub-graph matching category has been chosen. However, as graph and sub-graph inexact matching problems are of equivalent category, we will simply use the term *inexact graph matching* to refer to both of them. In addition, we have set an additional constraint that makes the problem even more restrictive: in order to consider a homomorphism as valid, all the vertices in the model graph will have at least a vertex in the data graph that has been matched to it. Examples on synthetic data and on two real applications are explained in more detail in Chapters 6 and 7.