REGION-BASED EXTRACTION AND ANALYSIS OF VISUAL OBJECTS INFORMATION

Verónica Vilaplana, Xavier Giró, Philippe Salembier, Ferran Marqués

Universitat Politècnica de Catalunya,
Jordi Girona, 1-3, 08034 Barcelona, SPAIN
{veronica,xgiro,philippe,ferran}@gps.tsc.upc.edu

ABSTRACT

In this paper, we propose a strategy to detect objects from still images that relies on combining two types of models: a perceptual and a structural model. The algorithms that are proposed for both types of models make use of a region-based description of the image relying on a Binary Partition Tree. Perceptual models link the low-level signal description with semantic classes of limited variability. Structural models represent the common structure of all instances by decomposing the semantic object into simpler objects and by defining the relations between them using a Description Graph.

1. INTRODUCTION

A common procedure to bridge the, so called, semantic gap in the indexing framework is to characterize semantic classes (abstract representations of objects) by means of a combination of low-level descriptors. In this context, low-level descriptors refer to features that can be directly evaluated on the signal (e.g.: color, texture, shape, etc.). The various low-level descriptors that characterize a semantic class have to be computed at different positions and scales within the image. Region-based approaches, as the one discussed in this paper, allow improving the robustness of this process as well as reducing the size of the search space.

The evaluation of low-level descriptors corresponds to a pure perceptual characterization. It is assumed that all instances of an object class (individual representations of an object) will be perceptually similar. However, some semantic classes are not compatible with this restriction. In this case, models that are not purely perceptual should be used. One way of dealing with the variability of the object instances is to analyze and to model the common structure of all instances.

We propose in this paper a strategy to detect objects from still images that relies on combining two types of models: a perceptual and a structural model. The algorithms that are proposed for both types of models make use of a region-based description of the image relying on a Binary Partition Tree. Perceptual models link the low-level signal description with semantic classes of limited variability. Structural models represent the common structure of all instances by decomposing the semantic object into simpler objects and by defining the relations between them.

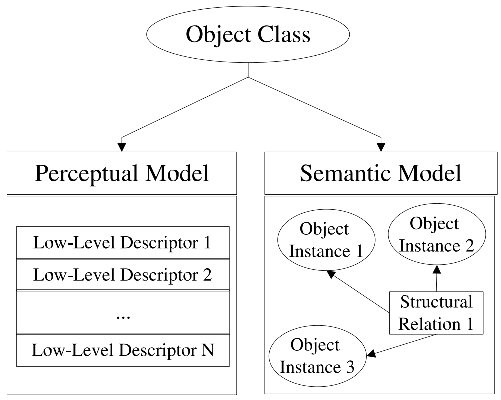

Fig. 1. Perceptual and structural models of an object class

This paper is structured as follows. Section 2 introduces the main concepts of the semantic class model to be used in this work. In Section 3, the usefulness of a region-based image representation is further discussed and the Binary Partition Tree is presented. Section 4 develops the bases of the perceptual model whereas Section 5 details the structural model. In both cases, the extraction of human frontal faces is used as example to illustrate the usefulness of the models. Finally, Section 6 presents the conclusions of this work.

2. OVERVIEW OF THE PROPOSED MODEL

In our work, a Semantic Class (SC) represents the abstraction of a semantic object. We use two types of models to describe a semantic class: a *Perceptual model* and a *Structural model*. A semantic class can be described by both types of models. This concept is illustrated in Fig. 1 where a semantic class (represented by a gray circle) is associated with both descriptions. Typically, semantic classes with a limited amount of perceptual variability can be handled with a

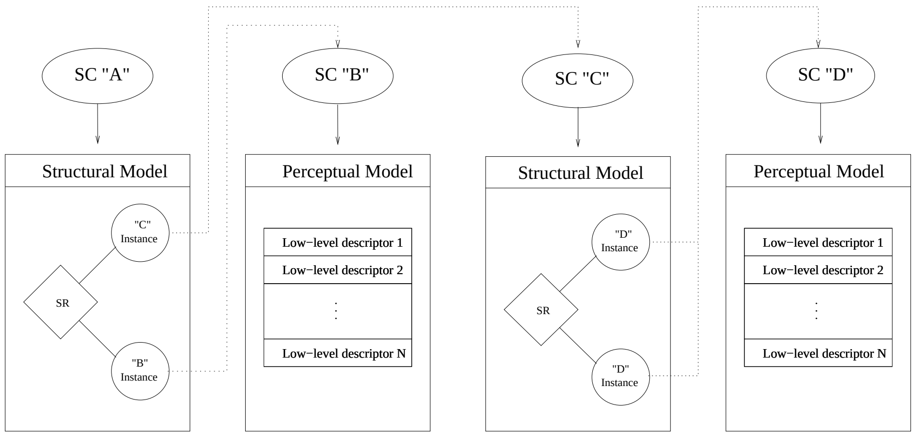

Fig. 2. Example of decomposition of a structural model into simpler structural and perceptual models.

perceptual model whereas more complex semantic classes require a structural model.

Perceptual model: A semantic class can be characterized by a set of low level visual descriptors defining the perceptual characteristics of all class instances. In this work, a low level descriptor is a descriptor that can be directly evaluated on the signal (e.g.: a histogram or the shape of a region). The perceptual model is a list of low-level descriptors whose combination defines the semantic class.

Structural model: In order to deal with the perceptual variability, a semantic class can be decomposed into its simpler parts (parts forming the object) and the relations among these parts. In turn, these simpler parts are instances of simpler semantic classes (e.g.: every wheel in the description of the semantic class ”car” is the instance of the simpler semantic class ”wheel”). In this work, the relations among simpler parts are assumed to be only structural. The structural model that represents the instances of these simpler semantic entities and their Structural Relations (SRs) is described by means of a graph, the so-called Description Graph. This concept is illustrated in Fig. 1, where an example of description graph is presented. In it, instances of simpler semantic classes are represented by white circles while rhombi correspond to structural relations.

The description of a semantic class, however complex, ultimately relies on a set of perceptual models. Every instance in a description graph is associated to a simpler semantic class that may be described by a perceptual or/and a structural model. If the simpler semantic class is described only by a structural model, the previous decomposition can be iterated until reaching the simplest possible level of semantic classes which can only be described by perceptual models. This concept is illustrated in Fig. 2.

3. REGION-BASED REPRESENTATION OF IMAGES: THE BINARY PARTITION TREE

In most object analysis tasks, one of the first difficulties to be faced is related to the raw representation of the original data built around a rectangular array of pixels. Detecting objects directly on this representation is difficult in particular because one has to detect not only the presence of the object but also its position and its scale. In this section, we discuss the interest of Binary Partition Trees [11] as a region-based representation that can be used for a large number of object analysis and recognition applications. The idea is to perform a first step of abstraction from the signal by defining a reduced set of regions at various scales of resolution that are moreover representative of the perceptual features of the image. Instead of looking at all possible pixel locations and all possible scales, the object recognition algorithm will base its analysis strategy on this reduced set of candidate regions.

An example of Binary Partition Tree is shown in Fig. 3. The lower part of the figure presents an original image (left) and an initial partition corresponding to this image (center). The image obtained by filling the regions of the initial partition with their mean gray value is shown in the right part of Fig. 3. As can be seen, almost all details of the original image are visible. Information about the similarity is encoded in the tree shown in the upper part of the figure. The tree leaves represent the regions of the initial partition. The remaining tree nodes are parent nodes. They represent regions that can be obtained by merging the regions represented by the child nodes. As a result, the similarity between regions represented in the lower (upper) part of the tree is very high (rather low).

Several approaches can be followed to create the tree. An attractive solution relies on a region-based merging algorithm that follows a bottom-up approach. Tree leaves represent the regions of an initial partition. The remaining

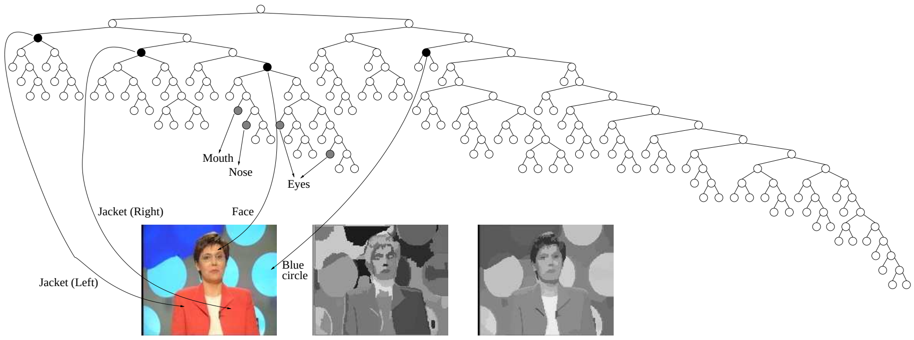

Fig. 3. Example of Binary Partition Tree (top) together with the original image (left), the initial partition with 100 regions (center) and the regions of the initial partition represented by their mean value (right).

nodes represent the regions that are obtained by merging the regions associated to the two children nodes.

Using region-based merging algorithms such as [2, 6], the Binary Partition Tree is created by keeping track of the regions that are merged at each iteration. The homogeneity criterion used in the example of Fig. 3 is based on color similarity. Note that in any cases, if the criteria used to create the tree are generic, it is unlikely that complex objects be represented as individual nodes. Only simple objects that are homogeneous in terms of the criteria used to compute the tree can be expected to be represented in the tree as single nodes. The notion of extended nodes discussed in section 4.3 can be used to tackle this issue.

Once the tree has been created, the remaining analysis steps directly work on its nodes. The number of nodes is dramatically smaller than the number of pixels and the set of nodes spans all possible scales in terms of regions.

4. PERCEPTUAL MODEL

The purpose of the perceptual modeling is to describe a semantic class, here an object, using features that can be directly measured on the signal. The detection of an instance of a given object is done by extracting and analyzing the low-level features, such as pixel distributions or geometrical features, that indicate the presence of the object of interest.

The definition of a semantic class by a perceptual model involves then the selection of a set of useful features followed by a learning stage where these attributes are described by statistical or other kind of models using sample data. Finally, a combination rule has to be defined to merge the information provided by the individual features.

The object detector based on these models analyzes different candidate regions: for each region it extracts the features modeling the semantic class, computes the likelihood value for each descriptor and then combines these values into a final class likelihood. The output of the detector is a list of regions that more likely belong to the class of interest.

4.1. Perceptual Model Definition

4.1.1. Selection of low-level features

The selection of a set of discriminant low-level visual features leads to the idea of seeking features that are invariant to certain transformations of the input signal (translation, rotation, scale, occlusions, projective distortion, non rigid deformations, illumination, etc.). The choice of distinguishing features is a critical design step that requires knowledge about the problem domain. Many different visual features may be employed to describe a semantic class:

Features related to the pixel distribution take into account the value of the pixel components and their relative positions in the area of analysis. The most common ones are color (for example, characterizing the pixel distribution in an area by the color histogram) and texture (for example, using a wavelet or a principal components analysis).

Geometrical features account for the structural characteristics of the area of support of the object being analyzed. The most common ones analyze the shape of the various connected components (for example, using a curvature scale space representation) and the pattern that these components may form (for example, using a graph to describe their relations).

4.1.2. Feature models and decision functions

The next steps in the definition of a perceptual model are the modeling of the feature values for a set of class representatives and the design of a function or decision rule to evaluate if a candidate region is an instance of the class.

The feature models may be built using prior knowledge about the class or by a learning process using training data. When learning is employed, models are usually expressed in terms of statistics or probability distributions associated to the values of the descriptors for the sample data. The training set has to be carefully selected to allow good generalization but to avoid over-training [4]. Following the classification proposed by Tax [13], the learning of the descriptor models may be approached with different techniques: Density methods [1], Boundary methods [13] and Reconstruction methods [1].

For each visual descriptor, a function f has to be inferred so that if x is the value of the descriptor for a given region, $f(x)$ is an estimate of the likelihood or probability that the region is an instance of the class.

Density methods estimate the probability density of the class $f(x) = p(x/\omega)$, where ω is the target class. On the other side, boundary and reconstruction methods fit a model to the data and define a distance between a test instance x and the model, $f(x) = d_\omega(x)$. In some applications, votes or binary outputs are preferred and the function f is an indicator function: $I(p(x/\omega) \geq \theta)$ or $I(d_\omega(x)) \leq \theta)$ where θ is a decision threshold.

4.1.3. **Combining rules**

As mentioned above, in many cases, it is unlikely that a single feature can be used to characterize a class optimally. Using the best feature (the feature that leads to the maximum likelihood) and overlooking the other descriptors might give poor results. To improve the algorithm performance, different descriptors can be combined [13].

The descriptors will be combined using likelihoods. For descriptors that provide distances $d_\omega(x)$ instead of probabilities, the distances must first be transformed into likelihoods. This transformation may be done by fitting sample descriptor values to some distribution or by applying a mapping like $\tilde{p}(x/\omega) = \frac{1}{c_1} exp(-d_\omega(x)/c_2)$, which models a Gaussian distribution around the model if $d_\omega(x)$ is a squared Euclidean distance. Typical combination rules are the weighted sum of estimated likelihoods and the product combination of likelihoods.

4.2. Object detection

For the detection, our strategy relies on a region-based approach. Images are segmented into homogeneous regions and a Binary Partition Tree is constructed from the initial partition. The set of candidates for the object detection are the regions represented by the nodes of the tree. For every node in the Binary Partition Tree, descriptor values and likelihoods are computed and likelihoods are combined into a global class probability. The more likely regions are considered as object instances. This region-based approach reduces the computational burden of an exhaustive search and increases the robustness of the feature extraction.

4.3. Example: face detection

The proposed approach is illustrated with a human face detector based on a perceptual model of the semantic class face (frontal faces). A face can be associated with a set of homogeneous regions. Consequently, it should be possible to find a face by properly selecting a set of regions from a segmented image [8].

4.3.1. **Selection of candidates with a BPT**

The initial image partition is created using a region growing technique, where regions are merged until a given PSNR is reached. Then a Binary Partition Tree is built. The merging order is based on a color similarity measure between regions. Although the use of color as a similarity measure helps to construct meaningful regions in the Binary Partition Tree, the presence of the desired regions (faces) as nodes is not ensured as they are not homogeneous in color.

To overcome this problem and provide the Binary Partition Tree with more flexibility, the tree analysis uses information from regions associated to tree nodes as well as from neighboring regions. The strategy relies on the notion of *extended nodes*. In an extended node, the area of support of a node is determined by the shape of the object to detect, frontal faces in this case. The region corresponding to a node is extended by enlarging its area of support. The new area is formed by the regions of the initial partition contained in a face shape model placed on the node regions. Fig. 6 illustrates this concept and demonstrates the convenience of the method: an extended node may represent objects that are not completely represented as individual nodes in the tree.

4.3.2. **Face class modeling**

In this work, the face class is defined with the following set of low-level visual descriptors. In all cases f_{feat} denotes the normalized decision function derived for the model of feature $feat$.

Color (f_c): The descriptor is the mean value of the region for U and V components in the YUV color space. The face color distribution is modeled using a Gaussian distribution in the (u, v) space.

Aspect ratio (f_{ar}): This descriptor measures the aspect ratio of the bounding box of the region. Its distribution is also modeled with a Gaussian.

Shape (f_{sh}): A face shape model (A) is compared to the shape of the extended node (B). This comparison relies on the modified Hausdorff distance between the contour points of both shapes:

$$H(A, B) = max(h(A, B), h(B, A)) \tag{1}$$

where $h(A, B)$ is the modified directed Hausdorff distance proposed in [3]

$$h(A, B) = \frac{1}{|A|} \sum_{a \in A} min_{b \in B} \|a - b\| \tag{2}$$

The last two descriptors are texture features that use Principal Component Analysis (PCA) to describe the global appearance of regions. Given a collection of n by m pixel training images represented as vectors of size $N = nm$ in an N-dimensional space, the PCA finds a set of orthonormal vectors that capture the variance of the data in an optimal way. The eigenvectors of the data covariance matrix are computed and those with largest eigenvalues are preserved. These vectors are used as basis vectors to map the data.

Distance in the feature space (f_{difs}): A Gaussian model is assumed for the face class in the subspace spanned by the first M eigenvectors of a PCA computed on the training dataset. The similarity measure between a candidate x and the face class is the Mahalanobis distance in the subspace (the distance between x and the sample mean $\bar{x}$).

$$d_{difs}(x) = \sum_{i=1}^{M} \frac{y_i^2}{\lambda_i} \tag{3}$$

where y_i is the projection of the mean normalized vector $x - \bar{x}$ on the i-eigenvector and λ_i is the i-eigenvalue [9].

Distance from the feature space (f_{dffs}): The face class is modeled as the subspace spanned by the first M eigenvectors of a PCA. Another similarity measure between a candidate and the face class is the reconstruction error, the Euclidean distance between the candidate and its projection on the subspace [9].

$$d_{dffs}(x) = \|x - \bar{x}\|^2 - \sum_{i=1}^{M} y_i^2 \tag{4}$$

The distances defined by these two descriptors are transformed into likelihoods by fitting them to the face class distributions obtained through the training data (a χ^2 distribution for the *difs* (f_{difs}) and a gamma distribution for the *dffs* (f_{dffs})).

Density methods are used to learn the feature models for color, shape, aspect ratio and *difs* descriptors, whereas *dffs* is based on a reconstruction method. The training of the feature models is performed with a subset of 400 images from the XM2VTS database [5]. $M = 5$ eigenvectors are used in *difs* and *dffs*. The descriptors are combined by a product combination of estimated likelihoods.

4.3.3. **Face detection**

The face detector computes the descriptors for the candidates defined by the Binary Partition Tree. As too small regions do not contain enough information , nodes smaller than a given threshold are not analyzed.

The likelihoods are combined and the candidates with highest confidence values are proposed as face instances. The result of this procedure is presented in Fig. 4 where the selected node in the example of Fig. 3 is shown. To show

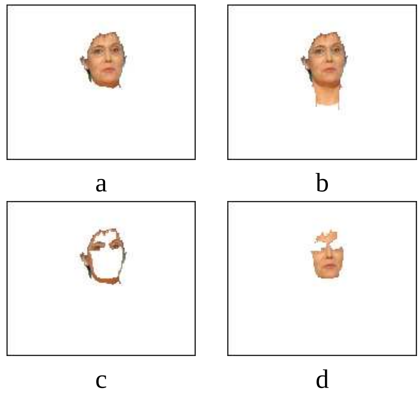

Fig. 4. Results of the face detection on the example of Fig. 3: a) detected face, b) region corresponding to the father node of the face region, c) and d) regions corresponding to the children nodes of the face region.

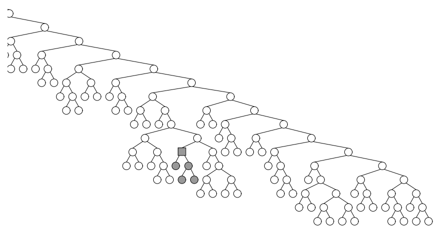

Fig. 5. Binary Partition Tree of the original image of Fig. 6.a. The shaded subtrees correspond to the two face regions (see Fig. 6).

the accuracy of the selected node, Fig. 4.b, c and d present the father and children nodes of the selected one. As can be seen, the selected node is the best representation of the face in the scene that can be obtained.

Fig. 5 and 6 show the Binary Partition Tree and the related images of another example respectively. Two faces are present in the original image. In the Binary Partition Tree of Fig. 5, the subtrees associated to the selected nodes are marked. In this case, the detection of the faces in Fig. 6 illustrates the usefulness of evaluating the extended nodes. The complete shape of the faces can be extracted thanks to the extension of the nodes since complete faces do not appear as single nodes in the tree.

5. STRUCTURAL MODEL

5.1. Definition of structural models with Description Graphs

A second approach for the modeling of semantic classes is to treat them as a structure of simpler semantic classes instead of as a whole. As shown in Fig. 1, a structural model of a semantic class is formed by instances of other semantic

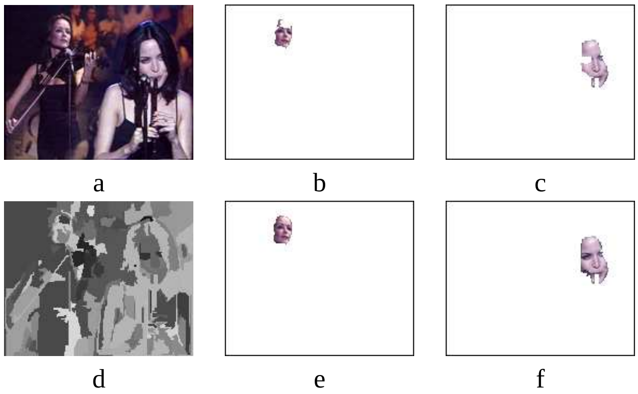

Fig. 6. Example of face detection: a) original image, b) and c) detected face regions corresponding to the shaded subtrees of Fig 5, d) original partition, e) and f) extended nodes of the two face regions.

classes that satisfy certain *Structural Relations (SR)* among them. The proposed approach organizes these instances in terms of a Description Graph (DG) [7]. A Description Graph assigns structural relations and instances of semantic classes to its vertices. A SC vertex is represented by a circle whereas a SR vertex is represented by a rhombus. Description graph edges create connections between the two types of vertices, creating a complex model that describes the common structure of the semantic class. The use of graphs as a tool to express meaning has been widely treated in the work of Sowa on conceptual graphs [12]. Previous experiences on object and event modelization (e.g.: [10]) have shown their applications on the indexing of visual data.

Description Graph vertices are classified into necessary and optional. Necessary vertices correspond to those parts of the object that must be represented in all instances. On the other hand, the presence of the optional vertices is not mandatory but it reinforces the probability of a correct detection. Following the "face" example, a Description Graph for the structural model of a face could be formed by two necessary instances of the SC "eye" and a necessary instance of the SC "mouth", all of them structured with a SR "triangle". The model might be completed with an optional instance of the SC "mustache" related with the eyes and mouth by two optional vertices of a SR "above". This example illustrates the fact that although any complete instance of a face must contain two eyes and a mouth and it may also include a mustache which increases the probability of a correct face detection.

5.2. Likelihood function

Analogously to the low-level descriptors of the perceptual model, Description Graphs also require a likelihood function to measure the *probability (f)* of a set of regions of being an instance of a semantic class. It combines the individual probabilities of each vertex with their weights. The *weight (w)* of a vertex expresses its relevance in the Description Graph, taking values between 0 and 1, where 0 denotes irrelevant and 1 very relevant. Weights can be set manually or as a result of a learning algorithm, and they should be considered as part of the model.

The proposed likelihood function for Description Graphs is shown in Equation 5. The global f is computed by combining the probabilities of the instance vertices weighted according to the model. The expression is normalized by the sum of the weights. Sums for the N necessary vertices and the O optional vertices are expressed separately for clarity.

$$f = \frac{\sum_{k=0}^{N} w_k f_k + \sum_{l=0}^{O} w_l f_l}{\sum_{k=0}^{N} w_k + \sum_{l=0}^{O} w_l} \tag{5}$$

While all necessary vertices must be included when computing f, there is no previous information to decide which optional vertices must be considered. A possible selection criterion is to consider only those optional vertices whose inclusion in the expression increases f. This condition is accomplished when the likelihood of an optional vertex is higher than the current f [7].

5.3. Extraction algorithm

The process of detecting an object in an image can be understood as a graph matching between a subset of Binary Partition Tree nodes and the Description Graph vertices describing a semantic class. For each semantic class vertex in the Description Graph, the likelihood of every node in the Binary Partition Tree to be its instance should be computed. Therefore, the number of possible combinations may be huge, making unadvisable to check every possibility one by one. For this reason, a heuristic algorithm is proposed.

The approach is based on searching first instances of the necessary vertices in the model. Only when this process is finished, the extraction algorithm starts looking for instances of the optional vertices. The algorithm starts analyzing the necessary SC vertex with highest weight. If the associated semantic class is defined by a perceptual model, the instance likelihood is computed directly on the low-level descriptors of the Binary Partition Tree node under analysis.

On the other hand, if the semantic class associated to the necessary SC vertex is defined by a structural model, the algorithm is iterated looking for instances of the vertices of this new structural model in the subtree below the node under analysis. Those Binary Partition Tree nodes whose subtrees contain nodes that match the structural model will be associated to the instance vertex.

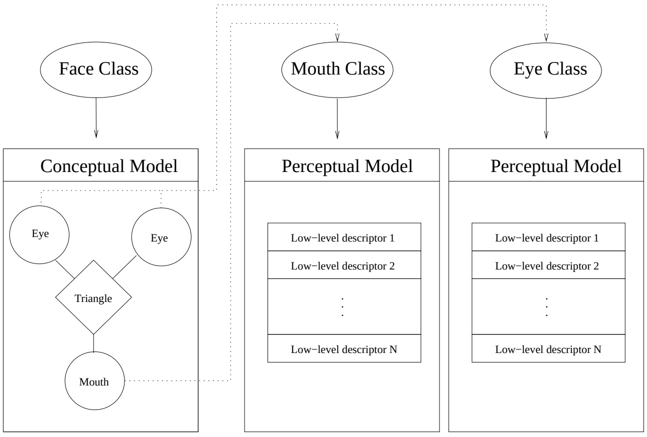

Fig. 7. Object instances in Description Graph refer to models of an object class

As shown in the face example of Fig. 7, structural and perceptual models complement each other to allow the description of complex semantic class.

Once the necessary SC vertex with highest weight has been analyzed, the analysis of the following SC vertex can be performed relying on the SR vertex information. Structural relations provide a prior knowledge that reduce the number of Binary Partition Tree nodes to consider, a restriction that decreases significantly the computation effort.

5.4. Example: face detection

This study case shows a structural approach for the automatic detection of faces in images. In this case, the detection of a face is based on the previous extraction of the individual facial features, a process driven according to the algorithm described in the previous section.

Fig. 8 shows the Description Graph of the considered structural model of face. “Mouth”, “eyes” and “skin” are chosen as necessary vertices, while “eyebrows” and “nostrils” are considered optional because they are not always visible in a human face image. All semantic classes in the Description Graph have their own perceptual model based on color and shape descriptors.

The search for the face starts from the most relevant among the necessary SC vertex of the Description Graph, in our case, first the mouth, secondly the eyes and then the skin. As the eyes must satisfy a triangular structural relation with respect to the mouth, those regions marked as mouth candidates are used as anchor points to find the eye candidates. This approach reduces drastically the total amount of node candidates to be considered. At this point, it is important to notice that the algorithm considers that a single Binary Partition Tree node cannot be part of two different instances (unless one of them is part of the other). This restriction is basic to discard many of the candidates. After

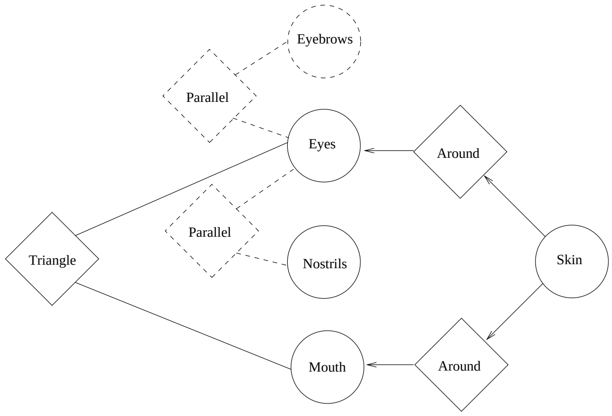

Fig. 8. Description Graph for SC “face” used in the example

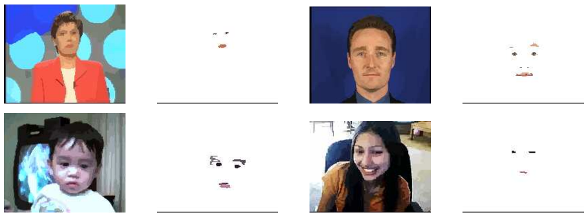

Fig. 9. Initial partitions and extracted facial features of four study cases

the eyes, the algorithm looks for the skin around the mouth and eyes. When the algorithm has been applied on all necessary vertices, the search for the optional ones starts. In this example, optional vertices are those related to eyebrows and nostrils parallel to the eyes. The result of the search is a list of instance candidates ordered according to their final probability. Finally, a threshold is applied to select those candidates accepted as valid instances of the SC “face”

Fig. 9 shows some results of the algorithm on four examples. The training of the weights was performed with a subset of 400 images from the XM2VTS database [5], initially fixing low values to necessary vertices and high values to the optional ones.

6. CONCLUSIONS

In this paper, we have presented a technique for object detection in images. It relies on a combination of two models for characterizing the semantic classes: A perceptual model based on the low-level features, and a structural model that exploits the decomposition of a semantic class into its simpler parts and their structural relations. The application of both models for object detection relies on a region-based description of the image.

The use of a region-based representation of the image allows reducing the set of candidate positions and scales to analyze within the image while improving the robustness of the analysis since all features are estimated on homogeneous regions. In this work, a Binary Partition Tree representation is chosen.

Complex objects are very unlike to appear in a Binary Partition Tree as nodes since, commonly, generic criteria are used for its creation. This difficulty is circumvented by separating the selection of the positions and scales that are to be analyzed (node selection) from the definition of the exact area of analysis (node extension).

The Binary Partition Tree representation is useful as well when using structural models. The characterization of a BPT node as an instance of a semantic class represented by a structural model requires the analysis of the regions that form this node. The Binary Partition Tree allows the analysis to be restricted to the sub-tree associated to this node.

Perceptual models are used in this work to bridge the semantic gap; that is, to characterize objects by means of low-level descriptors. In the examples presented in Section 4, it can be seen that, even for a structured object as a human face, perceptual models can be very useful.

Structural models are used in this work to cope with the variability of the instances of complex objects. Description graphs are used to model complex objects. In these models, the distinction between necessary and optional vertices and their non-linear combination lead to a very robust object characterization.

Finally, it should be stressed that the powerfulness of the proposed approach is based on the combination of both, perceptual and structural models.

7. REFERENCES

[1] C. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, Walton Street, Oxford OX2 6DP, 1995.

[2] J. Crespo, R.W. Shafer, J. Serra, C. Gratin, and F. Meyer. A flat zone approach: a general low-level region merging segmentation method. *Signal Processing*, 62(1):37–60, October 1997.

[3] M.P. Dubuisson and A.K. Jain. A modified haussdorff distance for object matching. In *ICPR-94*, volume A, pages 566–568, Jerusalem, Israel, 1994.

[4] R. Duda, P. Hart, and D. Stork. *Pattern Classification, Second Edition*. Wiley Interscience, 2001.

[5] K. Messer et Al. XM2VTSbd: The extended M2VTS database. In *Proc. 2nd. Conf. On Audio and Video Based Biometric Personal Verification*, New York, 1999. Springer Verlag.

[6] L. Garrido, P. Salembier, and D. Garcia. Extensive operators in partition lattices for image sequence analysis. *EURASIP Signal Processing*, 66(2):157–180, April 1998.

[7] X. Giró and F. Marqués. Semantic entity detection using description graphs. In *Workshop on Image Analysis for Multimedia Application Services (WIAMIS'03)*, pages 39–42, London, England, April 2003.

[8] F. Marqués and V. Vilaplana. Face segmentation and tracking based on connected operators and partition projection. *Pattern Recognition*, 35(3):601–614, 2002.

[9] B. Moghaddam and A. Pentland. Probabilistic visual learning for object representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):696–710, Jul 1997.

[10] M.R. Naphade, I.V. Kozintsev, and T.S. Huang. Factor graph framework for semantic video indexing. *IEEE Transactions on Circuits and Systems for Video Technology*, 12(1):40–52, January 2002.

[11] P. Salembier and L. Garrido. Binary partition tree as an efficient representation for image processing, segmentation and information retrieval. *IEEE Transactions on Image Processing*, 9(4):561–576, April, 2000.

[12] J.F. Sowa. *Knowledge representation: Logical, Philosophical, and Computational Foundations*. Brooks Cole Publishing Co., 2000.

[13] D. Tax. *One-Class Classification: Concept Learning in the Absence of Counter-Examples*. PhD thesis, TU Delft, 2001.