Function-described graphs for modelling objects represented by sets of attributed graphs

Francesc Serratosa[a,*], René Alquézar[b], Alberto Sanfeliu[c]

[a] *Department d'Enginyeria Informàtica i Matemàtiques, Universitat Rovira i Virgili, Campus Sescelades, Avinguda dels Països Catalans 26, 43007 Tarragona, Spain*

[b] *Dept. de Llenguatges i Sistemes Informàtics, Universitat Politècnica de Catalunya, Spain*

[c] *Institut de Robòtica i Informàtica Industrial, Universitat Politècnica de Catalunya, Spain*

Abstract

We present in this article the model function-described graph (FDG), which is a type of compact representation of a set of attributed graphs (AGs) that borrow from random graphs the capability of probabilistic modelling of structural and attribute information. We define the FDGs, their features and two distance measures between AGs (unclassified patterns) and FDGs (models or classes) and we also explain an efficient matching algorithm. Two applications of FDGs are presented: in the former, FDGs are used for modelling and matching 3D-objects described by multiple views, whereas in the latter, they are used for representing and recognising human faces, described also by several views.

Keywords: Attributed graphs; Error-tolerant graph matching; Function-described graphs; Random graphs; Clustering; Synthesis; 3D-object recognition and face identification

1. Introduction

A function-described graph (FDG) is a model that contains probabilistic and structural descriptions of a set of attributed graphs (AGs) to maintain, to the most part, the local features of the AGs that belong to the set and other AGs that are "near" them, as well as to allow the rejection of the AGs that do not belong to it or are "far" from them [1]. Let us consider, as an example, the 3D-object modelling and recognition problem. Fig. 1 shows schematically the FDG learning and classification processes. The basic idea is that only a single FDG is synthesised from the graphs that represent several views of a 3D-object. Therefore, in the recognition process, only one comparison is needed between each model represented by an FDG and the unclassified object (view of a 3D-object) represented by a graph.

The FDG techniques have been applied on the project "active vision system with automatic learning capacity for industrial applications" in which an autonomous mobile robot has been developed [2].

Random graphs [3–5] are one of the earliest approaches used to represent a set of AGs. In this approach, AGs are extended to include probabilistic information. Wong et al. first defined the general random graphs (GRGs) for modelling classes of patterns described by AGs through a joint probability space of random variables ranging over pattern primitives (vertices) and relation (arcs). Due to the computational intractability of GRGs, caused by the difficulty in estimating and handling the high-order joint probability distribution, first-order random graphs (FORGs) were proposed for real applications [4]. Strong simplifications were made in FORGs to allow the use of random graphs in

* Corresponding author. Tel.: +34-977-55-85-07; fax: +34-977-55-97-10.

E-mail addresses: francesc.serratosa@etse.urv.es (F. Serratosa), alquezar@lsi.upc.es (R. Alquézar), asanfeliu@iri.upc.es (A. Sanfeliu).

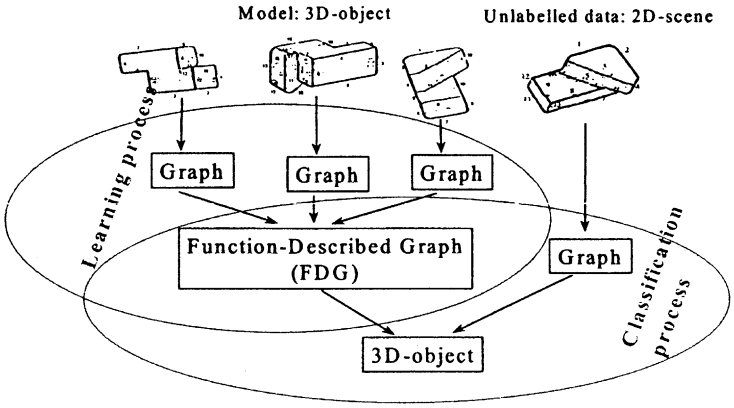


Fig. 1. FDG learning and classification processes applied to the 3D-object recognition.

practical cases; more precisely, the following assumptions were made [4]:

(1) the random vertices are mutually independent;
(2) given values for the random vertices, the random arcs are independent;
(3) the arcs are independent of the vertices except for the vertices that they connect.

FDGs can be seen as a type of simplification of the GRGs, different from FORGs, in which some structural constraints are recorded. A drawback of FORGs is that the strong assumptions about the statistical independence of nodes and arcs may lead to an excessive generalisation of the sample graphs when synthesising an FORG. To alleviate this weakness, a qualitative information of the joint probabilities of two nodes is incorporated into FDGs, thus improving the representional power of FORGs with a negligible increase of computational cost.

Other different approaches that define a model to represent a set of AGs are mentioned here. Winston [6] presented one of the earliest works in which a set of examples was used to learn the structures that represent the class. Later on, Stein [7] presented a system to identify the instances of the objects in the image. Other authors presented systems applied as well to the problem of 3D-object recognition from single-view graphs: for instance, the interesting approach to graph matching and classification [8], in which, from a training set of AGs corresponding to several classes, a single model graph called rulegraph is learnt (using evidence-based systems methods) which describes the full training set. In this rulegraph, vertices do not represent object parts but rules which depict attribute bounds and evidence for different classes. A new AG is then classified by rulegraph matching (subgraph isomorphism), with the advantage of a reduced search space, since the number of vertices in the rulegraph can be much lower than the number of nodes in the original AGs. And also, Kim and Kak [9] represented a 3D-object as an assembly of several parts, each represented as a node in a graph carrying information characterising the part. Sossa [10] indexes a library of graph-based models using the coefficients derived from the Laplacian matrix that describes the graph. However, the graphs are not attributed and the system has been shown to work only with simplistic objects comprising planar surfaces. Shapiro and Haralick [11] organise models in a tree according to an intermodel distance metric. In the approach by Messmer and Bunke [12–14], the model graphs are pre-processed generating a symbolic data structure, called network of models. This network is a compact representation of the models in the sense that multiple occurrences of the same sub-graph are represented only once. Consequently, such sub-graphs will be matched only once with the input and the computational effort will be reduced. A recent approach is the generalised attributed relational graph (GARG) [15], in which the set of AGs is represented by only one model or prototype, but differently from the network of models, the model has the same structure and attribute domain as the AGs that represent, except that a null value is introduced. GARGs are synthesised by a prototyping algorithm, based on inductive learning methodologies. Moreover, Chan proposed the fuzzy attributed graphs (FAGs) and the hard FAGs to represent objects and templates, respectively, [16]. A neural networks method [17] was proposed to encode an AG into a real-valued feature vector using recurrent neural network architectures and a generalised recursive neuron model. A feedforward network classifies the feature vectors yielded by the encoder network while returning an error signal to the encoder, which is then used to adapt the encoder weights. Note however that, in this case, there is no learnt model graph that represents a set of AGs, but a collection of learnt feature vectors, one for each AG, together with a trained neural classifier. Finally, Sengupta [18] presented a hierarchically structured approach to organising large structural model bases using an information theoretic criterion. Objects or patterns are modelled in the form of random

parametric structural descriptions and objects in the scenes are represented as parametric structural descriptions.

The structure of the rest of the paper is as follows. In Section 2, all the notation used throughout the paper is introduced, including the formal definitions of AG and FDG. The fundamental ideas behind FDGs are discussed in Section 3, which also includes an illustrative example. In Section 4, two distance measures between an AG and an FDG are presented, whereas the algorithms for computing or approximating them are sketched in Section 5. The experimental results of two applications of FDGs for object modelling and recognition are presented in Section 6: 3D-object modelling and recognition [19] and face identification [20]. Finally, some conclusions are given in Section 7.

We defined some methods and algorithms to synthesise an FDG using a set of classified AGs and also to cluster AGs and synthesise several FDGs that represent the clusters in Refs. [21,22]. Due to lack of space, these methods are out of the scope of this article.

2. Formal definitions and notation

Definition 1. 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 $\Sigma_v = \{v_k | k = 1, \ldots, n\}$ is a set of vertices (or nodes), $\Sigma_e = \{e_{ij} | i, j \in \{1, \ldots, n\}, i \neq j\}$ is a set of arcs (or edges), and the mappings $\gamma_v : \Sigma_v \to \Delta_v$ and $\gamma_e : \Sigma_e \to \Delta_e$ assign attribute values to vertices and arcs, respectively.

Definition 2. A *complete AG* is an AG with a complete graph structure (Σ_v, Σ_e), but possibly including null elements. An AG G with n vertices can be *extended* to form a complete AG G' with k vertices $k \geqslant n$, by adding vertices and arcs with null attribute values Φ. We call G' the *k-extension* of G.

Definition 3. 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 *function-described graph* F over (Δ_v, Δ_e) is defined to be a tuple $(\Sigma_v, \Sigma_e, \gamma_v, \gamma_e, P, R)$, where $\Sigma_v = \{\omega_k | k = 1, \ldots, n\}$ is a set of vertices, $\Sigma_e = \{\varepsilon_{ij} | i, j \in \{1, \ldots, n\}, i \neq j\}$ is a set of arcs, the mapping $\gamma_v : \Sigma_v \to \Omega_v$ associates each vertex $\omega_k \in \Sigma_v$ with a random variable $\alpha_k = \gamma_v(\omega_k)$ with values in Δ_v, and $\gamma_e : \Sigma_e \to \Omega_e$ associates each arc $\varepsilon_{ij} \in \Sigma_e$ with a random variable $\beta_k = \gamma_e(\varepsilon_{ij})$ with values in Δ_e. $P = (P_v, P_e)$ are two sets of marginal (or first-order) probability density functions for random vertices and edges, respectively. This is, $P_v = \{p_i(\mathbf{a}), 1 \leqslant i \leqslant n\}$ and $P_e = \{q_j(\mathbf{b}), 1 \leqslant j \leqslant m\}$, m being the number of edges, where $p_i(\mathbf{a}) = \Pr(\alpha_i = \mathbf{a})$ for all $\mathbf{a} \in \Delta_v$ and $q_j(\mathbf{b}) \equiv \Pr(\beta_j = \mathbf{b} | \alpha_{j1} \neq \Phi \wedge \alpha_{j2} \neq \Phi)$ for all $\mathbf{b} \in \Delta_e$ such that α_{j1}, α_{j2} refer to the random variables for the endpoints of the arc associated with β_j. Note that, due to the structural consistency requirements, there is no need to store the conditional probabilities $\Pr(\beta_j = \mathbf{b} | \alpha_{j1} = \Phi \vee \alpha_{j2} = \Phi)$, since by definition $\Pr(\beta_j = \Phi | \alpha_{j1} = \Phi \vee \alpha_{j2} = \Phi) = 1$.

$R = (A_v, O_v, E_v)$ is a collection of boolean functions defined over pairs of graph vertices (i.e. relations on the set of vertices) that allow the incorporation of qualitative second-order probability information. Thus, $A_v : \Sigma_v \times \Sigma_v \to \{0, 1\}$, defined by $A_v(\omega_i, \omega_j) = 1 \Leftrightarrow \Pr(\alpha_i \neq \Phi \wedge \alpha_j \neq \Phi) = 0$, is the so-called *vertex antagonism function*, which can be seen as a symmetric binary relation on the set Σ_v. In addition, $O_v : \Sigma_v \times \Sigma_v \to \{0, 1\}$, defined by $O_v(\omega_i, \omega_j) = 1 \Leftrightarrow \Pr(\alpha_i \neq \Phi \wedge \alpha_j = \Phi) = 0$, is the so-called *vertex occurrence function*, which can be seen as a reflexive and transitive relation (partial order) on the set Σ_v. Finally, $\Sigma_v : \Sigma_v \times \Sigma_v \to \{0, 1\}$, defined by $E_v(\omega_i, \omega_j) = 1 \Leftrightarrow \Pr(\alpha_i = \Phi \wedge \alpha_j = \Phi) = 0$, is the so-called *vertex existence function*, which can also be seen as a symmetric binary relation on Σ_v.

A random element α or β of an FDG is a *null random element* if its probability of instantiation to the null value is one, i.e. $\Pr(\alpha = \Phi) = 1$ or $\Pr(\beta = \Phi) = 1$.

Definition 4. A *complete FDG* is an FDG with a complete graph structure (Σ_v, Σ_e), but possibly including null random elements. An FDG F with n vertices can be *extended* to form a complete FDG F' with k vertices $k \geqslant n$, by adding null random vertices and null random arcs and extending appropriately the boolean functions of antagonism, occurrence and existence. We call F' the *k-extension* of F. Note that both F' and F represent the same model.

Definition 5. Any AG obtained by instantiating all random vertices and random arcs of an FDG in a way that satisfies all the relations of antagonism, occurrence and existence specified by the FDG is called an *outcome graph* of the FDG. Hence, similar to the case of random graphs, an FDG represents the set of all possible AGs that can be outcome graphs of it, according to an associated probability distribution.

3. A function-described graph as a model of a class of attributed graphs

FDGs are proposed here for modelling classes of patterns described by attributed graphs. An FDG is a probabilistic structural model that contains some functions to maintain, to the most part the local features of the AGs that belong to the class as well as to allow the rejection of the AGs that do not belong to it. In general, an FDG can be derived as a synthesis of a cluster or set of individual AGs, similar to the case of random graphs. A comparison between FDGs and FORGs is sketched in Section 3.1 to introduce the novel features that are included in FDGs to represent a set of AGs. Then, a simple example of representing a 3D-object using FDGs is provided for illustrative purposes in Section 3.2.

3.1. FDGs: a model lying between FORGs and general random graphs

In order to attain a compact representation of a set of AGs by means of a model, a probabilistic description of the ensemble is desirable to account for the variations of structural patterns in the reference set or sample. As recalled in Section 1, random graphs (RGs) provide such a representation. Nevertheless, when estimating the probability distribution of the structural patterns from an ensemble of AGs, it is impractical to consider the high-order probability distribution where all the graph elements are taken jointly. It was the random graph authors believe (and also ours), that general RGs cannot be applied in real applications, and for this reason, they proposed the FORGs.

The FORG approach, although simplifies the representation considerably, continues to be difficult to apply in real problems where there is a large number of vertices in the AGs and their attributes have an extensive domain. The main cause of this problem is the dependence of the arc attributes with respect to the attributes of the vertices that the arc connects (assumption 3 in Section 1). Although this supposition is useful to constrain the generalisation of the given set of AGs, it needs a huge amount of data to estimate the probability density functions and bears a high computational cost. On the other hand, an important drawback of FORGs, which is due to the probability independence assumptions 1 and 2 in Section 1, is that the structural information in a sample of AGs is not well preserved in the FORG synthesised from them. This is because an FORG represents an over-generalised prototype that may cover graph structures quite different from those in the sample. For example, if C is a set of AGs describing different perspective views of an object O, many of the outcome graphs of the FORG synthesised from C will represent impossible views of O (just from the topological point of view, without further consideration of the attributes of primitives and relations).

With the aim of offering a more practical approach, function-described graphs (FDGs) can be seen as a different type of simplification of the GRGs, in which another approximation of the joint probability P of the random elements is proposed. On the one hand, some independence assumptions (1) are considered, but on the other hand, some useful second-order functions (2) are included to constrain the generalisation of the structure.

(1) *Independence assumptions in the FDGs*:

1. The attributes in the vertices are independent of the other vertices and also of the arcs.
2. The attributes in the arcs are independent of the other arcs and also of the vertices. However, it is mandatory that all non-null arcs be linked to a non-null vertex at each extreme in every AG covered by an FDG. In other words, any outcome AG of the FDG has to be structurally consistent.

With these assumptions, the probability density functions are themselves independent since the attributes in the arcs do not depend on the attributes in the vertices that they connect, but only on the existence of the extreme vertices. Consequently, associated with each graph element in an FDG, there is a *random variable* that represents the semantic information distribution of the corresponding graph elements in the set of outcome AGs. A random variable has a probability density function defined over the same attribute domains of the AGs, including the null value Φ, that denotes the non-instantiation of an FDG graph element in an outcome AG.

(2) *Second-order functions in the FDGs*:

In order to tackle the problem of the over-generalisation of the sample, we introduce the *antagonism, occurrence* and *existence* relations in FDGs, which apply to pairs of vertices. In this way, random vertices are not assumed to be mutually independent, at least with regard to the structural information. These second-order relations, that suppose a little increase of the amount of data to be stored in the prototype, are useful for two reasons. Firstly, they constrain the set of outcome graphs covered by the prototype and tend to cut down notably the structural over-generalisation. Secondly, they reduce the size of the search space of the AG-to-FDG matching algorithm, decreasing the global temporal cost of the recognition [1]. Antagonism, occurrence and existence relations in the FDGs represent a *qualitative information* of the second-order joint probability functions of a pair of vertices. To define these second-order relations it is necessary to split the domain of the joint probabilities in four regions (see Fig. 2a). The first one is composed of the points that belong to the Cartesian product of the sets of actual attributes of the two elements, corresponding to the cases where both elements are defined in the initial non-extended AG and therefore their value is not null. The second and third regions are both straight lines in which only one of the elements has the null value. This covers the cases when one of the two elements does not belong to the initial AG and has been added in the extending process. Finally, the fourth region is the single point where both elements are null, which includes the cases when none of them appear in the initial AG. The second-order relations are defined as follows:

Antagonism relations: Two vertices of the FDG are antagonistic if the probabilities in the first region are all zero, $\Pr(\alpha_i \neq \Phi \wedge \alpha_j \neq \Phi) = 0$, which means that, although these vertices are included in the prototype as different elementary parts of the covered patterns, they have never taken place together in any AG of the reference set used to synthesise the FDG. Fig. 2b shows the joint probabilities of the vertices ω_i and ω_j defined as antagonistic.

Occurrence relations: There is an occurrence relation if the joint probability function equals zero in the second region, $\Pr(\alpha_i \neq \Phi \wedge \alpha_j = \Phi) = 0$. That is, it is possible to assure that if the element α_i does appear in any AG of the reference set, then the element α_j must appear too. The case of the third region is analogous to the second one with the only

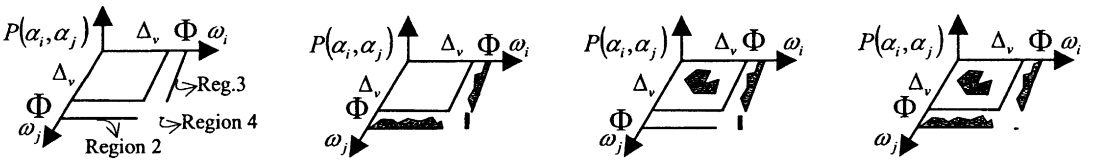


Fig. 2. (a) Split of the joint domain of two random vertices in four regions. Second-order density function of (b) two antagonistic vertices, (c) two occurrent vertices and (d) two existent vertices.

difference of swapping the elements. Fig. 2c shows the joint probabilities of the vertices ω_i and ω_j, there is an occurrence relation from ω_i to ω_j.

Existence relations: Finally, there is an existence relation between two vertices if the joint probability function equals zero in the fourth region, $\Pr(\alpha_1 = \Phi \wedge \alpha_j = \Phi) = 0$, that is, all the objects in the class described by the FDG have at least one of the two elements. Fig. 2d shows the joint probabilities of the vertices ω_i and ω_j.

3.1.1. Mapping joint probabilities to second-order relations

Fig. 3 shows 16 different combinations of the joint probability of two vertices if it is considered that the probabilities of the four regions can be zero or greater than zero. An X is written on a region if and only if the sum of the probabilities in that region is greater than zero. For each one of the 16 cases, the second-order relations obtained from the corresponding joint probability density function are shown.

Note that it is not "logical" for both occurrence and antagonism relations to be satisfied at the same time between two elements. If two elements cannot exist in the same AG (antagonism), one of them cannot always exist when the other exists (occurrence). This combination only appears in cases in which one of the elements is null (cases 1–5); that is, it is a synthetically created element. Moreover, the 16th combination is impossible in a correct FDG since the sum of the joint probability throughout the four regions equals 1. Therefore, the four second-order relations cannot appear between two graph elements at the same time.

3.2. An example of modelling 3D-object described by views using FDGs

In this simple example, FDGs are used to represent 3D-objects with planar faces. The knowledge of the 3D-object represented by an FDG depends on the views taken from it. The representation of 3D-objects by means of FDGs is a useful application to show the meaning of the probabilities in the vertices and arcs and also of second-order relations. Fig. 4 shows object A, four different perspective views and their corresponding AGs. Vertices and arcs represent faces and adjacency between faces, respectively. The attribute of the vertices is chosen to be the average hue of the planar face discretised in 7 colours and the attribute of the arcs is chosen to be the cyclic distance between hues discretised in 4 categories. The direction of the arcs (from tail to head) is set by the combination of both nodes that gives lower cyclic distance. In this example, we will suppose that the extraction of graph elements and their attributes is robust enough to obtain the same discrete value (colour) in all the vertices that represent the same face and the same discrete value (colour similarity of adjacent regions) in all the arcs that represent the same edge.

Similar to AGs, vertices and arcs of FDGs represent in this case faces and edges of the 3D-object. We assume that there is a given labelling between vertices and arcs of the AGs such that same faces in different views are matched. Then, vertices or arcs in the AGs that represent the same face or the same edge are used to build only one vertex or arc in the FDG (the synthesis of FDGs given a set of AGs with a common labelling is explained in Ref. [21]). We have to take into account that faces that cannot be seen together in a view appear as different vertices in the FDG. This feature of the 3D-object representation is characterised through the probability density functions and the second-order relations. When a face is not visible in a view, then its AG does not include its related vertex, but a null vertex, called v_Φ, is introduced in the extended AG with a null value, Φ (Section 2). Similarly, a null arc e_Φ is extended when at least one of the adjacent faces of the corresponding edge is occluded in a view. Hence, the probability of a face or edge to be occluded is represented by the probability of its related vertex or arc of being null. Two faces are antagonistic when it is not possible to see both in the same view. Translated to representations, two vertices are antagonistic when there is no AG that contains both of them. On the other hand, a face is occurrent with respect to another if always the former is visible (its vertex appears in the AG) and the latter is visible too. Finally, there is an existence relation between two faces (vertices) if one of them or both appear in all the views (in all the AGs).

The FDG F_a (Fig. 5) represents object A if we consider only View_1 and View_2. Because only six faces of the object have been seen, the object is considered to have only these faces and therefore F_a has six vertices. Some faces are visible in both views (i.e. face_1 and face_7), and thus, the probability of the related vertices in the FDG of taking their colour is 1. Other faces are only visible in one of both views (i.e. face_2) and for this reason, the probability of being null is $\frac{1}{2}$. There

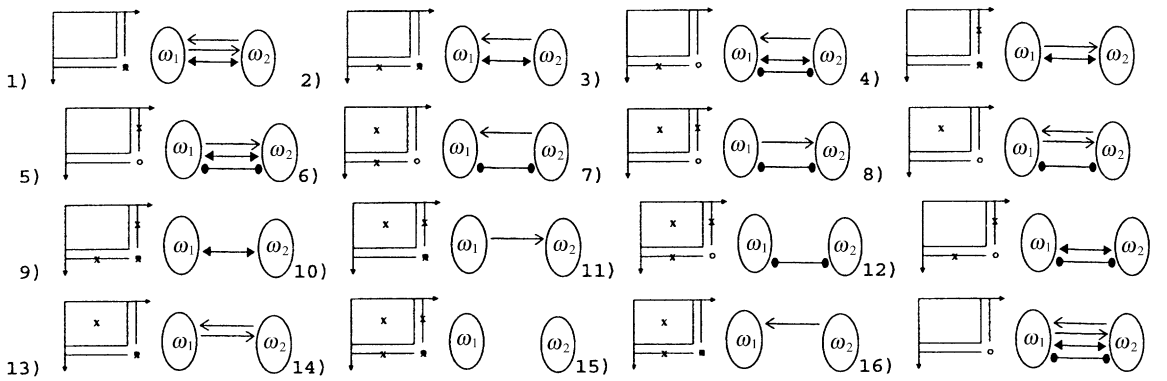
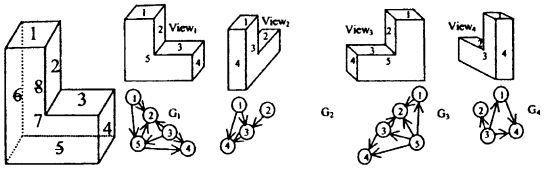


Fig. 3. The sixteen combinations of the joint probability. Antagonism, occurrence and existence relations are represented by $\leftrightarrow$, $\rightarrow$ and $\bullet$—$\bullet$, respectively.

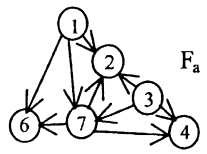


(a) (b)

AG G1	AG G1	AG G2	AG G2	AG G3	AG G3	AG G4	AG G4
edges	nodes	edges	nodes	edges	nodes	edges	nodes
e1,5=S	V1=C	e1,4=D	V1=C	e5,1=D	V1=C	e3,2=S	V1=C
e1,2=VD	V2=O	e1,3=S	V2=B	e5,2=D	V2=O	e3,4=VD	V2=B
e5,2=D	V3=B	e2,3=D	V3=G	e5,3=S	V3=B	e3,1=D	V3=M
e3,2=VD	V4=R	e3,4=S	V4=Y	e5,4=S	V4=R	e1,4=D	V4=Y
e3,5=D	V5=G			e1,2=VD	V5=M		
e3,4=D				e3,2=VD			
e5,4=VD				e3,4=D			

(c)

Fig. 4. (a) Object A. Dashed lines are crossed out numbers represent non-visible edges and faces, respectively. (b) Four prespective views of A and the obtained AGs $\{G1, G2, G3, G4\}$. (c) Attribute values of edge and nodes of the AGs: $G1, G2, G3, G4$. Attributes on the edges are: VS = very similar, S = similar, D = dissimilar and VD = very dissimilar. Attributes on the nodes are: C = cyan, B = blue, R = red, G = green, O = orange, Y = yellow and M = magenta.



(a)

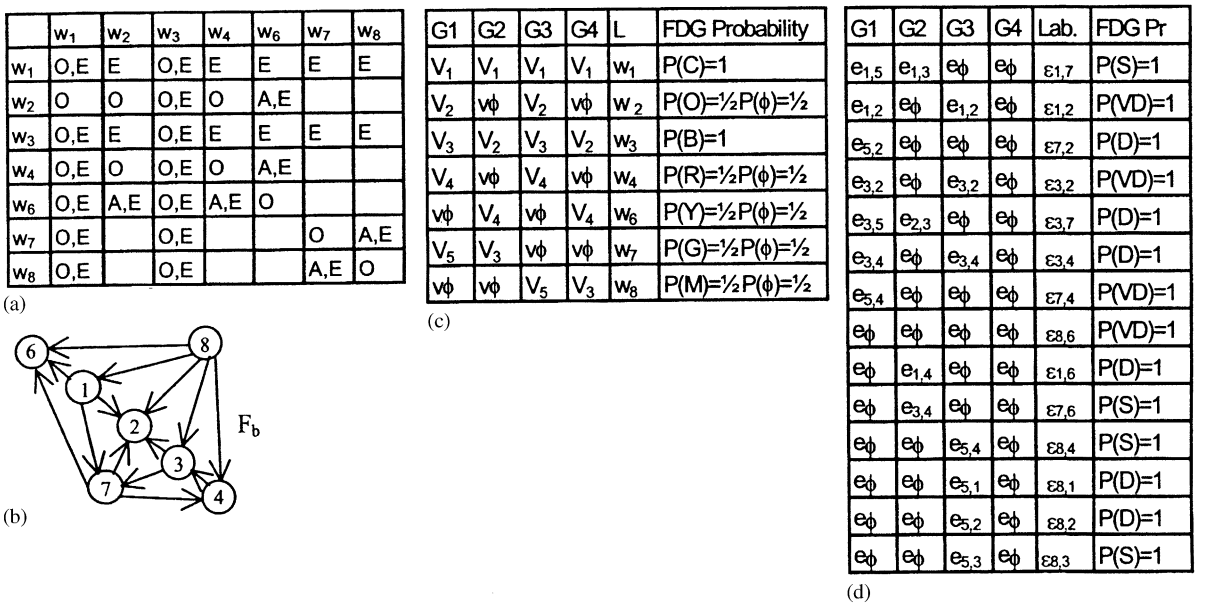
	w_1	w_2	w_3	w_4	w_6	w_7
w_1	O,E	E	O,E	E	E	O,E
w_2	O,E	O	O,E	O	A,E	O,E
w_3	O,E	E	O,E	E	E	O,E
w_4	O,E	O	O,E	O	A,E	O,E
w_6	O,E	A,E	O,E	A,E	O	O,E
w_7	O,E	E	O,E	E	E	O,E

(b)

Vertices				Edges			
G1	G2	Lab	FDG Probability	G1	G2	Lab.	FDG Pr.
V_1	V_1	w_1	P(C)=1	$e_{1,5}$	$e_{1,3}$	$\varepsilon_{1,7}$	P(S)=1
V_2	$V\phi$	w_2	P(O)=1/2, P(ϕ)=1/2	$e_{1,2}$	$e\phi$	$\varepsilon_{1,2}$	P(VD)=1
V_3	V_2	w_3	P(B)=1	$e_{5,2}$	$e\phi$	$\varepsilon_{7,2}$	P(D)=1
V_4	$V\phi$	w_4	P(R)=1/2, P(ϕ)=1/2	$e_{3,2}$	$e\phi$	$\varepsilon_{3,2}$	P(VD)=1
$V\phi$	V_4	w_6	P(Y)=1/2, P(ϕ)=1/2	$e_{3,5}$	$e_{2,3}$	$\varepsilon_{3,7}$	P(D)=1
V_5	V_3	w_7	P(G)=1	$e_{3,4}$	$e\phi$	$\varepsilon_{3,4}$	P(D)=1
				$e_{5,4}$	$e\phi$	$\varepsilon_{7,4}$	P(VD)=1
				$e\phi$	$e_{1,4}$	$\varepsilon_{1,6}$	P(D)=1
				$e\phi$	$e_{3,4}$	$\varepsilon_{7,6}$	P(S)=1

(c)

Fig. 5. FDG F_a: (a) structure, (b) second-order relations. A(row, col)=antagonism ($\omega_{row}, \omega_{col}$), E(row, col)=existence ($\omega_{row}, \omega_{col}$) and O(row, col)= occurrence ($\omega_{row}, \omega_{col}$). (c) Labelling between the two AGs: $G1$ and $G2$ and the obtained probabilities in the FDG. The 4 first columns are related to vertices and the other 4 are related to edges. The non-existence of a vertex or an edge in the AGs is represented by the null vertex v_Φ or the null edge e_Φ, respectively. The column "lab" shows the labelling between the vertices and edges of the two AGs:$G1$ and $G2$ and the ones in the FDG.



	w_1	w_2	w_3	w_4	w_6	w_7	w_8
w_1	O,E	E	O,E	E	E	E	E
w_2	O	O	O,E	O	A,E		
w_3	O,E	E	O,E	E	E	E	E
w_4	O,E	O	O,E	O	A,E		
w_6	O,E	A,E	O,E	A,E	O		
w_7	O,E		O,E			O	A,E
w_8	O,E		O,E			A,E	O

(a)

(b)

G1	G2	G3	G4	L	FDG Probability
V_1	V_1	V_1	V_1	w_1	P(C)=1
V_2	vϕ	V_2	vϕ	w_2	P(O)=½P(ϕ)=½
V_3	V_2	V_3	V_2	w_3	P(B)=1
V_4	vϕ	V_4	vϕ	w_4	P(R)=½P(ϕ)=½
vϕ	V_4	vϕ	V_4	w_6	P(Y)=½P(ϕ)=½
V_5	V_3	vϕ	vϕ	w_7	P(G)=½P(ϕ)=½
vϕ	vϕ	V_5	V_3	w_8	P(M)=½P(ϕ)=½

(c)

G1	G2	G3	G4	Lab.	FDG Pr
$e_{1,5}$	$e_{1,3}$	eϕ	eϕ	$\varepsilon_{1,7}$	P(S)=1
$e_{1,2}$	eϕ	$e_{1,2}$	eϕ	$\varepsilon_{1,2}$	P(VD)=1
$e_{5,2}$	eϕ	eϕ	eϕ	$\varepsilon_{7,2}$	P(D)=1
$e_{3,2}$	eϕ	$e_{3,2}$	eϕ	$\varepsilon_{3,2}$	P(VD)=1
$e_{3,5}$	$e_{2,3}$	eϕ	eϕ	$\varepsilon_{3,7}$	P(D)=1
$e_{3,4}$	eϕ	$e_{3,4}$	eϕ	$\varepsilon_{3,4}$	P(D)=1
$e_{5,4}$	eϕ	eϕ	eϕ	$\varepsilon_{7,4}$	P(VD)=1
eϕ	eϕ	eϕ	eϕ	$\varepsilon_{8,6}$	P(VD)=1
eϕ	$e_{1,4}$	eϕ	eϕ	$\varepsilon_{1,6}$	P(D)=1
eϕ	$e_{3,4}$	eϕ	eϕ	$\varepsilon_{7,6}$	P(S)=1
eϕ	eϕ	$e_{5,4}$	eϕ	$\varepsilon_{8,4}$	P(S)=1
eϕ	eϕ	$e_{5,1}$	eϕ	$\varepsilon_{8,1}$	P(D)=1
eϕ	eϕ	$e_{5,2}$	eϕ	$\varepsilon_{8,2}$	P(D)=1
eϕ	eϕ	$e_{5,3}$	eϕ	$\varepsilon_{8,3}$	P(S)=1

(d)

Fig. 6. FDG F_b generated by 4 views of object A or AGs: $G1, G2, G3$ and $G4$: (a) second-order relations; (b) structure; (c) probabilities of FDG nodes; (d) probabilities of FDG arcs.

is an antagonism between face$_2$ and face$_6$ because they have not been seen at the same time (they cannot be seen together indeed). There is an occurrence from face$_2$ to face$_7$ because when face$_2$ has not been occluded, neither face$_7$. There are existence relations between all the vertices, except ω_2 and ω_4 because both are occluded at view$_2$. Note that ω_2 and ω_6 are related with both antagonism and existence relations. It is proved in Ref. [1] that it is not possible that two vertices be related through the three second-order relations.

The FDG F_b (Fig. 6) represents again object A but considering now the four views (View$_1$–View$_4$). F_b incorporates a new vertex representing face$_8$ of the object, which is visible in View$_3$ and View$_4$. Face$_5$ is not visible in any of the four views, and therefore there is no vertex in the FDG representing this face. Face$_7$ is visible in View$_1$ and View$_2$ but it is occluded in View$_3$ and View$_4$. For this reason, Face$_7$ has a zero probability of being null in F_a but the probability of being null in F_b is $\frac{1}{2}$.

Moreover, some occurrence and existence relations disappear, e.g. the occurrences in which ω_7 is the second element. It is easily proved that, when new AGs are incorporated in FDGs, relations between previously existing vertices can be kept or disappear, but they cannot be incorporated. New relations appear only when at least one of the involved vertices has just been incorporated in the FDG.

4. Distance measure between AGs and FDGs

Two distance measures are presented in this section to provide a quantitative value of the match between an AG G (data graph) and an FDG F (model graph). They are somehow related to the maximum a posteriori probability $P(f|G)$ of a labelling function $f: G \to F$ given the measurements and structure of the data graph G. The Bayes theorem allows the maximisation of the product $P(G|f)P(f)$ instead of the posterior probability $P(f|G)$ to arrive at an optimal matching f^*. And assuming a uniform probability distribution among the valid labelling functions (those that satisfy the required structural constraints), the problem is reduced to that of filtering the invalid mappings and maximising $P(G|f)$ within the remaining set H of allowable configurations. Alternatively, we may attempt to minimise a *global cost* measure C_f of a morphism f in the set H, by taking the cost as a monotonic decreasing function of the conditional probability of the data graph given the labelling function, $C_f = \text{func}(P(G|f))$. For instance, $C_f = -\ln(P(G|f))$ would be a possible choice.

Once a cost measure C_f is defined, a measure of dissimilarity between G and F (from here, distance measure) and the optimal labelling f^* are defined respectively as

$$d = \min_{f \in H}\{C_f\} \quad \text{and} \quad f^* = \arg\min_{f \in H}\{C_f\}. \tag{1}$$

Note that the joint probability $P(G|f)$ cannot be estimated directly and has to be approximated by the product of the first-order probabilities of the elements,

$$P(G|f) = \prod_{\forall x} \Pr(\gamma(y) = \gamma(x) \,|\, f(x) = y), \tag{2}$$

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 mapping with null elements).

Thus, the previous choice for the global cost C_f gives

$$C_f = -\sum_{\forall x} \ln(Pr(\gamma(y) = \gamma(x) \mid f(x) = y)). \quad (3)$$

However, only if that one graph element had a probability of zero, the joint probability would be zero and C_f would be infinite. Since this may happen due to the noisy presence of an unexpected element (insertion) or the absence of a model's element (deletion), only if 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 $P(G|f) = 0$ should not be discarded since they could be the result of a noisy feature extraction and graph formation. In addition, a model (FDG) should match to a certain degree not only the objects (AGs) in its learning set but also the ones that are "near". Hence, for practical purposes it is more appropriate to decompose the global cost C_f into the sum of some *bounded* individual costs, one for each of the graph element matches. Although it has the major flaw that C_f is no longer a monotonically decreasing function of the probability $P(G|f)$ 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 graph elements, $C(x, y) = \text{func}(\Pr(\gamma(y) = \gamma(x)|f(x) = y))$, as far as it is bounded by some fixed constant, $C(x, y) \leqslant \text{Max}$, for instance $C(x, y) \leqslant 1$.

Moreover, for the sake of robustness, the mapping will not be defined from the initial AG which represents the pattern to the initial FDG which represents the class or model, 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. The number of vertices k in the extended graphs is theoretically 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 with an underlying structure (Σ_v, Σ_e) and F' be a k-extension of the FDG F with an underlying structure $(\Sigma_\omega, \Sigma_\varepsilon)$. 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). The labelling function $f : G' \rightarrow F'$ is actually defined as a pair of morphisms $f = (f_v, f_e)$, where $f_v : \Sigma_v \rightarrow \Sigma_\omega$ is a mapping defined on the vertices and $f_e : \Sigma_e \rightarrow \Sigma_\varepsilon$ is a mapping defined on the arcs.

Now, we define the individual cost of matching a pair of elements as a normalised function depending on their dissimilarity, 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,

$$C(x, y) = \begin{cases} \dfrac{-\ln(\Pr(\gamma(y) = \gamma(x) \mid f(x) = y))}{-\ln(K_{Pr})} & \text{if } \Pr(\gamma(y) = \gamma(x) \mid f(x) = y) \geqslant K_{Pr}, \\ 1 & \text{otherwise}, \end{cases} \quad (4)$$

where the cost $C(x, y)$ is bounded by $[0, 1]$, and the positive constant $K_{Pr} \in [0, 1]$ 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, i.e. $\Pr(\gamma(y) = \gamma(x)|f(x) = y) \leqslant K_{Pr}$.

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

$$C_{f_v}(v_i, \omega_q) = \begin{cases} \dfrac{-\ln(p_q(\mathbf{a}_i))}{-\ln(K_{Pr})} & \text{if } p_q(\mathbf{a}_i) \geqslant K_{Pr}, \\ 1 & \text{otherwise}, \end{cases} \quad (5)$$

where $f_v(v_i) = \omega_q$ and $\gamma_v(v_i) = \mathbf{a}_i$. And in the case of the arcs, we define the cost as

$$C_{f_e}(e_{ij}, \varepsilon_{ab}) = \begin{cases} \dfrac{-\ln(q_n(\mathbf{b}_m))}{-\ln(K_{Pr})} & \text{if } q_n(\mathbf{b}_m) \geqslant K_{Pr}, \\ 1 & \text{otherwise}, \end{cases} \quad (6)$$

where $f_e(e_{ij}) = \varepsilon_{ab}$, $\gamma_e(e_{ij}) = \mathbf{b}_m$ in the AG arc and $\gamma_\varepsilon(\varepsilon_{ab}) = \beta_n$ in the matched FDG arc.

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 it depends on the existence of the two extreme vertices. But if we consider that the AG is structurally correct, e_{ij} has to be a null arc in that case, then we apply $\Pr(\beta_n = \Phi | \alpha_a = \Phi \vee \alpha_b = \Phi) = 1$, which gives $C_{f_e}(e_{ij}, \varepsilon_{ab}) = 0$.

The global cost of a given mapping f based only on first-order information is therefore computed as the sum of the individual costs of all the matches between graph elements,

$$C_f = \sum_{\forall x} C(x, f(x)) = \sum_{\forall v_i \in \Sigma_v \text{ of } G'} C_{f_v}(v_i, f_v(v_i)) + \sum_{\forall e_{ij} \in \Sigma_e \text{ of } G'} C_{f_e}(e_{ij}, f_e(e_{ij})). \quad (7)$$

Now, in order to take into account the second-order information included in the FDG boolean functions, there are two options, which lead to the two different distance measures that are presented in the following subsections.

4.1. Distance measure between AGs and FDGs using second-order constraints

The first approach is to consider the antagonism, occurrence and existence relations in the FDG as second-order constraints that must be fulfilled by any valid morphism $f \in H$. We say that $f \in F_{v,e}$, if the morphism is bijective and structurally correct, and $f \in F_A$, $f \in F_O$ and $f \in F_E$, if the antagonism, occurrence and existence constraints are met, respectively. More precisely, the three second-order constraints are defined as follows:

$$\begin{aligned} f \in F_A \\ &\Leftrightarrow (A_v(\omega_p, \omega_q) = 1 \wedge f_v(v_i) = \omega_p \wedge f_v(v_j) = \omega_q) \\ &\Rightarrow (\mathbf{a}_i = \Phi \vee \mathbf{a}_j = \Phi \vee p_p(\Phi) = 1 \vee p_q(\Phi) = 1), \end{aligned} \quad (8)$$

$$\begin{aligned} f \in F_O \\ &\Leftrightarrow (O_v(\omega_p, \omega_q) = 1 \wedge f_v(v_i) = \omega_p \wedge f_v(v_j) = \omega_q) \\ &\Rightarrow (\mathbf{a}_i = \Phi \vee \mathbf{a}_j \neq \Phi \vee p_p(\Phi) = 1 \vee p_q(\Phi) = 1), \end{aligned} \quad (9)$$

$$\begin{aligned} f \in F_E \\ &\Leftrightarrow (E_v(\omega_p, \omega_q) = 1 \wedge f_v(v_i) = \omega_p \wedge f_v(v_j) = \omega_q) \\ &\Rightarrow (\mathbf{a}_i \neq \Phi \vee \mathbf{a}_j \neq \Phi \vee p_p(\Phi) = 1 \vee p_q(\Phi) = 1), \end{aligned} \quad (10)$$

where $\gamma_v(v_i) = \mathbf{a}_i$ and $\gamma_v(v_j) = \mathbf{a}_j$ refer to the attribute values of nodes v_i and v_j in the AG. The two right-most terms in the consequent of the above rules allow the match of a non-null vertex of the AG to an extended null vertex of the FDG (insertion) when a second-order relation involving the FDG null vertex is not satisfied. Otherwise, the FDG null vertices would never be matched to actual vertices in the AG and, consequently, they would be of no help in bringing some flexibility to the matching process.

Finally, the first distance measure between an AG and an FDG is defined as the minimum cost achieved by a valid morphism $f \in F_{v,e} \cap F_A \cap F_O \cap F_E$, i.e.

$$d_f = \min_{f \in \{F_{v,e} \cap F_A \cap F_O \cap F_E\}} \{C_f\},$$

where (11)

$$\begin{aligned} C_f = &\sum_{\forall v_i \in \Sigma_v \text{ of } G'} C_{f_v}(v_i, f_v(v_i)) \\ &+ \sum_{\forall e_{ij} \in \Sigma_e \text{ of } G'} C_{f_e}(e_{ij}, f_e(e_{ij})). \end{aligned} \quad (11)$$

A comparison of the distance above with a distance based on edit operations [14] is presented in Ref. [22]. In summary, substitution and deletion costs are variable depending on the probability density functions of the random elements, whereas the insertion cost is constant.

4.2. Distance between AGs and FDGs relaxing second-order constraints

In real applications, AGs can be distorted or noisy, and therefore, the constraints associated with the second-order relations have to be relaxed to avoid a noisy AG being misclassified due to non-fulfilment of any of them. For instance, the deletion of a strict non-null vertex of the FDG (with a zero probability of being null) will almost always imply the non-fulfilment of some of the existence or occurrence constraints induced by that strict non-null vertex.

To gain more flexibility and robustness, some non-negative costs can be added to the global cost of the labelling when second-order constraints are not met. Given that $f_v(v_i) = \omega_p$ and $f_v(v_j) = \omega_q$, Eqs. (12)–(14) show these additional costs, which can be only 1 or 0, associated with the three relations of antagonism, occurrence and existence between pairs of vertices. 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:

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

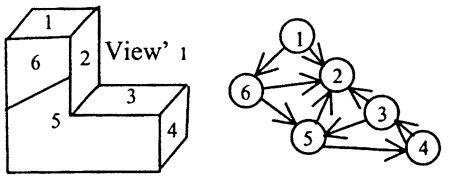
$$C_{O_v}(v_i, v_j, \omega_p, \omega_q) = \begin{cases} O_v(\omega_p, \omega_q) & \text{if } (\mathbf{a}_i \neq \Phi \wedge \mathbf{a}_j = \Phi \wedge p_p(\Phi) \neq 1 \wedge p_q(\Phi) \neq 1), \\ 0 & \text{otherwise,} \end{cases} \quad (13)$$

$$C_{E_v}(v_i, v_j, \omega_p, \omega_q) = \begin{cases} E_v(\omega_p, \omega_q) & \text{if } (\mathbf{a}_i = \Phi \wedge \mathbf{a}_j = \Phi \wedge p_p(\Phi) \neq 1 \wedge p_q(\Phi) \neq 1), \\ 0 & \text{otherwise.} \end{cases} \quad (14)$$

Now, the global cost C_f of the labelling function f is redefined by C_f^R with two terms that depend on the first-order probability information, which are the original ones, and three more terms that depend on the second-order costs:

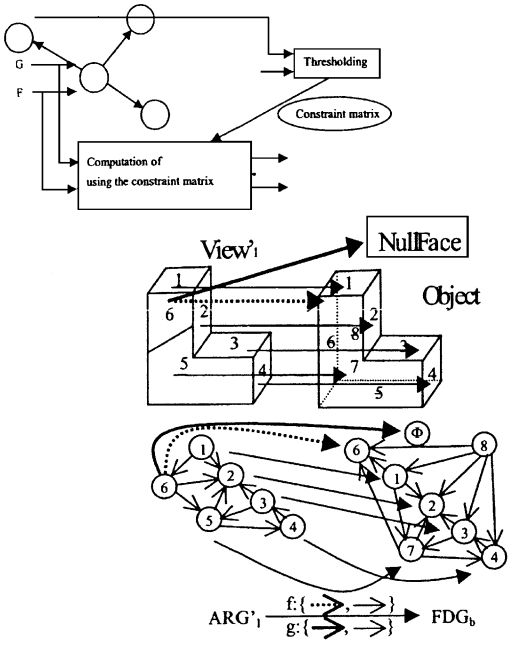
$$\begin{aligned} C_f^R = &K_v * \sum_{\forall v_i \in \Sigma_v \text{ of } G'} C_{f_v}(v_i, f_v(v_i)) \\ &+ K_e * \sum_{\forall e_{ij} \in \Sigma_e \text{ of } G'} C_{f_e}(e_{ij}, f_e(e_{ij})) \\ &+ K_A * \sum_{\forall v_i, v_j \in \Sigma_v \text{ of } G'} C_{A_v}(v_i, v_j, f_v(v_i), f_v(v_j)) \\ &+ K_O * \sum_{\forall v_i, v_j \in \Sigma_v \text{ of } G'} C_{O_v}(v_i, v_j, f_v(v_i), f_v(v_j)) \\ &+ K_E * \sum_{\forall v_i, v_j \in \Sigma_v \text{ of } G'} C_{E_v}(v_i, v_j, f_v(v_i), f_v(v_j)). \end{aligned} \quad (15)$$

The five terms are weighted by non-negative constants K_v, K_e, K_A, K_o and K_E, to compensate for the different num-



Vertex	Attrib.	Arc	Attrib.
V 1	Cyan	e 1 ,2	Very Dissimilar
V 2	Orange	e 5 ,2	Dissimilar
V 3	Blue	e 3 ,2	Very Dissimilar
V 4	Red	e 3 ,5	Dissimilar
V 5	Green	e 3 ,4	Dissimilar
V 6	Brown	e 5 ,4	Very Dissimilar
		e 1 ,6	Dissimilar
		e 6 ,2	Similar
		e 5 ,6	Similar

Fig. 7. View'$_1$, which is similar to View$_1$, with the spurious face$_6$ and the resulting AGG'_1.



G'1	labelling f	Prob(f)	Cost(f)	labelling g	Prob(g)	Cost(g)
V_1	w_1	1	0	w_1	1	0
V_2	w_2	½	0.301	w_2	½	0.301
V_3	w_3	1	0	w_3	1	0
V_4	w_4	½	0.301	w_4	½	0.301
V_5	w_7	½	0.301	w_7	½	0.301
V_6	w_6	0	1	w_ϕ	0	1
$e_{1,5}$ (ϕ)	$\varepsilon_{1,7}$	0	1	$\varepsilon_{1,7}$	1	0
$e_{1,2}$	$\varepsilon_{1,2}$	1	0	$\varepsilon_{1,2}$	1	0
$e_{5,2}$	$\varepsilon_{7,2}$	1	0	$\varepsilon_{7,2}$	1	0
$e_{3,2}$	$\varepsilon_{3,2}$	1	0	$\varepsilon_{3,2}$	1	0
$e_{3,5}$	$\varepsilon_{3,7}$	1	0	$\varepsilon_{3,7}$	1	0
$e_{3,4}$	$\varepsilon_{3,4}$	1	0	$\varepsilon_{3,4}$	1	0
$e_{5,4}$	$\varepsilon_{7,4}$	1	0	$\varepsilon_{7,4}$	1	0
$e_{6,2}$	$\varepsilon_{6,2}$ {P(ϕ)=1}	0	1	$\varepsilon_{1,\phi}$	0	1
$e_{1,6}$	$e_{1,6}$	1	0	$\varepsilon_{\phi,2}$	0	1
$e_{5,6}$	$\varepsilon_{7,6}$	1	0	$\varepsilon_{7,\phi}$	0	1
	Antag(w_2,w_6)		K_A			
		Total:	3.903+K_A		Total:	4.903

Fig. 8. Labellings f and g between G'_1 and F_b with their associated probabilities and costs.

ber 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. Finally, the second distance measure proposed between an AG and an FDG, d_f^R, is defined as the minimum cost C_f^R achieved by a bijective and structurally valid function f:

$$d_f^R = \min_{f \in F_{v,e}} \{C_f^R\}. \tag{16}$$

Note that both distance measures d_f and d_f^R will be equivalent in practice if we set $K_A = K_O = K_E =$ BigNumber, since in this case, the optimal labelling is forced to satisfy the second-order constraints, i.e. $f^* \in \{F_A \cap F_O \cap F_E\}$.

4.3. An example of matching a 3D-object using the proposed distance measure

We are interested in matching the AG G'_1, corresponding to View$'_1$ (Fig. 7) against the FDG F_b (Fig. 6). This view is similar to View$_1$ (Fig. 4) but the segmentation process has generated a new region represented by the spurious face$_6$. Its hue has been assigned to the average hue of their adjacent faces. Since this region has never appeared in any view used to learn the object B, it should be matched to an FDG null vertex in the optimal isomorphism.

When one attempts to recognise a 3D-object from a single view, the occluded parts of the object should not have an influence on the distance value (i.e. we are looking for a subgraph isomorphism rather than a graph isomorphism). In our approach this can be accomplished by not taking into account the costs of deleting vertices of the FDG (matching AG vertices to null FDG vertices).

Fig. 8 shows two sub-graph isomorphisms between G'_1 and F_b. Labelling f is represented by arrows $\rightarrow$ and $\dashrightarrow$ and labelling g is represented by $\rightarrow$ and $\Rightarrow$. The difference between both functions is in the matching of v_6. In f, it

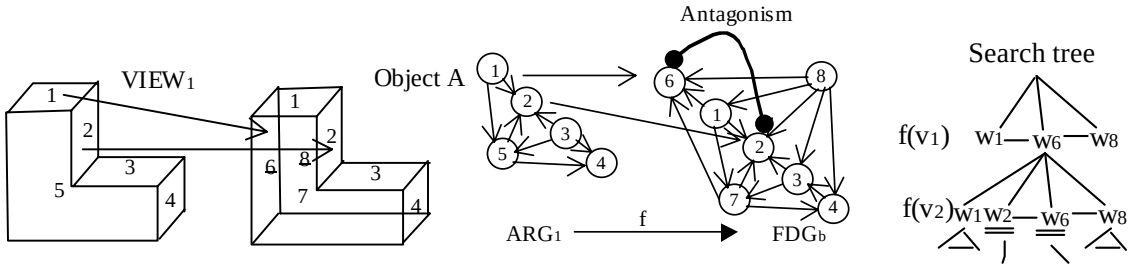


Fig. 9. Tha mapping (v_1, ω_6) and (v_2, ω_2) is rejected due to the antagonism relation, and the search tree is pruned.

is matched to ω_6, although $p_6(\text{brown}) = 0$, and in g, it is matched to a null FDG vertex.

The table shows the probabilities and costs of the matching elements for the sub-graph isomorphisms f and g. The threshold on the probabilities used to calculate the costs is $K_{pr} = 0.1$ (Eqs. (5) and (6)). The arcs $e_{1,5}$ and $\varepsilon_{6,2}$ are extended null arcs since the edge between face$_1$ and face$_5$ in View$'_1$ and the edge between face$_6$ and face$_2$ in the object, do not exist. In the case of the labelling f, vertices ω_2 and ω_6 are matched and then the antagonism between both vertices has to be considered with a cost K_A. If antagonisms are not considered, $K_A = 0$, then the optimal labelling is f with a cost of 3.903. But, if $K_A > 1$ then the cost of f is bigger than the cost of g and so f is discarded as the best labelling and the spurious Face$_6$ is rejected. In the case of $K_A = 1$, the computed labelling depends on the implementation of the algorithm since the minimum cost is 4.903 in both labellings.

5. Algorithms for computing the distance measures

An algorithm for error-tolerant graph matching that calculates the distance measure and the optimal morphism between an AG and an FDG was presented in Ref. [23]. We defined an approach based on a tree search by A* algorithm reminiscent of the algorithm presented in Ref. [5], where the search space is reduced by a *branch and bound* technique. We showed that by incorporating some second-order constraints (antagonism relations), the search tree can be somewhat pruned, although the exponential combinatorial complexity remains. To overcome this drawback, we presented in Ref. [19] an efficient algorithm that computes a sub-optimal distance between an AG and an FDG, together with its corresponding morphism, in polynomial time.

Nevertheless, we refer again to the 3D example to show that antagonism relations not only are useful for recognition purposes but also to prune the search tree computed in the matching algorithm and thus to decrease the time used to compute the distance. Fig. 9 shows a partial labelling between G_1 (Fig. 4) and F_b (Fig. 6) and the search tree computed by the matching algorithm. When the algorithm considers the mapping (v_1, ω_6), the mapping (v_2, ω_2) is rejected and the matching algorithm backtracks, since there is an antagonism between ω_2 and ω_6 and therefore, it is not possible that these two FDG vertices be matched with two AG vertices. One of the FDG vertices does not have to be matched (it is considered to be matched with a null AG vertex).

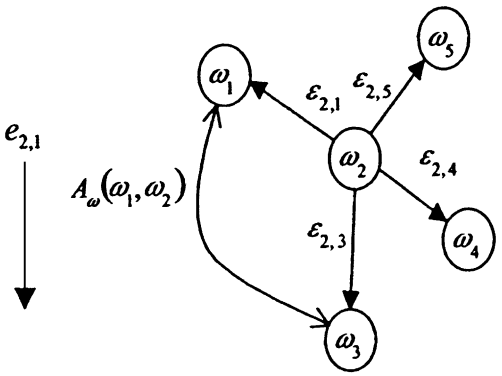


Fig. 10. Expanded vertex of an AG (a) and an FDG (b).

The *efficient algorithm* to compute the distance between AGs and FDGs reduces enormously the search space of our branch and bound algorithm [23] by initially discarding most of the mappings between vertices, using a function based on the distance between the sub-units of the graphs to be matched. The match of a pair of vertices is discarded if the distance between their related sub-units is higher than a threshold. Thus, the graphs are broken down into manageable sub-units that overlap, so that two adjacent sub-units contain information about each other through the mutual graph elements they contain, similar to the discrete relaxation method presented in Ref. [24]. We refer to these sub-units as *expanded vertices* (Fig. 10) and they are structured by a central vertex and all the adjacent vertices connected to it by an outgoing arc (of the central vertex).

Fig. 11 shows the basic scheme of the method proposed to compute a sub-optimal approximation of the distance measure $d_f^R(G,F)$ between an AG G and an FDG F, that will be denoted $\hat{d_f^R}(G,F)$. The method consists of three main

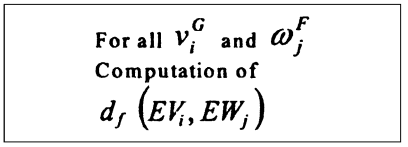


Fig. 11. Scheme of an efficient algorithm to compute the distance between AGs and FDGs.

modules. The first one computes all the distances between *expanded vertices* using the distance measure, $d_f(EV_i, EW_j)$. Note that an expanded vertex of an AG or an FDG is also an AG or an FDG and for this reason, the distance between AGs and FDGs can be computed. The thresholding module decides which matches of vertices are discarded depending on an externally imposed threshold τ. The last module computes the sub-optimal distance $d_f^R(G,F)$ using the branch and bound algorithm presented in Ref. [23] but only on the set of vertex mappings allowed by the thresholding module.

6. Application of FDGs for modelling and recognition of objects

We present two different applications of FDGs to assess the validity of our approach: 3D-object recognition and face identification. In both applications, there is a set of images of each class (of a 3D-object or a person) and an adjacency graph is extracted from each image. The learning process is based on synthesising an FDG per each class (or a set of adjacency graphs) and the classification process matches the unclassified pattern (represented by an adjacency graph) with a class (represented by an FDG).

6.1. Application of FDGs to 3D-object recognition

FDGs are applied here to 3D-object representation and recognition. Similar to the example that we have been using throughout the paper, views are structured by adjacency graphs. The attribute of the vertices is the average hue of the region (cyclic range from 0 to 49) and the attribute of the edges is the difference between the colours of the two neighbouring regions.

We first present an experimental validation of FDGs using artificial 3D-objects in which the adjacency graphs have been extracted manually and afterwards we present a real application on an image database in which the graphs have been extracted automatically. The advantages of the experimental validation are that the results do not depend on the segmentation process and that we can use a *supervised synthesis* [21], since we know which vertices of the AGs represent the same planar face of the object. Thus, we can discern between the effects of computing the distance measure using different values of the costs on the second-order relations. In the real application, we show the capacity of FDGs to keep the structural and semantic knowledge of an object despite the noise introduced by the segmentation process and an *automatic synthesis* [22].

In both experiments, we want to show the influence of the antagonism and occurrence costs on the recognition ratio and the time spent to classify the objects. Thus, we first tested combinations of the antagonism and occurrence costs considering three cases (Tables 1 and 3): with cost null ($K_3 = 0$ or $K_4 = 0$), with high costs ($K_3 = 1000$ or $K_4 = 1000$) and with moderate costs ($K_3 = 1$ or $K_4 = 1$). When these costs are null, the second-order relations are not considered and the system is similar to FORGs. When a high cost is imposed, the non-fulfilment of only one antagonism or occurrence restriction causes the labelling to be rejected. Conversely, when a moderate cost is used, the non-fulfilment of a second-order restriction causes only a slight increase of the global cost. The cost associated with the existence relations is not involved because no existence relations are synthesised in these experiments (there is no pair of nodes such that one or the other can be seen in all the views). From the recognition ratio results, we arrived at the conclusion that the best combination is the one that has both moderate costs. Then we compared FDGs to two other classical methods (Tables 2 and 5). The first one was the 3-nearest neighbours (3-NN) classifier: The edit-operations distance [25] was used as the distance between elements. In this case, the costs of insertion and deletions were set to 1 and the cost of substitution was 0 if $\text{cyclic\_dist}(\text{hue1}, \text{hue2}) = 0$, cost $= \frac{1}{2}$ if $\text{cyclic\_distance}(\text{hue1}, \text{hue2}) = 1$ and cost $= 1$ if $\text{cyclic\_dist}(\text{hue1}, \text{hue2}) > 1$. And the second classical method compared was random graphs.

6.1.1. Supervised synthesis on artificial objects

We designed five objects using a CAD program (Fig. 12). After that, we took five sets of views from these objects and from these views we extracted a total of 101 adjacency graphs. To obtain the AGs of the test set and of the reference set, we modified the attribute values of the vertices and arcs of the adjacency graphs by adding zero-mean Gaussian noise with different variances. Moreover, some vertices and arcs were inserted and deleted randomly in some cases. The FDGs were synthesised using the AGs that belonged to the same 3D-object and using the *synthesis given a common labelling from a set of AGs* described in Ref. [21]. Table 1 shows the ratio of correctness (%) for different levels of noise and applying several costs on the antagonisms and occurrences. We see that the best results appear always when we use moderate second-order costs. Furthermore, when noise increases, the recognition ratio decreases drastically when we use high costs but there is only a slight decrease when we use moderate costs. Moreover, in Table 2 we compare the FDGs (with second-order costs) with other two methods. The FDG classifier always obtains better results than the 3-NN and the random graph classifiers.

Table 1
FDGs ratio of correctness %

No. of vertices ins. and del.		0	0	0	0	0	1	2	1
Standard deviation		0	2	4	8	12	0	0	8
Cost on antag.	Cost on occu.								
Moderate	Moderate	100	98	97	95	92	89	85	83
High	Null	100	92	89	87	84	61	54	57
Null	High	100	91	89	88	85	62	59	59
High	High	100	95	90	86	80	60	53	56
Moderate	Null	100	92	91	91	87	80	75	75
Null	Moderate	100	95	92	91	86	81	77	76
Null	Null	100	90	89	88	86	70	67	68

Table 2
FDGs (moderate second-order costs), FORGs and 3-NN ratio of correctness (%)

No. of vertices ins. or del.	0	0	0	0	0	1	2	1
Standard deviation	0	2	4	8	12	0	0	8
FDGs (moderate costs)	100	98	97	95	92	89	85	83
Random graphs (FORGs)	100	90	89	88	86	70	67	68
3-NN (edit op. distance)	100	98	82	62	52	90	58	58

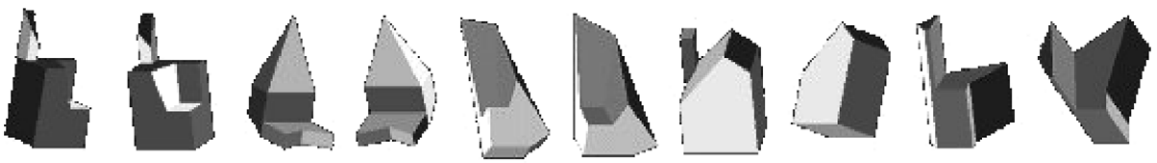


Fig. 12. Two views of 5 artificial objects designed by a CAD program.

The difference of the ratio between FDGs and the other two methods increases when the noise also increases. FORGs obtain better results than the 3-NN only when the noise is high.

6.1.2. Unsupervised synthesis on real life objects

Images were extracted from the database COIL-100 from Columbia University (www.cs.columbia.edu/CAVE/research/softlib/coil-100.html). It is composed of 100 isolated objects and for each object there are 72 views (one view each 5 degrees). Adjacency graphs are obtained by the segmentation process presented in Ref. [26]. Fig. 13 shows 20 objects at angle 100 and their segmented images with the adjacency graphs. The test set was composed of 36 views per object (taken at the angles 0,10, 20 and so on), whereas the reference of set was composed of the 36 remaining views (taken at the angels 5, 15, 25 and so on). FDGs were synthesised automatically using the AGs in the reference set that represent the same object. The method of *incremental synthesis*, in which the FDGs are updated while new AGs are sequentially presented, was applied [22]. We made 6 different experiments in which the number of FDGs that represents each 3D-object varied. If the 3D-object was represented by only one FDG, the 36 AGs from the reference set that represent the 3D-object were used to synthesise the FDG. If it was represented by 2 FDGs, the 18 first and consecutive AGs from the reference set were used to synthesise one of the FDGs and the other 18 AGs were used to synthesise the other FDG. A similar method was used for the other experiments with 3, 4, 6 and 9 FDGs per 3D-object.

Table 3 shows the ratio of correctness of the FDG classifier varying the number of FDGs per each object. Similar to the previous experimental results (Table 2), the correctness is higher when second-order relations are used with a moderate cost. The best result appears when each object is represented by 4 FDGs, that is, each FDG represents 90° of the 3D-object. When objects are represented by 9 FDGs, each FDG represents 40° of the 3D-object and 4 AGs per FDG, there is poor probabilistic knowledge and therefore the costs on the vertices and arcs are coarse. Moreover, when objects are represented by only 1 or 2 FDGs, there are too

Fig. 13. The 20 selected objects at angle 100 and the segmented images with the AGs.

Table 3
Ratio of correctness (%) using several costs on antagonisms and occurrences

Number of FDGs per object		9	6	4	3	2	1
Number of AGs per FDG		4	6	9	12	18	36
Cost on antag.	Cost on occu.						
Moderate	Moderate	65	69	79	70	60	54
High	Null	60	51	47	41	32	12
Null	High	59	49	45	37	25	10
High	High	52	41	43	33	18	8
Moderate	Null	62	60	65	61	52	45
Null	Moderate	61	59	63	57	45	42
(FORGs)		40	44	45	53	27	14
3-NN (edit op. distance)		63					

much spurious regions (produced in the segmentation process) to keep the structural and semantic knowledge of the object.

Moreover, Table 4 shows the average time in milliseconds spent to compute the classification of a new AG. The algorithms were implemented in Visual C++ and run on a Pentium 800 MHz. The higher the cost on the antagonisms (K_A), the search tree is more pruned and so, the faster is the computation of the distance. Moreover, when the number of FDGs per object decreases, the number of comparisons also decreases but the time spent to compute the distance increases since the FDGs are bigger.

Table 4
Average time in milliseconds for each AG

No. of FDGs per object	9	6	4	3	2	1
No. of AGs per FDG	4	6	9	12	18	36
FDG (high, high)	125	89	32	44	64	59
FDG (moderate, moderate)	150	101	47	56	72	82
FORG	187	123	253	392	814	1203
3-NN	93					

Fig. 14. Identification and location of the pre-established points and four selected faces.

Table 5
Correctness (%) for different representations of the 3D-object applying FORGs, FDGs and 3-NN. We used 3 levels of occlusion of the test images: 0%, 20% and 30%

Occlusion	0%						20%						40%					
Number of FDGs per object	9	6	4	3	2	1	9	6	4	3	2	1	9	6	4	3	2	1
Number of AGs per FDG	4	6	9	12	18	36	4	6	9	12	18	36	4	6	9	12	18	36
FDGs (moderate costs)	65	69	79	70	60	54	62	65	77	60	61	45	51	59	62	58	49	34
Random graphs (FORGs)	40	54	45	33	27	14	33	49	40	37	19	10	29	24	25	23	20	10
3-NN (edit op. distance)	63						54						33					

And finally, Table 5 shows that FDGs with moderate second-order costs obtain the best results independent of the occlusion on the test image. The best results using FORGs are obtained when 6 AGs are used to synthesise the FDGs, in contrast to the FDGs where the best results are obtained when 9 AGs are used. This is because FORGs do not have the second-order information. Note that FDGs obtained much better results than FORGs and the 3-NN.

6.2. *Application of FDGs to face identification*

In our approach, faces were defined by AGs, in which vertices and arcs represented pre-established points and the distances between them, respectively. Fig. 14 shows the identification of the pre-established points in one of the images and the location of 36 points, and also, four faces of the database.

In the learning process, FDGs were obtained by the *supervised synthesis* [21] and the AGs obtained by the faces of the same person. To carry out the identification experiments, we used images of 100 people and 10 images per person. Half of the images were composed of the reference set and the other half the test set. These images were taken from the database at the address http://cswww.essex.ac.uk/projects/vision/allfaces/faces94. See Ref. [20] for more details. We compared the FDG approach with the RG and the 5-NN approach, where the distance between elements was defined as the edit-operations distance between AGs [25]. Table 6 shows the correctness ratio varying the ratio of occlusion of the faces in the test set. In all the cases, the FDG classifier obtained better results than the other classical approaches.

Table 6
Correctness ratio (%) using FDGs, random graphs and 5-NN

Occlusion	Full face	20%	40%	60%	80%
FDGs (moderate second-order costs)	95	92	88	82	64
Random graphs (FORGs)	89	85	70	63	47
5-NN (edit op. distance)	91	90	83	78	61

7. Conclusions

FDGs are a type of compact representation of a set of AGs that borrow from *Random Graphs* the capability of probabilistic modelling of structural and attribute information, while improving the capacity of *FORGs* to record structural relationships that consistently appear through the data. This is achieved by incorporating some qualitative knowledge of the second-order probabilities of the elements that are expressed as relations (boolean functions) between pairs of vertices in the FDG.

If only the relations between vertices are considered, the space complexity of the FDG representation does not augment with respect to a first-order random graph, n^2, since

all the relations defined (antagonism, occurrence, existence) can be obtained easily from the first-order marginal probabilities and just one second-order probability for each pair of vertices (namely, the probability of both vertices being null at the same time). However, if arc relations were considered as well, the space complexity obviously would rise to a^2, where a is the number of arcs, and for dense graphs, this would imply a storage (and a time complexity of arc relation creation, maintenance and verification) of order n^4 instead of n^2, which would be a severe drawback in practice. This is the reason why second-order constraints between arcs have not been discussed in this work.

A distance measure between an AG and an FDG has been proposed to be used in the problem of matching an AG to an FDG for recognition or classification purposes. Although, in the origin, a distance measure was derived from the principle of maximum likelihood in a Bayesian framework [1], a more robust measure has been described by considering the effects of extraneous and missing elements only locally and relaxing the second-order hard constraints into moderate costs, at the expense of losing the theoretical link with the maximum likelihood source [27]. These second-order relations are useful not only for increasing the correctness in the classification process but also to reduce the computational time of the labelling process. Nevertheless, the combinatorial complexity of computing the presented distance measure remains. For this reason, an efficient algorithm that yields a suboptimal distance value is presented [23] to reduce dramatically the computational time at the expense of losing the optimal labelling.

In this work, two different methods to synthesise an FDG from a set of AGs were used in the experiments. These methods, together with some others to synthesise an FDG from a set of unlabelled AGs and to cluster, in a non-supervised manner, a set of AGs into a set of FDGs representing subclasses, have been reported elsewhere [21,22].

Some experimental tests were carried out in a 3D-object recognition problem for two different scenarios, an artificial one with a varying controlled noise level in feature extraction and a real-world one, in which the final results are heavily influenced by the image segmentation process. In both cases, an FDG model has been synthesised for each object from a set of views (AGs), including antagonism and occurrence relations between vertices (representing faces of the object). Moreover, some tests on face identification have also been carried out, where an FDG is built from a set of views of a person's face.

In all the experiments, FDGs obtained better results than the 3-NN classifier, which means that it is useful to represent the set of AGs (or object) by only one structure since this object is seen as a whole. However, in the real 3D-object recognition application, it has been shown that it is more effective and robust to partition a single 3D-object into a few FDGs for recognition purposes. This is due to the representation of the joint probability of vertices and arcs by only the first-order probabilities and second-order relations. Furthermore, FDGs obtained also better results than FORGs due to the use of the second-order relations. Since these relations are a simplification of the second-order probabilities, results show that it is better to use moderate second-order costs than high costs.

Finally, some artificial experiments were carried out in Ref. [1] to show the sensibility of the method to failure mismatch. In these experiments AGs were generated artificially and FDGs were built using the methods presented in Ref. [22].

References

[1] F. Serratosa, Function-described graphs for structural pattern recognition, Ph.D. Thesis Dissertation, Universitat Politècnica de Catalunya, Barcelona, Spain, 2000.

[2] A. Sanfeliu, R. Alquézar, J. Andrade, J. Climent, F. Serratosa, J. Vergés, Graph based representations and techniques for image processing and image analysis, Pattern Recognition 35 (2002) 639–650.

[3] A.K.C. Wong, J. Constant, M. You, Random graphs, in: Syntactic and Structural Pattern Recognition: Theory and Applications, World Scientific, Singapore, 1990, pp. 197–234.

[4] A.K.C. Wong, M. You, Entropy and distance of random graphs with application to structural pattern recognition, IEEE Trans. Pattern Anal. Mach. Intell. 7 (1985) 599–609.

[5] A.K.C. Wong, M. You, S.C. Chan, An algorithm for graph optimal monomorphism, IEEE Trans. Systems Man Cybern. 20 (1990) 628–636.

[6] P.H. Winston, Learning structural descriptions from examples, Technical Report MAC-TR-76, Department of Electrical Engineering and Computer Science-MIT, 1976.

[7] F. Stein, G. Medioni, Structural indexing: efficient 3D object recognition IEEE Trans. Pattern Anal. Mach. Intell. 14 (1992) 125–145.

[8] A. Pearce, T. Caelly, W.F. Bischof, Rulegraphs for graph matching in pattern recognition, Pattern Recognition 27 (9) (1994) 1231–1247.

[9] W. Kim, A. Kak, 3D object recognition using bipartite matching embedded in discrete relaxation, IEEE Trans. Pattern Anal. Mach. Intell 13 (1991) 224–251.

[10] K. Sossa, R. Horaud, Model indexing: the graph-hashing approach, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 1992, pp. 811–815.

[11] L.G. Shapiro, R.M. Haralick, Organization of relational models for scene analysis, IEEE Trans. Pattern Anal. Mach. Intell. 4 (1982) 595–602.

[12] A. Sanfeliu, K. Fu, A distance measure between attributed relational graphs for pattern recognition, IEEE Trans. Systems Man Cybern. 13 (1983) 353–362.

[13] A. Sanfeliu, F. Serratosa, R. Alquezar, Clustering of attributed graphs and unsupervised synthesis of function-described graphs, in: Proceedings of ICPR'2000, 15th International Conference on Pattern Recognition, Barcelona, Spain, Vol. 2, 2000, pp. 1026–1029.

[14] B.T. Messmer, H. Bunke, A decision tree approach to graph and subgraph isomorphism detection, Pattern Recognition 32 (1999) 1979–1998.

[15] P. Foggia, R. Genna, M. Vento, Learning in structured domains by attributed relational graphs, Proceedings of the IAPR Workshops SSPR'00 and SPR'00, Alacant, Spain, 2000.
[16] K.P. Chan, Learning templates from fuzzy examples in structural pattern recognition, IEEE Trans. Systems Man Cybern. 26 (1996) 118–123.
[17] A. Sperduti, A. Starita, Supervised neural networks for the classification of structures, IEEE Trans. Neural Networks 8 (3) (1997) 714–735.
[18] K. Sengupta, K. Boyer, Organizing large structural modelbases, IEEE Trans. Pattern Anal. Mach. Intell. 17 (1995) 321–332.
[19] F. Serratosa, A. Sanfeliu, Function-described graphs applied to 3D object recognition, Ninth International Conference on Image Analysis and Processing, Italy, Vol. I, 1997, pp. 701–708.
[20] J. Vergés, A. Sanfeliu, F. Serratosa, R. Alquézar, Face recognition: Graph matching versus neural techniques, Eighth National Symposium on Pattern Recognition and Image Analysis, Vol. I, 1999, pp. 259–266.
[21] R. Alquézar, A. Sanfeliu, F. Serratosa, Synthesis of function-described graphs, Proceedings of SSPR'98, Sydney, Australia, Lecture Notes in Computer Science, Vol. 1451, Spinger, Berlin, 1998, pp. 112–121.
[22] A. Sanfeliu, F. Serratosa, R. Alquezar, Clustering of attributed graphs and unsupervised synthesis of function-described graphs, Proceedings of ICPR'2000, 15th International Conference on Pattern Recognition, Barcelona, Spain, Vol. 2, 2000, pp. 1026–1029.
[23] F. Serratosa, R. Alquezar, A. Sanfeliu, Efficient algorithms for matching attributed graphs and function-described graphs, in: Proceedings of ICPR'2000, 15th International Conference on Pattern Recognition, Barcelona, Spain, Vol. 2, 2000, pp. 871–876.
[24] R.C. Wilson, E.R. Hancock, Structural matching by discrete relaxation, IEEE Trans. Pattern Anal. Mach. Intell. 19 (1997) 634–648.
[25] A. Sanfeliu, K. Fu, A distance measure between attributed relational graphs for pattern recognition, IEEE Trans. Systems Man Cybern. 13 (1983) 353–362.
[26] P.F. Felzenswalb, D.P. Huttenlocher, Image segmentation using local variation, Proceedings of the IEEE Conference on Computer Vision Pattern Recognition, 1998, pp. 98–104.
[27] R. Alquézar, F. Serratosa, A. Sanfeliu, Distance between attributed graphs and function-described graphs relaxing 2nd-order restrictions, Proceedings of the IAPR International Workshops SSPR'2000 and SPR'2000, Barcelona, Spain, Lecture Notes in Computer Science, Vol. 1876, Springer, Berlin, 2000, pp. 277–286.

About the Author—Dr. SANFELIU was born in Barcelona in 1954. He got the Industrial Engineering degree (speciality in Electrical Engineering) in 1978, and the Ph.D. degree in 1982 from the Universitat Politécnica de Catalunya (UPC). He is the General Manager in the Cognivision Research S.L. and Professor in the UPC. He has received the *Prize to the Technology given by the Generalitat de Catalonia* and he is *Fellow of the International Association for Pattern Recognition*. Dr. Sanfeliu has been the Scientific Co-ordinator and researcher in more than 18 scientific projects, five of them European Projects (*RAIM* (BRITE-EURAM BE96-3004), *HITEX* (BRITE-EURAM BE96-3505), *ACIL* (BRITE-EURAM E-7137), *MUVI* (BRITE-EURAM E-4330)) and TEXTERM (GRD1-2000-26817).

He is the co-author of 4 international patents in quality control techniques based on computer vision for the fibre and textile industry. He has been managing several industrial projects in the automation of classification and packaging lines for textile bobbin packages (Bobbin Vision) and also several projects in the automation of electronic management of documents.

Dr. Sanfeliu has published (edited) 10 books, 22 book chapters, 19 journal papers and around 100 conference papers. He is an associated editor of three scientific journals and reviewer of several top scientific journals in pattern recognition and computer vision ("IEEE Transactions on Pattern Analysis and Machine Intelligence", "Computer Vision and Image Understanding", "Image and Vision Computing", "Pattern Recognition", "Pattern Recognition Letters", "International Journal on Document Analysis and Recognition (IJDAR)", "Computación y Sistemas").

From 1979 to 1981, he was visiting researcher in the Department of Prof. K.S. Fu in Purdue University, USA. He joined the faculty of the Polytechnical University of Cataluyna in 1981 as associate professor. He is now professor in the department of "Ingeniería de Sistemas, Automática e Informática Industrial" in the UPC. From 1975 to 1995 he has been a researcher in the "Instituto de Cibernética", research institute which belongs to the National Council of Scientific Research (CSIC) and to the UPC. He is now researcher of the "Institut de Robótica i Informática Industrial (CSIC-UPC)".

Dr. Sanfeliu has been the President of the Spanish Association for Pattern Recognition and Image Analysis from 1984 to 2000, the *Chairman* of the Technical Committee on Syntactic and Structural Pattern Recognition of the International Association for Pattern Recognition and the General Co-Chairman of the International Conference on Pattern Recognition which was held in Barcelona in the year 2000.

About the Author—RENE ALQUEZAR received the licentiate and Ph.D. degrees in Computer Science from the Polytechnical University of Catalonia (UPC), Barcelona, Spain, in 1986 and 1997, respectively. From 1987 to 1991 he was with the Spanish company NTE as technical manager of R& D projects on image processing applications for the European Space Agency (ESA). From 1991 to 1994 he was with the "Institut de Cibernetica", Barcelona, holding a pre-doctoral research grant from the Government of Catalonia and completing the doctoral courses of the UPC on artificial intelligence. He joined the Department of "Llenguatges i Sistemes Informatics", UPC, in 1994 as an associate professor, and since March 2001 he has been professor in the same department. He has been co-chair of the local organising committee of the 15th ICPR held in Barcelona in September 2000.

His current research interests include neural networks, syntactic and structural pattern recognition and computer vision.

About the Author—FRANCESC SERRATOSA was born in Barcelona, 24 may 1967. He received his Computer Science Engineering degree from the Universitat Politècnica de Catalunya (Barcelona) in 1993. Since then, he has been active in research in the areas of computer vision, robotics and structural pattern recognition. He received his Ph.D. from the same university in 2000. He is currently associate professor of computer science at the Universitat Rovira i Virgili (Tarragona), he has published more than 30 papers and he is an active reviewer in some congresses and journals.