

Graph-based representations and techniques for image processing and image analysis[†]

A. Sanfeliu^a, R. Alquézar^b, J. Andrade^a, J. Climent^c, F. Serratosa^d and J. Vergés^a

^a Institut de Robòtica i Informàtica Industrial, Universitat Politècnica de Catalunya-CSIC, Barcelona, Spain

^b Dept. de Llenguatges i Sistemes Informàtics, Universitat Politècnica de Catalunya, Barcelona, Spain

^c Dept. d'Enginyeria de Sistemes, Automàtica i Informàtica Industrial, Universitat Politècnica de Catalunya, Barcelona, Spain

^d Dept. d'Enginyeria Informàtica, Universitat Rovira i Virgili, Tarragona, Spain

Abstract

In this paper we will discuss the use of some graph-based representations and techniques for image processing and analysis. Instead of making an extensive review of the graph techniques in this field, we will explain how we are using these techniques in an active vision system for an autonomous mobile robot developed in the *Institut de Robòtica i Informàtica Industrial* within the project “Active Vision System with Automatic Learning Capacity for Industrial Applications (CICYT TAP98-0473)”. Specifically we will discuss the use of graph-based representations and techniques for image segmentation, image perceptual grouping and object recognition. We first present a generalisation of a graph partitioning greedy algorithm for colour image segmentation. Next we describe a novel fusion of colour-based segmentation and depth from stereo that yields a graph representing every object in the scene. Finally we describe a new representation of a set of attributed graphs (AGs), denominated Function Described Graphs (FDGs), a distance measure for matching AGs with FDGs and some applications for robot vision.

Keywords: Structural pattern recognition; Graph-based representations; Object recognition; Colour image segmentation; Perceptual grouping; Data fusion; Depth from stereo; Attributed graphs; Function-described graphs; Distance measure between graphs

1. Introduction

Techniques that use graphs for the representation and manipulation of data [1, 2] have been widely investigated within the fields of image processing and image analysis. The use of graphs allows for a structural representation of object models extracted from images. However some difficulties in the applicability of graph techniques have reduced their use in practical applications. The basic problem is the time complexity of the algorithms. The search for a specific node or edge in a graph, or for the relationship between two nodes (i.e., shortest path) requires polynomial time. However, the search for a match of a scene model to an object model in a graph representation, known as the isomorphism problem, is untractable. This is, its computation has an exponential time complexity. Nevertheless, new developments in approximation algorithms for graph matching have allowed the reduction of this time complexity for sub-optimal solutions (see for example Refs. [3, 4]).

In this paper we will describe several applications of graph-based representation and techniques for image modelling, processing and analysis within the project “Active Vision System with Automatic Learning Capacity for Industrial Applications (CICYT TAP98-0473)”. Specifically, we will explain their use in image segmentation, image perceptual grouping, and object recognition.

A method to segment a colour image based on a graph partitioning greedy algorithm will be firstly described [5]. This algorithm, although uses a graph-based representation, has a polynomial time complexity. We will also explain how to combine this method for perceptual grouping using a depth map obtained from stereo disparity [6].

We will also describe a matching method for recognition based on a new graph representation scheme denominated function-described-graphs (FDGs) [7-9], which is a type of compact representation of a set of attributed graphs (AGs) that borrow from random graphs [10,11] the capability of probabilistic

[†] This work has been partially granted by the Ministerio de Educación y Cultura TAP1998-0473. Corresponding author: A. Sanfeliu, Tel. +34-93-401-5782, Fax: +34-93-401-5750, E-mail address: sanfeliu@iri.upc.es

modelling of structural and attributed information. This technique uses a synthesis graph method to acquire the prototype class from a set of attributed graphs (ARGs) [8,12]. We will also show two examples of application to face recognition and 3D object recognition, respectively.

2. Graph partitioning greedy algorithm for colour image segmentation

Segmentation is the most essential and important step of any low-level vision system. Segmentation is the process of partitioning an image into a set of non-intersecting regions such that each region is homogeneous and the transition from one region to another is sharp. Hundreds of segmentation techniques are present in the literature, but there is still no single method that is robust under any illumination conditions and for any application.

We have selected a graph-theoretical approach to cope with image segmentation because it has a good mathematical basis and because the image segmentation problem can easily be translated into graph-related problems by analogy. Moreover, in the graph theoretical approach, image region extraction and edges extraction are dual problems. An immediate consequence of this is that the region boundaries extracted are always closed. The worst disadvantage of these approaches, as can be seen in Refs. [13-15], is that these algorithms are very time consuming, which prohibits their implementation for real-time applications.

For this reason, we have chosen to solve the segmentation problem using a graph partition greedy algorithm with superlinear time complexity. The proof of the time complexity can be found in Ref. [16]. Nevertheless, after the analysis and implementation of this algorithm, two types of problems arose: the existence of hard-constraints which produce inadequate segmentations, and high-variability regions which are not well segmented.

In the graph-based approach to image segmentation, *undirected weighted graphs* (UWG) are used to represent both intensity or colour images. An UWG $G=(V,A)$ is defined from the set of pixels $P=\{p_i\}$ of an image $I=\{I(p) : p \in P\}$ in the following way: To each pixel $p \in P$ corresponds a vertex $v \in V$ and a *neighbourhood* $N_p=\{q \in P \mid 0 < d_p(p,q) \leq \delta\}$, with $N=\{N_p : p \in P\}$ and $d_p: P \times P \rightarrow \mathfrak{R}^+$ a distance between pixels, defining the set of arcs $A=\{a_{pq}=(p,q), q \in N_p\}$. A *weight function* ω defined among arcs gives a measure of similarity between two vertices:

$$\begin{aligned} \omega : A &\rightarrow \mathfrak{R}^+ \\ a_{pq} &\mapsto \omega(a_{pq}) = d_I(I(p), I(q)) = \omega_{pq} \end{aligned}$$

where d_I is a distance measure for image values, which can be intensity or colour. For intensity images, the distance measure used is $d_I : (I_a, I_b) \mapsto |I_a - I_b|$, and for colour images, $d_I : (I_a, I_b) \mapsto \|I_a - I_b\|_{RGB \text{ or } HSI}$, where *RGB* or *HSI* means the use of different colour spaces and their particular metrics. $\Omega = \{\omega(a) : a \in A\}$ is the set of all weights of the set of arcs in G .

A *segmentation* of G is defined as a subgraph $S=(C,F)$ where $C=\{C_i\}$ is the set of components which form a partition of V and $F=\{F_{C_i}\}$ is a *canonical forest*. Each *component* C_i is a set of vertices that are connected by a path of arcs in A . C_p is the component to which the vertex p belongs. A canonical forest F is a set of trees where each one $F_{C_i} \in F$ is a *spanning tree* of $C_i \in C$. We can now define the set Σ of all the segmentations S of G and an equivalence relation, \leq , between its elements which is *reflexive*, *anti-symmetric*, and *transitive*, $T \leq S \Leftrightarrow T \in R(S)$, where $R(S)=\{Q \in \Sigma : \forall C \in Q, \exists C' \in S \mid C \subseteq C'\}$ is a refinement of the segmentation $S \in \Sigma$.

The set (Σ, \leq) is a *partially ordered set* because T and $T' \in R(S)$ do not imply that $T \in R(T')$ or $T' \in R(T)$. Because of that, (Σ, \leq) is a *lattice*, and for any $T=(C,F)$ and $T'=(C',F')$, $T \cap T' \leq T$ and T', T and $T' \leq T \cup T'$, where $T \cap T'=(C \cap C', F_C \cap F_{C'})$ and $T \cup T'=(C \cup C', F_C \cup F_{C'})$. The *maximum* element of (Σ, \leq) is (P, F_p) , the *minimum* one is (\emptyset, \emptyset) and the *atomic* one is $(\{p \in P\}, \emptyset)$. If we follow an algorithm that in every step joins two components C and C' together, the resultant segmentation will be in ascendant order in (Σ, \leq) . This is the case of the class of greedy algorithms that use an ordering of the set of arcs A ; for example, the Kruskal algorithm to find the minimum spanning tree, or the algorithm in Ref. [16].

The problem of segmenting an image I can be translated into finding a “proper” segmentation S from a graph G among the set of all possible segmentations Σ . The issue at hand is to find a segmentation that fulfils a global property by a local search using a greedy algorithm. To do so, we follow the definitions of what is a *not over-segmented* and a *not sub-segmented* image given in [16].

We consider an image is not over-segmented if all the differences between any two adjacent components is greater than their similarities, i.e.,:

$$S \in \Sigma \text{ is not over-segmented} \equiv \forall C_i, C_j \in S \text{ and } C_i \neq C_j \text{ then } Dif(C_i, C_j) > Sim(C_i, C_j)$$

where Dif is a function that measures the *difference* between two adjacent components and Sim measures the *similarity* of these two components. With $NOS = \{T \in \Sigma \mid T \text{ is not over-segmented}\}$. In the same way, we define a not sub-segmented image as any image having the segmentation for which there is a proper refinement such that this refinement is not over-segmented. If $T < S$ means that $T \leq S$ and $T \neq S$, we define the following set $A_S = \{T \in \Sigma \text{ and } T < S \mid T \in NOS\}$. Then, S is *not sub-segmented* $\equiv A_S = \emptyset$.

The algorithm proposed in Ref. [16] (from now on, $F\&H$ algorithm) makes decisions based on local properties of the image, such as pixel differences, and yet the resulting segmentation reflects global properties of the image –not over-segmentation and not sub-segmentation-. But these constraints are *too hard*, causing growing defects in resultant segmentation components. The growing defects are of two kinds, one is related with the theoretical approach, and the other one has to do with its practical results.

$F\&H$ algorithm causes a resultant segmentation S that fulfils both the not over-segmented and the not sub-segmented constraints. Moreover, two successive segmentations S_i and S_{i+1} are $S_i \leq S_{i+1}$. Then, $A_S = \emptyset \wedge S \in NOS \Leftrightarrow S = \min_{T \in NOS} \{T\}$. This means that $F\&H$ algorithm stops at the first segmentation S that is not over-segmented, which is a quite arbitrary and hard constraint. Moreover, visually the segmentation S has still too many components, i.e., is still over-segmented. If we relax the not sub-segmented constraint, it could be possible to attain segmentations S' with less components ($S \leq S'$). We proceed in the following way: If S' is over-segmented, the algorithm will continue until a not over-segmentation S'' ($S' \leq S''$). On the other hand, if S' is not over-segmented, we can break again the constraint by joining together two components as above, or just stop at that segmentation, which will be, greater than S and also not over-segmented, as we were looking for.

In order to manage this leap but avoiding arbitrariness, we have reformulated the concept of the not over-segmented constraint as a problem of minimising an energy function U in the following way:

$$S \in \Sigma \text{ is not over-segmented} \equiv \forall C_i, C_j \in S \text{ and } C_i \neq C_j \text{ then } \Delta U_{S \rightarrow S'} > 0 \text{ where } S \leq S'$$

i.e., S is not over-segmented if a S' has more energy U than S ; otherwise, we could go to S' and minimise the energy U . Moreover, a not over-segmentation S is also a global minimum over the lattice of the energy function U . $\Delta U_{S \rightarrow S'}$ is the energy of the system involved in this transition between two different segmentation $S \leq S'$. If we must join C_i and C_j together, $\Delta U_{S \rightarrow S'} = \Delta U(C_i, C_j)$. In the particular formulation of $F\&H$ algorithm $\Delta U(C_i, C_j) = Dif(C_i, C_j) - Sim(C_i, C_j)$, which shows its equivalence. The resultant algorithm differs only with that of $F\&H$ in the following condition:

$$\text{if } a^q = a_{ij} \mid (C_i^{q-1} \neq C_j^{q-1}) \wedge (\Delta U(C_i^{q-1}, C_j^{q-1}) \leq 0) \text{ then } \begin{cases} C_i^{q-1} \cup C_j^{q-1} \\ F_{C_i^{q-1}} \cup F_{C_j^{q-1}} \cup \{a^q\} \end{cases}$$

A condition to be fulfilled for a certain energy function U to attain this global minimum over the lattice by means of this greedy algorithm is the following: for any arc $a = a^q = a_{ij}$ such that $C_i \neq C_j$, and $C_i^{q-1} \subseteq C_i$ and $C_j^{q-1} \subseteq C_j$ must be true that $\Delta U(C_i^{q-1}, C_j^{q-1}) > 0 \Rightarrow \Delta U(C_i, C_j) > 0$. Then, the segmentation produced by our algorithm is *not over-segmented*.

Proof. Be $a = a_{ij}$ any arc such that $C_i \neq C_j$, i.e., $a^q \notin S$. It occurs at position q in the ordering, then $a = a^q$. $C_i \neq C_j$ implies that $a^q \in S_q$, then $a^q \in S$, which is false. Thus the first condition, $C_i^{q-1} \neq C_j^{q-1}$, is true, which means the second one must be false, i.e., $\Delta U(C_i^{q-1}, C_j^{q-1}) > 0$. This implies that $\Delta U(C_i, C_j) > 0$ and, thus, S is not over-segmented. As an example of that, if we use $\Delta U(C_i, C_j) = Dif(C_i, C_j) - Sim(C_i, C_j)$, in [3] it is shown that $C_i^{q-1} = C_i$ and $C_j^{q-1} = C_j$, which hold for the above property about the energy function. \square

Moreover, using this new definition we can compute a probability of the event $S \rightarrow S'$ or of joining C_i and C_j in a similar way it is computed a certain step in a *simulated annealing* process using the *Metropolis* dynamics:

$$P[C_i \cup C_j] = e^{\left(\frac{-\max\{\Delta U(C_i, C_j), 0\}}{T}\right)}$$

If $\Delta U(C_i, C_j) \leq 0$ then $P[C_i \cup C_j] = 1$, otherwise $P[C_i \cup C_j]$ is compared to a *random number* to decide whether to joint or not. Thus, it is possible to find other not over-segmentations S' such that $S \leq S'$, with a great probability of being not over-segmented. Both the probabilistic and deterministic algorithms break the uniqueness of the solution claimed in Ref. [16].

Using the function $Sim(C) = \max\{\omega(a) : a \in F_C\}$ defined in Ref. [16] leads to difficulties; due to the fact that a component C will not grow for any arc a that $\omega(a) > Sim(C)$, and that $Sim(C) \geq \omega(a')$, $\forall a' \in F_C$, it is only possible that all of arcs in F_C have the same weight. Thus, a huge number of regions are formed, which is critical in homogeneous regions (small $Sim(C)$) because they can never join. To solve this defect, a correction function can be used to compute $Sim(C)$: $Sim(C) = \max_{a \in F_C} \{\omega(a)\} + \tau / |C|$, in such a way that $Sim(C)$ is greater in small components and decrease as these components grows. This helps homogeneous regions to grow, but dissimilar regions also grow, creating *fictitious border regions* or leaving textured regions badly segmented -unless the image is initially smoothed, which expands even more the borders-. The reason is the following: $\max\{\omega\}$ of homogeneous regions is relatively small and $|C|$ grows rapidly because these regions tend to be large. Thus, the correction functions effect over them disappears quickly. But in dissimilar ones, $\max\{\omega\}$ is relatively greater and these regions tend to be small, thus the correction functions effect takes place longer, allowing them to grow too much. The correction functions threshold controls region measure, whether homogeneous or not.

To cope with this pernicious effect, we identify all pixels that belong to these dissimilar regions -we only take under consideration fictitious border regions, not textured ones- by means of a *coefficient* I_C computed over every region C , which is directly proportional to the *compactness*, K_C , -border-shaped- and $\max\{\omega\}$ -high-variability regions-, and inversely proportional to the *area*, A_C , -small regions-:

$$I_C = \frac{K_C \max_{a \in C} \{\omega(a)\}}{A_C}$$

Once these regions have been identified, their pixels are joined randomly to the region with most neighbour pixels, i.e., $p \in C_i \Leftrightarrow C_i = \arg \max_{C \in S} \{\#\{q \in N(p) \mid q \in C\}\}$.

This step can cause some border distortions due the pixel distribution, then it is useful to have as few pixels as possible to distribute. This can be obtained inhibiting for some steps the over-segmentation constraint in the algorithm. Thus, it is granted that, at least, $Sim(C) \geq Threshold$, avoiding the case $Sim(C) = 0$, which do not let the homogeneous regions join together without a correction function. By combining these two steps, it is possible to let homogeneous regions grow, but not the dissimilar ones, with any use of correction functions.

An experimental result using the algorithm as presented in Ref. [5] can be seen in Fig. 1. Fig. 1(c) has been obtained using the probabilistic algorithm inhibiting the over-segmented constraint up to arc weights greater than 2.0. It seems very realistic, but has many fictitious border regions and needs a second step to eliminate them. In (b) a threshold of 0.5 to identify fictitious border regions has been used. In the final result (c), 29 regions have been detected.

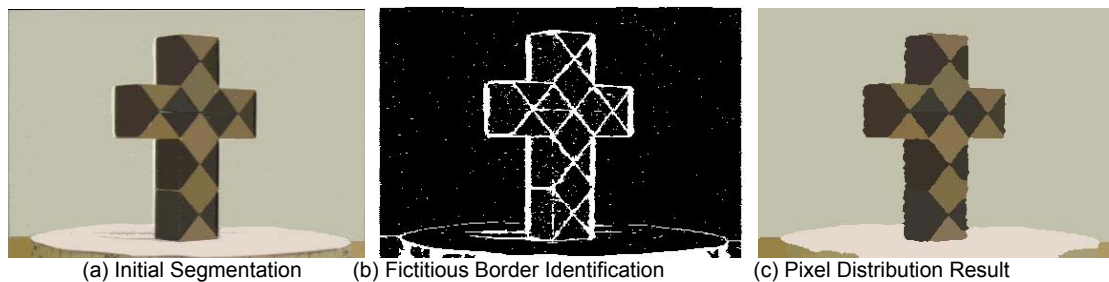


Fig. 1. Colour-based segmentation.

3. Fusion of colour-based segmentation and depth from stereo

One of the objectives of the project is to have a robust scene description using diverse techniques. We are presently working with the idea to integrate perceptual groupings extracted from colour-based segmentation and depth information from stereo. The use of depth cues as an aid to perceptual grouping allows for a robust segmentation of objects and background with similar reflectance properties, that otherwise would be impossible. The method is explained in Ref. [6] and it is summarised next.

The algorithm starts from a set of regions $\mathbf{R} = \{R_1, R_2 \dots R_L\}$ with good color continuity, which can be obtained from either the color segmentation algorithm described above, or any other method such as the one described in [3]. This set of regions is further divided in two subsets. If $|R_i| > t_a$ and $p_i^2/|R_i| < t_c \Rightarrow R_i \in \Omega$, otherwise $R_i \in \Gamma$. $\mathbf{R} = \Omega \cup \Gamma$. Ω represents the regions with area greater than t_a and compactness smaller than t_c , Γ represents the detail in the image, and p_i is the perimeter of R_i .

The two main issues that must be addressed during any perceptual grouping method are: (1) subsegmentation; when two or more entities should have been considered separate elements, but their distinguishable properties were not evident during segmentation, and (2) oversegmentation; when an element in the image was erroneously divided by the segmentation algorithm in smaller entities. Our data fusion algorithm combines depth information from stereo with the color segmentation results described above to solve these issues.

To solve for subsegmentation, in each region Ω the set of points $r_j = \{(x_i, y_i, z_i) | (u_i^d, v_i^d) \in \Omega_j\}$ is extracted from the depth map, and for each point $(u_i, v_i) \in \Omega$ a new point (x_i, y_i, z_i) is added to r_j , where the corresponding $d(u_i, v_i)$ is obtained from the average disparity from the points in the window $(u_i-1 \dots u_i+1, v_i-1 \dots v_i+1)$ with entry in r_j . The process is repeated iteratively until all pixels in Ω have a corresponding entry in r_j . This is, until all pixels in Ω have been assigned a depth estimate.

Smooth surface segments \mathbf{S}_i^j are recursively generated by starting at any point $\bar{\mathbf{p}}_i \in r_j$ and growing outwards while meeting the following two criteria for the neighbouring points $\bar{\mathbf{p}}_i$ and $\bar{\mathbf{p}}_k$: The ‘‘jump edge criterion’’ $\|\bar{\mathbf{p}}_i - \bar{\mathbf{p}}_k\| > t_J$, and the ‘‘curvature criterion’’ $\cos^{-1}(\hat{\mathbf{n}}_i^T \hat{\mathbf{n}}_k) / \|\bar{\mathbf{p}}_i - \bar{\mathbf{p}}_k\| > t_U$. The normals $\hat{\mathbf{n}}_i$ are computed minimizing the error of fitting a local planar patch in the vicinity of $\bar{\mathbf{p}}_i$. The region $R_i \in \Omega$ is then replaced in \mathbf{R} by the segments in \mathbf{S}^j . A new division of \mathbf{R} into Ω and Γ is necessary.

For the case of oversegmentation we first generate a set of initial classes $\omega_i = \{(u_i^d, v_i^d, d_i) | (u_i^d, v_i^d) \in \Omega_i\}$, and the sets of points to be classified $\gamma_m = \{(u_i^d, v_i^d, d_i) | (u_i^d, v_i^d) \in \Gamma_m\}$.

The task at hand is to associate each Γ_m to its parent region Ω_i based on their spatial proximity. This classification constitutes the merging of smooth and detail into spatially coherent entities. Consider one detail region Γ_m . We must compute the distance from the points $(u_i^d, v_i^d, d_i) \in \gamma_m$ to the classes in ω . We use a parametric distance measure for classification. The normalized distance from point $(u_i^d, v_i^d, d_i) \in \gamma_m$ to class ω_i is $d_{il} = \text{tr}(\Sigma_i^{-1}((u_i^d, v_i^d, d_i) - (\bar{u}_i^d, \bar{v}_i^d, \bar{d}_i))^T ((u_i^d, v_i^d, d_i) - (\bar{u}_i^d, \bar{v}_i^d, \bar{d}_i)))$, where $(\bar{u}_i^d, \bar{v}_i^d, \bar{d}_i)$ is the mean vector of class ω_i , and Σ_i the covariance matrix. The votes $V_i = \min_l(d_{il})$ are accumulated for each $(u_i^d, v_i^d, d_i) \in \gamma_m$, and the region in Ω associated to the class ω_i with most votes is considered the parent region for Γ_m .

The result is a new set of regions Ω' where each element $\Omega'_j = \{\Omega_j \cup \Gamma_m \dots \Gamma_n\}$ represents a region in the scene where color continuity and depth continuity are merged to constitute spatially coherent entities. A set of characteristics can be measured on each of these regions, such as position, normal orientation, curvature, level of detail, area, compactness, etc.

The time complexity of the depth from stereo module is $O(dkn)$, where k is the width of the kernel window, d is the maximum expected disparity, and n is the number of pixels in the image. If the edges in \mathbf{G} are sorted in linear time, the segmentation module is bounded by $O(n)$. The time required to compute the normalized distances is bounded by $O(m^2/a^2)$, where m is the number of points in the disparity map

associated to regions in Ω , and $a=|\Omega|$. Given that the disparity map is dense in the perimeter of Ω and negligible inside smooth regions, $m \approx \sum_a p_j k/2$, and from the compactness constraint $p_j < t_c |R_j|$.

The overall cost of the oversegmentation part of the algorithm is bounded by $O(m^2/a) \approx O(a\bar{p}_j^2 k^2) < O(ak^2 t_c |\Omega_j|) < O(k^2 t_c n)$. The time complexity of the algorithm is thus linear with respect to the number of pixels in the image, and is asymptotically comparable to that of the individual data acquisition modules.

Experimental results in a complicated scene can be seen in Fig. 2. The original left colour image from the stereo pair is shown in Fig. 2(a). Both the left and right images were rectified to meet the parallel epipolar constraint. The results of the colour segmentation algorithm applied to the rectified left image are shown in Fig. 2(b). In this figure, the textured regions represent the small segments that need to be classified as belonging to the nearby objects.

Fig. 2(c) shows the disparity map obtained from the stereo module. It is most clear from this image how depth information is dense at highly detailed areas in the scene, whereas smooth regions are poorly represented. Although the left-to-right right-to-left constraint could have been relaxed from equality to similarity when creating the depth map, this was not implemented; letting the segmentation module overcome the weaknesses of any typical depth-from-stereo module.

An initial pruning of the segmented image is done based purely on the mean disparity value of each segment. We eliminated from our three-dimensional region of interest those segments that fall too close or too far from the camera by computing their mean disparity. Also those segments with very low points-in-depth-map to segmented-area ratio were discarded, as they do not contain enough disparity information to accurately estimate their depth, and are not suitable for later attempts at object characterisation. Fig. 2(d) shows how the boxes behind the four objects of interest are virtually eliminated, as well as the table and the vertical bar in the back.

Results from the data fusion algorithm are shown in Fig. 2(e). The image shows our four objects of interest easily identifiable. These segment groups and their attributes can be used to characterise the objects they represent. Fig. 2(f) shows a set of graphs representing the hypothesised objects that are due for recognition.

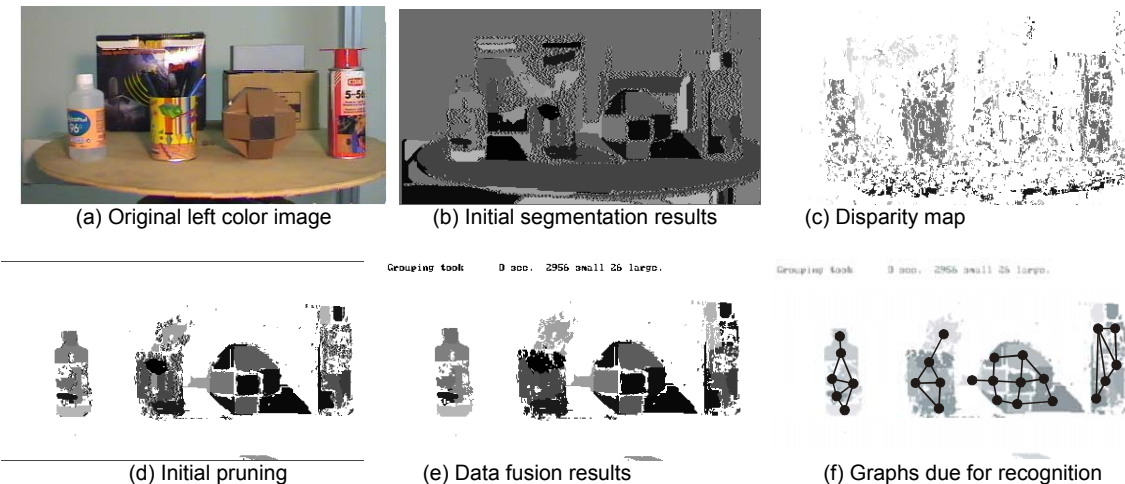


Fig. 2. Data fusion steps on a complicated scene.

4. Representation and recognition of objects using function-described graphs

A fundamental problem in pattern recognition is selecting suitable representations for objects and classes. In the decision-theoretic approach to pattern recognition, a pattern is represented by a set of numerical values, which forms a feature vector. Although, in many tasks, objects can be recognised successfully using only global features such as size and compactness, in some applications it is helpful to describe an object in terms of its basic parts and the relations between them. In these cases, the structural approach to pattern recognition can be applied [1,2].

Nevertheless, there are two major problems that practical applications using structural pattern recognition are confronted with. The first problem is the computational complexity of comparing two structures. The time required by any of the optimal algorithms may in the worst case become exponential in the size of the graphs. The approximate algorithms, on the other hand, have only polynomial time complexity, but do not guarantee to find the optimal solution. For some of the applications, this may not be acceptable. The second problem is the fact that there is more than one model graph that must be matched with an input graph, then the conventional graph matching algorithms must be applied to each model-input pair sequentially. As a consequence, the performance is linearly dependent on the size of the database of model graphs. For applications dealing with a large database, this may be prohibitive.

Attributed Graphs (AGs) are the basic tool for structural pattern recognition and have been widely used in the literature since they were proposed by Tsai and Fu in the latest seventies [17]. Whereas a graph consists of a set of nodes (or vertices) representing pattern primitives and a set of edges (or arcs) representing relations between the primitives, AGs incorporate semantic information about the properties of both the primitives and the relations, through the use of vertex and arc attributes, respectively.

Function-Described Graphs (FDGs) have been introduced recently as a new compact representation of an ensemble of AGs for structural pattern recognition [7-9]. FDGs can be seen as a type of simplification of the general random graphs proposed by Wong *et al.* [10], that borrow from them the ability to probabilistically model structural attribute information, while improving, with respect to first-order random graphs [10, 11], the capacity to record structural relationships that consistently appear throughout the data. FDGs do this by incorporating qualitative knowledge of the second-order probabilities of the graph elements, that are expressed as binary relations (Boolean functions) between vertices or arcs.

Three methods for building FDGs from a set of attributed graphs have been presented elsewhere. The first one synthesises an FDG in a supervised manner [8], while the other two use the dynamic and hierarchical clustering algorithms, respectively [12]. The problem of matching AGs to FDGs for recognition or classification purposes has been studied from a Bayesian perspective [18]. A distance measure between AGs and FDGs may be derived from the principle of maximum likelihood, but robustness is improved by considering only locally the effects of extraneous and missing elements. Hence, the distance measure that will be presented in this section considers the costs of the matching of vertices and arcs using the edit operation approach [19], but incorporates as well the second-order relations between vertices or arcs to enforce the structural properties that are common in the modelled set of AGs. A branch-and-bound algorithm has been proposed to compute this distance measure together with its corresponding optimal labelling [9]. Because of the exponential cost of this method in the worst case, three efficient but approximate algorithms have been also proposed and compared to compute sub-optimal distances (actually matching scores) between AGs and FDGs [20].

In the rest of this section, we recall the definition of both AGs and FDGs, describe the distance measure between them and present the results obtained in some experimental tests in two applications: face recognition and 3D-object recognition. Extended information about FDGs and the techniques used to synthesise FDGs from a set of AGs and to compute optimal or sub-optimal distances between AGs and FDGs can be found in [9].

4.1. Attributed graphs

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

Definition 1. An AG G over (Δ_v, Δ_e) with an underlying graph structure $H = (\Sigma_v, \Sigma_e)$ is defined to be a pair (V, A) where $V = (\Sigma_v, \gamma_v)$ is an *attributed vertex set* and $A = (\Sigma_e, \gamma_e)$ is an *attributed arc set*. The mappings $\gamma_v : \Sigma_v \rightarrow \Delta_v$ and $\gamma_e : \Sigma_e \rightarrow \Delta_e$ assign attribute values to vertices and arcs, respectively, where $\Delta_e = \Delta_v \cup \{\Phi\}$ and $\Delta_v = \Delta_e \cup \{\Phi\}$.

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

4.2. Function-described graphs

Definition 2. A FDG F over (Δ_v, Δ_e) with an underlying graph structure $H = (\Sigma_\omega, \Sigma_\varepsilon)$ is defined to be a tuple (W, B, P, R) such that

1. $W = (\Sigma_\omega, \gamma_\omega)$ is a *random vertex set* and $\gamma_\omega : \Sigma_\omega \rightarrow \Delta_\omega$ is a mapping that associates each vertex $\omega_i \in \Sigma_\omega$ with a random variable $\alpha_i = \gamma_\omega(\omega_i)$ with values in Δ_ω .
2. $B = (\Sigma_\varepsilon, \gamma_\varepsilon)$ is a *random arc set* and $\gamma_\varepsilon : \Sigma_\varepsilon \rightarrow \Delta_\varepsilon$ is a mapping that associates each arc $\varepsilon_{kl} \in \Sigma_\varepsilon$ with a random variable $\beta_j = \gamma_\varepsilon(\varepsilon_{kl})$ with values in Δ_ε .
3. $P = (P_\omega, P_\varepsilon)$ are two sets of marginal (or first-order) probability density functions for random vertices and edges, respectively. This is, $P_\omega = \{p_i(\mathbf{a}), 1 \leq i \leq n\}$ and $P_\varepsilon = \{q_j(\mathbf{b}), 1 \leq j \leq m\}$ (being m the number of edges), where $p_i(\mathbf{a}) \equiv \Pr(\alpha_i = \mathbf{a})$ for all $\mathbf{a} \in \Delta_\omega$ and $q_j(\mathbf{b}) \equiv \Pr(\beta_j = \mathbf{b} | \alpha_{j_1} \neq \Phi \wedge \alpha_{j_2} \neq \Phi)$ for all $\mathbf{b} \in \Delta_\varepsilon$ such that $\alpha_{j_1}, \alpha_{j_2}$ refer to the random variables for the endpoints of the random arc associated with β_j . By definition, $\Pr(\beta_j = \Phi | \alpha_{j_1} = \Phi \vee \alpha_{j_2} = \Phi) = 1$.
4. $R = (A_\omega, A_\varepsilon, O_\omega, O_\varepsilon, E_\omega, E_\varepsilon)$ is a collection of boolean functions defined over pairs of graph elements (i.e. relations on the sets of vertices and arcs) that allow the incorporation of qualitative second-order probability information. A_ω and A_ε are the *vertex antagonism* and *arc antagonism functions*, respectively, where $A_\omega : \Sigma_\omega \times \Sigma_\omega \rightarrow \{0,1\}$ is defined by $A_\omega(\omega_i, \omega_j) = 1 \Leftrightarrow \Pr(\alpha_i \neq \Phi \wedge \alpha_j \neq \Phi) = 0$, and similarly, $A_\varepsilon : \Sigma_\varepsilon \times \Sigma_\varepsilon \rightarrow \{0,1\}$ is defined by $A_\varepsilon(\varepsilon_{kl}, \varepsilon_{pq}) = 1 \Leftrightarrow \Pr(\beta_i \neq \Phi \wedge \beta_j \neq \Phi) = 0$, where $\beta_i = \gamma_\varepsilon(\varepsilon_{kl})$ and $\beta_j = \gamma_\varepsilon(\varepsilon_{pq})$. In addition, O_ω and O_ε are the *vertex occurrence* and *arc occurrence functions*, where $O_\omega : \Sigma_\omega \times \Sigma_\omega \rightarrow \{0,1\}$ is defined by $O_\omega(\omega_i, \omega_j) = 1 \Leftrightarrow \Pr(\alpha_i \neq \Phi \wedge \alpha_j = \Phi) = 0$, and $O_\varepsilon : \Sigma_\varepsilon \times \Sigma_\varepsilon \rightarrow \{0,1\}$ is defined by $O_\varepsilon(\varepsilon_{kl}, \varepsilon_{pq}) = 1 \Leftrightarrow \Pr(\beta_i \neq \Phi \wedge \beta_j = \Phi) = 0$. We say that two graph elements (of the same type) are *co-occurrent* if and only if the occurrence relation applies to them in both directions. Finally, E_ω and E_ε are the *vertex existence* and *arc existence functions*, where $E_\omega : \Sigma_\omega \times \Sigma_\omega \rightarrow \{0,1\}$ is defined by $E_\omega(\omega_i, \omega_j) = 1 \Leftrightarrow \Pr(\alpha_i = \Phi \wedge \alpha_j = \Phi) = 0$, and $E_\varepsilon : \Sigma_\varepsilon \times \Sigma_\varepsilon \rightarrow \{0,1\}$ is defined by $E_\varepsilon(\varepsilon_{kl}, \varepsilon_{pq}) = 1 \Leftrightarrow \Pr(\beta_i = \Phi \wedge \beta_j = \Phi) = 0$.

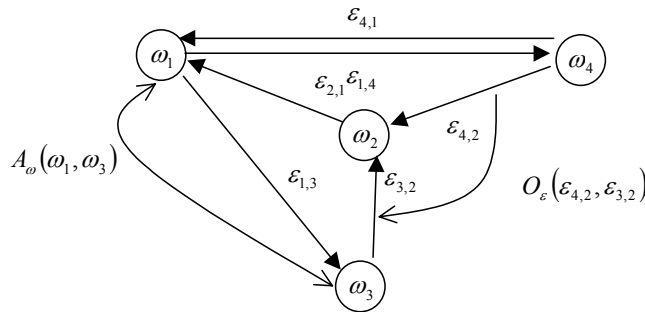


Fig. 3. An FDG with an underlying graph structure composed by 4 vertices and 6 arcs. There is also an antagonism relation between vertices ω_1 and ω_3 and an occurrence relation from arc $\varepsilon_{4,2}$ to arc $\varepsilon_{3,2}$. The probability density functions stored in the vertices and the arcs are not shown.

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

4.3. Distance measure between AGs and FDGs

Let G' be the k -extension of an AG G and F' be the k -extension of an FDG F , both defined over a common attribute domain (Δ_v, Δ_e) , where k is the sum of the number of vertices in G and F . Let $f: G' \rightarrow F'$ be a mapping that labels graph elements, which is actually defined as a pair of morphisms $f = (f_v, f_e)$, where $f_v: \Sigma_v \rightarrow \Sigma_w$ and $f_e: \Sigma_e \rightarrow \Sigma_\varepsilon$ apply on vertices and arcs, respectively.

A reasonable choice for the set H of valid morphisms f is to require a one-to-one mapping between vertices of the extended graphs and to determine the arc mapping from the vertex mapping assuming structural coherence. In addition, the relative order between arcs must be preserved when dealing with planar graphs. A *global cost* C_f^R can be associated with each f in H , and the distance measure between G and F is defined as the minimum of all such costs:

$$d_f^R = \min_{f \in H} \{ C_f^R \} \quad (1)$$

Now, the global cost C_f^R is defined with two terms that depend on the first-order probability information and six more terms that depend on the second-order constraints stored in the FDG,

$$C_f^R = \begin{cases} K_1 * \sum_{\forall v_i \in \Sigma_v \text{ of } G'} C_{f_v} (v_i, f_v(v_i)) + K_2 * \sum_{\forall e_{ij} \in \Sigma_e \text{ of } G'} C_{f_e} (e_{ij}, f_e(e_{ij})) + \\ K_3 * \sum_{\forall v_i, v_j \in \Sigma_v \text{ of } G'} C_{A_{\omega}} (v_i, v_j, f_v(v_i), f_v(v_j)) + K_4 * \sum_{\forall e_{ij}, e_{kt} \in \Sigma_e \text{ of } G'} C_{A_\varepsilon} (e_{ij}, e_{kt}, f_e(e_{ij}), f_e(e_{kt})) + \\ K_5 * \sum_{\forall v_i, v_j \in \Sigma_v \text{ of } G'} C_{O_{\omega}} (v_i, v_j, f_v(v_i), f_v(v_j)) + K_6 * \sum_{\forall e_{ij}, e_{kt} \in \Sigma_e \text{ of } G'} C_{O_\varepsilon} (e_{ij}, e_{kt}, f_e(e_{ij}), f_e(e_{kt})) + \\ K_7 * \sum_{\forall v_i, v_j \in \Sigma_v \text{ of } G'} C_{E_{\omega}} (v_i, v_j, f_v(v_i), f_v(v_j)) + K_8 * \sum_{\forall e_{ij}, e_{kt} \in \Sigma_e \text{ of } G'} C_{E_\varepsilon} (e_{ij}, e_{kt}, f_e(e_{ij}), f_e(e_{kt})) + \end{cases} \quad (2)$$

These eight terms are weighted by non-negative constants K_1 to K_8 , to compensate for the different number of elements in the additions as well as to balance the influence of second-order costs with respect to first-order costs in the overall value.

The individual costs caused by vertices are defined using the probabilities stored in the FDG as

$$C_{f_v}(v_i, \omega_q) = \begin{cases} \frac{-\ln(p_q(\mathbf{a}_i))}{-\ln(K_{pr})} & \text{if } p_q(\mathbf{a}_i) \geq K_{pr} \\ 1 & \text{otherwise} \end{cases} \quad (3)$$

where K_{pr} is a threshold on low probabilities that is introduced to avoid the case $\ln(0)$. In the case of the arcs, the individual costs are defined using the arc conditional probabilities as follows. Let $\gamma_e(e_{ij}) = \mathbf{b}_m$ in the AG arc and let $\gamma_\varepsilon(\varepsilon_{ab}) = \beta_n$ in the matched FDG arc. Then, in general,

$$C_{f_e}(e_{ij}, \varepsilon_{ab}) = \begin{cases} \frac{-\ln(q_n(\mathbf{b}_m))}{-\ln(K_{pr})} & \text{if } q_n(\mathbf{b}_m) \geq K_{pr} \\ 1 & \text{otherwise} \end{cases} \quad (4)$$

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 and must be replaced by the conditional probability $\Pr(\beta_n = \mathbf{b}_m | \alpha_a = \Phi \vee \alpha_b = \Phi)$, which is 1 if $\mathbf{b}_m = \Phi$ and 0 otherwise.

Some second-order costs are added to the global cost of the labelling when second-order constraints (antagonism, occurrence, existence) are broken. Eqs. (5)-(7) show the second-order costs, which can be only 1 or 0, associated with the three relations of antagonism, occurrence and existence between pairs of vertices. The costs derived from the second-order relations between arcs are similarly defined [9].

$$C_{A_{\omega}}(v_i, v_j, \omega_p, \omega_q) = \begin{cases} A_{\omega}(\omega_p, \omega_q) & \text{if } \mathbf{a}_i \neq \Phi \wedge \mathbf{a}_j \neq \Phi \wedge p_p(\Phi) \neq 1 \wedge p_q(\Phi) \neq 1 \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

$$C_{O_{\omega}}(v_i, v_j, \omega_p, \omega_q) = \begin{cases} O_{\omega}(\omega_p, \omega_q) & \text{if } \mathbf{a}_i \neq \Phi \wedge \mathbf{a}_j = \Phi \wedge p_p(\Phi) \neq 1 \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

$$C_{E_{\omega}}(v_i, v_j, \omega_p, \omega_q) = \begin{cases} E_{\omega}(\omega_p, \omega_q) & \text{if } \mathbf{a}_i = \Phi \wedge \mathbf{a}_j = \Phi \wedge p_p(\Phi) \neq 1 \wedge p_q(\Phi) \neq 1 \\ 0 & \text{otherwise} \end{cases} \quad (7)$$

4.4. Experimental tests and results

The previous matching measure between AGs and FDGs has been applied in different contexts, for example, for face recognition [21] and for 3D-object recognition [7,9]. Next, some tests and results on these two applications are summarised.

4.4.1. Face recognition

Fig. 4 displays the node set of the graph for face representation and Fig. 5 shows the result of the feature extraction process from which the graph information is obtained. Some of the distances between the extracted nodes are represented as arc attributes.

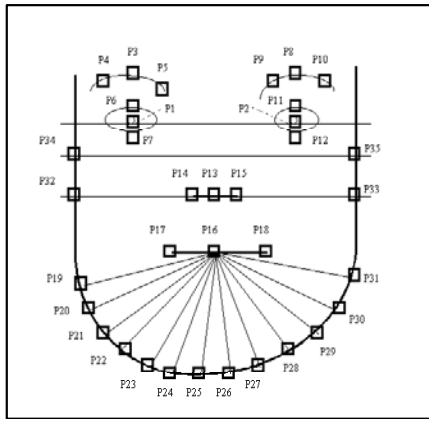


Fig. 4. Node set for face representation.



Fig. 5. Feature Extraction.

Recognition ratio (%)	Full face data	20% occlusion	40% occlusion	60% occlusion	80% occlusion
Data Set 1 LS 1.1/TS 1.2	95.2	90.0	86.8	80.0	58.8
Data Set 2 LS 2.1/TS 2.2	95.2	93.6	90.0	85.6	69.6
Average	95.2	91.8	88.4	82.8	64.2

(a) FDG synthesis and matching using fuzzy discretization

Recognition ratio (%)	Full face data	20% occlusion	40% occlusion	60% occlusion	80% occlusion
Data Set 1 LS 1.1/TS 1.2	90.0	87.6	79.6	75.6	57.2
Data Set 2 LS 2.1/TS 2.2	92.0	92.4	87.6	82.0	65.6
Average	91.0	90.0	83.6	78.8	61.4

(b) 5 – Nearest neighbours

Fig. 6. FDG and 5-NN results.

In the case of face recognition, to carry out identification experiments we used images of a hundred people. These people were divided into two groups of 50 people each (a and b) and starting from each one, training sample FDGs (a1 and b1) and a test sample FDGs (a2 and b2) were generated using five images per person. These images come from the Libor Spacek's database at the Department of Computer Science, University of Essex.

Images are 24-bit colour, 180x200 pixels ones, which were shot at a fixed distance between the video camera and the subject, with a homogenous background and a constant artificial illumination. Samples present people of any sex, several races with or without beard and/or glasses. Training was done using samples a1 and b1, and tests using samples a2 and b2. The classification results obtained by using the graph matching technique based on FDGs are displayed in Figure 6 and compared to the results achieved by means of the k -nearest neighbour technique with $k=5$.

As it can be seen in Fig. 6, the recognition ratio using the graph matching technique was of 95.2% using all the face data (the whole graph) and gracefully degraded by occluding face parts (removing subgraphs) down to a 64.2% when an 80% of the face was occluded (just one of the five subgraph components was used). These results were significantly better than those obtained by the 5-NN technique.

4.4.2. 3D-object recognition

We took sixteen views from four 3D colour objects. In each view, the angle was incremented 22.5 degrees. The views taken with the angles 0, 45, 90, 135, 180, 225, 270 and 315 were used to synthesise the FDGs in the learning process. The other views, 22.5, 67.5, 115.5, 157.5, 202.5, 247.5, 292.5 and 337.5 composed the test set. Images were segmented in regions and an adjacency graph was extracted in which vertices represent these regions with the average hue as the attribute. There was an arc between each pair of adjacent regions as shown in Fig. 7.

The classification results are displayed in Table 1. The FDGs were synthesised using two different approaches: dynamic and hierarchical clustering. The results of the former technique were much better because the presentation order was relevant as given by the rotating view, whereas this information is not exploited in the latter technique. Again, the FDG-based approach outperformed the k -nearest neighbour technique (for $k=1$ and $k=3$).

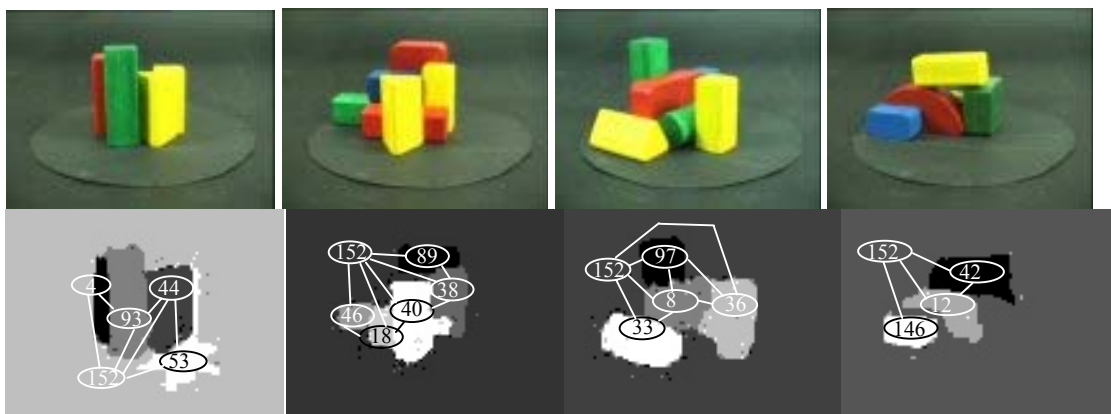


Figure 7. The 4 objects, their segmented views and their corresponding AGs.

Table 1

Classification correctness of the FDGs synthesised with two different learning methods and the k -nearest neighbours.

	Correctness
FDG dynamic synthesis	78%
FDG hierarchical synth.	51%
1- Nearest Neighbour	56%
3- Nearest Neighbour	59%

5. Conclusions

In this paper we have presented several graph-based representations and techniques applied to image modelling, processing and analysis. Our main interest has been to show how these techniques can be used in practical cases for robot vision, covering the stages from low level image segmentation to high level 3D object recognition. Graph techniques permit to represent image objects and scenes in very natural way, without losing the critical information of each one of the parts that belongs to the image object or scene. The graph-based representation and techniques presented here avoid the computational complexity of typical graph problems while offering a good performance in practice. Some experimental results have been shown for image segmentation, perceptual grouping and object recognition.

References

- [1] H. Bunke, Recent developments in graph matching, Proc. 15th Int. Conf. on Pattern Recognition, Barcelona, Vol. 2, 2000, pp. 117-124.
- [2] H. Bunke and A. Sanfeliu (eds.), Syntactic and Structural Pattern Recognition: Theory and Applications, World Scientific, 1990.
- [3] R.C. Wilson and E.R. Hancock, Structural matching by discrete relaxation, IEEE Trans. on Pattern Analysis and Machine Intelligence, 19 (1997) 634-648.
- [4] J. Kittler, W.J. Christmas and M. Petrou, Probabilistic relaxation for matching problems in machine vision, Proc. 4th International Conference on Computer Vision, 1993, pp. 666-674.
- [5] J. Vergés and A. Sanfeliu, Colour image segmentation solving hard-constraints on graph partitioning greedy algorithms, Proc. 15th Int. Conf. on Pattern Recognition, Barcelona, Vol. 3, 2000, pp. 629-632.
- [6] J. Andrade and A. Sanfeliu, Integration of perceptual grouping and depth, Proc. 15th Int. Conf. on Pattern Recognition, Barcelona, Vol. 1, 2000, pp. 295-298.
- [7] F. Serratoso and A. Sanfeliu, Function-described graphs applied to 3D object recognition, Proc. 9th Int. Conf. Image Analysis and Processing, Firenze, Vol. 1, 1997, pp. 701-708.
- [8] R. Alquézar, A. Sanfeliu and F. Serratoso, Synthesis of function-described graphs, Advances in Pattern Recognition, Proc. Joint IAPR Int. Workshops SSPR'98 and SPR'98, Sydney, Springer Lecture Notes in Computer Science, Vol. 1451, 1998, pp. 112-121.
- [9] F. Serratoso, Function-Described Graphs for Structural Pattern Recognition, Ph.D. thesis dissertation, Universitat Politècnica de Catalunya, July 2000.
- [10] A.K.C. Wong, J. Constant and M. You, Random graphs, in: H. Bunke and A. Sanfeliu (eds.), Syntactic and Structural Pattern Recognition: Theory and Applications, World Scientific, 1990, pp. 197-234.
- [11] A.K.C. Wong and M. You, Entropy and distance of random graphs with application to structural pattern recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence, 7 (1985) 599-609.
- [12] A. Sanfeliu, F. Serratoso and R. Alquézar, Clustering of attributed graphs and unsupervised synthesis of function-described graphs, Proc. 15th Int. Conf. on Pattern Recognition, Barcelona, Vol. 2, 2000, pp. 1026-1029.
- [13] Z. Wu and R. Leahy, An optimal graph theoretic approach to data clustering: theory and its application to image segmentation, IEEE Trans. on Pattern Analysis and Machine Intelligence, 15(11) (1993) 1101-1113.
- [14] T. Vlachos and A.G. Constantinides, Graph-theoretical approach to colour picture segmentation and contour classification, IEE Proceedings, Part I, 140 (1), 1993 36-45.
- [15] Y. Xu and E.C. Uberbacher, 2D image segmentation using minimum spanning trees, Image and Vision Computing, 15 (1997) 47-57.
- [16] P.F. Felzenswalb and D.P. Huttenlocher, Image segmentation using local variation, Proc. IEEE Computer Soc. Conf. on Computer Vision and Pattern Recognition, 1998, pp. 98-104.
- [17] W.H. Tsai and K.S. Fu, Error-correcting isomorphism of attributed relational graphs for pattern analysis, IEEE Transactions on Systems, Man and Cybernetics, 9 (1979) 757-768.
- [18] F. Serratoso, A. Sanfeliu, and R. Alquézar, Function-described graphs: a measure of similarity based on probabilities, in: M.I.Torres and A.Sanfeliu (eds.), Pattern Recognition and Applications, IOS Press, Amsterdam, 2000, pp. 59-68.
- [19] A. Sanfeliu and K. Fu, A distance measure between attributed relational graphs for pattern recognition, IEEE Transactions on Systems, Man and Cybernetics, 13 (1983), 353-362.
- [20] F. Serratoso, R. Alquézar and A. Sanfeliu, Efficient algorithms for matching attributed graphs and function-described graphs, Proc. 15th Int. Conf. on Pattern Recognition, Barcelona, Vol. 2, 2000, pp. 871-876.
- [21] J. Vergés, A. Sanfeliu, F. Serratoso and R. Alquézar, Face recognition: graph matching versus neural techniques, Proc. 8th Spanish Symposium on Pattern Recognition and Image Analysis, Bilbao, Vol. 1, 1999, pp. 259-266.