MORENO-GARCÍA, C.F., CORTÉS, X. and SERRATOSA, F. 2016. A graph repository for learning error-tolerant graph matching. *In Robles-Kelly, A., Loog, M., Biggio, B., Escolano, F. and Wilson, R. (eds.) Structural, syntactic and statistical pattern recognition: proceedings of 2016 Joint International Association of Pattern Recognition (IAPR) Structural and syntactic pattern recognition internaional workshops (SSPR 2016) and Statistical techniques in pattern recognition (SPR 2016) (S+SSPR 2016), 20 November - 2 December 2016, Mérida, Mexico*. Lecture notes in computer science, 10029. Cham: Springer [online], pages 519-529. Available from: https://doi.org/10.1007/978-3-319-49055-7_46

A graph repository for learning error-tolerant graph matching.

MORENO-GARCÍA, C.F., CORTÉS, X. and SERRATOSA, F.

2016

A Graph Repository for Learning Error-Tolerant Graph Matching

Carlos Francisco Moreno-García(*), Xavier Cortés, and Francesc Serratosa

Departament d'Enginyeria Informàtica i Matemàtiques, Universitat Rovira I Virgili, Tarragona, Spain
carlosfransisco.moreno@estudiants.urv.cat, xavier.cortes@urv.cat, francesc.serratosa@urv.cat

Abstract. In the last years, efforts in the pattern recognition field have been especially focused on developing systems that use graph based representations. To that aim, some graph repositories have been presented to test graph-matching algorithms or to learn some parameters needed on such algorithms. The aim of these tests has always been to increase the recognition ratio in a classification framework. Nevertheless, some graph-matching applications are not solely intended for classification purposes, but to detect similarities between the local parts of the objects that they represent. Thus, current state of the art repositories provide insufficient information. We present a graph repository structure such that each register is not only composed of a graph and its class, but also of a pair of graphs and a ground-truth correspondence between them, as well as their class. This repository structure is useful to analyse and develop graph-matching algorithms and to learn their parameters in a broadly manner. We present seven different databases, which are publicly available, with these structure and present some quality measures experimented on them.

Keywords: Graph database Graph-matching algorithm Graph-learning algorithm

1 Introduction

In pattern recognition, benchmarking is the process of measuring the quality of the representation of the objects, or the quality of the algorithms involved on comparing, classifying or clustering these objects. The objective of benchmarking is to improve performance of the involved object representations and pattern recognition algorithms. Pattern recognition, through graph-based representations, has been developed through the last forty years with great success and acknowledgement. Interesting surveys about this subject are [1, 2] or [3]. The first error-tolerant graph matching algorithms were published in 1983, [4, 5], and since then, several new algorithms have been presented.

This research is supported by projects DPI2013-42458-P TIN2013-47245-C2-2-R and by Consejo Nacional de Ciencia y Tecnología (CONACyT México).

For this reason, in 2008, a specific database to perform benchmarking on graph databases was published for the first time [6]. As authors reported, they presented such database and published its paper with the aim of providing to the scientific community a public and general framework to evaluate graph representations and graph algorithms [7–9], such as error-tolerant graph matching, [10–15] learning the consensus of several correspondences, [16–20], image registration based on graphs, [21, 22], learning graph-matching parameters [23, 24], and so on. Note that a huge amount of methods has been presented, and the previous list is simply a small sample of them. For a detailed list of methods, we refer to the aforementioned surveys [1–3]. This database, called IAM [25], has been largely cited and used to develop new algorithms. It is composed of twelve datasets containing diverse attributed graphs, for instance, proteins, fingerprints, hand written characters, among others.

With the same idea, another graph database had been previously published in 2001 [26, 27]. Nevertheless, the aim of this database [28] is to perform exact isomorphism benchmarking and cannot be used to test error-tolerant graph matching since nodes and edges are unattributed. It contains 166’000 graphs with very diverse graph sizes. Most recently in 2015 [29], a new graph repository [30] was presented in order to compare exact graph edit distance (GED) calculation methods, where data from [26, 31] was collected and enhanced using low-level information.

Note that other papers have presented with new graph-based methodologies and, with the aim of experimental reproducibility, reported their self-made databases and made them public. This is the case of the one first presented in 2006 [32, 33]. It is composed of attributed graphs extracted from image sequences taken from the CMU repository [34]. Graph nodes represent salient points of some images and graph edges have been generated through Delaunay triangulation or represent shape edges.

Registers of the aforementioned databases are composed of a graph and its class (except for the one in [29] that incorporates some additional information). Thus, the only quality measures that we can extract from the algorithms applied to these databases are related on classification purposes. For instance, the usual measures are the false positives, the false negatives and the recognition ratio.

In this paper, we present a new graph-database structure. Registers on this database are composed of a pair of graphs, a ground-truth correspondence between them as well as the class of these graphs. This ground-truth is independent of the graph-matching algorithm and also on their specific parameters, since it has been imposed by a human or an optimal automatic technique. Therefore, the quality measures that we can extract not only are the ones related on classification, but also the ones related on the ground-truth correspondence, such as the Hamming distance (HD) between the obtained correspondence and the ground-truth correspondence. Moreover, some graph-matching learning algorithms that need a given ground-truth correspondence [19, 33, 35–37] could be applied and evaluated. We concretise this structure on seven different databases, and we present some quality measures experimented on them.

Similar to the case of the IAM graph database repository [25], we divide the databases in three sets, viz. learning, test and validation. In machine learning applications, the learning set is used to learn the database knowledge that is usually materialised on the algorithms’ input parameters. The validation set is used for regularisation purposes, that is, to tune the over-fitting or under-fitting of the learned parameters. Finally, the test set is used to test the quality measures of the methods learned through the learning and the validation sets.

The rest of the paper is structured in two other sections. In the first one, we present the graph repository and its benchmarks. In the second one, we conclude the paper.

2 The Graph Repository

The "Tarragona Repository" (publicly available at [38]) is described in this section, which is divided into three sub-sections. In the first one, the general structure of the whole databases is described. In the second one, we describe the current databases in the repository. Note the aim of this paper is to define a new method to structure graph databases and therefore, other databases could be included by the authors or other researches in a near future. In the third sub-section, we summarise the main features of each database and we present some experimental results performed on them.

2.1 General Structure

Databases in the "Tarragona repository" are composed of registers with a format G^i; $G^{0i}; f^i; C^i$. Attributed graphs G^i and G^{0i} need to be defined in the same attribute domain, but may have different orders. The ground-truth correspondence f^i between the nodes of G^i and G^{0i} may have some nodes of G^i mapped to nodes of G^{0i}, and other ones mapped to a null node. Nevertheless, two nodes of G^i cannot be mapped to the same node of G^{0i}. The null node is a mechanism to represent that a node of G^i do not have to be mapped to any node of G^{0i} [10]. Note some nodes of G^{0i} may not have been mapped to any node of G^i through f^i. Moreover, we impose both graphs to belong to the same class. This is because we consider it has no sense to map local parts of objects that belong to different classes. For instance, if graphs represent hand-written characters, there is no ground-truth correspondence between an "A" and a "J".

Our databases are composed of five terms: Name, Description, Learning, Test and Validation. Name and Description are obvious, and Learning, Test and Validation are the three common datasets to perform benchmarking.

We present in [38], together with these databases, the following Matlab functions:

- *Load Register Database*; *Set*; *Register*: Returns the register *Register* in the database *Database* and the set *Set* that accepts three values: *Learning*, *Test* or *Validation*. The output has the format $G^i; G^{0i}; C^i; f^i; I^i; I^{0i}$. G^i and G^{0i} are both graphs with their class C^i, f^i is the ground-truth correspondence, and values I^i and I^{0i} are the indices of graphs G^i and G^{0i} respectively. These indices are useful to know which graphs have been mapped to other ones since any given graph can appear in several registers although each time has to be mapped to a different graph.
- *Load Graph Database*; *Set*; *Index*: it returns the graph in position *Index*. This function is useful to test the classification ratio.
- *Classification Database*; *Set*1; *Set*2; K_v; K_e: Returns the classification ratio and the average Hamming distance given sets *Set*1 and *Set*2 in *Database*. The fast bipartite

graph matching (FBP) [13] has been used to compute the GED [10] and the correspondences. Parameter K_v is the insertion and deletion costs on the nodes, and parameter K_e is the insertion and deletion cost on the edges.
- *Plot Graph Graph*; *Image*: Plots the graph over the image where it was extracted from, in the case that the graph represents an object on an image. This function assumes that the first two node attributes are the image coordinates $x; y$.

With the aim of reducing the memory space, the *Learning*, *Test* and *Validation* sets of each database have been logically structured as shown in Fig. 1. There is a main vector, where each cell is composed of a structure of three elements. The first one contains a graph, the second one assigns a class to this graph, and the third one describes the correspondences from this graph to the rest of graphs. Considering the graphs, the set of nodes and edges are defined as numerical matrices. The order of each graph is N and nodes have A attributes. Graphs can have different orders N, but they have the same number of attributes A given the whole database. Edges do not have attributes. The existence of an edge is represented by a 1, and the non-existence is represented by a 0. Classes are defined as string of characters. Each correspondence cell $f^{i,a}$ maps the original graph G^i to another graph G^a and it is composed of a structure of two elements that are the index of the input graph and the node-to-node mapping vector. In the node-to-node mapping vector, there are natural numbers representing the index node, and the value 1, which can appear in several positions of the correspondence, represents a mapping to a null node.

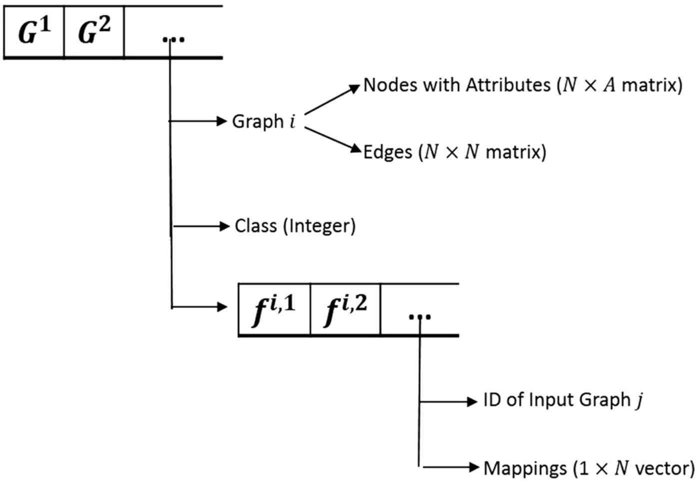

Fig. 1. Scheme representing the distribution of the information contained in each set (learning, validation or test) of a database.

2.2 Databases

The databases that are currently available are:

2.2.1 Rotation Zoom

This database contains graphs that have been extracted from 5 classes that have 10 images of outdoors scenes. Per each class, images were taken from different angles and positions. We were able to generate a correspondence between all the generated graphs by using the image homography, which was provided on the original image database [39]. Each node represents a salient point of the image. It is attributed with the position of the salient point in the image x; y and also a 64-size feature vector obtained by the SIFT extractor [40]. Edges are conformed using the Delaunay triangulation and do not have attributes. An example with a graph of each class is shown in Fig. 2.

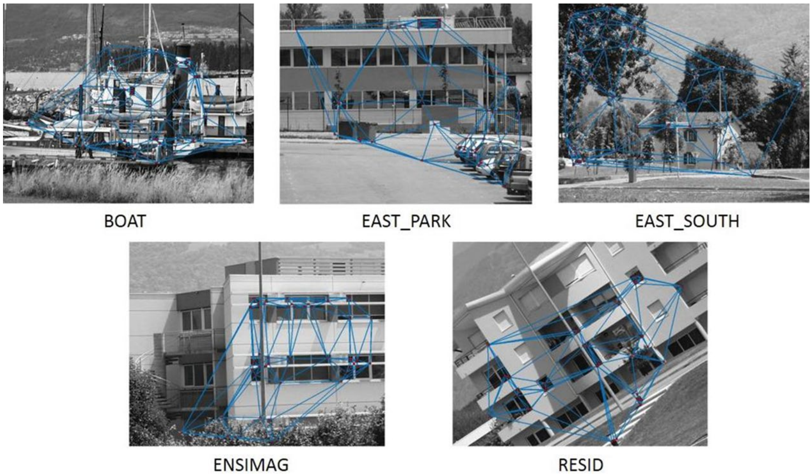

Fig. 2. The first image of each of the 5 classes and their graphs.

2.2.2 Palmprint

In order to construct this database, we used palmprint images contained in the Tsinghua 500 DPI Database [41], which currently has more than 150 subjects whose right and left palm has been scanned a total of 8 times each. Using the first 20 palms of the original database (10 right hands and 10 left hands), this database is constituted by a total of 20 classes of 8 graphs each. Minutiae were extracted using the algorithm proposed in [42] and graphs were constructed with each node representing a minutia. Node attributes contain information such as the minutiae position, angle, type (termination or bifurcation) and quality (good or poor). Edges are conformed using the Delaunay triangulation and do not have attributes. Finally, a correspondence between all graphs of the same class is generated using a greedy matching algorithm based on the Hough transform [43]. An example of a palmprint image and its graph is provided in Fig. 3.

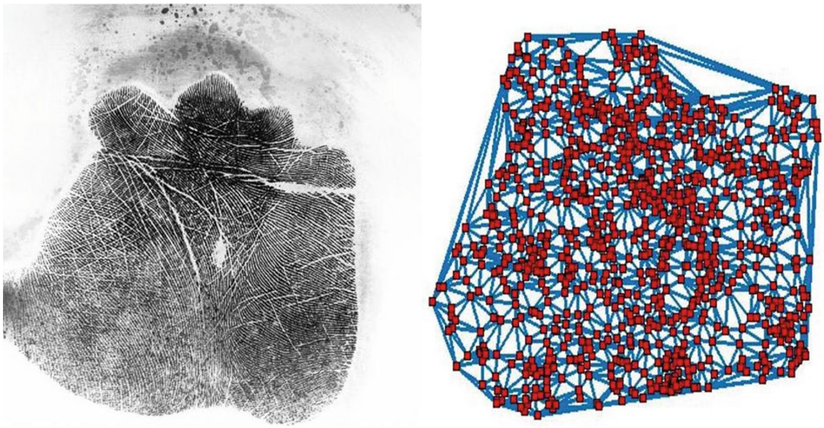

Fig. 3. A palmprint and its graph.

2.2.3 Letters

The *Letters* graph database originally presented in [6] consists on a set of graphs that represent artificially distorted letters of the Latin alphabet. For each class, a prototype line drawing was manually constructed. These prototype drawings are then converted into prototype graphs by representing the lines through undirected edges, and the ending points of such lines through nodes. Attributes on nodes are only the bi-dimensional position of the junctions and edges do not have attributes. Figure 4 shows four samples of letter A.

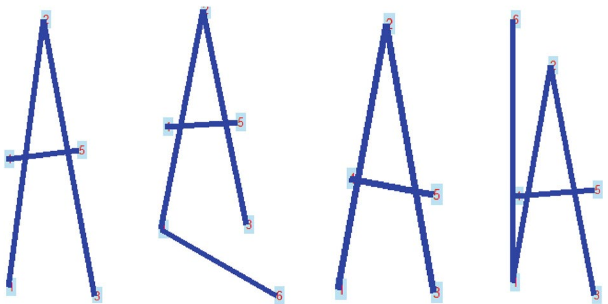

Fig. 4. Different instances of letter A.

There are three variants of the database depending on the degree of distortion with respect to the original prototype (adding, deleting and moving nodes and edges), viz. low, medium and high. The ground-truth correspondence between the nodes is well-known, because graphs of each class are generated from an original prototype.

2.2.4 Sagrada Familia 3D

The *Sagrada Familia 3D* database consist of a set of graphs, where each one represents a cloud of 3D points with structural relations between them. Nodes represent 3D points and their attributes are the 3D position. Edges represent proximity and do not have attributes. These points have been extracted as follows. First, a sequence of 473 photos were taken from different positions around the Sagrada Familia church in Barcelona (Catalonia, Spain), pointing the camera at the centre of it. Using the whole sequence of 2D images, a 3D model of the monument was built through the Bundler method [44, 45]. This method deducts a global cloud of 3D points of a central object using the salient points of the set of 2D images. Moreover, it also returns the correspondence between the 3D points of the resultant model and the salient points of the 2D images. Each graph in the database represents the 3D information of the salient points that appear in each image. Figure 5 shows the process to generate the graphs. Red points are the 3D model of Sagrada Familia, blue points are the different poses of the camera that has captured the images of the model and black points represent the salient points of images.

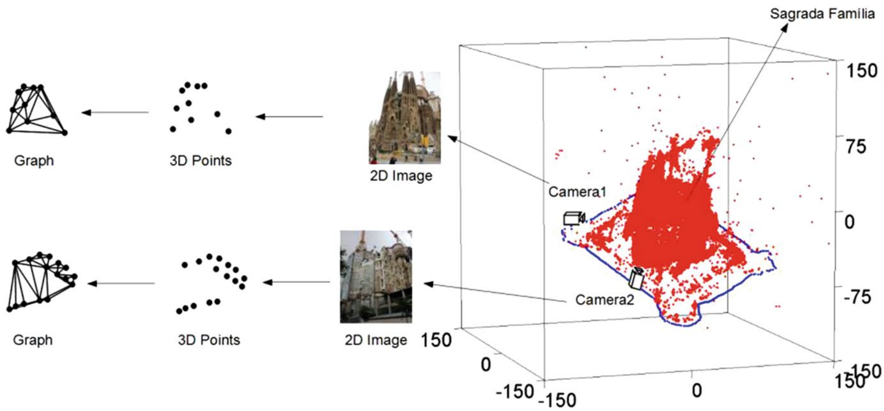

Fig. 5. The process to generate *Sagrada Familia 3D* database. (Color figure online)

2.2.5 House-Hotel

The original CMU "house" and "hotel" databases consist of 111 graphs corresponding to a toy house and 101 graphs corresponding to a hotel [46]. Each frame of these sequences has the same 30 hand-marked salient points identified and labelled with some attributes. Therefore, nodes in the graphs represent the salient points, with their position in the image plus a 60-size feature vector using Context Shape (CS) as attributes. Edges are unattributed and were constructed using the Delaunay triangulation. In this database there are three sets of pairs of frames, considering as baseline the number of frames of separation in the video sequence (Fig. 6).

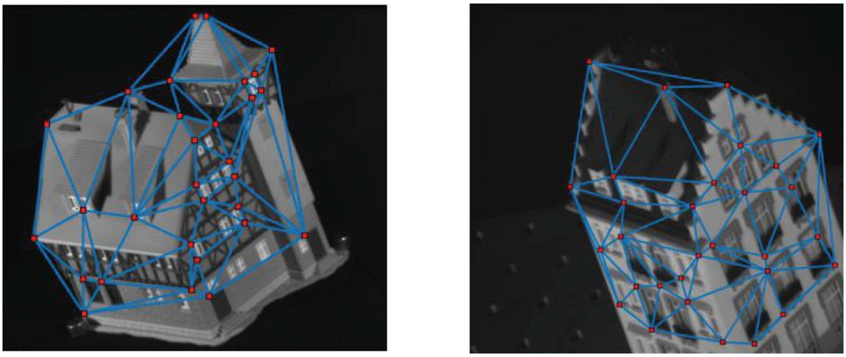

Fig. 6. Different images of each of the two classes and their graphs.

2.3 Repository Summary

Table 1 summarises the main characteristics of the repository. The databases contained have been selected due to the variability on their characteristics, such as the number of nodes and edges, the number of classes, the type of attributes or the number of nodes that the ground-truth correspondences maps to the null node. These differences directly influence on the behaviour of the implemented algorithms and therefore, these databases can be used to analyse different situations and arrive to interesting conclusions, such as whether the functionality of certain methodology could be better than another, given a determined situation.

Table 1. Summary of the characteristics of each database.

Database		*Rotation zoom*	*Palmprint*	*Letter*			*Sagrada Familia*	*House-Hotel*
				Low	*Med*	*High*		
Number of graphs	Train	20	80	750	750	750	136	71
	Validation	10	0	750	750	750	136	71
	Test	20	80	750	750	750	135	70
Number of correspondences	Train	80	320	37500	37500	37500	18496	2627
	Validation	40	0	37500	37500	37500	18255	2627
	Test	80	320	37500	37500	37500	18255	2590
Number of classes		5	20	15	15	15	1	2
Number of node attributes		66	5	2	2	2	3	62
Attributes' description		*(x,y)* 64 SIFT	*(x,y)* 1 Angle 1 Type 1 Quality	*(x,y)*			*(x,y,z)*	*(x,y)* 60 CS
Avg. nodes		50	836.3	4.6	4.6	4.6	39.3	30
Avg. edges		277.4	4971.2	6.2	6.4	9	456.5	154.4
Avg. null correspondences		31.6	152.1	0.4	0.4	0.4	30.1	0
Max. nodes		50	1505	8	9	9	141	30
Max. edges		284	8962	12	14	18	1918	158
Max. null correspondences		50	619	4	5	5	139	0

Table 2 shows the classification ratio and the average Hamming distance between the computed correspondences and the ground-truth correspondences. It is the result of running the Matlab function *Classification Database*; *Test*; *Reference*; K_v; K_e available in [36] (explained in Sect. 2.1). As commented, the FBP [13] has been used to compute the GEDs [10] and the correspondences. Insertion and deletion cost on nodes, K_v, and insertion and deletion cost on edges, K_e, have been deducted through the learning algorithm presented in [37]. The aim of this table is not to report the best achieved results but simply to show an example of a specific graph-matching algorithm and learning algorithm. We encourage other researches to share their results, while showing these ones as a starting point.

Table 2. Classification ratio and HD obtained with the FBP [13] given edit costs K_v and K_e, which have been learned by a correspondence-based learning algorithm [37].

Database		Edit costs		Classification ratio	Hamming distance
		K_v	K_e		
Rotation zoom		0.0325	−0.0027	1	0.8598
Palmprint		210	5	0.85	0.4763
Letter	*Low*	1	1	0.9453	0.9096
	Med	1	1	0.8667	0.8382
	High	1	1	0.8080	0.8303
Sagrada Familia		0.05	0.05	–	0.7439
House-Hotel		1000	1	1	0.8598

3 Conclusions

We have presented a publicly available graph repository to perform benchmarking on graph algorithms such as graph matching, graph clustering, leaning consensus correspondence or parameter learning. The main feature of this repository is that registers of these databases do not have the classical structure composed of a graph and its class, but are composed of a pair of graphs, their class and the ground-truth correspondence. We want this repository not to be seen as a concluded project, but a dynamic one, in which other researches contribute with more graph databases. Moreover, we have presented some classification ratios and Hamming distance on these databases, given some specific algorithms and parameterisations. For this aspect as well, we invite other researches to contribute with more results and therefore, to extend and disseminate the results obtained so far.

References

1. Conte, D., Foggia, P., Sansone, C., Vento, M.: Thirty years of graph matching in pattern recognition. Int. J. Pattern Recogn. Artif. Intell. 18(3), 265–298 (2004)
2. Foggia, P., Percannella, G., Vento, M.: Graph matching and learning in pattern recognition in the last ten years. Int. J. Pattern Recogn. Artif. Intell. 28(1), 1450001 (2014)
3. Vento, M.: A long trip in the charming world of graphs for pattern recognition. Pattern Recogn. 48(2), 291–301 (2015)
4. Sanfeliu, A., Fu, K.S.: A distance measure between attributed relational graphs for pattern recognition. IEEE Trans. Syst. Man Cybern. 13(3), 353–362 (1983)
5. Bunke, H.: Inexact graph matching for structural pattern recognition. Pattern Recogn. Lett. 1, 245–253 (1983)
6. Riesen, K., Bunke, H.: IAM graph database repository for graph based pattern recognition and machine learning. In: da Vitoria Lobo, N., Kasparis, T., Roli, F., Kwok, J.T., Georgiopoulos, M., Anagnostopoulos, G.C., Loog, M. (eds.) SSPR & SPR 2008. LNCS, vol. 5342, pp. 287–297. Springer, Heidelberg (2008)
7. Wong, A., You, M.: Entropy and distance of random graphs with application to structural pattern recognition. IEEE Trans. Pattern Anal. Mach. Intell. 7(5), 599–609 (1985)
8. Serratosa, F., Alquézar, R., Sanfeliu, A.: Function-described graphs for modelling objects represented by attributed graphs. Pattern Recogn. 36(3), 781–798 (2003)
9. Sanfeliu, A., Serratosa, F., Alquézar, R.: Second-order random graphs for modelling sets of attributed graphs and their application to object learning and recognition. Int. J. Pattern Recogn. Artif. Intell. 18(3), 375–396 (2004)
10. Solé, A., Serratosa, F., Sanfeliu, A.: On the graph edit distance cost: properties and applications. Int. J. Pattern Recogn. Artif. Intell. 26(5), 1260004 (2012)
11. Lladós, J., Martí, E., Villanueva, J.: Symbol recognition by error-tolerant subgraph matching between region adjacency graphs. Trans. Pattern Anal. Mach. Intell. 23(10), 1137–1143 (2001)
12. Riesen, K., Bunke, H.: Approximate graph edit distance computation by means of bipartite graph matching. Image Vis. Comput. 27, 950–959 (2009)
13. Serratosa, F.: Fast computation of bipartite graph matching. Pattern Recogn. Lett. 45, 244–250 (2014)
14. Serratosa, F.: Computation of graph edit distance: reasoning about optimality and speed-up. Image Vis. Comput. 40, 38–48 (2015)
15. Serratosa, F.: Speeding up fast bipartite graph matching through a new cost matrix. Int. J. Pattern Recogn. Artif. Intell. 29(2), 1550010 (2015)
16. Moreno-García, C.F., Serratosa, F.: Consensus of two sets of correspondences through optimisation functions. Pattern Anal. Appl. 2015, 1–13 (2015)
17. Moreno-García, C.F., Serratosa, F.: Consensus of multiple correspondences between sets of elements. Comput. Vis. Image Underst. 142, 50–64 (2015)
18. Moreno-García, C.F., Serratosa, F., Cortés, X.: Consensus of two graph correspondences through a generalization of the bipartite graph matching. In: Liu, C.-L., Luo, B., Kropatsch, Walter G., Cheng, J. (eds.) GbRPR 2015. LNCS, vol. 9069, pp. 87–97. Springer, Heidelberg (2015)
19. Moreno-García, C.F., Serratosa, F.: Online learning the consensus of multiple correspondences between sets. Knowl. Based Syst. 90, 49–57 (2015)
20. Moreno-García, C.F., Serratosa, F.: Obtaining the consensus of multiple correspondences between graphs through online learning. Pattern Recogn. Lett. (2016). doi:10.1016/j.patrec.2016.09.003

21. Sanromà, G., Alquézar, R., Serratosa, F., Herrera, B.: Smooth point-set registration using neighbouring constraints. Pattern Recogn. Lett. 33, 2029–2037 (2012)
22. Sanromà, G., Alquézar, R., Serratosa, F.: A new graph matching method for point-set correspondence using the EM algorithm and softassign. Comput. Vis. Image Underst. 116 (2), 292–304 (2012)
23. Neuhaus, M., Bunke, H.: Automatic learning of cost functions for graph edit distance. Inf. Sci. 177(1), 239–247 (2006)
24. Neuhaus, M., Bunke, H.: Self-organizing maps for learning the edit costs in graph matching. IEEE Trans. Syst. Man Cybern. Part B 35(3), 503–514 (2005)
25. http://www.iam.unibe.ch/fki/databases/iam-graph-database
26. Foggia, P., Sansone, C., Vento, M.: A database of graphs for isomorphism and subgraph isomorphism benchmarking. In: Proceedings of 3rd International Workshop on Graph Based Representations in Pattern Recognition, pp. 176–187 (2001)
27. De Santo, M., Foggia, P., Sansone, C., Vento, M.: A large database of graphs and its use for benchmarking graph isomorphism algorithms. Pattern Recogn. Lett. 24, 1067–1079 (2003)
28. http://iapr-tc15.greyc.fr/links.html
29. Abu-Aisheh, Z., Raveaux, R., Ramel, J.Y.: A graph database repository and performance evaluation metrics for graph edit distance. In: IAPR International Workshop on Graph Based Representation (2015)
30. http://www.rfai.li.univ-tours.fr/PublicData/GDR4GED/home.html
31. Gao, X., Xiao, B., Tao, D., Li, X.: A survey of graph edit distance. Pattern Anal. Appl. 13 (1), 113–129 (2010)
32. Caetano, T., Caelli, T., Schuurmans, D., Barone, D.: Graphical models and point pattern matching. IEEE Trans. Pattern Anal. Mach. Intell. 28(10), 1646–1663 (2006)
33. Caetano, T., et al.: Learning graph matching. Trans. Pattern Anal. Mach. Intell. 31(6), 1048–1058 (2009)
34. http://www.cs.cmu.edu/afs/cs/project/vision/vasc/idb/www/html_permanent//index.html
35. Cortés, X., Serratosa, F.: Learning graph matching substitution weights based on the ground-truth node correspondence. Int. J. Pattern Recogn. Artif. Intell. 30(2), 1650005 (2016)
36. Leordeanu, M., Sukthankar, R., Hebert, M.: Unsupervised learning for graph matching. Int. J. Comput. Vis. 96(1), 28–45 (2012)
37. Cortés, X., Serratosa, F.: Learning graph-matching edit-costs based on the optimality of the oracle's node correspondences. Pattern Recogn. Lett. 56, 22–29 (2015)
38. http://deim.urv.cat/☆francesc.serratosa/databases/
39. http://www.featurespace.org
40. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vis. 60(2), 91–110 (2004)
41. Dai, J., Feng, J., Zhou, J.: Robust and efficient ridge based palmprint matching. IEEE Trans. Pattern Anal. Mach. Intell. 34(8), 1618–1632 (2012)
42. Dai, J., Zhou, J.: Multi-feature based high-resolution palmprint recognition. IEEE Trans. Pattern Anal. Mach. Intell. 33(5), 945–957 (2011)
43. Ratha, N.K., Karu, K., Chen, S., Jain, A.K.: A real-time matching system for large fingerprint databases. IEEE Trans. Pattern Anal. Mach. Intell. 18(8), 799–813 (1996)
44. http://www.cs.cornell.edu/☆snavely/bundler/
45. Snavely, N., Todorovic, S.: From contours to 3D object detection and pose estimation. In: International Congress on Computer Vision (ICVV), pp. 983–990 (2011)
46. http://vasc.ri.cmu.edu/idb/html/motion/