Distance between Attributed Graphs and Function-Described Graphs Relaxing 2nd Order Restrictions

René Alquézar[1], Francesc Serratosa[2], and Alberto Sanfeliu[3],

[1] Dept. Llenguatges i Sistemes Informàtics, Universitat Politècnica de Catalunya,
alquezar@lsi.upc.es

[2] Dept. Enginyeria Informàtica i Matemàtiques, Universitat Rovira i Virgili,
Francesc.Serratosa@etse.urv.es

[3] Inst. de Robòtica i Informàtica, Universitat Politècnica de Catalunya,
asanfeliu@iri.upc.es

Abstract. *Function-Described Graphs* (FDGs) have been introduced as a representation of an ensemble of *Attributed Graphs* (AGs) for structural pattern recognition and a distance measure using restrictions between AGs and FDGs has been reported. Nevertheless, in real applications, AGs can be distorted by some external noise, and therefore some constraints have to be relaxed. To gain more flexibility and robustness, some local costs may be added to the global cost of the labelling depending on the fulfilment of the graph element constraints of the FDG instead of applying hard binary constraints.

1 Introduction

Function-Described Graphs (FDGs) were introduced in [2] and redefined in [3] as a representation of an ensemble of *Attributed Graphs* (AGs) for structural pattern recognition different from Random Graphs [5]. Some 2nd order relations (antagonism, existence and occurrence of a pair of vertices or a pair of arcs) are introduced to the FDGs to keep, to the most, the structure of the ensemble of the AGs. The synthesis of FDGs was studied in [1]. Here, a new distance measure, relaxing second order restrictions is presented.

Relations of second order defined on the FDGs are useful to constrain the set of possible labellings while computing the *distance with restrictions* between AGs and FDGs. This is aimed at reaching the best labelling function, taking into account the second order information obtained from the structure of the cluster of AGs that was used to synthesise the FDG. Nevertheless, in real applications, AGs can be distorted by some external noise, and therefore, the constraints associated with the second order relations have to be relaxed to avoid that a noisy AG be misclassified due to non-

F.J. Ferri et al. (Eds.): SSPR&SPR 2000, LNCS 1876, pp. 277-286, 2000.

fulfilment of any of these constraints. A *distance relaxing 2^{nd} order restrictions* is presented here. To gain more flexibility and robustness, some local non-negative costs may be added to the global cost of the labelling depending on the second-order probabilities of the graph elements, instead of applying hard binary constraints.

The organisation of this paper is as follows: AGs and FDGs are reviewed in sections 2 and 3, respectively. The new distance is proposed in section 4 and applied to the 3D-object recognition problem in section 5. Finally, some conclusions are sketched in section 6.

2 Attributed Graphs

Let $H=(\Sigma_v,\Sigma_e)$ be a directed graph structure of order *n* where $\Sigma_v=\{v_k \mid k=1,...,n\}$ is a set of vertices (or nodes) and $\Sigma_e=\{e_{ij} \mid i,j\in\{1,...,n\}, i\neq j\}$ is a set of edges (or arcs). We use the term *graph element* to refer to either a vertex or an edge. Let Δ_v and Δ_e be the global domains of possible values for non-null attributed vertices and arcs, respectively. A *null value* of a graph element is represented by Φ.

An *attributed graph G* over (Δ_v,Δ_e) with an underlying graph structure $H=(\Sigma_v,\Sigma_e)$ is defined to be a pair (V,A) where $V=(\Sigma_v,\gamma_v)$ is an *attributed vertex set* and $A=(\Sigma_e,\gamma_e)$ is an *attributed arc set.* The mappings $\gamma_v:\Sigma_v\rightarrow\Delta_\omega$ and $\gamma_e:\Sigma_e\rightarrow\Delta_\varepsilon$ assign attribute values to vertices and arcs, respectively, where $\Delta_\varepsilon=\Delta_e\cup\{\Phi\}$ and $\Delta_\omega=\Delta_v\cup\{\Phi\}$.

A *complete AG* is an AG with a complete graph structure *H* (but possibly including null elements). An attributed graph $G=(V,A)$ of order *n* can be *extended* to form a complete AG $G'=(V',A')$ of order k, $k\geq n$, by adding vertices and arcs with null attribute values Φ. We call *G'* the *k-extension* of *G*.

3 Function-Described Graphs

A *function-described graph F* over (Δ_v,Δ_e) with an underlying graph structure $H=(\Sigma_\omega,\Sigma_\varepsilon)$ is defined to be a tuple (W,B,P,R) such that

1. $W=(\Sigma_\omega,\gamma_\omega)$ is a *random vertex set* and $\gamma_\omega:\Sigma_\omega\rightarrow\Omega_\omega$ is a mapping that associates each vertex $\omega_i\in\Sigma_\omega$ with a random variable $\alpha_i=\gamma_\omega(\omega_i)$ with values in Δ_ω.

2. $B=(\Sigma_\varepsilon,\gamma_\varepsilon)$ is a *random arc set* and $\gamma_\varepsilon:\Sigma_\varepsilon\rightarrow\Omega_\varepsilon$ is a mapping that associates each arc $\varepsilon_{kl}\in\Sigma_\varepsilon$ with a random variable $\beta_j=\gamma_\varepsilon(\varepsilon_{kl})$ with values in Δ_ε.

3. $P=(P_\omega,P_\varepsilon)$ are two sets of marginal (or first-order) probability density functions for random vertices and edges, respectively. This is, $P_\omega=\{p_i(\mathbf{a}), 1\leq i\leq n\}$ and

$P_\varepsilon = \{q_j(\mathbf{b}), 1 \le j \le m\}$ (being m the number of edges), where $p_i(\mathbf{a}) \equiv \Pr(\alpha_i = \mathbf{a})$ for all $\mathbf{a} \in \Delta_\omega$ and $q_j(\mathbf{b}) \equiv \Pr(\beta_j = \mathbf{b} \mid \alpha_{j1} \ne \Phi \wedge \alpha_{j2} \ne \Phi)$ for all $\mathbf{b} \in \Delta_\varepsilon$ such that α_{j1}, α_{j2} refer to the random variables for the endpoints of the random arc associated with β_j. By definition, $\Pr(\beta_j = \Phi \mid \alpha_{j1} = \Phi \vee \alpha_{j2} = \Phi) = 1$.

4. $R = (A_\omega, A_\varepsilon, O_\omega, O_\varepsilon, E_\omega, E_\varepsilon)$ is a collection of boolean functions defined over pairs of graph elements (i.e. relations on the sets of vertices and arcs) that allow the incorporation of qualitative second-order probability information. A_ω and A_ε are the *vertex antagonism* and *arc antagonism functions*, respectively, where $A_\omega : \Sigma_\omega \times \Sigma_\omega \to \{0,1\}$ is defined by $A_\omega(\omega_i, \omega_j) = 1 \Leftrightarrow \Pr(\alpha_i \ne \Phi \wedge \alpha_j \ne \Phi) = 0$, and similarly, $A_\varepsilon : \Sigma_\varepsilon \times \Sigma_\varepsilon \to \{0,1\}$ is defined by $A_\varepsilon(\varepsilon_{kl}, \varepsilon_{pq}) = 1 \Leftrightarrow \Pr(\beta_i \ne \Phi \wedge \beta_j \ne \Phi) = 0$, where $\beta_i = \gamma_\varepsilon(\varepsilon_{kl})$ and $\beta_j = \gamma_\varepsilon(\varepsilon_{pq})$. In addition, O_ω and O_ε are the *vertex occurrence* and *arc occurrence functions*, where $O_\omega : \Sigma_\omega \times \Sigma_\omega \to \{0,1\}$ is defined by $O_\omega(\omega_i, \omega_j) = 1 \Leftrightarrow \Pr(\alpha_i \ne \Phi \wedge \alpha_j = \Phi) = 0$, and $O_\varepsilon : \Sigma_\varepsilon \times \Sigma_\varepsilon \to \{0,1\}$ is defined by $O_\varepsilon(\varepsilon_{kl}, \varepsilon_{pq}) = 1 \Leftrightarrow \Pr(\beta_i \ne \Phi \wedge \beta_j = \Phi) = 0$. We say that two graph elements (of the same type) are *co-occurrent* if and only if the occurrence relation applies to them in both directions. Finally, E_ω and E_ε are the *vertex existence* and *arc existence functions*, where $E_\omega : \Sigma_\omega \times \Sigma_\omega \to \{0,1\}$ is defined by $E_\omega(\omega_i, \omega_j) = 1 \Leftrightarrow \Pr(\alpha_i = \Phi \wedge \alpha_j = \Phi) = 0$, and $E_\varepsilon : \Sigma_\varepsilon \times \Sigma_\varepsilon \to \{0,1\}$ is defined by $E_\varepsilon(\varepsilon_{kl}, \varepsilon_{pq}) = 1 \Leftrightarrow \Pr(\beta_i = \Phi \wedge \beta_j = \Phi) = 0$.

A random element δ of an FDG is a *null random element* if its probability of instantiation to the null value is one, $\Pr(\delta = \Phi) = 1$. A *complete FDG* is an FDG with a complete graph structure *H*. An FDG $F = (W, B, P, R)$ of order *n* can be *extended* to form a complete FDG $F' = (W', B', P', R')$ of order $k, k \ge n$, by adding null vertices and null arcs and extending appropriately both the set of probability density functions and the boolean functions that relate graph elements. We call *F'* the *k-extension* of *F.*

4 Distance between AGs and FDGs Using 1st and 2nd Order Costs

We require a fine but robust matching cost that makes powerful use of the measurement information in the data graphs (attribute values) and in the prototypes (random variable distributions) as well as an effective way of constraining the possible matches, if we want the system to have the capability of discerning between prototypes. The matching measure must be soft for two reasons: first, because it is assumed that in real applications the patterns are distorted by noise, and second, because a prototype has to represent not only the objects in the reference set but also the ones that are "near" them.

First of all, and for the sake of robustness, the mapping h is not defined from the initial AG that represents the pattern to the initial FDG that represents the class, but from the *k*-extended AG to the *k*-extended FDG, to contemplate the possibility of some missing graph elements or some extraneous graph elements introduced by noisy effects. A missing element in the AG will be represented by a null element in the extended AG, and an extraneous element in the AG should be mapped to a null element in the extended FDG. Since it is desired to allow a priori all the isomorphisms, the number of vertices k in the extended graphs is set to the sum of the number of vertices in both initial graphs. Hence, the limit situations in which all the graph elements in the FDG are missing in the AG or all the graph elements in the AG are extraneous are covered.

Let G' be a *k*-extension of the AG G and F' be a *k*-extension of the FDG F. Then, G' and F' are structurally isomorphic and complete with the same number of vertices k, and they also share a common attribute domain (Δ_v, Δ_e). Now, the labelling function is defined as a mapping $h: G' \rightarrow F'$. Since graphs do not have any predetermined orientation and each orientation is given by a morphism h, a *global cost* C_h is associated with each h in a set of valid mappings H, and the measure of dissimilarity is defined as the minimum of all such costs,

$$d = \min_{h \in H} \{C_h\} \tag{1}$$

In addition, an optimal labelling h_d is given by

$$h_d = \arg \min_{h \in H} \{C_h\} \tag{2}$$

The set of valid mappings H contains all the bijective functions that are coherent structurally (i.e. the arc labelling is totally determined by the vertex labelling).

We want the global cost C_h to provide a quantitative idea of the match quality through the mapping h based on the joint conditional probability that the AG is generated from the FDG given labelling h, this is, $C_h = func(P(G|h))$ as presented in [3]. For instance, $C_h = -\ln(P(G|h))$ would be a possible choice, but it is not the most adequate because of its high sensitivity to noise. Only that one of the probabilities was zero, then the obtained distance would be ∞. Note that the joint probability $P(G|h)$ cannot be estimated directly and has to be approximated by the product of the first-order probabilities of the elements. In this case, the previous choice is equivalent to

$$C_h = -\sum_{\forall x} \ln(\Pr(\gamma(y) = \gamma(x) | h(x) = y)) \tag{3}$$

where x and y are graph elements in the AG and the FDG respectively, $\gamma(y)$ is the random variable associated with y, $\gamma(x)$ is the attribute value in x, and all the elements of both graphs have to appear in the productory (possibly by extending the domain and range of the mapping with null elements).

However, only that one graph element had a probability of zero, the joint probability would be zero and C_h would be infinite. Since this may happen due to the noisy presence of an unexpected element (insertion) or the absence of a prototype element (deletion), only that one graph element were not properly mapped due to clutter, the involved graphs would be wrongly considered to be completely different.

Hence, it is better to decompose the global cost C_h into the sum of bounded individual costs associated with the element matches. Although it has the major flaw that the joint probability is not considered as a whole, it has the advantage that clutter affects only locally the global cost. An *individual cost* $C(x, y)$ represents the dissimilarity between two mapped elements x and y, and it could be based still on the first-order probabilities of the elements, $C(x, y) = func\left(\Pr\left(\gamma(y) = \gamma(x) \middle| h(x) = y\right)\right)$, as far as is bounded by some fixed constant, $C(x, y) \leq \mathrm{Max}$, for instance $C(x, y) \leq 1$.

The global cost is therefore computed as the sum of the individual costs of all the matches between graph elements,

$$C_h = \sum_{\forall x} C(x, h(x)) \tag{4}$$

The main concepts underlying the definition of the distance measures between AGs and FDGs have been introduced above. To define now the different specific measures, it is only needed to define the set of valid mappings H and the individual costs $C(x, y)$.

Individual Costs of Matching Elements

We now turn our attention into the individual cost of matching a pair of elements, one from an AG and one from an FDG. It is defined as a normalised function depending on the dissimilarity between the two mapped elements, as given by the negative logarithm of the probability of instantiating the random element of the FDG to the corresponding attribute value in the AG, this is

$$C(x, y) = \begin{cases} \dfrac{-\ln\left(\Pr\left(\gamma(y) = \gamma(x) \middle| h(x) = y\right)\right)}{-\ln\left(K_{\Pr}\right)} & \text{if } \Pr\left(\gamma(y) = \gamma(x) \middle| h(x) = y\right) \geq K_{\Pr} \\ 1 & \text{otherwise} \end{cases} \tag{5}$$

where the cost $C(x,y)$ is bounded by $[0,1]$, and the positive constant $K_{\mathrm{Pr}} \in [0,1]$ is a threshold on low probabilities that is introduced to avoid the case $\ln(0)$, which gives negative infinity. Hence, $C(x,y)=1$ will be the cost of matching a null element of the FDG to a non-null element of the AG or matching an FDG element to an AG element whose attribute value has a very low probability of instantiation, that is $\Pr(\gamma(y)=\gamma(x) \mid h(x)=y) \leq K_{\mathrm{Pr}}$.

In the case of the vertices, the individual cost is defined using the probabilities stored in the FDG as

$$C_{f_v}(v_i,\omega_q)=\begin{cases} \dfrac{-\ln(p_q(\mathbf{a}_i))}{-\ln(K_{\mathrm{Pr}})} & \text{if } p_q(\mathbf{a}_i) \geq K_{\mathrm{Pr}} \\ 1 & \text{otherwise} \end{cases} \tag{6}$$

And in the case of the arcs, the individual cost is defined using the arc conditional probabilities as follows. Let $\gamma_e(e_{ij})=\mathbf{b}_m$ in the AG arc and let $\gamma_\varepsilon(\varepsilon_{ab})=\beta_n$ in the matched FDG arc. Then, in general,

$$C_{f_e}(e_{ij},\varepsilon_{ab})=\begin{cases} \dfrac{-\ln(q_n(\mathbf{b}_m))}{-\ln(K_{\mathrm{Pr}})} & \text{if } q_n(\mathbf{b}_m) \geq K_{\mathrm{Pr}} \\ 1 & \text{otherwise} \end{cases} \tag{7}$$

However, if either v_i or v_j is a null extended vertex in the AG, then the conditional probability $q_n(\mathbf{b}_m)$ is not applicable, since depends on the existence of the two extreme vertices, and must be replaced by the conditional probability $\Pr(\beta_n=\mathbf{b}_m \mid \alpha_a=\Phi \vee \alpha_b=\Phi)$, which is 1 if $\mathbf{b}_m=\Phi$ and 0 otherwise.

Second Order Costs of Matching Elements

The second order costs could be defined for the vertices as shown in equations (8) to (10), where it is assumed that $h(v_i)=\omega_p$ and $h(v_j)=\omega_q$. These equations cover respectively the three following qualitative cases: presence of two vertices in the AG, presence of only one of them, and absence of both vertices. Note that, the second-order costs induced artificially by FDG null vertices are not taken into account.

$$C_{A_\omega}(v_i,v_j,\omega_p,\omega_q)=\begin{cases} 1-\Pr(\alpha_p\neq\Phi\wedge\alpha_q\neq\Phi) & if \begin{pmatrix} \mathbf{a}_i\neq\Phi\wedge\mathbf{a}_j\neq\Phi\wedge \\ p_p(\Phi)\neq1\wedge p_q(\Phi)\neq1 \end{pmatrix} \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

$$C_{O_\omega}\left(v_i,v_j,\omega_p,\omega_q\right)=\begin{cases}1-\Pr\left(\alpha_p\neq\Phi\wedge\alpha_q=\Phi\right) & \mathit{if}\begin{pmatrix}\mathbf{a}_i\neq\Phi\wedge\mathbf{a}_j=\Phi\wedge\\ p_p(\Phi)\neq 1\end{pmatrix}\\ 0 & \text{otherwise}\end{cases}\quad(9)$$

$$C_{E_\omega}\left(v_i,v_j,\omega_p,\omega_q\right)=\begin{cases}1-\Pr\left(\alpha_p=\Phi\wedge\alpha_q=\Phi\right) & \mathit{if}\begin{pmatrix}\mathbf{a}_i=\Phi\wedge\mathbf{a}_j=\Phi\wedge\\ p_p(\Phi)\neq 1\wedge p_q(\Phi)\neq 1\end{pmatrix}\\ 0 & \text{otherwise}\end{cases}\quad(10)$$

The definition of the costs on the arcs, $C_{A_\varepsilon}\left(e_{ij},e_{kt},\varepsilon_{ab},\varepsilon_{cd}\right)$, $C_{O_\varepsilon}\left(e_{ij},e_{kt},\varepsilon_{ab},\varepsilon_{cd}\right)$ and $C_{E_\varepsilon}\left(e_{ij},e_{kt},\varepsilon_{ab},\varepsilon_{cd}\right)$ are similar than the costs on the vertices (See [3] for more details).

Since the second-order probabilities are not actually stored in the FDGs, they are replaced by the second-order relations, thus obtaining costs that are coarser. This is, some second-order non-negative costs are added to the global cost of the labelling when second-order constraints (antagonism, occurrence, existence) are broken. Equations (11) to (13) show the final second-order costs, which can be only 1 or 0, associated with the three relations of antagonism, occurrence and existence between pairs of vertices.

$$C_{A_\omega}\left(v_i,v_j,\omega_p,\omega_q\right)=\begin{cases}A_\omega\left(\omega_p,\omega_q\right) & \mathit{if}\begin{pmatrix}\mathbf{a}_i\neq\Phi\wedge\mathbf{a}_j\neq\Phi\wedge\\ p_p(\Phi)\neq 1\wedge p_q(\Phi)\neq 1\end{pmatrix}\\ 0 & \text{otherwise}\end{cases}\quad(11)$$

$$C_{O_\omega}\left(v_i,v_j,\omega_p,\omega_q\right)=\begin{cases}O_\omega\left(\omega_p,\omega_q\right) & \mathit{if}\ \mathbf{a}_i\neq\Phi\wedge\mathbf{a}_j=\Phi\wedge p_p(\Phi)\neq 1\\ 0 & \text{otherwise}\end{cases}\quad(12)$$

$$C_{E_\omega}\left(v_i,v_j,\omega_p,\omega_q\right)=\begin{cases}E_\omega\left(\omega_p,\omega_q\right) & \mathit{if}\begin{pmatrix}\mathbf{a}_i=\Phi\wedge\mathbf{a}_j=\Phi\wedge\\ p_p(\Phi)\neq 1\wedge p_q(\Phi)\neq 1\end{pmatrix}\\ 0 & \text{otherwise}\end{cases}\quad(13)$$

Global Cost

The global cost on the labelling function C_h is defined with two terms that depend on the first-order probability information, and six more terms that depend on the second-order constraints:

$$C_h = \begin{cases} K_1 * \sum\limits_{\forall v_i \in \Sigma_v \text{of} G'} C_h\left(v_i, h(v_i)\right) + K_2 * \sum\limits_{\forall e_{ij} \in \Sigma_e \text{of} G'} C_h\left(e_{ij}, h(e_{ij})\right) + \\ K_3 * \sum\limits_{\forall v_i, v_j \in \Sigma_v \text{of} G'} C_{A_\omega}\left(v_i, v_j, h(v_i), h(v_j)\right) + K_4 * \sum\limits_{\forall e_{ij}, e_{kt} \in \Sigma_e \text{of} G'} C_{A_\varepsilon}\left(e_{ij}, e_{kt}, h(e_{ij}), h(e_{kt})\right) + \\ K_5 * \sum\limits_{\forall v_i, v_j \in \Sigma_v \text{of} G'} C_{O_\omega}\left(v_i, v_j, h(v_i), h(v_j)\right) + K_6 * \sum\limits_{\forall e_{ij}, e_{kt} \in \Sigma_e \text{of} G'} C_{O_\varepsilon}\left(e_{ij}, e_{kt}, h(e_{ij}), h(e_{kt})\right) + \\ K_7 * \sum\limits_{\forall v_i, v_j \in \Sigma_v \text{of} G'} C_{E_\omega}\left(v_i, v_j, h(v_i), h(v_j)\right) + K_8 * \sum\limits_{\forall e_{ij}, e_{kt} \in \Sigma_e \text{of} G'} C_{E_\varepsilon}\left(e_{ij}, e_{kt}, h(e_{ij}), h(e_{kt})\right) + \end{cases} \tag{14}$$

The eight terms are weighted by non-negative constants K_1 to K_8, to compensate for the different number of elements in the additions as well as to balance the influence of second-order costs with respect to first-order costs in the overall value. Note that if $K_i = \infty : i = 3..8$ there are strict constraints associated with the second order relations and so the distance with 2nd order restrictions is obtained.

5 Results

The contribution of FDGs to structural pattern recognition is illustrated by the three-dimensional object recognition problem. The original data is composed by 101 AGs, which represent the semantic and structural information of the views taken from five objects. Figure 1 shows a selected view of each object.

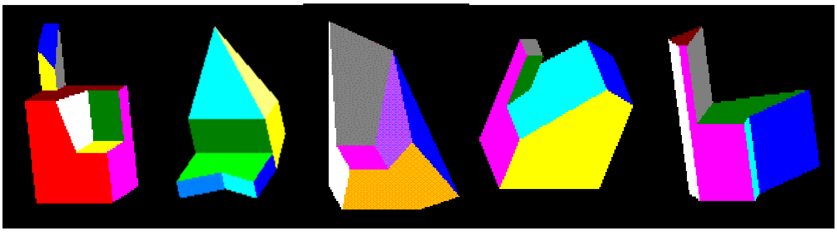

Figure 1.

Vertices in the AGs represent the faces, with one attribute, which is the area of the face. Arcs represent the edges between faces, with one attribute, which is the length of the edge. Five FDGs were built from the AGs that represent their views using a supervised synthesis method. An antagonism relation between two graph elements appears when these elements have never seen together in the same view. On the other hand, an occurrence relation appears when a graph element is visible in all the views in which another one is visible too. There is not any existence relation because there is no pair of faces such that at least one of the two faces is visible in all views. See [3] for more details. The object of the tests presented here is to assess the effects of the application of the antagonism and occurrence relations between vertices in the computation of our distance. To that aim, some tests have been carried out with different weights on these relations (K_3 , K_5). The other weights have been set as follows: $K_1 = 1$ (vertices), $K_2 = 1/2$ (arcs), $K_7 = 0$ (existence on the vertices), $K_4 = K_6 = K_8 = 0$ (second order relations on the arcs). The distance (optimal cost) presented in Section 4 was computed by means of a branch-and-bound algorithm [3].

The test set was composed by random AGs, which were the previous AGs modified by some structural or semantic noise. Results shown are the average of the correctness of the classification tests performed 20 times. The semantic noise, which is added to the attribute values of the vertices and arcs, is obtained by a random number generation with a median of 0.0 and a standard deviation: 0.0, 4.0, 8.0 and 12.0. The structural noise, also obtained by a random number generator, deletes or includes 0, 1 or 2 vertices, which represent the 0%, 20% and 40% of the average structure, respectively. The experimental results are summarised in Table 1.

# noise vertices	0	0	0	0	0	1	2	1
Standard Deviation	0.0	2.0	4.0	8.0	12.0	0.0	0.0	8.0
$K_3 = 0$ ， $K_5 = 0$	100	90.1	89.7	88.6	86.3	70.8	67.7	68.7
$K_3 = \infty$ ， $K_5 = 0$	100	92.5	89.3	87.0	84.9	61.6	54.4	57.4
$K_3 = 0$ ， $K_5 = \infty$	100	91.9	89.9	88.2	85.2	62.5	59.5	59.5
$K_3 = \infty$ ， $K_5 = \infty$	100	95.1	90.2	86.6	80.8	60.7	53.2	56.2
$K_3 = 1$ ， $K_5 = 0$	100	92.3	91.5	91.3	87.2	80.5	75.3	75.5
$K_3 = 0$ ， $K_5 = 1$	100	95.6	92.4	91.5	86.4	81.2	77.2	76.4
$K_3 = 1$ ， $K_5 = 1$	100	98.7	97.1	95.0	92.5	89.2	85.2	83.6
Nearest neighbour	100	98.9	82.6	62.6	52.4	90.0	58.6	58.6

Table 1. Recognition ratio (%) obtained by the FDGs and by the nearest-neighbour classifier (using Sanfeliu's distance between AGs [4]) resulted from applying different levels of noise.

The classification correctness is higher applying strict relations ($K_3 = \infty$ and $K_5 = \infty$) than without applying these two relations ($K_3 = 0$ and $K_5 = 0$) when the semantic or structural noise is low, but, when the noise increases, best results appear when no relations are applied. On the contrary, the distance with both second-order costs ($K_3 = 1$ and $K_5 = 1$) always obtains higher results than if one of the relations is not taken into account. The FDG classifier only obtains worse results than the nearest-neighbour classifier when the structural or semantic noise is very low.

6 Conclusions

A new distance between AGs and FDGs has been reported. The aim of this distance is to gain more flexibility and robustness throughout relaxing the second order constraints. Some local costs have been added to the global cost depending on the second order probabilities instead of applying hard binary constraints. Results show that the distance with 2nd order costs obtains better results than the distance with strict 2nd order restrictions or without considering them.

The main problem of computing a distance between graphs associated with an optimal match is the exponential cost. While a branch-and-bound algorithm [3] was used in the reported experiments, a more efficient but sub-optimal method has been presented recently [6].

Acknowledgements

This work has been partially supported by the Spanish "Comisión Interministerial de Ciencia y Tecnología", under project TAP98-0473.

References

1. R. Alquezar, A. Sanfeliu and F. Serratosa, "Synthesis of Function-Described Graphs". *Advances in Pattern Recognition, Proc. Joint IAPR Int. Workshops SSPR'98 and SPR'98,* Sydney, Australia, Springer LNCS-1451, pp. 112-121, 1998.
2. F. Serratosa and A. Sanfeliu, "Function-Described Graphs applied to 3D object recognition". In *Proceedings ICIAP'97, 9th Int. Conf. Image Analysis and Processing,* Firenze, Italy, Vol. I, pp. 701-708, 1997.
3. F. Serratosa, R. Alquézar and A. Sanfeliu, "Function-Described Graphs for structural pattern recognition", *Technical report DEIM-RR-99-006, Universitat Rovira i Virgili, Tarragona, Spain,* 1999.
4. 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.
5. A.K.C. Wong and M. You, "Entropy and distance of random graphs with application to structural pattern recognition", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 7, pp. 599-609, 1985.
6. F. Serratosa, R. Alquézar and A. Sanfeliu, "Efficient algorithms for matching attributed graphs and function-described graphs", *Proceedings 15th Int. Conf. Pattern Recognition, Barcelona, Spain,* 2000.