A Significant Improvement of Softassign with Diffusion Kernels

Miguel Angel Lozano and Francisco Escolano

Robot Vision Group
Departamento de Ciencia de la Computación e Inteligencia Artificial
Universidad de Alicante, Spain
{malozano,sco}.dccia.ua.es
http://rvg.ua.es

Abstract. In this paper we propose a simple way of significantly improving the performance of the Softassign graph-matching algorithm of Gold and Rangarajan. Exploiting recent theoretical results in spectral graph theory we use diffusion kernels to transform a matching problem between unweighted graphs into a matching between weighted ones in which the weights rely on the entropies of the probability distributions associated to the vertices after kernel computation. In our experiments, we report that weighting the original quadratic cost function results in a notable improvement of the matching performance, even in medium and high noise conditions.

1 Introduction

Energy-minimization approaches to graph matching [4][5][8] rely on transforming the discrete search space into a continuous one and then exploiting optimization techniques to find a, typically approximate, solution. One of the first algorithms, Softassign, the well-known graduated assignment method introduced by Gold and Rangarajan [4], optimizes a quadratic cost function through a low-order computational complexity process which updates the assignment variables encoding the matching proposals. However, it has been reported that the performance of the algorithm decays significantly at mid and high levels of structural corruption, and also that such a decay can be attenuated by optimizing an alternative non-quadratic energy function [5]. In this paper we report comparable results by weighting the quadratic cost function properly. This is due to the fact that we transform the original matching problem between two non-attributed graphs into a matching problem between attributed ones and then these attributes are used to weight the original cost function. The practical effect of this weighting is that it yields a good characterization of the local structure, which in turn helps to choose the proper attractor in a context of high ambiguity.

We address the key point of extracting good attributes for the nodes of the non-attributed graphs by exploiting recent theoretical results in spectral graph theory [1]: the definition of diffusion kernels on graphs [6] and their generalization to other families of kernels [13]. These latter works have transferred to

A. Fred et al. (Eds.): SSPR&SPR 2004, LNCS 3138, pp. 76–84, 2004.

the discrete domain of graphs the concept of a *kernel*, originally defined in the vector domain (see [3] for a survey on kernels for structures like strings, trees and graphs). Kernels, are key concepts in the context of statistical learning theory[2][12][7] which capture the structure of a domain by defining a similarity measure between two input elements in the domain. Such a similarity measure relies on the inner product of the results of mapping both inputs to a, usually higher dimensional, Hilbert space. Due to the so-called *kernel trick* such a mapping is implicitly defined once the kernel is specified, and the benefit of such a transformation consists on transforming non-linear relations between the inputs in the original domain into linear relations after the mapping. For instance, in the context of support-vector machines (in general we can talk about kernel machines), the task of classifying two non-linearly separable inputs is accomplished by using a suitable kernel to map them to another space in which these inputs are linearly separable (it works in the well-known two-spirals example).

When applied to graphs, kernels provide a similarity measure between the vertices of the same graph. In the case of diffusion kernels, such a similarity can be seen as the sum of probabilities of all paths connecting such vertices, and it is computed from the matrix exponentiation of the Laplacian of the adjacency matrix (section 2). As the Laplacian encodes information about the local structure of the graph, the global structure emerges in the kernel. However, we do not use directly the probabilities of connecting paths because they may change very easily when the graph is edited or corrupted, and, consequently, they are not useful for finding corresponding vertices. What we do is to is to compute a characteristic measure of the distribution of probabilities associated to paths emanating from a given vertex, the entropy of such a distribution, and use it as attribute for that vertex. The entropy of the probabilities associated to connecting paths is more stable and allows us to find correct matches (section 3). In section 4 we present the *kernelized* version of the quadratic cost function and its implications in the Softassign process. Our results are showed in section 5 and in 6 we present our conclusions and future work.

2 Diffusion Kernels on Graphs

Given a undirected and unweighted graph $G = (V, E)$ with vertex-set V of size m, and edge-set $E = \{(i, j)|(i, j) \in V \times V, i \neq j\}$, its respective adjacency matrix is defined as usual:

$$A_{ij} = \begin{cases} 1 & \text{if } (i,j) \in E \\ 0 & \text{otherwise} \end{cases}$$

and the diagonal degree matrix is defined by

$$D_{ij} = \begin{cases} \sum_{j=1}^{n} A_{ij} & \text{if } i = j \\ 0 & \text{otherwise .} \end{cases}$$

Then, the Laplacian of G is defined as $L = D - A$, that is,

$$L_{ij} = \begin{cases} -1 & \text{if } (i,j) \in E \\ D_{ii} & \text{if } i = j \\ 0 & \text{otherwise .} \end{cases}$$

Following [6] the associated diffusion kernel K is the result of the matrix exponentation

$$K = e^{-\beta L} = \lim_{n \to \infty} \left(1 - \frac{\beta L}{n}\right)^n, \tag{1}$$

and after solving the latter limit we obtain

$$e^{-\beta L} = I_m + L + \frac{1}{2!}L^2 + \frac{1}{3!}L^3 + \ldots, \tag{2}$$

where I_m is the $m \times m$ identity matrix. Moreover, $e^{-\beta L}$ is the solution of the heat equation [1]

$$\frac{d}{d\beta}K_\beta = -LK_\beta. \tag{3}$$

As L is symmetric, the solution $K = e^{-\beta L}$, the Gram matrix, satisfies the positive semi-definiteness condition for kernels. Although in this paper we will focus on diffusion kernels, this framework is generalized in [13] where a family of graph kernels is proposed in the context of regularization.

3 Diffusion Kernels and Node Entropy

On behalf of the so-called *kernel trick* the $m \times m$ matrix K defines a real-valued function between pairs of vertices, and K_{ij} can be interpreted as the inner product of the mappings of both vertices to a Hilbert space [12]. This means that such a inner product encodes the similarity between pairs of vertices in a possibly high-dimensional space. But, from the point of view of discrete structures what is interesting of such similarity is that as L encodes the local structure of V in G, the global structure emerges in K.

More precisely, and due to the fact that K is the solution of the heat equation, the diffusion kernel K is the version for discrete spaces of the Gaussian kernel for $\mathbb{R}^m$ with variance $\sigma^2 = 2\beta$, that is, the value of K_{ij} decays exponentially with the *distance* between i and j. But, how to apply this idea to a graph? From the point of view of random fields, the diffusion kernel K relies on the covariance matrix of a stochastic process in which each vertex has attached a random variable of zero mean an variance σ^2 and each variable sends a small fraction of its value to its neighbors. In this regard, K_{ij} can be interpreted as the amount of substance accumulated at vertex j after a given amount of time after injecting the substance at i and let it diffuse through the edges of the graph. The more *distant* are i and j the less amount we have.

In terms of random walks, K_{ij} can be regarded as the sum of probabilities that a *lazy* random walk takes each path from i to j [6]. A lazy random walk over the undirected graph G and with parameter β is a stochastic process which will take each of the edges emanating from i with a fixed probability β and will remain

in i with probability $1 - \beta D_{ii}$, being $\beta \leq 1/(max_i D_{ii})$. From this point of view, the final value of K_{ij} depends on the edge distribution and branching process between i and j. If j is an isolated node, then $K_{ij} = 0\ \forall i \neq j$ and $K_{jj} = 1$. Moreover, as each row i of K satisfies that $0 \leq K_{ij} \leq 1\ \forall j$ and $\sum_{j=1}^{m} K_{ij} = 1$, then we can consider each row as a probability distribution associated to vertex i. This allows us to build a proper attribute for each vertex in terms of the shape of the corresponding distribution. In our initial experiments we have found that as edit operations or noise addition on the graph will give a different kernel in terms of the number of nodes and edges, and obviously in terms of the diffusion process, building attributes in the properties of the distributions yields more stability that building such attributes in individual values of K_{ij}. This is why we retain as attribute for node i the entropy of the distribution

$$H_i^K = -\sum_{j=1}^{m} K_{ij} \log K_{ij} \ . \tag{4}$$

As we will see in the following sections, although this attribute does not provide a good discrimination between vertices it is very helpful in the continuation process in which Softassign relies. In fact, the kernel approach is closely related to the use of distance matrices in matching and tests for isomorphism [11], and, more recently, to the use of powers of the adjacency matrix [14].

In order to clarify the concept of kernel and node entropy, in Fig. 1 we show two graphs in which the smaller one X is a subgraph of the other, Y. We show the kernels of both of them and the distribution of the vertex 1 of Y.

4 Kernelizing Softassign

Given two graphs $G_X = (V_X, E_X)$, with nodes $a \in V_X$ and edges $(a, b) \in E_X$, and $G_Y = (V_Y, E_Y)$, with nodes $i \in V_Y$ and edges $(i, j) \in E_Y$, their adjacency matrices X and Y are defined by

$$X_{ab} = \begin{cases} 1 & \text{if } (a,b) \in E_X \\ 0 & \text{otherwise} \end{cases} \quad \text{and} \quad Y_{ij} = \begin{cases} 1 & \text{if } (i,j) \in E_Y \\ 0 & \text{otherwise} \ . \end{cases}$$

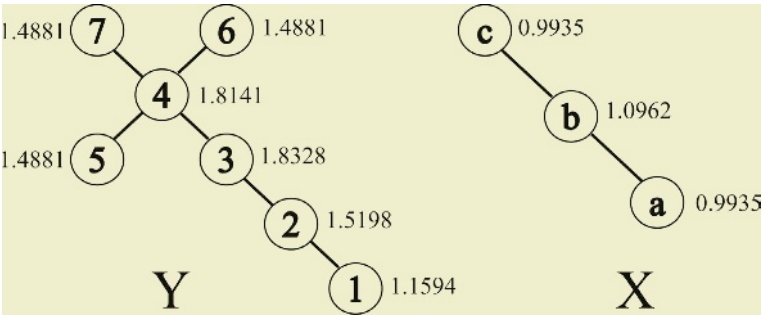
A feasible solution to the graph matching problem between G_X and G_Y is encoded by a matrix M of size $m \times n$, being $m = |V_X|$ and $n = |V_Y|$, with binary variables

$$M_{ai} = \begin{cases} 1 & \text{if } a \in V_X \text{ matches } i \in V_Y \\ 0 & \text{otherwise} \end{cases}$$

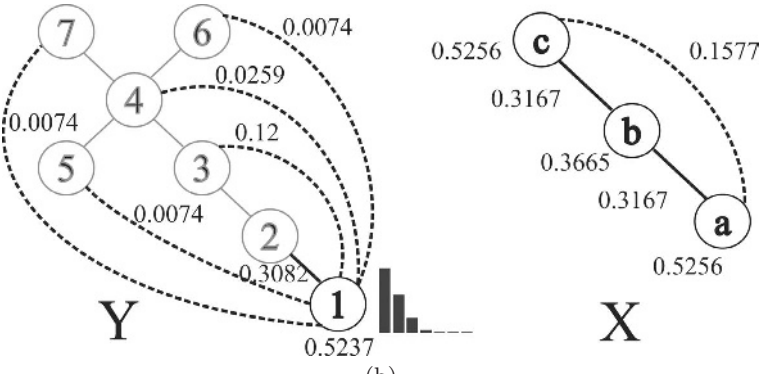
satisfying the constraints defined respectively over the rows and columns of M

$$\sum_{i=1}^{m+1} M_{ai} = 1, \forall a \quad \text{and} \quad \sum_{a=1}^{n+1} M_{ai} = 1, \forall i \ , \tag{5}$$

where equality comes from introducing slack variables for registering outliers.



(a)



(b)

$$K_Y = \begin{bmatrix} .5237 & .3082 & .1200 & .0259 & .0074 & .0074 & .0074 \\ .3082 & .3356 & .2140 & .0645 & .0259 & .0259 & .0259 \\ .1200 & .2140 & .2800 & .1369 & .0830 & .0830 & .0830 \\ .0259 & .0645 & .1369 & .1906 & .1940 & .1940 & .1940 \\ .0074 & .0259 & .0830 & .1940 & .4752 & .1073 & .1073 \\ .0074 & .0259 & .0830 & .1940 & .1073 & .4752 & .1073 \\ .0074 & .0259 & .0830 & .1940 & .1073 & .1073 & .4752 \end{bmatrix}$$

(c)

$$K_X = \begin{bmatrix} .5256 & .3167 & .1577 \\ .3167 & .3665 & .3167 \\ .1577 & .3167 & .5256 \end{bmatrix}$$

(d)

Fig. 1. Illustrating graph kernels and entropy. Example graphs X and Y where nodes are labelled with their entropies (a). Kernel values and distribution for vertex 1 of graph Y, and kernel values for all vertices in graph X (b). Kernel K_Y (c) and kernel K_X (d).

Following the Gold and Rangarajan formulation we are interested in finding the feasible solution M that minimizes the following cost function,

$$F(M) = -\frac{1}{2}\sum_{a=1}^{m}\sum_{i=1}^{n}\sum_{b=1}^{m}\sum_{j=1}^{n} M_{ai}M_{bj}C_{aibj} \ , \tag{6}$$

where typically $C_{aibj} = X_{ab}Y_{ij}$, that is, when $a \in V_X$ matches $i \in V_Y$, it is desirable that nodes b adjacent to a (with $X_{ab} \neq 0$) and nodes j adjacent to

i (with $Y_{ij} \neq 0$) also match, that is $M_{ai} = M_{bj} = 1$. This is the well known rectangle rule (in maximization terms we want to obtain as more rectangles as possible). Furthermore, considering the entropies defined in the previous section a simple way of *kernelizing* the latter energy function is to redefine C_{aibj} as

$$C^K_{aibj} = X_{ab}Y_{ij} \exp -[(H^{K_X}_a - H^{K_Y}_i)^2 + (H^{K_X}_b - H^{K_Y}_j)^2] \,, \tag{7}$$

where the entropies H^{K_X} and H^{K_Y} are associated to the kernels

$$K_X = e^{-\frac{\beta}{m}L_X} \text{ and } K_Y = e^{-\frac{\beta}{n}L_Y} \,,$$

that is, we normalize the decays by the number of nodes in each graph in order to make both diffusion processes, and consequently both kernels, comparable. This normalization is useful in big graphs, where it contributes to avoid the tendency of the diffusion process towards uniform distributions, but makes no sense in small graphs. But, normalization apart, the latter definition of C^K_{aibj} ensures that $C^K_{aibj} \leq C_{aibj}$, and the equality is only verified when nodes a and i have similar entropies, and the same for nodes b and j. In practice, this weights the rectangles in such a way that rectangles with compatible entropies in their opposite vertices are preferred, and otherwise they are underweighted.

Paying now attention to the deterministic annealing process implemented by Softassign, the assignment variables are updated by

$$M_{ai} = \exp\left[-\frac{1}{T}\frac{\partial F}{\partial M_{ai}}\right] = \exp\left[\frac{1}{2T}\sum_{i=a}^{m} M_{bj}C^K_{aibj}\right] \,,$$

where T is the temperature control parameter. Then, these assignments feed a Sinkhorn process [10], which iteratively normalizes rows and columns. After this process we obtain a doubly stochastic matrix, decrease T and a new iteration begins. The final doubly stochastic matrix is transformed into a permutation matrix by a proper clean-up process.

To see intuitively the difference between the classical Softassign and the kernelized one, in Fig. 2 we show the evolution of both algorithms for the two example graphs showed in Fig. 1. The classical Softassign prefers clearly the assignment $(b, 4)$ which is consistent with the cardinality heuristic(notable vertices in X prefer notable vertices in Y. However, a and c can be assigned either with 3, 5, 6 or 7 (ambiguity). On the other hand, in the kernelized case, the assignment $(b, 2)$ is clearly preferred and a and c may be assigned either to 1 or 3. The cardinality heuristic is inhibited in favor of a structural compatibility heuristic.

5 Experiments

We have performed several matching experiments with graphs of 50 nodes, and considering two levels of edge density: 25% and 50%. These levels of edge density are relatively high because we want to study the performance of the kernelized Softassign which it is assumed to have more problems in this context, because

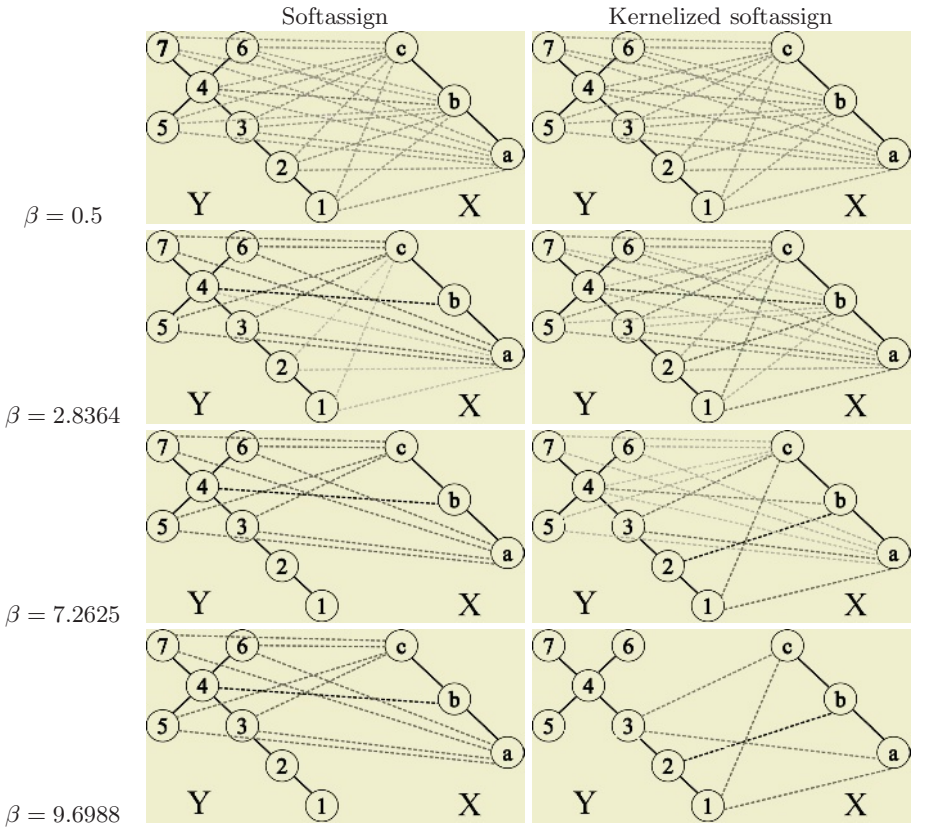


Fig. 2. Evolution of the algorithm for a simple matching problem. Matching matrices for many values of β for the classical Softassign and the kernelized version.

the difussion processes tend to generate uniform distributions. In all cases we use the classical initialization of Softassign. Each point corresponds to the averaged result for 100 graphs randomly generated. We have considered different noise levels: from 0% (isomorphism) to 50%. We have registered both the fraction of complete graphs successfully matched and the fraction of nodes successfully matched. In all cases the kernelized version outperforms significantly the classical one. Moreover, the kernelized version is also better than an attributed one with

$$C_{aibj} = X_{ab} Y_{ij} \exp\left[-\frac{\left| \sum_{b=1}^{m} X_{ab} - \sum_{j=1}^{n} Y_{ij} \right|}{\min(m,n)} \right] ,$$

that is, a Softassign version relying on node cardinality.

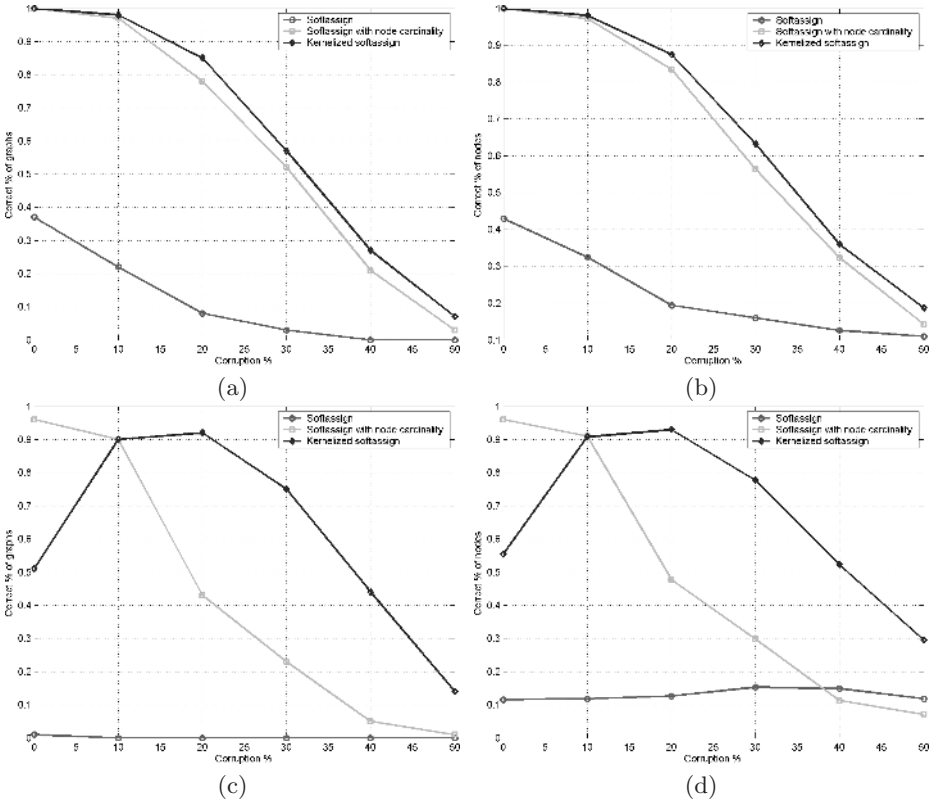


Fig. 3. Matching results. Graphs (a) and nodes (b) successfully matched with an edge density of 25%. Graphs (c) and nodes (d) successfully matched with an edge density of 50% (b).

6 Conclusions and Future Work

In this paper we have introduced a simple way of improving the performance of the Softassign graph-matching algorithm through the kernelization of the classical quadratic cost function. Our experimental results indicate that such an improvement is significant even in medium and high noise levels. Current and future work in this context includes the kernelization of other energy minimization and state-space algorithms, the formalization of the edit distance in terms of kernels, and the comparison with other approaches relying on node-neighborhood attributes.

Acknowledgements

This work was partially supported by grant $TIC2002-02792$ funded by *Ministerio de Ciencia y Tecnología* of the Spanish Government and by *FEDER*.

References

1. Chung, F.R.K.: Spectral Graph Theory. Conference Board of the Mathematical Sciences (CBMS) **92**. American Mathematical Society (1997)
2. Cristianini, N., Shawe-Taylor, J.: An Introduction to Support Vector Machines Cambridge University Press (2000)
3. Gärtner: A Survey of Kernels for Structured Data. ACM SIGKDD Explorations Newsletter **5**(1) (2003) 49–58
4. Gold, S., Rangarajan, A.: A Graduated Assignment Algorithm for Graph Matching. IEEE Transactions on Pattern Analysis and Machine Intelligence **18** (4) (1996) 377-388
5. Finch, A.M., Wilson, R.C., Hancock, E.: An Energy Function and Continuous Edit Process for Graph Matching. Neural Computation, **10** (7) (1998) 1873-1894
6. Kondor, R.I., Lafferty, J.: Diffusion Kernels on Graphs and other Discrete Input Spaces. In: Sammut, C., and Hoffmann, A. G. (eds) Machine Learning, Proceedings of the Nineteenth International Conference (ICML 2002). Morgan Kaufmann (2002) 315–322
7. Müller, K.-R., Mika, S., Räshc, Tsuda, K., Schölkopf, B.: An Introduction to Kernel-based Learning Algorithms. IEEE Transactions on Neural Networks, **12**(2) (2001) 181–201.
8. Pelillo, M.: Replicator Equations, Maximal Cliques, and Graph Isomorphism. Neural Computation **11** (1999) 1933–1955
9. Robles-Kelly, A., Hancock, E.: Graph Matching Using Spectral Seriation. In: Rangaraja, A., Figueiredo, M., and Zerubia, J. (eds) Energy Minimization Methods in Computer Vision and Pattern Recognition, Proceedings of the 4th International Workshop, EMMCVPR 2003. Lecture Notes in Computer Science. Springer. Vol **2683** (2003) 517–532
10. R. Sinkhorn.: A Relationship Between Arbitrary Positive Matrices and Doubly Stochastic Matrices. Annals of Mathematical Statistics **35** (1964) 876–879
11. Schmidt, D.C., Druffel, L.E.: A Fast Backtracking Algorithm to Test Direct Graphs for Isomorphism Using Distance Matrices. Journal of the ACM **23** (3) (1976) 433-445
12. Schölkopf, B., Smola, A.: Learning with Kernels. MIT Press (2002).
13. Smola, A., Kondor, R.I.: Kernels and Regularization on Graphs. In: Schölkopf,B., and Warmuth, M. K. (eds) Computational Learning Theory and Kernel Machines, 16th Annual Conference on Computational Learning Theory and 7th Kernel Workshop, COLT/Kernel 2003. Lecture Notes in Computer Science. Springer. Vol. **2777** (2003) 144–158
14. DePiero, F.W., Trivedi, M., Serbin, S.: Graph Matching Using a Direct Classification of Node Attendance. Pattern Recognition, Vol. **29**(6) (1996) 1031–1048
15. Ozer, B., Wolf, W., Akansu, A.N.: A Graph Based Object Description for Information Retrieval in Digital Image and Video Libraries. In: Proceedings of the IEEE Workshop on Content-Based Access of Image and Video Libraries (1999) 79–83