

Function-Described Graphs Applied to 3D Object Representation

Francesc Serratosa
Departament d'Enginyeria Informàtica
Universitat Rovira i Virgili
email fserrato@etse.urv.es

Alberto Sanfeliu
Institut de Robòtica i Informàtica Industrial
Universitat Politècnica de Catalunya
email asanfeliu@iri.upc.es

Abstract

The aim of this work is the characterization of a new structure called Function-Described Graphs (FDG) which can be used to represent objects in computer vision. The FDGs are useful in synthesizing structural information from a set of objects described through their structure. The FDG nodes and arcs are characterized by the probability distribution of the attributes of the ARGs nodes and arcs from where they have been synthesized. The FDG incorporates information of the family of the synthesized ARG and of the antagonistic node and arcs. In this work we apply this new structure to 3D object labeling.

1 Introduction

High level computer vision is used to analyze a scene and carry out subsequent reasoning tasks, such as identifying the objects in a scene. Attributed Relational Graphs (ARGs) [7] have been used in scene analysis to represent complex objects. Examples of ARGs and their use can be seen in the identification of English letters [2], hand written symbols [3], aerial road images [4,9] or 3-D objects [5,6]. When the number of ARGs is high, the number of matching becomes an important issue from the complexity point of view. Random Graphs [2] have been suggested as a solution for this problem, however they can not represent the complete structural information due to their representation weakness.

In this paper, we present a new representation graph scheme called a Function-Described Graph (FDG) which extends the capabilities of Random Graphs by means of adding *control functions* into the nodes and arcs of a classical ARG. Moreover we apply a labeling method to identify nodes of the FDG with nodes of an ARG. The same methodology used for the labeling process can be used for matching and synthesis. In order to reduce the time complexity we use a gradient ascendant labeling technique to find local optimal solutions, different from the probabilistic relaxation [1],[8],[9].

2 The New Structure

The new Function-Described Graphs structure arose from the idea of representing objects for computer vision. The FDG is a general structure, for this reason is applicable in any type of pattern. For instance, voice recognition, 3D objects, hand written characters, etc. An FDG is composed by two main parts. The first one is a relational structure that has the local and structural information of the object. The second part is composed by the *control functions* which control and generate the

structure information. These functions are applied during the labeling , synthesis and matching processes.

2.1 FDG Relational Structure

The relational structure is composed by a set of nodes and a set of arc relations between the nodes. The attributes of an FDG node or arc consists in local properties that come from the node or arc attributes of the synthesized ARG nodes and arcs. Each FDG node attribute or FDG arc attribute value is a probability distribution. We have characterized these probability distribution by means of the average, the standard deviation and the number of elements.

2.2 FDG Control Functions

Control Functions are used in the labeling, synthesis and matching processes of the FDG. We have defined six Control Functions which are the following ones :

Node Compatibility Function (R_n) and Arc Compatibility Function (R_e): R_n and R_e are the compatibility functions that are used in the diverse processes (labeling, ...) to support or no support mappings between node (or arc) of an ARG and an FDG. Take into account that always we compare an ARG attribute with the probability distribution of an FDG attribute. See section “mathematical representation” for additional details. These functions are applied to the attribute values of the ARG nodes and arcs. See section 5 for an application example.

Node Population Function (F_n) and Arc Population Function (F_e): The Population Functions are used to incorporate a new element in an FDG probability distribution or create new ones. This element is the attribute of an ARG node or arc. Before this process, the compatibility between the elements have been tested with the compatibility functions. Take into account that a probability distribution can be empty.

Node Antagonistic Function (A_n) and Arc Antagonistic Function (A_e): These functions are used to describe the antagonism between nodes and arcs in an FDG. For example, if two vertices of an object can not be seen in a single perspective view, they are antagonistic. If the vertices are the nodes of an FDG, then both nodes will be antagonistic.

2.3 Mathematical Representation

As it has been said, two types of relational structures are used in this system. The first one is the Attributed Relational Graphs (ARG) that have the information of the features of the input object. The mathematical representation of an Attributed Relational Graphs is as follows:

An ARG is $(\Sigma_v, \Sigma_e, \Delta_v, \Delta_e, \gamma_v, \gamma_e)$ where $\Sigma_v = \{v^i\}_{i=1..n}$ is a set of nodes. Assume there are n nodes. $\Sigma_e = \{e^{ij}\}_{i,j=1..n}$ is a set of arcs. $\Delta_v = \{a^i\}_{i=1..n}$ and $\Delta_e = \{b^{ij}\}_{i,j=1..n}$ are the set of attributes of the nodes and the arcs. The attributes of the nodes and of the arcs are $a^i = (a^i_1, a^i_2, \dots, a^i_t)$ and $b^i = (b^i_1, b^i_2, \dots, b^i_s)$. Assume t attributes in the nodes and s

attributes in the arcs. The domains are $a_k = \{D_{vk}\}_{k=1..t}$ and $b_k = \{D_{ek}\}_{k=1..s}$. Finally, $\gamma_v: \Sigma_v \rightarrow \Delta_v$ and $\gamma_e: \Sigma_e \rightarrow \Delta_e$ are two applications that assign attributes to the nodes and to the arcs. $\forall v^i: \gamma_v(v^i) = (a^i_1, a^i_2, \dots, a^i_t)$ and $\forall e^{ij}: \gamma_e(e^{ij}) = (b^{ij}_1, b^{ij}_2, \dots, b^{ij}_s)$

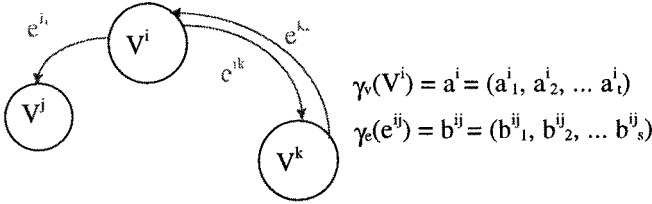


Figure 1. An ARG.

The second relational structure is the Function-Described Graphs (FDG), that contains the synthesis of a set of prototype ARGs. A FDG is composed by $(\Sigma_\omega, \Sigma_e, \Delta_\omega, \Delta_e, \gamma_\omega, \gamma_e, F_\omega, F_e, A_\omega, A_e, R_\omega, R_e)$ where $\Sigma_\omega = \{\omega^\lambda\}_{\lambda=1..m}$ is a set of nodes. Assume there are m nodes. $\Sigma_e = \{\varepsilon^{\lambda\lambda'}\}_{\lambda, \lambda'=1..m}$ is a set of arcs. $\Delta_\omega = \{\alpha^\lambda\}_{\lambda=1..m}$ and $\Delta_e = \{\beta^{\lambda\lambda'}\}_{\lambda, \lambda'=1..m}$ are the set of attributes of the nodes and the arcs. Attributes in the nodes and in the arcs are composed by $\alpha = (\alpha_1, \alpha_2, \dots, \alpha_\tau)$ and $\beta = (\beta_1, \beta_2, \dots, \beta_s)$. Assume τ attributes in the nodes and σ attributes in the arcs. The domains are $\alpha_k = \{D_{\omega k}\}_{k=1..t}$ and $\beta_k = \{D_{ek}\}_{k=1..s}$. Finally, $\gamma_\omega: \Sigma_\omega \rightarrow \Delta_\omega$ and $\gamma_e: \Sigma_e \rightarrow \Delta_e$ are two applications that assign attributes to the nodes and to the arcs. $\forall \omega^\lambda: \gamma_\omega(\omega^\lambda) = (\alpha^\lambda_1, \alpha^\lambda_2, \dots, \alpha^\lambda_\tau)$ and $\forall \varepsilon^{\lambda\lambda'}: \gamma_e(\varepsilon^{\lambda\lambda'}) = (\beta^{\lambda\lambda'}_1, \beta^{\lambda\lambda'}_2, \dots, \beta^{\lambda\lambda'}_\sigma)$

The second part is composed by six sets of structure control functions. ($F_\omega, F_e, A_\omega, A_e, R_\omega, R_e$)

Node Antagonistic Function

$A_\omega(\omega^\lambda, \omega^{\lambda'}) = 1$ (Antagonistic) 0 (Non antagonistic)

Arc Antagonistic Function

$A_e(\varepsilon^{\lambda\lambda'}, \varepsilon^{\lambda'\lambda''}) = 1$ (Antagonistic) 0 (Non antagonistic)

Node Population Function:

$F_\omega = \{F_{\omega k}\}_{k=1..t}$;

$F_{\omega k}(\{a_k^{i(p)}\}_{p=1..|\text{ARGI}|}, \{x_{\omega k}, \Gamma_{\omega k}, n_{\omega k}\}^\lambda) = \{x_{\omega k}, \Gamma_{\omega k}, n_{\omega k}\}^\lambda_{k=1..t, \lambda=1..m; i=1..n}$

$F_{\omega k}: D_{vk}^{(1)} \times D_{vk}^{(2)} \times \dots \times D_{vk}^{(p)} \times D_{\omega k} \rightarrow D_{\omega k, p=1..|\text{ARGI}|, k=1..t}$

Arc Population Function:

$F_e = \{F_{ek}\}_{k=1..s}$;

$F_{ek}(\{b_k^{ij(p)}\}_{p=1..|\text{ARGI}|}, \{x_{ek}, \Gamma_{ek}, n_{ek}\}^\lambda) = \{x_{ek}, \Gamma_{ek}, n_{ek}\}^\lambda_{k=1..s; \lambda, \lambda'=1..m; i, j=1..n}$

$F_{ek}: D_{ek}^{(1)} \times D_{ek}^{(2)} \times \dots \times D_{ek}^{(p)} \times D_{ek} \rightarrow D_{ek, p=1..|\text{ARGI}|, k=1..s}$

Node Compatibility Function:

$R_\omega = \{R_{\omega k}\}_{k=1..t}; R_{\omega k}(v^i, \omega^\lambda) \in \{0, 1\}_{k=1..t, \lambda=1..m, i=1..n}$

$R_{\omega k}: \Sigma_v \times \Sigma_\omega \rightarrow \{0, 1\}_{k=1..t}$

Arc Compatibility Function:

$R_e = \{R_{ek}\}_{k=1..s}; R_{ek}(e^{ij}, \varepsilon^{\lambda\lambda'}) \in \{0, 1\}_{k=1..s; \lambda, \lambda'=1..m; i, j=1..n}$

$R_{ek}: \Sigma_e \times \Sigma_e \rightarrow \{0, 1\}_{k=1..s}$

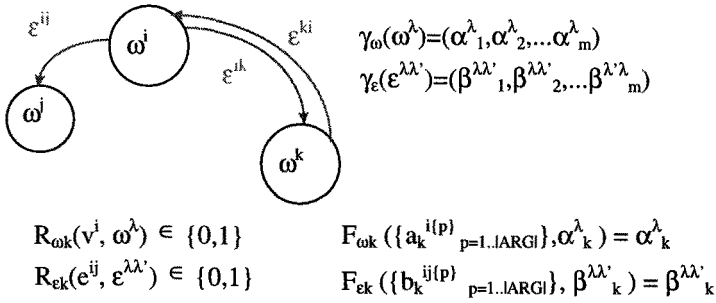


Figure 2. An FDG.

3 Application to 3D Object Recognition through perspective views

We have tested our relational structure with 3D objects with plane faces. All the object vertices have three faces. The aim of the application is to recognize them. As the synthesis process has not been studied yet, a data base with some FDGs has been created manually [Figure 3]. We have scenes with some objects and we assume parallel perspective. We have extracted their local features and studied the labeling process behavior. We have also studied the system behavior with different Control Functions and their influence on the representation power of the FDG. Integration of this application in the new structure is explained in the following sections: 1) Extracted features from the objects, 2) Integration of these features in the Attributed Relational Graph, 3) Description of the Function-Described Graph and finally 4) Results.

3.1 Extracted features from the objects

Three features of the object have been extracted of the scenes. As the objects can be partially occluded and it is not possible to have the whole object view on the scene, local features are used. The first feature is the edge type. There are three kinds of edges in a scene: convex, concave and occluded[10]. An occluded edge appears when a face is not visible from the camera point of view. The second feature is the vertex type. There are twelve types of vertices[10] depending on the type vertex junction. And finally, the third feature is the orthographic invariant.

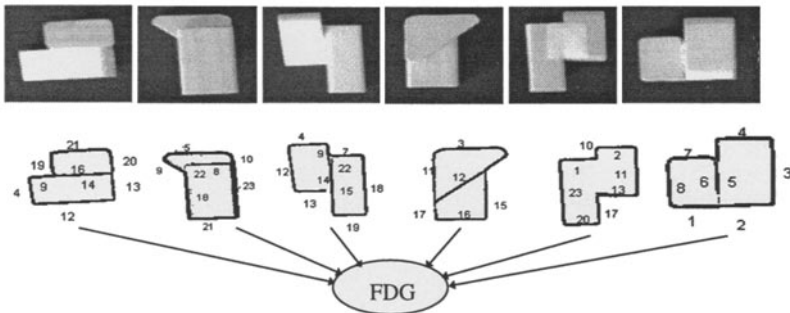


Figure 3. The six pattern views used to create an FDG.

3.2 Attributed Relational Graph

The three aforementioned features have been structured on an Attributed Relational Graph [Figure 4]. The nodes of the Graph represent straight edges whereas the arcs represent relations between them. The nodes have one attribute and the arcs have two attributes. These attributes are:

The attribute of the node v^i is a_1^i = Edge type. If the feature of the straight segment is concave or convex, then the attribute of the node is the same, whereas if the feature is occluded the node attribute will be convex.

The first attribute of the arcs is b_1^{ij} = orthographic invariant. Orthographic invariant is applicable when the straight segments are parallels. This attribute will be “not parallels (Ω)” if the straight segments are not parallels.

The second attribute of the arcs is b_2^{ij} = Type of vertex. Arcs that represent vertices well extracted, their property will be the vertex type. Whereas vertices partially extracted (they have at most two edges), their property will be “not classified (θ)”. Also when there is not a vertex, their property will be “not a vertex (Ω)”.

Domain of first node attribute: $D_{v1} = \{ \text{Concave, Convex} \}$

Domain of first arc attribute: $D_{e1} = R \cup \{ \Omega \}$

Domain of second arc attribute: $D_{e2} = \{ \{ \text{Vertex type} \} \cup \{ \Omega, \theta \} \}$

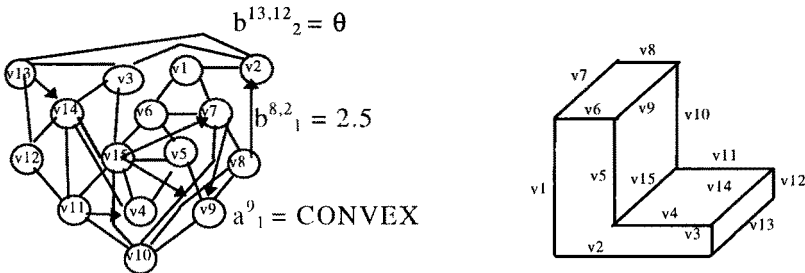


Figure 4. An Attributed Relational Graph and the represented object.

3.3 Function-Described Graph

Each Function-Described Graph has the synthesized information of an object, therefore the information of the nodes or of the arcs between them have to include properties of the object, if possible invariant to the point of view. Since an object can be partially occluded, these properties will usually be local properties. No global features are considered. The relational structure and the control functions of the FDG are as follows. Figure 5 shows 3 objects and their synthesized FDG.

3.3.1 Relational Structure

The nodes of the structure are represented by the probability distributions of the attributes of the ARG nodes and arcs. The domain of the first arc attribute is $D_{e1} = \{ x_{e1}, \Gamma_{e1}, n_{e1} \}$ and the second domain is $D_{e2} = \{ x_{e2}, \Gamma_{e2}, n_{e2} \}$. The domain of the node attribute is $D_{\omega1} = \{ x_{\omega1}, \Gamma_{\omega1}, n_{\omega1} \}$

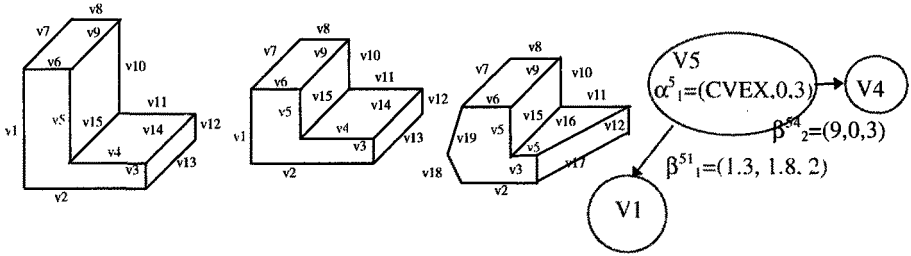


Figure 5. Three objects that belong to a class. Three nodes of the Function-Described Graph that represents their class.

3.3.2 Population Functions

As we have mentioned earlier, we have not studied the synthesis process yet therefore the probability distributions domains are known but their Population Functions have not been studied.

3.3.3 Compatibility Functions

Depending on the probability distributions and the attributes of the ARG nodes and ARG arcs, there are three Compatibility Functions. The Edge Type Compatibility Function (R_{ω_1}) in the nodes, the Orthographic Invariant Compatibility Function (R_{ϵ_1}) on the arcs and the Vertex Type Compatibility Function (R_{ϵ_2}) on the arcs:

$$R_{\omega_1}(v^i, \omega^\lambda) = 1 \text{ if } a^i_1 = x^\lambda_{\omega_1} \text{ and } n^\lambda_{\omega_1} \neq 0$$

$$= 0 \text{ Otherwise}$$

$$R_{\epsilon_1}(e^{ij}, \epsilon^{\lambda\lambda'}) = 1 \text{ if Mahalanobis-Dist.}(b^{ij}_1, x^{\lambda\lambda'}_{\omega_1}, \Gamma^{\lambda\lambda'}_{\omega_1}) < \text{Threshold}$$

$$\text{ \& } n^{\lambda\lambda'}_{\omega_1} \neq 0$$

$$= 0 \text{ Otherwise}$$

$$R_{\epsilon_2}(e^{ij}, \epsilon^{\lambda\lambda'}) = 1 \text{ if } b^{ij}_2 = x^{\lambda\lambda'}_{\omega_2} \text{ and } n^{\lambda\lambda'}_{\omega_2} \neq 0$$

$$= 0 \text{ Otherwise}$$

3.4 Results

The process consists on combining different Compatibility Functions. We have used a Goodness Function $A(R_i)$ that gives information about the labeling process. $A(R_i) \in [0,1]$. If $A(R_i)=1$ all the nodes have been correctly labeled and if $A(R_i)=0$ there is no labeled node. We have used an iterative process based on a gradient descendent method to update the probabilities in the labeling process. A class represented by an FDG has been created taking as samples six perpendiculars views of the object of figure 3. The application of Consistent Labeling to the 8 views of figure 6 are shown in table 1 where the columns represents: Column 1: $A(R_i)$ when the Compatibility Function Type of edge is applied. Column 2: $A(R_i)$ when the Compatibility Function Orthographic Invariant is applied. Column 3: $A(R_i)$ when the Compatibility Function Vertex Type is applied. Column 4: The $A(R_i)$ when the three Compatibility Functions are applied. Column 5: The number of edges totally labeled and column 6: The number of nodes of each graph.

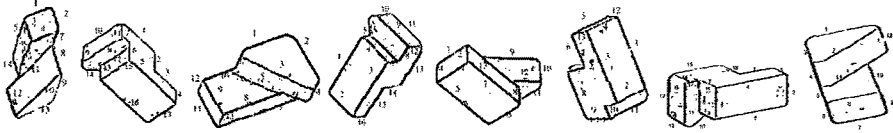


Figure 6. Eight points of view ARG1..ARG8 used to test the FDG.

A summary of the results is the following. When only one Compatibility Function is used in the labeling process, the results are poor. Whereas, when some Compatibility Functions are combined, the results are quite good. However, there are cases that the results are low (ARG 3, ARG5, ARG8) since the individual Compatibility Functions have not been capable to obtain good results. We have not studied the Validation process yet that will improve these results.

In the ARG5 case, although there is not any labeled node, the goodness function is not 0. This is due to the fact that any ARG node is correctly labeled but some labeling hypothesis have been discarded .

	A(h,R ₀₁)	A(h,R ₁)	A(h,R ₂)	A(h,R ₀₁ ,R ₁ ,R ₂)	L(h,R ₀₁ ,R ₁ ,R ₂)	N(R ₀₁ ,R ₁ ,R ₂)
ARG1	0.01	0.22	0.09	0.75	3	17
ARG2	0.07	0.27	0.52	1.00	17	17
ARG3	0.01	0.07	0.09	0.50	2	13
ARG4	0.08	0.16	0.54	0.89	12	16
ARG5	0.01	0.04	0.06	0.34	0	13
ARG6	0.01	0.37	0.07	1.00	14	14
ARG7	0.07	0.35	0.51	0.98	16	17
ARG8	0.01	0.05	0.09	0.41	1	12

Table 1. Results from eight points of view.

We have considered, also, a view partially occluded [Figure 7]. This means that there are some edges (therefore nodes of the graph) that have disappeared and some edges that their attributes have changed (therefore the node still remains in the graphs but the attribute values have changed). We have studied the results using the three Compatibility Functions. We have to read the results [Table 2], having in mind that the type of vertex and the edge type is not possible to be changed by the occlusion but the length of the edge can be modified.

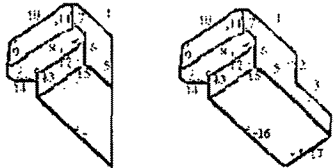


Figure 7. On the left, view partially occluded and on the right the same view without occluding. These views have been structured in the graphs ARG2.1 and ARG2. Five edges have been totally occluded and three edges have been partially occluded

	A(h,R ₀₁)	A(h,R ₁)	A(h,R ₂)	A(h,R ₀₁ ,R ₁ ,R ₂)	L(h,R ₀₁ ,R ₁ ,R ₂)	N(R ₀₁ ,R ₁ ,R ₂)
ARG2	0.07	0.27	0.52	1.00	17	17
ARG2.1	0.09	0.19	0.52	0.68	5	14

Table 2. Results from the partially occluded view.

4 Conclusions

We have defined a new representation graph scheme called Function-Described Graph which allows to integrate diverse ARG into one unified structure. This new structure incorporates all the information of the ARG including the antagonistic information. We have used the FDG to synthesize projective views of 3D polyhedral objects. Scene objects represented by ARGs have been labeled with objects represented by FDGs assuming parallel perspective. Examples of them can be seen in table 1 and 2.

References

- [1] J.T.L.WANG, K.ZHANG, G-W. CHIRN, "The approximate Graph Matching Problem", IEEE pp. 284-288 , 1994.
- [2] A.K.C.WONG, J.CONSTANT and M.L.YOU, "Random graphs", (H. BUNKE and A. SANFELIU eds.). "Syntactic and structural pattern recognition: Theory and applications", World Scientific Publishing Co. Pte. Ltd pp 179-195, 1990.
- [3] A.SANFELIU and K.FU," A Distance Measure Between Attributed Relational Graphs for Pattern Recognition", IEEE Trans. on Sys. man and cybern. Vol. smc 13 No 3 May/June, 1983.
- [4] R.C.WILSON, A.N.EVANS and E.R.HANCOCK, "Relational matching by discrete relaxation", Image and vision computing, Vol. 13, pp 411-421, 1995.
- [5] W.Y.KIM and A.C.KAK, "3D object recognition using bipartite matching embedded in discrete relaxation", IEEE Trans. Pattern anal. mach. intell. 13 pp 224-251, 1991.
- [6] L.SHAPIRO and R.M.HARALICK, "Matching relational structures using discrete relaxation", (H. BUNKE and A. SANFELIU eds.). "Syntactic and structural pattern recognition: Theory and applications", World Scientific Publishing Co. Pte. Ltd pp 179-195, 1990.
- [7] A. SANFELIU, "Matching Methods", 8th Scandinavian conference on image analysis, 1993.
- [8] E.R.HANCOCK and J.KITTLER, "Discrete relaxation", Pattern recognition Vol. 23 pp.711-733, 1990.
- [9] W.J.CHRISTMAS, J.KITTLER and M.PETROU, "Structural matching in computer vision using probabilistic relaxation", PAMI, Vol. 17, No 8 pp 749-764, 1995.
- [10] D.H.BALLARD and C.M.BROWN, "Computer Vision", Prentice Hall 1982.