Attributed Graph Matching using Local Descriptions

Salim Jouili, Salvatore Tabbone

Attributed Graph Matching using Local Descriptions

Salim Jouili and Salvatore Tabbone

LORIA UMR 7503 - University of Nancy 2
BP 239, 54506 Vandoeuvre-lès-Nancy Cedex, France
{salim.jouili,tabbone}@loria.fr

Abstract. In the pattern recognition context, objects can be represented as graphs with attributed nodes and edges involving their relations. Consequently, matching attributed graphs plays an important role in objects recognition. In this paper, a node signatures extraction is combined with an optimal assignment method for matching attributed graphs. In particular, we show how local descriptions are used to define a node-to-node cost in an assignment problem using the Hungarian method. Moreover, we propose a distance formula to compute the distance between attributed graphs. The experiments demonstrate that the newly presented algorithm is well-suited to pattern recognition applications. Compared with well-known methods, our algorithm gives good results for retrieving images.

1 Introduction

Recently, graphs become commonly used as an adequate representations for documents, and many recognition problems can be formulated as an attributed graph matching problem, where nodes of the graphs correspond to local features of the document and edges correspond to relational aspects between features. Therefore, attributed graphs matching imply establishing correspondences between nodes of two graphs as consistently as possible. In the last decades, there have been many researches on defining efficient and fast graph matching algorithms [8]. The major approaches for matching attributed graphs include edit distance minimization [10, 2, 3], spectral approach [7], Bayesian approaches [1], probing technique [6], probabilistic relaxation [13]. According to [11], most of these approaches, the attributed graph matching are implemented as a following 2-steps procedure: Firstly, similarities between every pair of nodes in two graphs, forming a distance matrix, are computed using a predefined measure. Secondly, the matching between nodes is based on the distance matrix by using an approximate algorithm such as the bipartite matching [3]. Therefore, the attributed graph matching problem is mathematically formulated as an assignment problem.

In this paper, we propose a new efficient algorithm for matching and computing the distance between attributed graphs. We introduce a new *vector-based node signature* as a local description in the attributed graph (AG). Each node

is associated with a vector where components are a collection of degrees, the attributes of the node and the incident edge attributes. To compute a distance between two node signatures, we use the Heterogeneous Euclidean Overlap Metric (HEOM) which handles numeric and symbolic attributes. Afterwards, using the node signatures and the HEOM distance, a cost matrix is constructed. The cost matrix describes the matching costs between nodes in two graphs, it is a (n,m) matrix where n and m are the sizes of the two graphs. An element (i,j) in this matrix gives the distance between the ith node signature in the first graph and the jth node signature in the second graph. To find the optimum matching, we consider this problem as an instance of the assignment problem, which can be solved by the Hungarian method [14]. We also introduce a new metric to compute the distance between graphs.

The remainder of this paper is organized as follow: in the next section (§2), local descriptions for graphs and the distance between these local descriptions are described. In the Section 3, the proposed matching algorithm is described and the distance between two graphs is also introduced. This algorithm is used to find correct node-to-node correspondences between two graphs, and to retrieve graphs in data-sets. We have compared our method with the Umeyama method [12] and the Zass's probabilistic method [13] for the matching task and with the BGMEDG [3] for the retrieving task (section 4).

2   Local descriptions of AG

In this paper, we present an algorithm for reducing the problem of graph matching to a bipartite graph matching problem by means of node signatures. We have taken inspiration from literature, to use an assignment-based algorithm for graph matching [3, 9, 18, 19] by making use of a new node signature. To compute the distance between graphs, a framework is proposed in this section to extract node signatures and compute distance between these signatures.

2.1   Node signatures

In the literature, the major part of proposed AG matching algorithms deal with global-based representation of graphs. Then, the graph is handled as one entity which can be only one vector[6], a matrix [7] or a string [15]. In few previous work, the concept of node signature has been introduced in [20, 19, 17], here the node signatures have been computed by making use of spectral approach, decomposition approach and random walks approach. These methods using node signatures descibe the graph locally.

In this paper, we propose local-based descriptions instead of global-based description of graphs. Henceforth, each graph is represented by a set of local descriptions which are related to the node features and used to compute the node-to-node distance. In the following, we denote the local descriptions as node signatures. Contrary to the previous works in the literature, our node signature is a simple vector and computed straightforwardly from the adjacency matrix.

In order to construct a signature for a node in an attributed graph, we use all available information into the graph and related to this node. These information are the node attribute(s), the node degree and the attributes of the incident edges to this node. The collection of these informations should be refined into an adequate structure which can provides distances between different node signatures. In this perspective, we define the node signature as a set composed by three subsets which represent the node attribute, the node degree and the attributes of the incident edges to this node. Given a graph G = (V, E, A) where V is the vertex set, E is the edge set, and A is the attribute set that contains unary attribute a_i (linked to each node n_i) and binary attribute a_{ij} (linked to each edge e_k=(n_i,n_j) ∈E), the node signature is formulated as follows:

$$\mathrm{Ns}(n_i) = \{\{a_i\}, \mathrm{d}^o(n_i), \{a_{ij}\}_{\forall n_j \in E}\}$$

Where $n_i \in$ V, $\{a_i\}$ is the attribute of the node n_i, $\mathrm{d}^o(n_i)$ gives the degree of n_i, and $\{a_{ij}\}$ is the set of the attributes of the incident edges to n_i.

The set of these node signatures (vectors) describing nodes in an attributed graph is a collection of local descriptions. So, local changes of the graph will modify only a subset of vectors while leaving the rest unchanged.

2.2   Distance metric between node signatures

Classically, to determine the similarity between two entities in multidimensional feature space, a distance metric is required. Although several distance metrics have been proposed [22], the most commonly used metrics are suitable only for either symbolic *or* numeric attributes. These include the *Euclidean* and *Manhattan* distance metrics for numeric attributes, and the Overlap distance for symbolic attributes. In our case, the node signature can be expected to encounter a spectrum of different types of attributes including numeric and symbolic data that require more complex metrics. Wilson and al. [22] review a list of well-known metrics based on heterogenous distance function which handle multiple data type. We can classify these metrics into two family. On the one hand, the distances based on the value difference metric (e.g. *Heterogeneous Value Difference Metric*) and on the other hand the Euclidean-based distance (e.g. *Heterogeneous Euclidean Overlap Metric*). The metrics in the first family are only used in the classification context, by introducing class information into the distance formula. Therefore, we use the *Heterogeneous Euclidean Overlap Metric* (HEOM) to compute the distance between two node signatures. The HEOM uses the *overlap* metric for symbolic attributes and the normalized Euclidean distance for numeric attributes. The overall distance between two heterogeneous node signatures $\boldsymbol{i}$ and $\boldsymbol{j}$ is given by the function HEOM($\boldsymbol{i,j}$):

$$HEOM(i,j) = \sqrt{\sum_{a=0}^{A} \delta(i_a, j_a)^2} \tag{1}$$

Here a refers to one attributes of A. And $\delta(i_a, j_a)$ is defined as:

$$\delta(i_a,j_a) = \begin{cases} 1 & \text{if } i_a \text{ or } j_a \text{ are missing} \\ Overlap(i_a,j_a) & \text{if } a \text{ is symbolic} \\ rn\_diff_a(i_a,j_a) & \text{if } a \text{ is numeric} \end{cases}$$

Missing attribute values are handled by returning an attribute distance of 1 (a maximal distance) if either of the attribute values is missed. The function $Overlap$ and the rang-normalized difference rn_diff_a are defined as:

$$Overlap(i_a,j_a) = \begin{cases} 0 & \text{if } i_a = j_a \\ 1 & \text{otherwise} \end{cases}$$

$$rn\_diff_a(i_a,j_a) = \frac{|i_a - j_a|}{range_a}$$

The value $range_a$ is used to normalize the attributes. This normalization scales the attributes down to the point where differences are almost less than one [22]. Therefore, we can remark that the definition of δ guarantees a value in the interval [0 1].

3   Proposed AG matching algorithm

3.1   Algorithm

As described in Section 1, two attributed graphs can be matched by using the 2-steps procedure [11]. In the proposed algorithm, we adopt the use of node signatures and an assignment problem. Note that, the distances between node signatures and the Hungarian method [14] (for solving the assignment problem) correspond to the first and the second steps of the 2-steps procedure, respectively. We improve this 2-steps procedure, by adding a new step which consists in computing the distance between the attributed graphs in a metric framework.

So, in a first step, distances between every pair of nodes in two attributed graphs are computed using the distance defined in the previous section. These distances form a cost matrix which defines a node-to-node assignment for a pair of graphs. This task can be seen as an instance of the assignment problem which is the second step in our algorithm. The assignment problem can be solved by the Hungarian method, running in $O(n^3)$ time[14] where n is the size of the biggest graph. The permutation matrix P, obtained by applying the Hungarian method to the cost matrix, defines the optimum matching between two given graphs. As third step, based on the permutation matrix P, we define a matching function M as follow :

$$M(x_i) = \begin{cases} y_j, & \text{if } P_{i,j}=1 & \text{(2a)} \\ 0, & \text{else} & \text{(2b)} \end{cases}$$

where x_i and y_j are the nodes, respectively, in the first and the second graph. Using this matching function we compute the distance between two attributed graphs. But before introducing the distance formula we denote by:

- $|M|$: the size of the matching function M which is the number of matching operations. In any case, when two attributed graphs are matched the number of the matching operations is the size of the smaller one.
- $\hat{M} = \sum D_n(\text{Ns(x)}, \text{Ns(M(x))})$: the matching cost which is the sum of the matching operation costs, for two attributed graphs matched by M.

We define the distance between two attributed graphs g_i and g_j as follows:

$$D(g_i, g_j) = \frac{\hat{M}}{|M|} + ||g_i| - |g_j|| \tag{3}$$

This distance represents the matching cost normalized by the matching size, and is increased by the difference of sizes of the two graphs ($|g_i|$ is the size (number of nodes) of the graph g_i). We can demonstrate that this distance is a metric satisfying non-negativity, identity of indiscernible, and symmetry triangle inequality conditions.
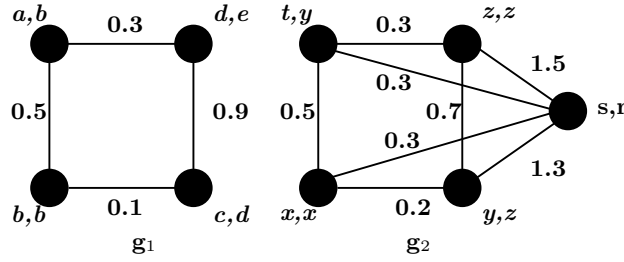
3.2 Attributed graph matching example

Fig. 1. Two attributed graphs.

Here, we present an example of attributed graph matching in order to clarify the proposed algorithm. Figure 1 shows two attributed graphs to be matched, g_1 with four nodes and g_2 with five nodes, and each node in the two graphs is attributed by two symbolic attributes and each edge by a numeric attribute. First, The node signatures in the two graphs must be computed. Using the previous definition, the graph g_1 has the following node signatures:

$$g_1:\{ \text{Ns}(n_{1_1})=\{\{\mathbf{a,b}\},\mathbf{2}, \{0.5, 0.3\}\},$$
$$\text{Ns}(n_{1_2})=\{\{\mathbf{b,b}\},\mathbf{2}, \{0.5, 0.1\}\},$$
$$\text{Ns}(n_{1_3})=\{\{\mathbf{c,d}\},\mathbf{2}, \{0.9, 0.1\}\},$$
$$\text{Ns}(n_{1_4})=\{\{\mathbf{d,e}\},\mathbf{2}, \{0.9, 0.3\}\}\}$$

and the graph g_2 has the following node signatures:

$$g_2:\{ \text{Ns}(n_{2_1})=\{\{\mathbf{t,y}\},\mathbf{3},\ \{0.5,\ 0.3,\ 0.3\}\},$$
$$\text{Ns}(n_{2_2})=\{\{\mathbf{x,x}\},\mathbf{3},\ \{0.5,\ 0.3,\ 0.2\}\},$$
$$\text{Ns}(n_{2_3})=\{\{\mathbf{y,z}\},\mathbf{3},\ \{1.3,\ 0.7,\ 0.2\}\},$$
$$\text{Ns}(n_{2_4})=\{\{\mathbf{z,z}\},\mathbf{3},\ \{1.5,\ 0.7,\ 0.3\}\ \}$$
$$\text{Ns}(n_{2_5})=\{\{\mathbf{s,r}\},\mathbf{4},\ \{1.5,\ 1.3,\ 0.3,\ 0.3\}\}\}$$

Next, we compute the cost matrix between nodes in the two graphs g_1 and g_2 making use of the distance described in §2. For example, the distance between the node signatures of n_{1_1} and n_{2_1} is given by:

$$\text{HEOM}(n_{1_1},n_{2_1}) = \sqrt{\sum_{a=0}^{5} \delta(n_{1_1a}, n_{2_1a})^2} = 1.7578$$

The computed cost matrix C between all the node signatures in g_1 and g_2 is:

	n_{2_1}	n_{2_2}	n_{2_3}	n_{2_4}	n_{2_5}
n_{1_1}	1.7578	1.7578	2.0221	2.0320	2.2764
n_{1_2}	2.0011	2.0011	2.0276	2.0374	2.2959
n_{1_3}	2.0055	2.0055	2.0144	2.0199	2.2804
n_{1_4}	2.0044	2.0055	2.0189	2.0144	2.2696

and then the permutation matrix P, obtained by applying the hungarian algorithm to the cost matrix is:

	n_{2_1}	n_{2_2}	n_{2_3}	n_{2_4}	n_{2_5}
n_{1_1}	1	0	0	0	0
n_{1_2}	0	1	0	0	0
n_{1_3}	0	0	1	0	0
n_{1_4}	0	0	0	1	0

Therefore, the correspondences between nodes in g_1 and g_2 can be established from P, and the distance between g_1 and g_2 is:

$$\text{D}(g_1,g_2)= \tfrac{1.7578+2.0011+2.0144+2.0144}{4}+|5-4| = 2.9469.$$

4   Experimental results

In this section, we provide some experimental results of the new attributed graph matching method. We check our method with two kinds of graphs attributes : graphs with only numeric attributes(weighted graphs) and graphs with symbolic and numeric attributes. We start with a matching problem using real world data and numeric attributes. The aim here is to evaluate how the new algorithm recovers the node-to-node matching under structural changes and to compare it with the well-known Umeyama method [12]. Afterward, we evaluate our graph distance by performing a graph retrieval task. Here, we provide a comparison with the method published by Riesen and al. [3] for two different data sets (images and molecules).

4.1   Node-to-node matching

We provide a comparison between our algorithm, the Umeyama method for inexact graph matching [12] and the Zass's probabilistic method [13] because both methods provide a explicit correspondence between nodes in two graphs using Hungarian Method [14].

The Umeyama [12] method matches nodes between two graphs by performing eigendecomposition of their adjacency matrices, and then computes the permutation matrix that brings the nodes of the graph into correspondence by applying the Hungarian Method [14] on the outer-product of the left singular vectors of the adjacency matrices. This method works only for graphs with same numbers of nodes.

Zass and al. [13] derived the graph matching problem in a probabilistic setting, which is solved via convex optimization and based on an algebraic relation between the hyperedges, the global optimum of the matching is found via an iterative successive projection algorithm.

Firstly, to provide a comparison with the Umeyama's algorithm, we have selected 23 images from the same class in the COIL-100 database [21] which contains the same number of corner points. Here, we are concerned with matching the Delaunay triangulations of corner-features, where each edge is weighted by a numeric attribute representing the Euclidean distance between two points. We use the Harris corner detector [16] to extract point features. Figure 2 (image of 50th object rotated in 320° and image of 50th object rotated in 325°) shows the correspondences between the corners as lines between the two images using our algorithm (Fig.2(a)) and the Umeyama algorithm (Fig.2(b)). The results are summarized in Table 1. From these results, our new method provides higher correct correspondence rate from the compared algorithms. Contrary to the Umeyama method our algorithm deals with graphs with different sizes and take into account both numeric and symbolic attributes. In the next section, we repeat this set of experiments using graphs with different sizes.
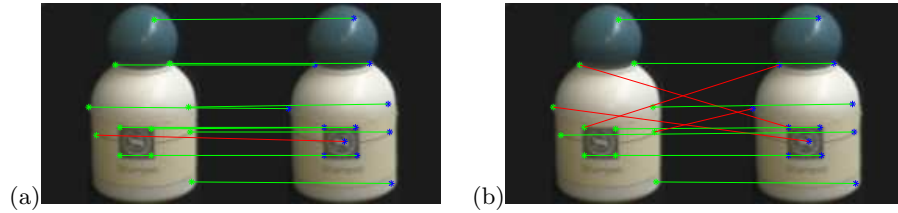

(a)   (b)

Fig. 2. (a) Correspondences with our algorithm. (b) Correspondences with the Umeyama algorithm. (green lines correspond to correct correspondences and red lines to false ones)

Secondly, we use the CMU/VASC model-house sequence database which contains 9 images corresponding to different camera viewing directions. The graphs

Algorithms	Correct correspondences	False correspondences	CCR
Umeyama	2120	916	69.83%
Zass	2222	814	73.19%
Our method	2525	511	**83.17%**

Table 1. Comparison of the two Matching Algorithms.

are obtained by the Delaunay triangulations of the detected corner points by the Harris [16]. Figure 3 shows an example of the results obtained when we match two images using our method and the Zass method. The results are given in Table 2. From these results, it is clear that the new method returns considerably better matches.

From the two previous set of experiments, our method has shown good flexibility and robustness among different data sets. In fact, in the model-house sequence database there are clearly significant structural differences in the graphs in comparison to the selected subset from the COIL-100 database. For these two graph sets our new method provides good results.
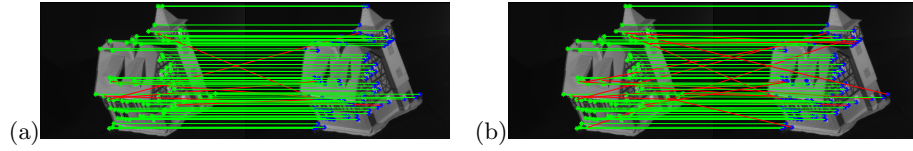
Fig. 3. (a) Correspondences with our algorithm. (b) Correspondences with the Zass algorithm. (green lines correspond to correct correspondences and red lines to false ones)

Algorithms	Correct correspondences	False correspondences	CCR
Zass	1474	1162	55.92%
Our method	1658	978	**62.90%**

Table 2. Comparison of the two Matching Algorithms.(CCR: correct correspondence rate)

4.2   Graph retrieval

Here, the retrieval performance is evaluated on four databases from the IAM graph database repository [4]:

- The COIL-RAG database (7200 images, 72 classes) consists of COIL-100 database where images are transformed into region adjacency graphs. Each region corresponds to a node linked with attributes specifying the color histogram, and adjacent regions correspond to the edges which are attributed with the length of the common border.
- The Mutagenicity database consists of 4337 graphs (2 classes) representing molecular compounds, the nodes represent the atoms labeled with the corresponding chemical symbol and edges by valence of linkage.
- The Letter database (3000 graphs, 15 classes) involves graphs that represent distorted letter drawings. Each distorted letter correspond to a graph by representing lines by edges and ending points of lines by nodes. The nodes are labeled by two-dimensional attribute giving its position.
- The GREC database [5](1100 images, 22 classes) which consists of graphs representing symbols from architectural and electronic drawings. Here the ending points (ie corners, intersections and circles) are represented by nodes which are connected by undirected edges and labeled as lines or arcs.

In Table 3, we synthesize the types of the attributes (symbolic or numeric) present in the edges and the nodes of the graphs in each database.

	COIL-RAG	Mutagenicity	Letter	GREC
Symbolic		X (N)		X (E,N)
Numeric	X (E,N)	X (E)	X (N)	X (E,N)

Table 3. Synthesis of attributes types in the used databases.(E: Edge, N: Node)

In these experiments, the receiver-operating curve (ROC) is used to measure retrieval performances. The ROC curve is formed by Precision rate against Recall rate. Precision rate is the ratio of the number of correct images to the number of retrieved images. Recall is the ratio of the number of correct images to the total number of correct images in the database. We provide a comparison between our method and the Riesen and al. method [3] which consider (as our approach) the graph matching as an instance of an assignment problem. However, in their method (called BGMEDG) a bipartite graph matching is proposed to compute the edit distance of graphs. More precisely, the method computes the edit distance between two graphs based on a bipartite graph matching by means of the Hungarian algorithm and provides only sub-optimal edit distance results. Therefore, this algorithm requires a predefined cost function to define the node-to-node costs. Then, the Hungarian algorithm is applied to this matrix to find an edit path which consists in the minimum-cost node assignment.

Figure 4 shows the Precision-Recall curves obtained by applying the new method and the BGMEDG method on the different databases. The cost functions of the BGMEDG method have been defined empirically. These results note that our method outperforms the BGMEDG on three databases among four.
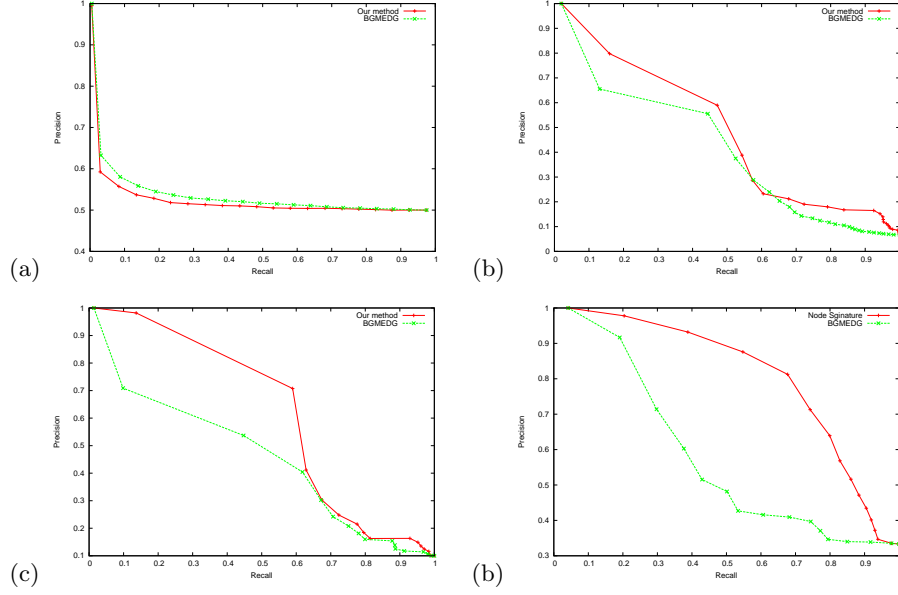
Fig. 4. Precision-Recall curves on databases :(a) Mutagenicity. (b) Letter. (c) COIL-RAG (d) GREC.

Especially, the results on the GREC database, when nodes and edges contain combined symbolic and numeric attributes, demonstrate that our node signatures are flexible and robust against the different type of attributes. However, our method fails on the Mutagenicity database, but the performance is quite similar to the other approaches

From this set of experiments, we can note that the extracted node signatures provide good structural local descriptions of the graphs. In addition, when both nodes and edges are labeled by combined symbolic and numeric attributes, the concept of node signature becomes more significant and its stored information becomes more discriminant.

5 Conclusion

In this work, we propose a new attributed graph matching technique based on node signatures describing local information in the graphs. We construct a cost matrix based on the distance between each pair of nodes in two graphs. To compute the distance between two graphs defined with symbolic and numeric attributes we have used the Heterogeneous Euclidean Overlap Metric. The optimum matching is computed using the Hungarian algorithm and based on a proposed metric graph distance. Experimentally, we have proved that our method performs node-to-node correspondences between two graphs, and provides good results to retrieve different kind of images represented by attributed graphs.

References

1. R. Myers, R. C. Wilson, and E. R. Hancock, Bayesian Graph Edit Distance, IEEE Trans. Pattern Anal. Mach. Intell., vol. 22, no. 6, pp. 628-635, 2000.
2. M. Neuhaus, and H. Bunke, Edit distance-based kernel functions for structural pattern classification, Pattern Recognition, vol. 39, no. 10, pp. 1852-1863, 2006.
3. K. Riesen, and H. Bunke, "Approximate graph edit distance computation by means of bipartite graph matching," Image Vis. Comput. (2008), doi:10.1016/j.imavis.2008.04.004
4. K. Riesen and H. Bunke, IAM Graph Database Repository for Graph Based Pattern Recognition and Machine Learning, IAPR Workshop SSPR & SPR, pp. 287-297, 2008.
5. P. Dosch and E. Valveny, "Report on the Second Symbol Recognition Contest," Proc. 6th IAPR Workshop on Graphics Recognition, pp. 381-397, 2005.
6. D. Lopresti, and G. Wilfong, A fast technique for comparing graph representations with applications to performance evaluation, International Journal on Document Analysis and Recognition, vol. 6, no. 4, pp. 219-229, 2004.
7. J. Shi and J. Malik, Self Inducing Relational Distance and Its Application to Image Segmentation, Proceedings of the 5th European Conference on Computer Vision, Vol. 1, pp. 528-543, 1998.
8. D. Conte P. Foggia C. Sansone and M. Vento, Thirty Years of Graph Matching in Pattern Recognition, Int'l J. Pattern Recognition and Artificial Intelligence, vol. 18, no. 3, pp. 265-298, 2004.
9. S. Jouili and S. Tabbone, Graph matching using node signatures,A. Torsello, F. Escolano, and L. Brun (Eds.), GbRPR 2009, LNCS 5534, pp. 154-163, Springer-Verlag Berlin Heidelberg 2009.
10. A. Sanfeliu and K.-S. Fu, A Distance Measure between Attributed Relational Graphs for Pattern Recognition, IEEE Transactions on Systems, Man and Cybernetics, vol.13, pp.353-362, 1983.
11. D. H. Kim , I. D. Yun and S. U. Lee, A New Attributed Relational Graph Matching Algorithm Using the Nested Structure of Earth Mover's Distance, 17th International Conference on Pattern Recognition, p. 48-51, 2004.
12. S. Umeyama, An eigendecomposition approach to weighted graph matching problems," IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 10, no. 5, pp. 695-703, 1988.
13. R. Zass, A. Shashua, Probabilistic graph and hypergraph matching, Computer Vision and Pattern Recognition, IEEE Computer Society Conference on, pp. 1-8, 2008.
14. H. W. Kuhn, The Hungarian method for the assignment problem. Naval Research Logistic Quarterly, vol. 2, pp. 83-97, 1955.
15. Robles-Kelly, A. and E. R. Hancock, Graph edit distance from spectral seriation, IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 27, no. 3, pp. 365-378, 2005.
16. C. Harris and M. Stephens, A combined corner and edge detection, Proceeding of the 4th Alvey Vision Conference, pp. 189-192, 1988.

17. M. Gori, M. Maggini, and L. Sarti, Exact and Approximate graph matching using random walks, IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 27,no. 7, pp. 1100-1111, 2005.
18. S. Gold, and A. Rangarajan, "A graduated assignment algorithm for graph matching," IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 18, no. 4, pp. 377-388, 1996
19. A. Shokoufandeh and S. Dickinson, Applications of Bipartite Matching to Problems in Object Recognition, Proceedings, ICCV Workshop on Graph Algorithms and Computer Vision, September 21, 1999
20. M. A. Eshera and K. S. Fu, A graph distance measure for image analysis, IEEE Trans. Syst. Man Cybern., vol. 14, 398408, 1984.
21. S.A. Nene, S.K. Nayar, and H. Murase, Columbia Object Image Library (COIL-100), technical report, Columbia Univ., 1996.
22. D.R. Wilson, T.R. Martinez, Improved heterogeneous distance functions, Journal of Artificial Intelligence Research, vol. 6, no. 1, pp. 1-34, 1997.