LEARNING AND RECOGNISING 3D MODELS REPRESENTED BY MULTIPLE VIEWS BY MEANS OF METHODS BASED ON RANDOM GRAPHS

Alberto Sanfeliu¹ and Francesc Serratosa²

¹ Universitat Politècnica de Catalunya, Inst. de Robòtica i Informàtica Ind., Spain
sanfeliu@iri.upc.es

² Universitat Rovira i Virgili, Dept. d'Enginyeria Informàtica i Matemàtiques, Spain
Francesc.Serratosa@etsi.urv.es

ABSTRACT
The aim of this article is to describe and compare the methods based on random graphs (RGs) which are applied to learn and recognize 3D objects represented by multiple views. These methods are based on modelling the objects by means of probabilistic structures that keep 1ˢᵗ and 2ⁿᵈ order probabilities. That is, multiple views of a 3D object are represented by few RGs. The most important probabilistic structures presented in the literature are First-Order Random Graphs (FORGs), Function-Described Graphs (FDGs) and Second-Order Random Graphs (SORGs).

In the learning process, each one of the 3D-object views are represented by an attribute graph (AG), and a group of AGs are synthesized in a RG. In the recognizing process, the view of the object is represented by an AG and then it is compared with the RG that model each one of the 3D-object prototypes. In this paper, it is explained the modelling of the 3D-objects and the methods of learning and recognition based on FORGs, FDGs and SORGs. We show some results of the methods for real 3D objects.

1. INTRODUCTION
Attributed Graphs (AGs) has been used to solve computer vision problems for decades and in many applications. Some examples include recognition of graphical symbols, character recognition, shape analysis, 3D-object recognition [3, 4, 8] and video and image database indexing. In these applications, AGs represent both unclassified objects (unknown input patterns) and prototypes. Moreover, these AGs are typically used in the context of nearest-neighbor classification. That is, an unknown input pattern is compared with a number of prototypes stored in the database. The unknown input is then assigned to the same class as the most similar prototype.

Nevertheless, the main drawback of representing the data and prototypes by AGs is the computational complexity of comparing two AGs. The time required by any of the optimal algorithms may in the worst case become exponential in the size of the AGs. The approximate algorithms, on the other hand, have only polynomial time complexity, but do not guarantee to find the optimal solution. For some applications, this may not be acceptable. Moreover, in some applications, the classes of objects are represented explicitly by a set of prototypes which means that a huge amount of model AGs must be matched with the input AG and so the conventional error-tolerant graph matching algorithms must be applied to each model-input pair sequentially. As a consequence, the total computational cost is linearly dependent on the size of the database of model graphs and exponential (or polynomial in subgraph methods) with the size of the AGs. For applications dealing with large databases, this may be prohibitive.

To alleviate these problems, some attempts have been made to try to reduce the computational time of matching the unknown input patterns to the whole set of models from the database. Assuming that the AGs that represent a cluster or class are not completely dissimilar in the database, only one structural model is defined from the AGs that represent the cluster, and thus, only one comparison is needed for each cluster (figure 1).

![Figure 1. Learning and classification processes in the classifiers that use only one structural representation per cluster.](image-url)

We distinguish two different methodologies depending on whether they keep probabilistic information in the structure that represent the cluster of AGs or not. In this
paper, we study and compare the probabilistic ones. For
other methods see [4].
In the probabilistic methods, the models, which are
usually called Random Graphs (RG), are described in the
most general case through a joint probability space of
random variables ranging over graph vertices and arcs.
They are the union of the AGs in the cluster, according to
some synthesis process, together with its associated
probability distribution. In this manner, a structural
pattern can be explicitly represented in the form of an AG
and an ensemble of such representations can be
considered as a set of outcomes of the RG. In this paper,
we briefly recall the three most important probabilistic
methods presented elsewhere and their use in the 3D-
object recognition problem, which are First-Order
Random Graphs (FORGs) [5], Function-Described
Graphs (FDGs) [3,4], and finally, Second-Order Random
Graphs (SORGs) [2]. The approach presented in the paper
by Sengupta et al. [7] can be regarded as similar to the
FORG approach.
In the remaining of the paper, we first explain the method
used to extract the 3D information. Then, we introduce the
RG methods and we concretise in the three methods
commented before: FORGs, FDGs and SORGs. Finally,
we present a practical comparison between the three
methods and some conclusions.

2. REPRESENTATION OF 3D-OBJECTS
The 3D-object learning proceeds is done as follows: A
mobile robot (figure 2) takes a sequence of images from
an object in a clockwise or counter-clockwise direction
maintaining a close distance. Each image is segmented
and an AG is extracted. Then the structure that represents
the 3D-object is updated using an incremental synthesis
algorithm [4].

![Incremental synthesis of the structure that represents a 3D-object](image)

**Figure 2. Incremental synthesis of the structure that represents a 3D-object.**

3. RANDOM-GRAPH REPRESENTATIONS
If we want to represent the cluster of AGs by a probability
distribution it is impractical to consider the high order
probability distribution where all vertices and arcs in
the AGs are taken jointly. For this reason, FORGs, FDGs and
SORGs are more practical approaches that propose
different approximations. All of them take into account in
some manner the incidence relations between attributed
vertices and arcs, i.e. assume some sort of dependence of
an arc on its connecting vertices. Also, a common
ordering (or labelling) scheme is needed that relates
vertices and arcs of all the involved AGs, which is
obtained through an optimal graph mapping process
called synthesis of the RG representation.
Moreover, in the classification process, a distance
measure has to be defined that provides a quantitative
value of the match between an AG G (data graph) and a
RG S (model graph). It is usually related to the probability
of G according to a labelling function \( \mu : G \rightarrow S \),
denoted \( p(G|\mu) \). It may attempt to minimise a global cost
measure \( C \) of the morphism \( \mu \) in the set \( H \) of allowable
configurations, by taking the cost as a monotonic
decreasing function of the conditional probability of the
data graph given the labelling function, \( C = \text{cost}(P(G|\mu)) \).
In the following sections, we summarise the three main
such approaches, FORGs, FDGs and SORGs and we show
that the main practical differences are the storage space
and the definition of the probability \( P(G|\mu) \).
We define that the domain of the random vertices \((\alpha_i)\)
and arcs \((\beta_j)\) are \( \alpha_i \in \Delta_i \) and \( \beta_j \in \Delta_j \), respectively, and also that
the number of elements of these domains are \( N \) and \( M \).
Finally, the number of arcs and vertices is \( n \) and \( m \).

4. FIRST-ORDER RANDOM GRAPHS
Wong and You [5] proposed the First-Order Random
Graphs (FORGs), in which strong simplifications are
made so that RGs can be used in practice. They
introduced three suppositions about the probabilistic
independence between vertices and arcs:
1) The random vertices are mutually independent;
2) The random arcs are independent given values for the
random vertices;
3) The arcs are independent of the vertices except for the
vertices that they connect.
Based on these assumptions, for a FORG \( R \), the
probability \( P(G|\mu) \) becomes

\[
P(G|\mu) = \prod_{i=1}^{n} p_i(\alpha_i) \prod_{j=1}^{m} q_j(\beta_j | \alpha_{j1}, \alpha_{j2})
\]

(1)

where \( p_i(\alpha) = \text{Pr}(\alpha = \alpha) \) are the marginal probability
density functions for vertices and \( q_j(\beta | \alpha_{j1}, \alpha_{j2}) \) are the conditional probability
functions for the arcs, where \( \alpha_{j1}, \alpha_{j2} \) refer to the random
vertices for the endpoints of the random arc \( \beta_j \).
The storage space of FORGs is \( O(n^2 + mn^2) \).
5. FUNCTION-DESCRIBED GRAPHS

Serratosa et al. [3,4] proposed the Function-Described Graphs (FDGs), which lead to another approximation of the joint probability \( P(G|\mu) \) of the random elements. On one hand, some independence assumptions (a) are considered, but on the other hand, some useful 2nd-order functions (b) are included to constrain the generalisation of the structure.

(a) Independence assumptions in the FDGs

1) The attributes in the vertices are independent of the other vertices and 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.

(b) 2nd-order functions in the FDGs

In order to tackle the problem of the over-generalisation of the sample, the antagonism, occurrence and existence relations are introduced in FDGs, which apply to pairs of vertices or arcs. In this way, random vertices and arcs are not assumed to be mutually independent, at least with regards to the structural information, since the above relations represent a qualitative information of the 2nd-order joint probability functions of a pair of vertices or arcs.

Based on these assumptions, for an FDG \( F, P(G|\mu) \) is,

\[
P(G|\mu) = \prod_{i=1}^{n} p_i(a_i) \prod_{j=1}^{m} q_j(b_j) \tag{2}
\]

where \( p_i(a_i) \) is defined as in FORGs and \( q_j(b_j) \) is \( \Pr(b_j = b_i | \alpha_j, \alpha_j' \neq \Phi, \alpha_j'' \neq \Phi) \). The null value is \( \Phi \).

However, the isomorphism \( \mu \) only not has to be structurally coherent but also has to fulfil the 2nd-order constraints (antagonism, existence and occurrence). Otherwise, \( P(G|\mu) \) is considered to be zero. The basic idea of these constraints is the satisfaction by an AG to be matched of the antagonism, occurrence and existence relations inferred from the set of AGs used to synthesise the FDG. The storage space of FDGs is \( \Omega_{nN+mM+n'M'} \).

6. SECOND-ORDER RANDOM GRAPHS

Finally, Serratosa et al. [2] proposed the Second-Order Random Graphs (SORGs), which lead to the last approximation of the joint probability \( P(G|\mu) \) of the random elements. The basic idea is to keep more structural and semantic information than FORGs and FDGs. In the SORGs, the second-order probabilities between vertices and arcs is obtained in the synthesis process and maintained in the structure. In the classification process, the first and second-order probabilities are considered in the probability \( P(G|\mu) \) as,

\[
P(G|\mu) = p(d_1, \ldots, d_n) \prod_{i=1}^{n} p_i(d_i) \prod_{j=1}^{m} q_j(d_j) \tag{3}
\]

where \( p_i(d_i) \) are the marginal probabilities of the random vertices or arcs, \( d_i \in \Delta_N \cup \Delta_M \), and \( r_{ij} \) are the Peleg compatibility coefficients [6] that take into account both the marginal and 2nd-order joint probabilities,

\[
r_{ij}(d_i, d_j) = \frac{\Pr(r_j = d_j \land r_i = d_i)}{p_i(d_i)p_j(d_j)} \tag{4}
\]

The Peleg coefficient, with a non-negative range, is related to the "degree" of dependence between two random variables. If they are independent, the joint probability is defined as the product of the marginal ones, thus, \( r_{ij} = 1 \) (or a value close to 1 if the probability functions are estimated). If one of the marginal probabilities is null, the joint probability is also null. In this case, the indecisiveness 0/0 is solved as 1, since this do not affect the global joint probability, which is null. Eq. (3) is obtained by assuming independence in the conditional probabilities and rearranging the joint probability expression with the Bayes rule [2].

The storage space of SORGs is \( \Omega_{nN+mM+n'M'} \).

7. RESULTS

We present a real application to recognize coloured objects using 2D images. Images were extracted from the database COIL-100 from Columbia University (www.cs.columbia.edu/CAVE/research/softlib/coil-100.html). It is composed by 100 isolated objects and for each object there are 72 views (one view each 5 degrees). Figure 3 shows some objects at angle 100 and their segmented images with the adjacency graphs. These graphs have from 6 to 18 vertices and the average number is 10. The test set was composed by 36 views per object (taken for the angles 0, 10, 20 and so on), whereas the reference set was composed by the 36 remaining views (taken at the angles 5, 15, 25 and so on). We compared the probabilistic models FORGs, FDGs and SORGs, and also, the Nearest-Neighbour classifier (AG-AG) with an edit-operation distance measure between AGs.

![Figure 3. Some objects at angle 100 and the segmented images with the AGs.](image-url)

We made 6 different experiments in which the number of clusters that represents each 3D-object varied. If the 3D-object was represented by only one cluster, the 36
AGs from the reference set that represent the 3D-object were used to synthesise the SORGs, FORGs or FDGs. If it was represented by 2 clusters, the 18 first and consecutive AGs from the reference set were used to synthesise one of the SORGs, FORGs or FDGs and the other 18 AGs were used to synthesise the other ones. A similar method was used for the other experiments with 3, 4, 6 and 9 clusters per 3D-object.

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

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

When the best classification is reached, FDGs have less run-time than SORGs but lower recognition ratio. This is due to the fact that the algorithm to compute the distance in the FDG classifier prunes harder the search tree than the SORGs since it uses a qualitative information of the 2nd-order relation. So, the time spent to search the best labelling decreases but the optimal one may not be found.

8. REFERENCES