

Distance Measures between Attributed Graphs and Second-Order Random Graphs

Francesc Serratosa¹ and Alberto Sanfeliu²

¹ Universitat Rovira I Virgili, Dept. d'Enginyeria Informàtica i Matemàtiques, Spain
Francesc.Serratosa@etse.urv.es
<http://www.etse.urv.es/~fserrato>

² Universitat Politècnica de Catalunya, Institut de Robòtica i Informàtica Industrial, Spain
sanfeliu@iri.upc.es

Abstract. The aim of this article is to propose a distance measure between Attributed Graphs (AGs) and Second-Order Random Graphs (SORGs) for recognition and classification purposes. The basic feature of SORGs is that they include both marginal probability functions and joint probability functions of graph elements (vertices or arcs). This allows a more precise description of both the structural and semantic information contents in a set (or cluster) of AGs and, consequently, an expected improvement in graph matching and object recognition. The distance measure is derived from the probability of instantiating a SORG into an AG.

SORGs are shown to improve the performance of other random graph models such as FORGs and FDGs and also the direct AG-to-AG matching in two experimental recognition tasks.

1 Introduction

Some attempts have been made to try to reduce the computational time of matching the unknown input patterns to the whole set of models from the database. Assuming that the AGs that represent a cluster or class are not completely dissimilar in the database, only one structural model is defined from the AGs that represent the cluster, and thus, only one comparison is needed for each cluster.

One of the most common methodologies are based on keeping the probabilistic information in the structure that represent the cluster of AGs. These models, which are usually called *Random Graphs* (RGs), are described in the most general case through a joint probability space of random variables ranging over graph vertices and arcs. They are the union of the AGs in the cluster, according to some synthesis process, together with its associated probability distribution. In this manner, a structural pattern can be explicitly represented in the form of an AG and an ensemble of such representations can be considered as a set of outcomes of the RG. The most important probabilistic methods are First-Order Random Graphs (FORGs) [4], the Sengupta method [3], Function-Described Graphs (FDGs) [1,6,7] and Second-Order Random Graphs (SORGs), which can be seen as a generalisation of both of them [5].

In the following section, we introduce the formal definitions used throughout the paper. In section 3, we recall the general formulation for estimating the joint prob-

ability of the random elements in a RG synthesised from a set of AGs. In section 4, we present the new distance measure between AGs and SORGs derived from the joint probabilities of the random elements. Finally, we present a comparative study between SORGs and FORGs, FDGs and direct AG-to-AG matching. In the last section, we provide some discussion about our distance measure.

2 Formal Definitions of Random-Graph Representation

Definition 1: Attributed Graph (AG). Let Δ_v and Δ_e denote the domains of possible values for attributed vertices and arcs, respectively. These domains are assumed to include a special value Φ that represents a *null value* of a vertex or arc. An AG G over (Δ_v, Δ_e) is defined to be a four-tuple $G = (\Sigma_v, \Sigma_e, \gamma_v, \gamma_e)$, where $\Sigma_v = \{v_k \mid k = 1, \dots, n\}$ is a set of vertices (or nodes), $\Sigma_e = \{e_{ij} \mid i, j \in \{1, \dots, n\}, i \neq j\}$ is a set of arcs (or edges), and the mappings $\gamma_v : \Sigma_v \rightarrow \Delta_v$ and $\gamma_e : \Sigma_e \rightarrow \Delta_e$ assign attribute values to vertices and arcs, respectively.

Definition 2: Random Graph (RG). Let Ω_v and Ω_e be two sets of random variables with values in Δ_v (random vertices) and in Δ_e (random arcs), respectively. A *random-graph structure* R over (Δ_v, Δ_e) is defined to be a tuple $(\Sigma_v, \Sigma_e, \gamma_v, \gamma_e, P)$, where $\Sigma_v = \{\omega_k \mid k = 1, \dots, n\}$ is a set of vertices, $\Sigma_e = \{e_{ij} \mid i, j \in \{1, \dots, n\}, i \neq j\}$ is a set of arcs, the mapping $\gamma_v : \Sigma_v \rightarrow \Omega_v$ associates each vertex $\omega_k \in \Sigma_v$ with a random variable $\alpha_k = \gamma_v(\omega_k)$ with values in Δ_v , and $\gamma_e : \Sigma_e \rightarrow \Omega_e$ associates each arc $e_{ij} \in \Sigma_e$ with a random variable $\beta_k = \gamma_e(e_{ij})$ with values in Δ_e . And, finally, P is a joint probability distribution $P(\alpha_1, \dots, \alpha_n, \beta_1, \dots, \beta_m)$ of all the random vertices $\{\alpha_i \mid \alpha_i = \gamma_v(\omega_i), 1 \leq i \leq n\}$ and random arcs $\{\beta_j \mid \beta_j = \gamma_e(e_{kl}), 1 \leq j \leq m\}$.

Definition 3: Probability of a RG instantiation. Given an AG G and a RG R , the joint probability of random vertices and arcs is defined over an instantiation that produces G , and such instantiation is associated with a structural isomorphism $\mu : G' \rightarrow R$, where G' is the extension of G to the order of R . G' represents the same object than G but some vertices or arcs have been added with the null value Φ to be μ bijective. Let G be oriented with respect to R by the structurally coherent isomorphism μ ; for each vertex ω_i in R , let $\mathbf{a}_i = \gamma_v(\mu^{-1}(\omega_i))$ be the corresponding attribute value in G' , and similarly, for each arc e_{kl} in R (associated with random variable β_j) let $\mathbf{b}_j = \gamma_e(\mu^{-1}(e_{kl}))$ be the corresponding attribute value in G' . Then the *probability of G according to* (or given by) *the orientation μ* , denoted by $P_R(G|\mu)$, is defined as

$$P_R(G|\mu) = \Pr\left(\bigwedge_{i=1}^n (\alpha_i = \mathbf{a}_i) \wedge \bigwedge_{j=1}^m (\beta_j = \mathbf{b}_j)\right) = p(\mathbf{a}_1, \dots, \mathbf{a}_n, \mathbf{b}_1, \dots, \mathbf{b}_m) \quad (1)$$

We define \mathbf{d}_i to represent a vertex or arc attribute value (\mathbf{a}_i or \mathbf{b}_i). Thus, if s is the number of vertices and arcs, $s=m+n$, eq. (1) can be rewritten as,

$$P_R(G|\mu) = p(\mathbf{d}_1, \dots, \mathbf{d}_s) \tag{2}$$

3 Second-Order Random-Graph Representation

If we want to represent the cluster of AGs by a RG, it is impractical to consider the high order probability distribution defined in the RGs $P(\alpha_1, \dots, \alpha_n, \beta_1, \dots, \beta_m)$ (definition 2), where all components and their relations in the structural patterns are taken jointly due to time and space costs. For this reason, some other more practical approaches have been presented that propose different approximations [3,4,5,7]. All of them take into account in some manner the incidence relations between attributed vertices and arcs, i.e. assume some sort of dependence of an arc on its connecting vertices. Also, a common ordering (or labelling) scheme is needed that relates vertices and arcs of all the involved AGs, which is obtained through an optimal graph mapping process called synthesis of the random graph representation. We showed in [5] that all the approximations in the literature of the joint probability of an instantiation of the random elements in a RG (eq. 1) can be described in a general form as follows:

$$P_R(G|\mu) = p(\mathbf{a}_1, \dots, \mathbf{a}_n, \mathbf{b}_1, \dots, \mathbf{b}_m) = \prod_{i=1}^n p_i(\mathbf{a}_i) \prod_{i=1}^m p_i(\mathbf{b}_i) \prod_{i=1}^{n-1} \prod_{j=i+1}^n r_{ij}(\mathbf{a}_i, \mathbf{a}_j) \prod_{i=1}^n \prod_{j=1}^m r_{ij}(\mathbf{a}_i, \mathbf{b}_j) \prod_{i=1}^{m-1} \prod_{j=i+1}^m r_{ij}(\mathbf{b}_i, \mathbf{b}_j) \tag{3}$$

where p_i are the marginal probabilities of the s random elements γ_i , (vertices or arcs) and r_{ij} are the Peleg compatibility coefficients [2] that take into account both the marginal and 2nd-order joint probabilities of random vertices and arcs.

According to eq. (2), we can generalise the joint probability as,

$$P_R(G|\mu) = p(\mathbf{d}_1, \dots, \mathbf{d}_s) = \prod_{i=1}^s p_i(\mathbf{d}_i) \prod_{i=1}^s \prod_{j=i+1}^s r_{ij}(\mathbf{d}_i, \mathbf{d}_j) \tag{4}$$

and define the Peleg coefficient,

$$r_{ij}(\mathbf{d}_i, \mathbf{d}_j) = \frac{p_{ij}(\mathbf{d}_i, \mathbf{d}_j)}{p_i(\mathbf{d}_i)p_j(\mathbf{d}_j)} \tag{5}$$

The Peleg coefficient, with a non-negative range, is related to the “degree” of dependence between two random variables. If they are independent, the joint probability, p_{ij} , is defined as the product of the marginal ones, thus, $r_{ij} = 1$ (or a value close to 1 if the probability functions are estimated). If one of the marginal probabilities is null, the joint probability is also null. In this case, the indecisiveness $0/0$ is solved as 1, since this do not affect the global joint probability, which is null.

4 Distance Measure between AGs and SORGs

The distance measure presented in this section provides a quantitative value of the match between an AG G (data graph) and a SORG S (model graph) similar to the one

presented in [1]. It is related to the probability of G according to the labelling function $\mu : G \rightarrow S$, denoted $P(G|\mu)$ in eq. (4). We may attempt to minimise a *global cost* measure C of the morphism μ in the set H of allowable configurations, by taking the cost as a monotonic decreasing function of the conditional probability of the data graph given the labelling function, $C = f(P(G|\mu))$. For instance, $C = -\ln(P(G|\mu))$ would be a possible choice. Thus, considering eq. (4),

$$C(G|\mu) = -\ln\left(\prod_{i=1}^s p_i(\mathbf{d}_i) \prod_{i=1}^{s-1} \prod_{j=i+1}^s r_{ij}(\mathbf{d}_i, \mathbf{d}_j)\right) = -\sum_{i=1}^s \ln(p_i(\mathbf{d}_i)) - \sum_{i=1}^{s-1} \sum_{j=i+1}^s \ln(r_{ij}(\mathbf{d}_i, \mathbf{d}_j)) \quad (6)$$

and using the definition of the Peleg coefficient (eq. 5) we obtain the following equation in the case that $p_i(\mathbf{d}_i) > 0$ and $p_j(\mathbf{d}_j) > 0$

$$C(G|\mu) = -\sum_{i=1}^s \ln(p_i(\mathbf{d}_i)) - \sum_{i=1}^{s-1} \sum_{j=i+1}^s [\ln(p_{ij}(\mathbf{d}_i, \mathbf{d}_j)) - \ln(p_i(\mathbf{d}_i)) - \ln(p_j(\mathbf{d}_j))] \quad (7)$$

Rearranging the second term of the expression we arrive at the equation

$$C(G|\mu) = -\sum_{i=1}^s \ln(p_i(\mathbf{d}_i)) + (s-1) \sum_{i=1}^s \ln(p_i(\mathbf{d}_i)) - \sum_{i=1}^{s-1} \sum_{j=i+1}^s [\ln(p_{ij}(\mathbf{d}_i, \mathbf{d}_j))] \quad (8)$$

And we obtain the final expression in which the cost of the labelling monotonically depends on the probabilities provided that $\forall i : p_i(\mathbf{d}_i) > 0$.

$$C(G|\mu) = (s-2) \sum_{i=1}^s \ln(p_i(\mathbf{d}_i)) - \sum_{i=1}^{s-1} \sum_{j=i+1}^s [\ln(p_{ij}(\mathbf{d}_i, \mathbf{d}_j))] \quad (9)$$

In the case that there is only one random element any joint probability is not defined and $s=1$. Then, the global cost of the matching is $C(G|\mu) = -\ln(p_i(\mathbf{d}_i))$. Moreover, in the case that there is a couple of random elements ($s=2$), the cost depends only on the joint probability, $C(G|\mu) = -\ln(p_{12}(\mathbf{d}_1, \mathbf{d}_2))$, although it has to be considered that $p_1(\mathbf{d}_1) > 0$ and $p_2(\mathbf{d}_2) > 0$. And finally, in the case that there are 3 random elements ($s=3$), the cost is defined as, $C(G|\mu) = \ln(p_1(\mathbf{d}_1)) + \ln(p_2(\mathbf{d}_2)) + \ln(p_3(\mathbf{d}_3)) - \ln(p_{12}(\mathbf{d}_1, \mathbf{d}_2)) - \ln(p_{13}(\mathbf{d}_1, \mathbf{d}_3)) - \ln(p_{23}(\mathbf{d}_2, \mathbf{d}_3))$. In the last case (and all the cases that $s \geq 3$, the marginal probabilities make the global cost decrease and the joint probabilities make the cost increase.

4.1 Assuming Independence between Random Elements

In the case that the random elements are independent, the joint probability is defined as the product of the marginal ones, $p_{ij}(\mathbf{d}_i, \mathbf{d}_j) = p_i(\mathbf{d}_i)p_j(\mathbf{d}_j)$. Thus, eq. (9) is rewritten as,

$$\begin{aligned}
 C(G|\mu) &= (s-2)\sum_{i=1}^s \ln(p_i(\mathbf{d}_i)) - \sum_{i=1}^{s-1} \sum_{j=i+1}^s [\ln(p_i(\mathbf{d}_i)p_j(\mathbf{d}_j))] = \\
 &= (s-2)\sum_{i=1}^s \ln(p_i(\mathbf{d}_i)) - \sum_{i=1}^{s-1} \sum_{j=i+1}^s [\ln(p_i(\mathbf{d}_i)) + \ln(p_j(\mathbf{d}_j))] = \tag{10} \\
 &= (s-2)\sum_{i=1}^s \ln(p_i(\mathbf{d}_i)) - (s-1)\sum_{i=1}^s \ln(p_i(\mathbf{d}_i))
 \end{aligned}$$

Thus, the final equation is,

$$C(G|\mu) = -\sum_{i=1}^s \ln(p_i(\mathbf{d}_i)) \tag{11}$$

Note that this expression could be obtained by considering eq. (4) and (6) with $r_{ij} = 1$, that is, assuming independence between the random elements.

4.2 Approximating the Distance Using Bounded Individual Costs

Using the above distance, only that one graph element had a probability of zero, the global joint probability would be zero and C would be infinite. Since this may happen due to the noisy presence of an unexpected element or the absence of a model’s element, only that one graph element were not properly mapped, the involved graphs would be wrongly considered to be completely different. We must therefore admit the possibility of both extraneous and missing elements in the data graphs, since the data extracted from the information sources (e.g. images) will usually be noisy, incomplete or uncertain. As a consequence, the matches for which $P(G|\mu) = 0$ should not be discarded since they could be the result of a noisy feature extraction and graph formation. In addition, a model (SORG) should match to a certain degree not only the objects (AGs) in its learning set but also the ones that are “near”.

Hence, it is more appropriate for practical purposes to decompose the global cost C into the sum of some *bounded* individual costs, one for each of the graph element matches (first-order costs on the marginal probabilities) and one for each relation between a pair of element matches (second-order costs on the joint probabilities)

$$C(G|\mu) = -(s-2)\sum_{i=1}^s C_i^1(\mathbf{d}_i) + \sum_{i=1}^{s-1} \sum_{j=i+1}^s C_{i,j}^2(\mathbf{d}_i, \mathbf{d}_j) \tag{12}$$

where first-order costs are given by

$$C_i^1(\mathbf{d}_i) = Cost(p_i(\mathbf{d}_i)) \tag{13}$$

and second-order costs are given by

$$C_{i,j}^2(\mathbf{d}_i, \mathbf{d}_j) = Cost(p_{i,j}(\mathbf{d}_i, \mathbf{d}_j)) \tag{14}$$

and the function $Cost(Pr)$ yields a bounded normalised cost value between 0 and 1 depending on the negative logarithm of a given probability Pr and parameterised by a positive constant $K_{pr} \in [0,1]$, which is a threshold on low probabilities that is introduced to avoid the case $\ln(0)$, which would give negative infinity. This is,

$$Cost(\text{Pr}) = \begin{cases} \frac{-\ln(\text{Pr})}{-\ln(K_{\text{pr}})} & \text{if } \text{Pr} \geq K_{\text{pr}} \\ 1 & \text{otherwise} \end{cases} \quad (15)$$

Once a cost measure C is defined, a distance measure between an AG and a SORG and the optimal labelling μ^* are defined respectively as

$$d = \min_{\mu \in H} \{C(G|\mu)\} \quad \text{and} \quad \mu^* = \arg \min_{\mu \in H} \{C(G|\mu)\} \quad (16)$$

The algorithm we use to calculate d and μ^* is a classical recursive tree search procedure, where the search space is reduced by a *branch and bound* technique (not described here due to lack of space).

5 Results

We carried out two different types of experiments to assess the usefulness of our new representation and to compare it with some other representations presented in the literature. In the first experiments, the AGs were synthetically generated varying some parameters such as the number of vertices or the distance between the AGs in their clusters. In the second experiments, we used a real application in which AGs represent coloured 3D objects. They were extracted and recognised from some 2D images. The first experiments are useful to study our representation from the theoretical point of view and the second ones are useful to apply our methods on noisy, real and complex images.

We present the experiments in the following three sections. In each experiment, we compare SORGs with three other methods: FDGs, FORGs and AG-to-AG matching. First, we show some information of the AGs and the structures obtained in the synthesis process and then we show the run time and ratio of correctness of the classification processes for each method. SORGs, FDGs and FORGs were synthesised using the *dynamic clustering* in which the models are incrementally updated from a sequence of AGs that represent the same cluster or 3D-object [6] (We used the order of presentation of AGs that obtained the best results). In the SORG method, AGs were classified using the distance measure described in this paper. In the FDG method, the AGs were classified applying the *distance measure between AGs and FDGs relaxing second-order constraints* (moderate costs on the antagonisms, existences and occurrences), without the *efficient module*, presented in [7]. FORGs were compared using the methods presented in [4]. Finally, in the direct AG-to-AG matching method, we used the *edit-operations distance* between AGs presented in [8]. The algorithms presented here were implemented in visual C++ and run on a Pentium IV (1.6Ghz).

5.1 Experiments with Randomly Generated AGs

The AGs used in this section were generated by the random graph generator process shown in figure 1 (this graph generator was also used and explained in depth in [6]).

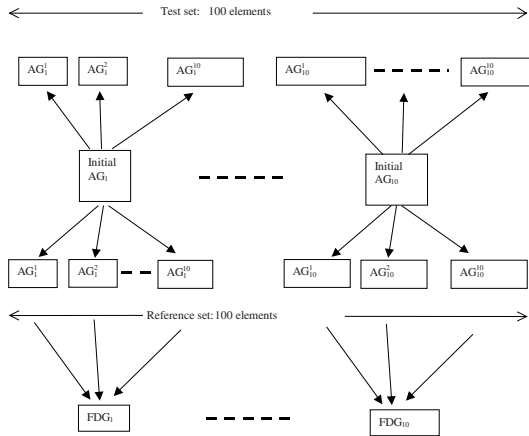


Fig. 1. Random generation of reference and test sets and FDG synthesis.

We first generated 10 initial AGs randomly, one for each model, that had 15 vertices and 5 arcs per vertex. From these AGs, the reference and test sets were derived in the following way. For each initial AG, a reference and a test set of 10 AGs was built by randomly deleting 3 vertices and replacing the attribute of the other vertices by adding gaussian noise with variance V to the attribute values. Then, from each set of 10 reference AGs, an FDG was synthesised.

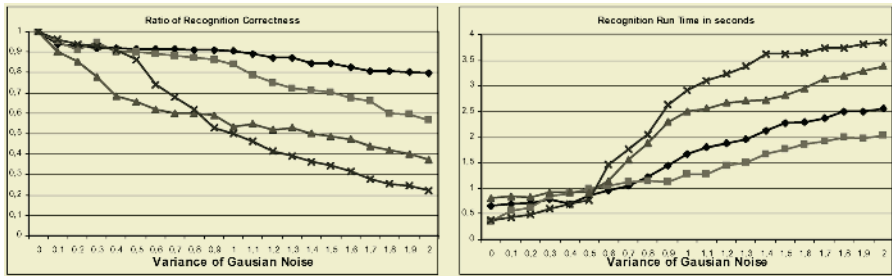


Fig. 2. (a) Ratio of recognition correctness (b) run time spent in the classification. SORG: ; FDG: ; FORG: ; AG-AG:

Figure 2 shows in (a) the ratio of recognition correctness and in (b) the time in seconds spent to compute an AG classification in average applying 4 different classification methods: SORGs, FDGs, FORGs and direct AG-AG matching. We have seen that the second-order knowledge kept in the SORGs is higher than in the FDGs and than in the FORGs. We see, through the results, that this knowledge is useful to represent the cluster of AGs and so to increase the recognition ratio. The direct AG-AG matching methods have similar results than SORGs and FDGS only when there is few noise in the test set. When the variance of the noise increases, the AGs in the tests set are very different from the AGs in the reference sets and then the ratio of classification decreases. While considering the run time, we see that the higher differences appear when the variance of the noise is large. FDGs is the fastest method since the

antagonisms are useful to prune the search tree (see [10] for more details). Nevertheless, the Peleg coefficients computed in the distance between AGs and SORGs are also useful to prune the search tree. For this reason, SORGs obtain better results than FORGs. Finally, the direct AG to AG matching is the slowest method when the variance is bigger than 0.6. This is due to the fact that the AGs in the test set are very different to those in the reference set and so the *branch and bound* algorithm can scarcely prune the search tree.



Fig. 3. The 20 selected objects at angle 100 and the segmented images with the AGs.

5.2 Application of Graph Structures to 3D Object Recognition

Finally, we present a real application to recognise coloured objects using 2D images. Images were extracted from the database COIL-100 from Columbia University (www.cs.columbia.edu/CAVE/research/softlib/coil-100.html). It is composed by 100 isolated objects and for each object there are 72 views (one view each 5 degrees). AGs are obtained by the segmentation process presented in [9]. AG nodes represent regions and their attribute value is their average hue and arcs represent adjacent regions and their attribute value is the distance between average hues. Figure 3 shows the 20 objects at angle 100 and their segmented images with the AGs. These AGs have from 6 to 18 vertices and the average number is 10. The test set was composed by 36 views per object (taken at the angles 0, 10, 20 and so on), whereas the reference set was composed by the 36 remaining views (taken at the angles 5, 15, 25 and so on). We made 6 different experiments in which the number of clusters that represents each 3D-object varied. If the 3D-object was represented by only one cluster, the 36 AGs from the reference set that represent the 3D-object were used to synthesise the SORGs, FORGs or FDGs. If it was represented by 2 clusters, the 18 first and consecutive AGs from the reference set were used to synthesise one of the SORGs,

FORGs or FDGs and the other 18 AGs were used to synthesise the other ones. A similar method was used for the other experiments with 3, 4, 6 and 9 clusters per 3D-object.

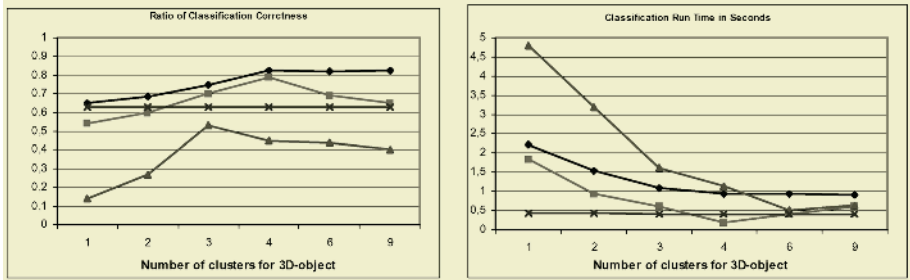






Fig. 4. (a) Ratio of recognition correctness (b) run time spent in the classification. SORG: ; FDG: ; FORG: ; AG-AG: 

Figure 4.a shows the ratio of correctness of the four classifiers varying the number of clusters per each object. When objects are represented by only 1 or 2 clusters, there are too much spurious regions (produced in the segmentation process) to keep the structural and semantic knowledge of the object. For this reason, different regions or faces (or vertices in the AGs) of different views (that is, AGs) are considered to be the same face (or vertex in the AGs). The best result appears when each object is represented by 3 or 4 clusters, that is, each cluster represents 90 degrees of the 3D-object. When objects are represented by 9 clusters, each cluster represents 40 degree views of the 3D-object and 4 AGs per cluster, there is poor probabilistic knowledge and therefore there is a lack of discrimination between objects.

Figure 4.b shows the average run time spent to compute the classification. When the number of clusters per object decreases, the number of total comparisons also decreases but the time spent to compute the distance increases since the structures that represent the clusters (SORGs, FORGs or FDGs) are bigger.

6 Conclusions and Future Work

SORGs are a general formulation of an approximation of the joint probability of random elements in a RG, that describes a set of AGs, based on 2nd order joint probabilities and marginal ones. FORG and FDG approaches are two specific cases of SORGs. A new distance measure between AGs and SORGs has been presented. It is related to the probability of the AG according to the function that matches the graph elements. We have commented the features of the new distance measure and we have applied to two pattern recognition applications. We show that in both cases the use of the 2nd order probabilities is useful to increase the recognition ratio and decrease the run time.

References

1. R. Alquézar, F. Serratosà, A. Sanfeliu, "Distance between Attributed Graphs and Function-Described Graphs relaxing 2nd order restrictions". *Proc. SSPR'2000 and SPR'2000*, Barcelona, Spain, Springer LNCS-1876, pp. 277-286, 2000.
2. S. Peleg and A. Rosenfeld, "Determining compatibility coefficients for curve enhancement relaxation processes", *IEEE Transactions on Systems, Man and Cybernetics*, vol. 8, pp. 548-555, 1978.
3. K. Sengupta and K. Boyer, "Organizing large structural model bases", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 17, pp. 321-332, 1995.
4. A.K.C. Wong and M. You, "Entropy and distance of random graphs with application to structural pattern recognition", *IEEE Trans. on PAMI.*, vol. 7, pp. 599-609, 1985.
5. F. Serratosà, R. Alquézar y A. Sanfeliu, "Estimating the Joint Probability Distribution of Random Vertices and Arcs by means of Second-order Random Graphs", *Proc. Syntactic and Structural Pattern Recognition, SSPR'2002, LNCS 2396*, Windsor, Canada, pp: 252-262, 2002.
6. F. Serratosà, R. Alquézar y A. Sanfeliu, "Synthesis of function-described graphs and clustering of attributed graphs", *International Journal of Pattern Recognition and Artificial Intelligence*, Vol. 16, No 6, pp.621-655, 2002.
7. F. Serratosà, R. Alquézar y A. Sanfeliu, "Function-described for modeling objects represented by attributed graphs", *Pattern Recognition*, 36 (3), pp. 781-798, 2003.
8. A. Sanfeliu and K. Fu, "A distance measure between attributed relational graphs for pattern recognition", *IEEE Transactions on Systems, Man and Cybernetics*, vol. 13, pp. 353-362, 1983.
9. P.F. Felzenswalb and D.P. Huttenlocher, "Image Segmentation Using Local Variation", *Proc. of the IEEE Computer Soc. Conf. on Computer Vision and Pattern Recognition*, pp. 98-104, 1998