Modelling and recognising 3D-objects described by multiple views using Function-Described Graphs

Francesc Serratosa
Dept. d’Enginyeria Informàtica
Universitat Rovira i Virgili
Francesc.Serratosa@etsc.urv.es

Alberto Sanfeliu
Inst. de Robòtica i Informàtica
Univ. Politècnica de Catalunya
asanfeliu@iri.upc.es

René Alquézar
Dept. Lleng. i Sist. Informàtics
Univ. Politècnica de Catalunya
alquezar@lsi.upc.es

Abstract
In this paper we study the application of Function-Described Graphs (FDGs) for 3D-object modeling and recognition. From a set of topological different 2D-views taken of an object, FDGs are synthesized from the attributed adjacency graphs that are extracted for each view. It is shown that, by keeping in the object representation (an FDG) a qualitative information of the 2nd-order joint probabilities between vertices, the object recognition ratio increases while the run time of the classification process decreases.

1. Introduction
A Function-Described Graph (FDG) is a model graph introduced by the authors that contains 1st-order probabilities of attributes and 2nd-order structural information (relations between vertices) to describe a set of Attributed Graphs (AGs) [1-3]. FDGs are synthesized from a learning set of AGs, but they also allow the matching and recognition of graphs that are similar, both structurally and semantically, to those in the learning set. Figure 1 shows schematically the FDG learning and classification processes for the 3D-object modelling and recognition problem. The basic idea is that only a single FDG is synthesized from the graphs that represent several 2D-views of a 3D-object. Thus, in the recognition process, only one comparison is needed between each model represented by an FDG and the unclassified object (2D-view of a 3D-object) represented by an attributed graph.

AGs through a joint probability space of random variables ranging over nodes and 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, in which strong simplifications were made. Sengupta [7] presented also probabilistic methods of synthesising sets of AGs. Moreover, Bunke [8] presented a model of sets of graphs, called network of models, in which all the graphs are pre-processed generating a symbolic data structure.

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 a FORG. To alleviate this weakness, a qualitative information of the joint probabilities of two nodes is incorporated into FDGs. As results show, they improve the representational power of FORGs, decrease the computational cost and with a negligible increase of spatial cost.

In this article, we show the usefulness of the FDG’s 2nd-order relations applied to 3D-object modelling to increase the ratio of recognition as well as to decrease the run time of the recognition process. The formal definition of FDGs is presented elsewhere [2,3]. The synthesis of FDGs from a labeled set of AGs and from the result of clustering a set of unlabelled AGs was studied in [1] and [2], respectively. A distance measure between an AG and an FDG for error-tolerant graph matching was reported in [3].

2. A probabilistic model with 2nd-order constraints
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 the introduction, FORGs provide such a representation but they are difficult to be applied in real problems where there is a large number of vertices in the AGs and their attributes have an extensive domain.

With the aim of offering a more practical approach, 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 one hand, some independence assumptions are considered, but...
on the other hand, some useful 2nd-order functions are included to constrain the generalisation of the structure.

2.1 Independence assumptions in the FDGs

The following independence assumptions are considered:

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. That is, any 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 \( \Phi \), that denotes the non-instantiation of an FDG node or arc in an AG that was used to synthesise the FDG.

2.2 2nd-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 regards to the structural information. These 2nd-order relations, that suppose a little increase of the amount of data to be stored in the prototype, are useful for two reasons. The former, they constrain the set of outcome graphs covered by the prototype and tend to cut down notably the structural over-generalisation. The latter, they reduce the size of the search space of the AG-to-FDG matching algorithm, decreasing the global temporal cost of the recognition [3].

Antagonism, Occurrence and Existence relations in the FDGs represent a qualitative information of the 2nd-order joint probability functions of a pair of vertices. To illustrate the meaning of the FDG 2nd-order relations it is convenient to split the domain of the joint probabilities in four regions (see Figure 2.a).

The first one is composed by the points that belong to the Cartesian product of the domains of actual attributes of the two vertices (which are assumed to be the same domain, \( \Delta_i \)) 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 one-dimensional (shown as straight lines) in which only one of the vertices 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 vertices are null, which includes the cases when none of them appear in the initial AG.

The 2nd-order relations are defined as follows: (We assume that \( \omega_i \) and \( \omega_j \) are two FDG vertices and that \( \alpha_i \) and \( \alpha_j \) are their corresponding random variables).

Antagonism relations: \( \omega_i \) and \( \omega_j \) are antagonistic if the probabilities in the first region are all zero (figure 2.b). In the 3D-object modelling case, two faces are antagonistic if it is not possible to see both in the same view. 

\[
A_{\omega_i}(\alpha_i, \alpha_j) = 1 \iff \Pr(\alpha_i \neq \Phi \land \alpha_j = \Phi) = 0
\]

Occurrence relations: There is an occurrence relation if the joint probability function equals zero in the second region (figure 2.c). The case of the third region is analogous to the second one with the only difference of swapping the elements. In the 3D-object modelling case, a face is “occurent” with respect to another if always that the former is visible, the latter is invisible too. 

\[
O_{\omega_i}(\alpha_i, \alpha_j) = 1 \iff \Pr(\alpha_i = \Phi \land \alpha_j \neq \Phi) = 0
\]

Existence relations: Finally, there is an existence relation between two vertices if the joint probability function equals zero in the fourth region (figure 2.d). In the 3D-object modelling case, there is an existence relation between two faces if one of them or both appear in all the views used to synthesise the model of the object. 

\[
E_{\omega_i}(\alpha_i, \alpha_j) = 1 \iff \Pr(\alpha_i = \Phi \land \alpha_j = \Phi) = 0
\]

Figure 3 shows the 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 2nd-order relations obtained from the corresponding joint probability density function are shown. 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 2nd-order relations cannot appear between two vertices at the same time.
3. Computing the 1st and 2nd-order functions

The probability density function, \( p_{\omega}(\omega) \), for an FDG vertex \( \omega \), can be estimated separately, in the maximum likelihood sense, using frequencies of attribute values. That is, the probability of \( a \) in the FDG vertex \( \omega \) is estimated as the number of AGs, \( g \), such that the attribute value in their corresponding vertex is \( a \), divided by the size of the learning set. The labellings between vertices of the AGs and the FDG, \( h' \), have been previously defined or found.

\[
p_{\omega}(\omega) = \frac{\sum_{g=1}^{G} \text{labelling}(\omega, \omega')} {G}
\]

The probabilities in the FDG arcs are computed similarly [1]. The 2nd-order functions are given by the logical AND of each of the antagonism, occurrence and existence relations, for all the AGs in the learning set. There is an antagonism if any of the AGs used to synthesize the FDG has both vertices (they are not null), which is the same than the probability in the first region is null.

\[
A_{\omega} = \begin{cases} 1 & \text{if } \forall g: 1 \leq g \leq z: h'(\omega') \neq \Phi \land h'(\omega) \neq \Phi \\ 0 & \text{otherwise} \end{cases}
\]

There is an occurrence if in all the AGs used to synthesize the FDG, the first node is non-null and the second node is null, that is, the probability in the second region is null.

\[
\text{occ}_{\omega} = \begin{cases} 1 & \text{if } \forall g: 1 \leq g \leq z: \neg(h'(\omega') \neq \Phi \lor h'(\omega) \neq \Phi) \\ 0 & \text{otherwise} \end{cases}
\]

There is an existence if in all the AGs used to synthesize the FDG, both nodes are nulls, that is, the probability in the fourth region is null.

\[
\text{exist}_{\omega} = \begin{cases} 1 & \text{if } \forall g: 1 \leq g \leq z: \neg(h'(\omega') = \Phi \land h'(\omega) = \Phi) \\ 0 & \text{otherwise} \end{cases}
\]

4. Application to 3D-object modelling

In this section we present some experimental results of an application of FDGs to a 3D-object representation and recognition problem. As already mentioned, an attributed adjacency graph is extracted first for each 2D-view (a color image) of a 3D-object. 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 (average hues) of the two neighbouring regions.

We want to assess the influence of the antagonism and occurrence costs on the correct classification ratio and the time spent to classify the views. When these costs are null, the 2nd-order relations are not considered and the system is similar to FORGs. When a high cost is used, the non-fulfilment of just one antagonism or occurrence constraint causes the labelling to be rejected. Conversely, when a moderate cost is used, the non-fulfilment of a 2nd-order relation causes only a slight increase of the global cost associated with the labelling. The cost associated with the existence relations is not involved because no existence relations were synthesised in the experiments (there is no pair of nodes such that one or the other can be seen in all the views).

Figure 4 shows 4 objects at angle 100 and their segmented images with the adjacency graphs. The test set was composed by 36 views per object (taken at the angles 0, 10, 20 and so on), whereas the learning set was composed by the 36 remaining views (taken at the angles 5, 15, 25 and so on). FDGs were synthesised automatically using the AGs in the learning 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 [2]. 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 learning set that represent this 3D-object were used to synthesise the FDG. If it was represented by 2 FDGs, the 18 first and consecutive AGs from the learning 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.
We compared FDGs to two other classification methods: a classical 3-Nearest Neighbours classifier and a graph matching approach based on first-order random graphs. In the former, the edit-operations distance [5] was used as the distance between elements (AGs). In that case, the costs of insertion and deletions were set to 1 and the cost of substitution was 0 if cyclic_distance\(\text{hue1, hue2}=0\), cost=1/2 if \(\text{cyclic_distance} (\text{hue1, hue2})=1\) and cost=1 if \(\text{cyclic_distance} (\text{hue1, hue2})>1\).

Table 1 shows the ratio of correctness of the FDG classifier varying the number of FDGs per each object. The correctness is higher when 2\textsuperscript{nd}-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 degrees of the 3D-object. When objects are represented by 9 FDGs, each FDG represents 40 degrees of the 3D-object and 4 AGs per FDG, there is poor increase in space due to the storage of 2\textsuperscript{nd}-order relations in FDGs is clearly compensated by both a notable increase in the recognition ratio and a significant decrease in the computation time.

Table 1. Ratio of classification correctness (%).

<table>
<thead>
<tr>
<th>Number of FDGs per object</th>
<th>9</th>
<th>6</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of AGs per FDG</td>
<td>4</td>
<td>6</td>
<td>9</td>
<td>12</td>
<td>18</td>
<td>36</td>
</tr>
<tr>
<td>Cost Antag.</td>
<td>Moderate</td>
<td>Moderate</td>
<td>65</td>
<td>69</td>
<td>79</td>
<td>70</td>
</tr>
<tr>
<td></td>
<td>High</td>
<td>Null</td>
<td>60</td>
<td>51</td>
<td>47</td>
<td>41</td>
</tr>
<tr>
<td></td>
<td>Null</td>
<td>High</td>
<td>59</td>
<td>49</td>
<td>45</td>
<td>37</td>
</tr>
<tr>
<td></td>
<td>High</td>
<td>High</td>
<td>52</td>
<td>41</td>
<td>43</td>
<td>33</td>
</tr>
<tr>
<td></td>
<td>Moderate</td>
<td>Null</td>
<td>62</td>
<td>60</td>
<td>65</td>
<td>61</td>
</tr>
<tr>
<td></td>
<td>Null</td>
<td>Moderate</td>
<td>61</td>
<td>59</td>
<td>63</td>
<td>57</td>
</tr>
<tr>
<td>(FORGs)</td>
<td></td>
<td></td>
<td>40</td>
<td>44</td>
<td>45</td>
<td>53</td>
</tr>
<tr>
<td>3-N.N. (Edit Op. Distance)</td>
<td></td>
<td></td>
<td>63</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 2 displays 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 at 800Mhz. Higher is the cost on the antagonisms, the search tree is more pruned and hence, faster is the computation of the distance. In addition, it can be noted that when the number of FDGs per object is lower than 4, the number of comparisons decreases but the time spent to compute the distance increases since the FDGs are bigger.

<table>
<thead>
<tr>
<th>Number of FDGs per object</th>
<th>9</th>
<th>6</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of AGs per FDG</td>
<td>4</td>
<td>6</td>
<td>9</td>
<td>12</td>
<td>18</td>
<td>36</td>
</tr>
<tr>
<td>FDCG: Moderate costs</td>
<td>125</td>
<td>89</td>
<td>32</td>
<td>44</td>
<td>64</td>
<td>59</td>
</tr>
<tr>
<td>FDCG: High costs</td>
<td>150</td>
<td>101</td>
<td>47</td>
<td>56</td>
<td>72</td>
<td>82</td>
</tr>
<tr>
<td>FORG</td>
<td>187</td>
<td>123</td>
<td>253</td>
<td>392</td>
<td>814</td>
<td>1203</td>
</tr>
<tr>
<td>3-N. N.</td>
<td>93</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 3. Correctness (%) for different representations of the 3D-object: FDGs, FORGs and 3-N.N with several occlusions.

<table>
<thead>
<tr>
<th>Occlusion</th>
<th>0%</th>
<th>20%</th>
<th>40%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of FDGs per object</td>
<td>9</td>
<td>6</td>
<td>4</td>
</tr>
<tr>
<td>Number of AGs per FDG</td>
<td>4</td>
<td>6</td>
<td>9</td>
</tr>
<tr>
<td>FDCGs (Moderate costs)</td>
<td>65</td>
<td>69</td>
<td>79</td>
</tr>
<tr>
<td>FORGs</td>
<td>51</td>
<td>59</td>
<td>62</td>
</tr>
<tr>
<td>3-N.N. (Edit Op. Distance)</td>
<td>63</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

3-N. N. (Edit Op. Distance) | 63 |    |    |    |    |    | 54 |    |    |    |    |    |

References


