Graph Matching based on Node Signatures*

Salim Jouili and Salvatore Tabbone

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

Abstract. We present an algorithm for graph matching in a pattern recognition context. This algorithm deals with weighted graphs, based on new structural and topological node signatures. Using these signatures, we compute an optimum solution for node-to-node assignment with the Hungarian method and propose a distance formula to compute the distance between weighted graphs. The experiments demonstrate that the newly presented algorithm is well suited to pattern recognition applications. Compared with four well-known methods, our algorithm gives good results for clustering and retrieving images. A sensitivity analysis reveals that the proposed method is also insensitive to weak structural changes.

Key words: graph representation, graph matching, graph clustering.

1 Introduction

In image processing applications, it is often required to match different images of the same object or similar objects based on structural descriptions constructed from these images. If the structural descriptions of objects are represented by graphs, different images can be matched by performing some kind of graph matching. Graph matching is the process of finding a correspondence between nodes and edges of two graphs that satisfies some constraints ensuring that similar substructures in one graph are mapped to similar substructures in the other. Many approaches have been proposed to solve the graph matching problem [1, 5, 15]. Matching by minimizing the edit distance [4, 11, 13, 14] is attractive since it gauges the distance between graphs by counting the least cost of edit operations needed to make two graphs isomorphic. Moreover the graph edit distance has tolerance to noise and distortion. The main drawback of graph edit distance is its computational complexity, which is exponential in the number of nodes of the involved graphs. To reduce the complexity, Apostolos [14] gives a fast edit distance based on matching specific graphs by using the *sorted graph histogram*. Equivalently, Lopresti [12] gives an equivalence test procedure that allows to quantify the similarity between graphs. Other methods based on spectral approaches [2, 3, 16], give an elegant matrix representation for graphs that ensure

* This work is partially supported by the French National Research Agency project NAVIDOMASS referenced under ANR-06-MCDA-012 and Lorraine region.

an approximate solutions for graphs matching in polynomial time. Among the pioneering works related to graph matching using the spectral techniques we quote the paper of Umeyama [3], in which the *Weighted Graph Isomorphism Problem* is addressed by an eigendecomposition. However, this method can only be applied for graphs with the same number of nodes. More recent works [17, 18] extend this approach for graphs with different sizes but with a higher complexity.

In this paper, we propose a new efficient algorithm for matching and computing the distance between weighted graphs. We introduce a new *vector-based node signature* to reduce the problem of graph matching to a bipartite graph matching problem. Each node is associated with a vector where components are the collection of the node degree and the incident edge weights. Using these node signatures 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 Manhattan 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 [6–8], which can be solved by the Hungarian method [19]. We introduce also a new metric to compute the distance between graphs. The concept of node signature has been studied previously in [10, 8, 15] where the node signatures are computed using spectral, decomposition and random walks approaches. On the contrary, our node signature is a vector and it is computed straightforwardly from the adjacency matrix.

The remainder of this paper is organized as follows: in the next section (§2), the proposed matching algorithm is described and also the distance between two graphs. This distance is used to cluster and retrieve graph data sets. The proposed algorithm is validated within images clustering and content-based image retrieval applications. We have compared our results with the Umeyama method [3], the graph edit distance from spectral seriation [2], the graph histograms distance [14], and the graph probing technique [12] (section 3). Finally, in section 4, some conclusions are drawn.

2 Graph Matching algorithm

In this section we describe our algorithm, firstly for the graph matching problem (exact and inexact), and then for computing a metric distance between graphs.

Graph matching method. In order to obtain a set of local descriptions describing a weighted graph, each node is associated to a signature (vector). As it will be seen later, these node signatures are used to determine if two nodes in different graphs can be matched. Therefore, the construction of the node signature is a crucial stage in the graph matching process. For this aim, two kinds of information are available to describe the nodes. The first one is the degree of the node and the second one is the weights of the incident edges of the node. By combining these two informations, the valued neighborhood relations can be drawn as well as the topological features of one node in the graph. We introduce

a node signature in the context of weighted and unweighted graphs. For weighted graphs, the signature is defined as the degree of the node and the weights of all the incident edges. Given a graph G = (X, E), the node signature is formulated as follows:

$$\mathrm{Vs(x)}=\{\mathrm{d(x)},w_0,w_1,w_2\ ...\}$$

Where x ∈ X, d(x) gives the degree of x, and w_i are the weights of the incident edges to x. For unweighted graphs, the weights of any incident edges are fixed to 1. The set of node signatures (vectors) describing nodes in a graph is a collection of local descriptions. So, local changes in the graph will modify only a subset of vectors while leaving the rest unchanged. Moreover, the computational cost of the construction of these signatures is low since it is computed straightforwardly from the adjacency matrix. Based on these node signatures, a cost matrix C is defined by:

$$C_{g_i,g_j}(i,j) = L_1(\gamma(i),\gamma(j)) \tag{1}$$

where i and j are, respectively, the nodes of g_i and g_j, and $L_1(.,.)$ the Manhattan distance. $\gamma(i)$ is the vector $V_s(i)$ sorted only for the weights in a decreasing order. Finally, since the graphs have different size, the γ vectors are padded by zeros to keep the same size of vectors.

The cost matrix defines a vertex-to-vertex assignment for a pair of graphs. This task can be seen as an instance of the assignment problem, and can be solved by the Hungarian method, running in $O(n^3)$ time [19] 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. Based on the permutation matrix P, we define a matching function M as follow :

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

where x_i and y_j are the nodes, respectively, in the first and the second graph.

Distance formula. Before introducing the distance formula we denote:

- $|M|$: the size of the matching function M which is the number of matching operations. In any case, when two graphs are matched the number of the matching operations is the size of the smaller one.
- $\hat{M} = \sum \mathrm{L_1(\gamma(x),\ \gamma(M(x)))}$: the matching cost which is the sum of the matching operation costs, for two graphs matched by M.

We define the distance between two 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 the result is increased by the difference of sizes of the two graphs. We can demonstrate that this distance is a metric satisfying non-negativity, identity of indiscernible, and symmetry triangle inequality conditions.

3 Experiments

To show the utility of our method in pattern recognition applications and the robustness to structural changes, we drawn different experiments.

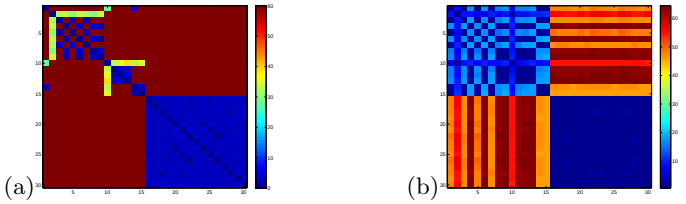


Fig. 1. Graph distance matrices. (a) results from Umeyama approach; (b) results from our approach.

Graph clustering application. Firstly, we compare our method with the Umeyama's algorithm for inexact graph matching [3]. The reason of selecting this method is that since we have applied the Hungarian algorithm to the cost matrix to find the optimum matching, we choose to compare our approach with a similar one. Since this method needs weighted graphs with the same number of nodes, we use only two classes from the GREC database, both have 15 graphs and 8 nodes per graph [22, 21]. The GREC data set consists in graphs representing symbols from architectural and electronic drawings classified into 22 classes. Graphs in each class are obtained by distorting original GREC images and the extracted graphs[21].

We compute the distance matrices (Fig. 1) for the two methods. The size of each matrix is 30x30. Each class of images corresponds to a block in these matrices. Images labeled between 1 and 15 correspond to the first class, and images between 16 and 30 correspond to the second class. The row and column index the distances between graphs, an element (i,j) corresponds to the distance between the ith and the jth image. Two blocks along the diagonal present the within-class distance and other blocks present the between-class distance. In Fig. 1(a), there are three blocks instead of two blocks along diagonal, and in the same block there are higher intensities; thus the within-class distance has a high value. In contrast, Fig. 1(b) shows two marked blocks, so a higher difference between within-class and between-classes distances.

Furthermore, we have performed the multidimensional scaling (MDS)[26] and the minimum spanning tree (MST) clustering [25]. Generally speaking, the MDS pictures the structure of a set of objects from data that define the distances between pairs of objects. Each object is represented by a point in a multidimensional space. The points are arranged in this space so that the distances between pairs of points have the strongest possible relation to the similarities among pairs of objects. We show the MDS results corresponding to the Umeyama method (Fig. 2(a)) and the results of our method (Fig. 2(b)). In Fig. 2(a), the

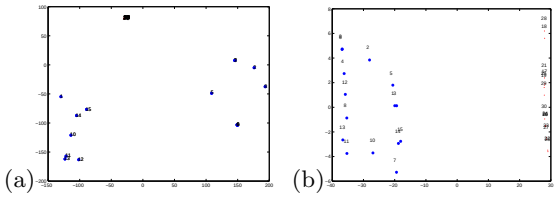


Fig. 2. MDS for each distance matrices. (a) MDS of Umeyama approach. (b) MDS of our graph distance.

two classes can not clearly be separated, since some points of diverse classes are mixed together. In Fig. 2(b), two classes of images can be clustered clearly and are distributed more compactly.

The MST method is a well known clustering method from the graph theory analysis. By this approach, a minimum spanning tree for the complete graph is generated, whose nodes are images and edge weights are the distance measures between images (graphs in our experiments). By cutting all edges with weights greater than a specified threshold, subtrees are created and each subtree represents a cluster. We use the distance matrices obtained previously to implement the MST clustering and for each method a threshold that optimizes its results is selected (see Table. **1**). The MST clustering is evaluated by the *Rand index* [27] and the *Dunn index* [28]. The *Rand index* measures how closely the clusters created by the clustering algorithm match the ground truth. The *Dunn index* is a measure of the compactness and separation of the clusters and unlike the *Rand index*, the *Dunn index* is not normalized. When the distance measure is the Umeyama distance, many images of second class are clustered into the first class and three classes are detected by MST clustering. When our method is used, two classes are detected and all images are clustered correctly. These results coincide with the MDS results. In addition, the results of *Dunn index* and the *Rand index* show that the clustering using our method obtains a better separation of the graphs into compact clusters. The time consumed by our method is 39.14% less than the Umeyama one (see Table. **1**).

	Cluster			Execution time (s)	Rand Index	Dunn Index
	1	2	3			
	Images					
Umeyama's Method	3, 20, 11, 14, 5, 2, 21, 24, 4, 15, 30, 8, 6, 7, 10, 13, 1, 9, 12	16, 25, 26, 27, 28	17, 18, 22, 19, 23	5.751	0.69	0.002
Our Method	1, 5, 3, 7, 14, 15, 2, 10, 4, 12, 6, 9, 8, 11, 13	16, 20, 23, 27, 22, 26, 19, 29, 30, 24, 17, 21, 25, 18, 28		2.251	1	2.32

Table 1. MST clustering with our graph distance and Umeyama's approach

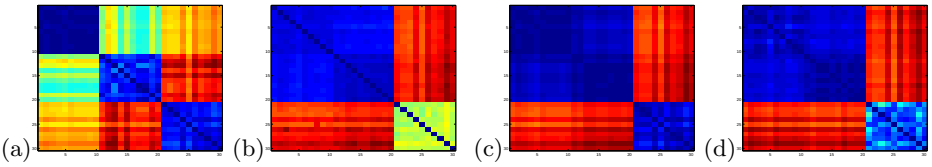


Fig. 3. Graph distance matrices. (a) results from our method; (b) results from GED from spectral seriation; (c) results from graph histograms method; (d)results from graph probing method.

Secondly, we have compared our method with the GED from spectral seriation [2], the graph histograms [14] and the graph probing [12]. The experiments consist on applying the previous tests (MDS and MST) on a database derived from COIL-100 [20] which contains different views of 3D objects. We have used three classes chosen randomly, with ten images per class. Two consecutive images in the same class represent the same object rotated by 5^o. The images are converted into graphs by feature points extraction using the Harris interest points [23] and Delaunay triangulation [24]. Finally, in order to get weighted graphs, each edge is weighted by the euclidean distance between the two points that it connect. The size of the graphs ranges from 5 to 128 nodes.

The distance matrix in Fig. 3(a) show clearly three blocks along the diagonal; thus the within-class and between-class distances are not close to each other. Whereas, in the other matrices (Fig. 3.b-d) the intensity of the first two blocks along the diagonal is close to the neighbor blocks. In addition, the MDS (see Fig. 4) and the MST clustering results (see Table. **2**) show that with our method three classes are clearly separated and the *Rand index* gets a value of 1. However, the evaluation of the separability and the compactness of the created clusters show that the graph histograms [14] has the best *Dunn index* but with two detected classes only (instead of three classes) and the graph probing has the best execution time.

From table. **2**, we can note that contrary to our method the first two classes are merged for the three methods (spectral seriation, graph histograms and graph probing). Each of these approaches uses a global description to represent graphs: the probing [12] and the graph histograms [14] methods represent each graph with only one vector, and the spectral seriation method [2] uses a string representation for graphs. Therefore, these global descriptions can not distinct differences when the graphs share similar global characteristics but not local.

Graph retrieval application. Firstly, the retrieval performance on the face expression database of Carnegie Mellon University [29] are evaluated. Secondly, the effectiveness of the proposed node signature is evaluated by performing a graph retrieval application with the GREC database [21, 22]. In the two experiments, given a query image, the system retrieves the ten most similar images from the database. The receiver-operating curve (ROC) is used to measure re-

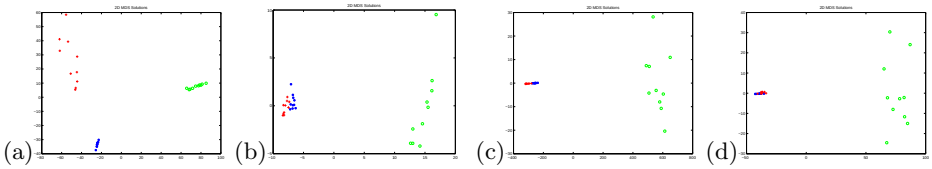


Fig. 4. MDS. (a) results from our method; (b) results from GED from spectral seriation; (c) results from graph histograms method; (d)results from graph probing method.

	Cluster			Execution time (s)	Rand Index	Dunn Index
	1	2	3			
	Images					
Spectral Seriation	18, 20, 13, 14, 17, 19, 16, 15, 11, 12, 1, 4, 9, 2, 6, 3, 10, 7, 8, 5	21, 22, 27, 23, 25, 24, 28, 26, 29, 30		1195.4	0.77	1.23
Histograms method	14, 18, 13, 17, 20, 11, 15, 16, 19, 1, 4, 7, 8, 10, 9, 5, 2, 3, 6, 12	21, 27, 22, 23, 25, 24, 28, 26, 30, 29		25.60	0.77	4.54
Graph Probing	14, 18, 13, 20, 19, 16, 17, 11, 15, 12, 2, 4, 7, 3, 6, 10, 9, 8, 1, 5	21, 29, 22, 25, 23, 24, 27, 26, 28, 30		19.46	0.77	1.78
Our method	3, 6, 2, 1, 9, 4, 7, 8, 10, 5	11, 19, 14, 17, 18, 20, 16, 12, 13, 15	21, 22, 23, 25, 24, 28, 26, 30, 27, 29	329.02	1	1.54

Table 2. MST clustering in three classes from COIL-100 : images 1-10 belong to first class, images 11-20 to the second class and images 21-30 to the third class.

trieval performances. The ROC curve is formed by Precision rate against Recall rate.

Figure **5** gives the retrieving results of our methods compared with the three methods used previously on the face database which contains 13 subjects and each subject has 75 images showing different expressions. The graphs are constructed with same manner as the previous experiment (graph clustering). The size of the graphs ranges from 4 to 17. Even though our method provides better results, the results in the figure **5** have a low performance. We can conclude that the way of the construction of the graphs is not appropriated for this kind of data.

	Node Signature	Node signature without node degree	Node signature without edge weights
A.R	60.19%	56.25 %	50.30 %

Table 3. Accuracy rate (A.R) in the GREC database

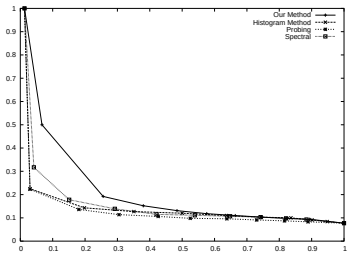
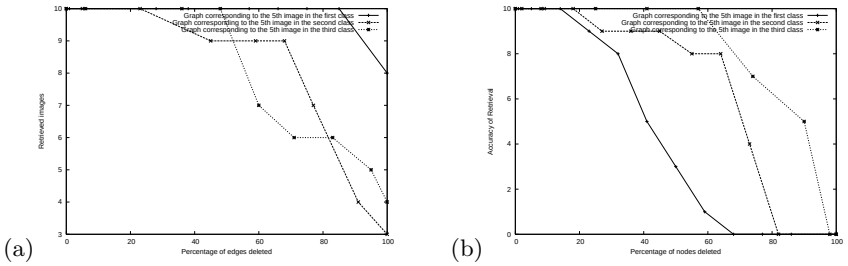


Fig. 5. Precision-Recall curves

Table **3** shows the accuracy rate of the retrieval on the GREC database making use of our graph distance as a function of the node signature. The aim of this experiment is to show the behavior of our metric when the signature about each node is defined of one of the two features either the degree of the node or the weights of the incident edges. From this experiment, we can remark that the use of the combination of the degree and the weights improves the accuracy rate. Moreover, the incident edge weights feature seems to affect more strongly the behavior of our metric because this feature provides a good specification to characterize the nodes compared with only the node degree.

Sensitivity Analysis. The aim in this section is to investigate the sensitivity of our matching method to structural differences in the graphs. Here, we have taken three classes from the COIL-100 database, each one contains 10 images. The structural errors are simulated by randomly deleting nodes and edges in the graph. The query graphs are the distorted version of the original graph representing the 5^{th} image in each class.



(a) (b)

Fig. 6. Effect of Noise for similarity queries. (b) Edges Deletion. (a) Nodes deletion

Figure 6 shows the retrieval accuracy as a function of the percentage of edges deletion (Fig. 6-a) and nodes deletion (Fig. 6-b). The retrieval accuracy degrades when the percent of edge deletion is around 22% (Fig. 6-a) and 20% of node deletion (Fig. 6-b). The main feature to denote from these plots is that our graph matching method is most robust to edge deletion, because the

edge deletion does not imply an important structural changes into the graph. It changes only some elements in the node signatures of the incident nodes of the deleted edge. In fact, the node signature procedure describes the nodes from different localization in the graph, e.g. all informations about the connected edge to the node is given. Therefore, the performance of the retrieval task is more sensitive to node deletion compared to the edge deletion.

4 Conclusion

In this work, we propose a new graph matching technique based on node signatures describing local information in the graphs. The cost matrix between two graphs is based on these signatures and the optimum matching is computed using the Hungarian algorithm. Based on this matching, we have also proposed a metric graph distance. From the experimental results, we have implicitly shown, that the nodes are well differentiated by their valence and the weights of the incident edges (considered as an unordered set) and therefore, our method provides good results to cluster and retrieve images represented by 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. A. Robles-Kelly, 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.
3. 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.
4. H. Bunke, and K. Shearer, "A graph distance metric based on the maximal common subgraph," Pattern Recognition Letters, vol. 19, pp. 255-259, 1998.
5. H. Bunke, A. Munger, and X. Jiang, "Combinatorial Search vs.Genetic Algorithms: A Case Study Based on the Generalized Median Graph Problem," Pattern Recognition Letters, vol. 20, no. 11-13, pp. 1271-1279, 1999.
6. 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
7. 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
8. 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
9. A. Shokoufandeh and S. Dickinson, "A unified framework for indexing and matching hierarchical shape structures," Lecture Notes in Computer Science, Vol. 2059, pp. 6784, 2001.
10. M. A. Eshera and K. S. Fu, "A graph distance measure for image analysis," IEEE Trans. Syst. Man Cybern., vol. 14, 398-408, 1984.

11. S. Sorlin, C. Solnon , J.M. Jolion, " A Generic Multivalent Graph Distance Measure Based on Multivalent Matchings," Applied Graph Theory in Computer Vision and Pattern Recognition, vol. 52, pp. 151-181, 2007.
12. 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.
13. A. Sanfeliu and K.S. Fu, "A Distance Measure between Attributed Relational Graphs for Pattern Recognition." IEEE Trans. Systems, Man, and Cybernetics, vol. 13, no. 353-362, 1983.
14. N. P. Apostolos, and M. Yannis, "Structure-Based Similarity Search with Graph Histograms," Proc. of the 10th International Workshop on Database & Expert Systems Applications, 1999.
15. 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.
16. R. K. Chung FAN , "Spectral Graph Theory," AMS Publications, 1997.
17. L. Xu, and I. King, "A PCA approach for fast retrieval of structural patterns in attributed graphs," IEEE Trans. Systems, Man, and Cybernetics, vol. 31, no. 5, pp. 812-817, 2001.
18. B. Luo and E.R. Hancock, "Structural Graph Matching Using the EM Algorithm and Singular Value Decomposition,"IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 23, no. 10, pp. 1120-1136, 2001.
19. H. W. Kuhn, "The Hungarian method for the assignment problem." Naval Research Logistic Quarterly, vol. 2, pp. 83-97, 1955.
20. S.A. Nene, S.K. Nayar, and H. Murase, "Columbia Object Image Library (COIL-100)," technical report, Columbia Univ., 1996.
21. 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.
22. P. Dosch and E. Valveny, "Report on the Second Symbol Recognition Contest," Proc. 6th IAPR Workshop on Graphics Recognition, pp. 381-397, 2005.
23. C. Harris and M. Stephens, "A combined corner and edge detection, " Proc. 4th Alvey Vision Conf., pp. 189-192, 1988.
24. S. Fortune, "Voronoi diagrams and Delaunay triangulations," Computing in Euclidean Geometry, pp. 193-233, 1992.
25. C.T. Zahn, "Graph-theoretical methods for detecting and describing Gestalt clusters," IEEE Trans. on Computers, vol. C-20, pp. 68-86, 1971.
26. T. Hofmann and J.M. Buhmann, "Multidimensional Scaling and Data Clustering," Advances in Neural Information Processing Systems (NIPS 7), Morgan Kaufmann Publishers, pp. 459-466, 1995.
27. W.M. Rand, "Objective criteria for the evaluation of clustering methods," Journal of the American Statistical Association 66, pp. 846-850, 1971.
28. J. Dunn, "Well separated clusters and optimal fuzzy partitions," Journal of Cybernetics, Vol. 4, no. 1, pp. 95-104, 1974.
29. Carnegie Mellon University face expression database: http://amp.ece.cmu.edu/downloads.htm.