OBJECT DETECTION AND SEGMENTATION ON A HIERARCHICAL REGION-BASED IMAGE REPRESENTATION

Veronica Vilaplana, Ferran Marques, Miriam Leon, Antoni Gasull

Technical University of Catalonia (UPC), Barcelona, Spain
veronica.vilaplana@upc.edu

ABSTRACT

In this paper we present a general framework for object detection and segmentation. Using a bottom-up unsupervised merging algorithm, a region-based hierarchy that represents the image at different resolution levels is created. Next, top-down, object class knowledge is used to select and combine regions from the hierarchy, in order to define the exact object shape. We illustrate the usefulness of the approach with four different object classes: sky, caption text, traffic signs and faces.

Index Terms— Object detection, Image segmentation, Image representations, Image region analysis.

1. INTRODUCTION

The most common approaches to object detection are block-based: the image is scanned at multiple scales with a sliding window of fixed size and shape -typically rectangular-, and the contents of the window are input to a classifier.

One limitation of these approaches, due to the large amount of candidates to evaluate, is their computational cost. Different speed up strategies have been proposed, like the use of heuristics, cascades of classifiers [1] or branch and bound schemes [2].

But the most important limitation is that the output of these systems is typically a bounding box surrounding the object. This is unsatisfactory as a final result because a bounding box does not capture the true shape of the object. The box may contain many non-object pixels, or may lack some object parts. The accurate segmentation of the object requires a post-processing stage.

In the last years, alternative methods have been proposed that incorporate unsupervised image segmentation into an object detection and segmentation framework, assuming that objects are defined by one or several regions in a segmentation of the image.

Regions are appealing for several reasons. One is that they simplify the problem, since the image is represented with a number of elements (the regions) much lower than the number of blocks. Also, the computation of region features is not affected by clutter from outside the region. Finally, regions may provide an accurate representation of the object shape.

Ideally, region boundaries should correspond to object boundaries. However, segmenting semantic objects is a very challenging problem. It is extremely difficult to partition an image into semantically meaningful elements, not just blobs of similar color or texture. Objects may be over-segmented (formed by several regions) or under-segmented (a region may include pixels from multiple objects). Moreover, regions in a partition may be produced by illumination discontinuities or may be artifacts introduced by the segmentation algorithm.

To overcome these limitations, some recent works use multiple segmentations of the same image (varying parameters or even the segmentation algorithm), assuming that the object to detect is correctly segmented in at least one of them [3], or integrate the information from multiple segmentations by classifying each region (in each partition) and combining the results into an object mask [4].

In this paper we present a general framework for object detection and segmentation. To address the weaknesses in image segmentation, instead of working with a single partition we build a hierarchy that represents the image at different resolution levels, the Binary Partition Tree (BPT) [5]. This representation is created using a bottom-up unsupervised merging algorithm. The goal is that nodes in the tree represent objects in the scene (or a very good approximation of these objects). Next, top-down, object class knowledge is used to select and combine regions from the hierarchy, in order to define the exact object shape. This way, the problem of selecting the right scale and location is reduced: the search is performed only at the scales (region size) and locations (region position) defined by the tree nodes. This framework is assessed on four different object detection scenarios: sky, caption text, traffic signs and faces.

After this introduction, Section 2 describes the system, while Section 3 illustrates the usefulness of our approach for different semantic objects. Some conclusions are presented in Section 4.

2. SYSTEM OVERVIEW

Figure 1 shows a general scheme of the system. Each input image is represented with a Binary Partition Tree (*image model*), and tree nodes are described by a set of simple geometric, color and texture features. These features are computed for all the nodes when creating the BPT and stored to be used later, in the detection of different classes of objects.

In the training mode, *object models* that characterize the different classes in terms of several region-based descriptors are learned. An object model may be formed by different kinds of descriptors; we arrange them in three groups according to their use in the system: generic descriptors, a shape descriptor and specific descriptors. Their use in the detection mode is described below.

In the detection mode, for a particular object model, the tree is analyzed. Initially, some nodes are rejected (*Simplification*) based on the information provided by the *generic descriptors* (that were already computed when creating the BPT). For the remaining nodes, a second set of descriptors (*specific descriptors*) is computed for the final *Classification and Decision*. Between these two steps, there is an optional *Shape Fitting* stage. This step is useful for somewhat rigid objects with a known shape. It modifies the area of support of the node to conform to a reference object shape (*shape descriptor*).

This work was partially founded by the projects CENIT-2006 i3media and TEC2007-66858/TCM PROVEC of the Spanish Government.

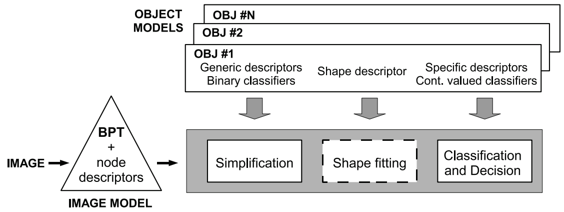
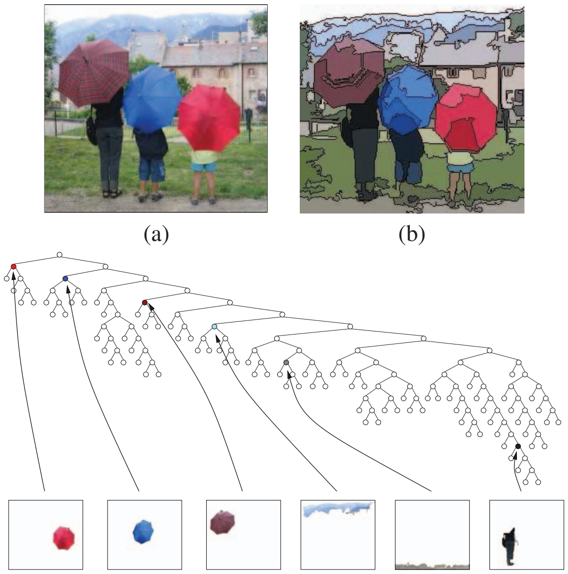


Fig. 1: General detection system


(a) (b)

(c)

Fig. 2: Example of BPT: (a) Original image, (b) Initial partition where each region has been filled with its mean color, (c) Binary Partition Tree and examples of nodes representing objects in the scene.

2.1. The image model

The Binary Partition Tree (BPT) [5] proposes a hierarchy of regions created by a merging algorithm. Starting from a given partition (with any number of regions; we may even assume that each pixel or flat zone is a region), the algorithm proceeds iteratively by (1) computing a similarity measure for all pair of neighbor regions, (2) selecting the most similar pair of regions and merging them into a new region and (3) updating the neighborhood and the similarity measures. The algorithm iterates steps (2) and (3) until all regions are merged into a single region. The BPT stores the whole merging sequence from an initial partition to the one-single region representation. The leaves in the tree are the regions in the initial partition. A merging is represented by creating a parent node (the new region resulting from the merging) and linking it to its two children nodes (the pair of regions that are merged).

The BPT represents a set of regions at different scales of resolution. At lower scales (close to the initial partition) we find a large number of small regions, while at higher scales, regions are larger and possibly more meaningful.

For object detection, we would like the nodes to represent complete or nearly complete objects. As shown in [6] this is possible for many kinds of objects (compact or fairly homogeneous) if the initial partition is fine enough and a similarity measure that takes into account both color similarity and contour complexity is used for the mergings. While chrominance is useful to define homogeneous regions, contour information favors the merging of regions with partial or total inclusions (unless they are very different in color), leading to regions with simple contours. Details on the construction of the tree can be found in [6]. An example of BPT illustrating its ability to represent objects is shown in Figure 2.

2.2. The object model

The goal is to find a description of each object class that has tolerance to intra-class variations and to a certain degree of illumination or viewpoint changes.

Each class is described at two levels. First, at a very general level, with a set of low-level features (which we call generic descriptors) associated with very simple classifiers, each based on one single feature. Second, at a more specific level, using more complex features and classifiers (specific descriptors). The first description is used to simplify the search, eliminating many of the candidate nodes, while the second, more costly, is applied only to the remaining nodes.

Generic descriptors: They are associated with low-level visual attributes which are common to any object. They are simple and relatively easy to measure, and they are pre-computed for all the nodes when creating the tree. We associate each descriptor with a simple threshold classifier which is trained on sample data. In order to cover the four scenarios proposed in this paper we work with the following descriptors. Note, however, that the list can be expanded with new descriptors to deal with other objects.

- *Position:* the mass center of the region.
- *Size:* the number of pixels that form the region.
- *Orientation:* computed using the region central moments.
- *Color mean:* the mean value of the pixels within the region in each color component in the YC_bC_r color space.
- *Aspect Ratio and Oriented Aspect Ratio:* of the bounding box and the oriented bounding box of the region, respectively.
- *Compactness:* the quotient between the region size and the size of its bounding box.
- *Circularity:* the quotient between squared perimeter and region size.
- *Homogeneity:* measured as the power of the coefficients within the region in the Haar wavelet transform (LH, HL and HH subbands, 2 levels).

Specific descriptors: They are more complex and costly than generic descriptors, but they are computed on a few regions. Each specific descriptor is associated with a classifier that outputs a real value which is an estimate of a distance or a resemblance between the region and the object class. We work with the following set (that can be enlarged to model new object classes):

- *Dominant Colors:* the N dominant colors of a region R are: $Dc(R) = \{\{(\mathbf{c}_i, p_i)\}_{i=1,\ldots,N}\}$ (with $N = 8$). Each dominant color is a vector c_i in the YCbCr space, p_i is the fraction of pixels in the region with color c_i, and $\sum_i p_i = 1$.
- *Histogram:* in the YCbCr color space.
- *Symmetry:* the reflectional symmetry of a region is measured with respect to an axis with the orientation of the region and passing through its center of mass.
- *Hausdorff distance:* a reference shape model is compared to the shape of the region. The comparison relies on the modified Hausdorff distance between the contour points of both shapes.

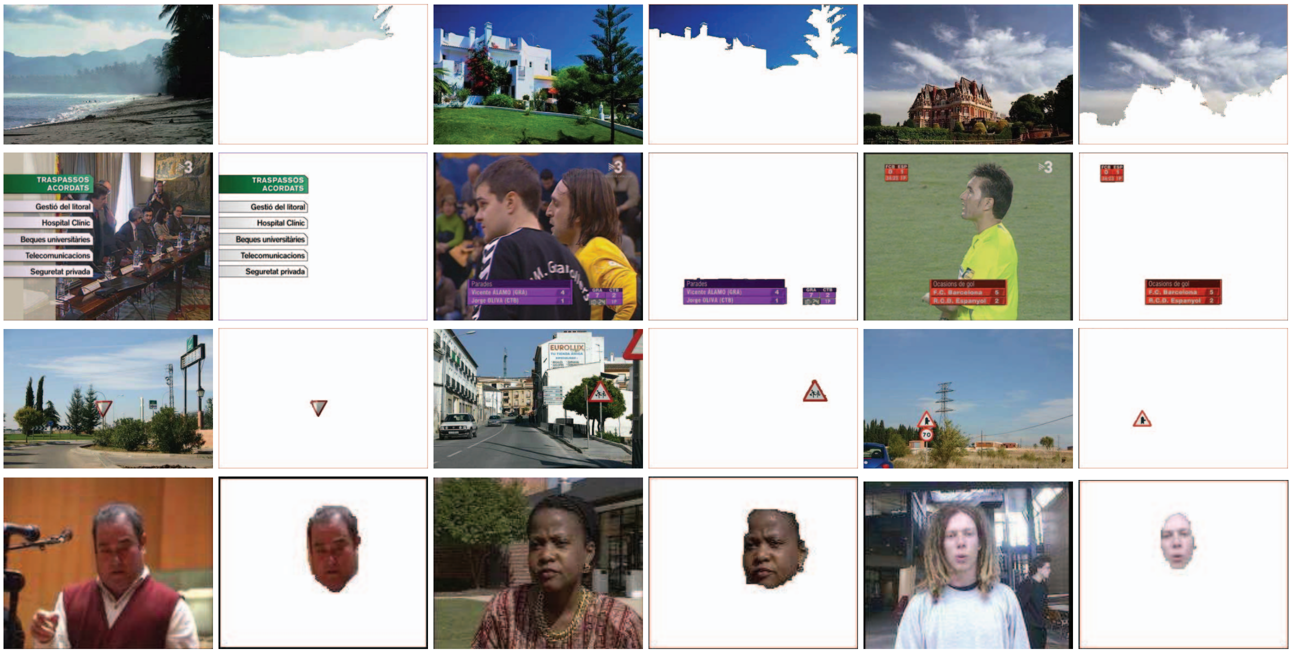


Fig. 3: Different object detection problems: Sky (first row), Caption text (second row), Warning traffic signs (third row) and Faces (fourth row)

- *PCA coefficients:* the region is projected on the subspace spanned by the first M eigenvectors of the PCA (computed on a training set of object samples).
- *Haar coefficients:* the coefficients of the Haar wavelet transform of the region, $N = 3$ levels.

The classifiers are trained with different techniques. *Hausdorff distance* and *Symmetry* descriptors already output a distance and a likelihood, respectively, and the χ^2 distance is used for the color *Histogram*. For the other descriptors, a density method (Gaussian distribution) is used for the *dominant colors*, reconstruction error and Support Vector Data Description [7] for *PCA projections*, and Real-Adaboost for *Haar coefficients*.

Note that different object classes may be described by different attributes. For example, shape is a key attribute to detect warning traffic signs, but is useless to describe sky. For each object class, we have to select the set of descriptors that best describes that class.

2.3. Simplification

The goal of this stage is to discard as many nodes as possible by a fast analysis of the regions associated with them. Each node passes through a cascade of very simple classifiers. Each classifier is trained on one of the generic descriptors and its output is binary. If a node is rejected by one classifier it is not analyzed by the next classifiers in the cascade. This is a very useful stage since, on average, between 85 and 95 percent of the nodes are rejected.

2.4. Shape fitting

This is an optional step that can be used to improve the representation given by the tree nodes. It is useful when the object has a known, rather rigid shape, to solve possible segmentation errors, such as leakages in the regions created in the segmentation.

A model of the object shape is fitted to each active node, and the area of support of the node is modified by adding or removing small regions from the initial set of regions that form the node. The fitting is performed with a shape matching technique based on distance transforms, by finding the parameters of the transformation (translation, scale and rotation) that minimize a similarity measure between the transformed reference shape and the node. The usefulness of this stage was demonstrated in [6].

2.5. Classification and decision

After the simplification and shape fitting stage, a small number of nodes are still active. For each active node we compute the specific descriptors and classify the node.

When several classifiers are used, their (continuous valued) outputs are combined into a global likelihood using logistic regression.

Once likelihoods are estimated, a last stage is performed to select the best representation of the object in the image taking into account both likelihoods and the inclusion and neighborhood information encoded in the BPT structure.

3. OBJECT DETECTION AND SEGMENTATION

In this section we illustrate the usefulness of our approach for the detection and segmentation of different objects. For each class, we briefly explain the interest in the detection task, analyze the attributes describing the class and list the descriptors used to train the object model. Results are presented in Subsec. 3.5.

3.1. Sky detection

Sky detection can contribute to image understanding by indoor /outdoor classification. Moreover, accurate sky segmentation can be used for content-based manipulation, like picture quality improvement by color enhancement, or background detection for 3D depth-map generation.

Attributes: Sky can have different appearances: clear, cloudy or overcast, and colors may cover a broad range from saturated blue to gray. Sky may appear as one or several connected components which are more likely to be found on the top of the image, and to present a smooth texture.

Object model: Given the previous attributes, the class ”sky” is represented using the generic descriptors *Position*, *Size*, *Color mean* and *Homogeneity* and the specific descriptor *Dominant colors*. Note that, in this case, the shape fitting step is not applied.

Typically, after classification there are several nodes, classified as sky, which represent different parts of the sky (some regions may overlap). A last stage analyzes the tree structure to select the set of largest connected components classified as sky (the top sky nodes in a subtree). All these components form the final sky area.

3.2. Caption text detection

The next application we address is the detection of caption text in key frames of a video sequence. Caption text is text artificially superimposed on the video at the time of editing and it usually underscores or summarizes the video content. It is, therefore, particularly useful for semantic video indexing.

Attributes: Caption text can be broadly described as text typed inside a rectangular box, horizontally aligned, highly contrasted with respect to the background and with a characteristic textured aspect.

Object model: The previous attributes can be translated into constraints on the following generic descriptors: *Aspect ratio*, *Compactness* and *Homogeneity* [8]. Selected regions are extended by fitting their shape to a rectangle, and classified using *Haar coefficients*. Finally, the tree structure is analyzed to obtain disjoint representations of the text lines (see Fig.3). This is a necessary step to correctly binarize the region and input the result to an OCR system.

3.3. Traffic sign detection

Traffic sign detection is a key point on the development of automatically driven vehicles and safety systems in human-driven vehicles.

Attributes: In this work, we focus on the detection of warning traffic signals. Such signals are characterized by their triangular shape and the color distribution: a white triangle (containing some black structures) surrounded by a red frame.

Object model: Given the previous attributes, the class of warning traffic signals is represented using the generic descriptors *Aspect ratio*, *Compactness* and *Homogeneity* and *Size*. In this case, the shape descriptor is a triangle whereas the following specific descriptors are used: *Color histogram* and *Hausdorff distance*. Before computing the Hausdorff distance, the reference shape model is transformed using the parameters found in the shape fitting stage. This way, we can detect signs with different orientations (see Fig. 3).

3.4. Face detection

Human face detection is an initial step in any face recognition security system. Furthermore, face detection is useful for content-based image retrieval and indexing, selective video coding, crowd surveillance, security, and intelligent human- computer interfaces.

Attributes: We focus on frontal or nearly frontal views of human faces, which can be described in terms of skin-like color areas with an elliptical shape. Moreover, faces present a specific texture pattern given by the relative positions of eyes, nose and mouth.

Object model: The generic descriptors used in this case are *Color mean*, *Size*, *Aspect ratio*, *Compactness*, *Circularity* and *Homogeneity*. Region fitting is carried out using an elliptical shape model. Finally, the following specific descriptors are used: *Dominant Colors*, *Hausdorff distance*, *Symmetry*, *PCA coefficients* and *Haar coefficients* (only one strong classifier with 200 features).

3.5. Results

We work with the following datasets: COREL for sky, TV news key frames for text, [9] for traffic signs and XM2VTS, Banca and MPEG7 for faces. Fig. 3 illustrates the type of results obtained using this approach. Objects are detected and accurately segmented in spite of their variability in color and shape (see first row), their presence in clutter background (see second and third rows) or their low contrast with the background (see mainly fourth row).

Table 1 summarizes the results in terms of recall, precision and segmentation accuracy. The segmentation accuracy between a ground truth region R_{GT} and a detected region R_D measures the area of overlap: $Acc = \frac{|R_{GT} \cap R_D|}{|R_{GT} \cup R_D|}$. The Acc is averaged over all segmented objects for which there is a ground truth region.

	SKY	**TEXT**	**SIGN**	**FACE**
Objects for training	50	100	30	2000
Img. with+without obj.	$300 + 100$	$100 + 50$	$70 + 20$	$550 + 50$
Objects for detection	300	249	70	558
Recall	0.97	0.86	0.94	0.97
Precision	0.93	0.89	0.99	0.95
Obj. with ground truth	200	130	60	158
Segmentation Accuracy	0.92	0.89	0.98	0.88

Table 1: For each class: number of training objects, number of images with and without objects, number of objects in the images with objects, recall, precision, number of objects for which the ground truth segmentation is available and average segmentation accuracy.

4. CONCLUSIONS

We have presented a general framework for object detection and segmentation which relies on a hierarchical representation of the image and has a flexible structure. New object models can be easily learned by selecting a suitable set of descriptors and classifiers. The performance is very good in the four object classes considered.

5. REFERENCES

[1] M. Jones and P. Viola, “Fast multi-view face detection,” in *Proc. IEEE CVPR*, 2003.

[2] C. Lampert, M. Blaschko, and T. Hofmann, “Beyond sliding windows: object localization by efficient subwindow search,” in *Proc. IEEE CVPR*, 2008.

[3] B. C. Russell, A. A. Efros, J. Sivic, W. T. Freeman, and A. Zisserman, “Using multiple segmentations to discover objects and their extent in image collections,” in *Proc. IEEE CVPR*, 2006.

[4] Caroline Pantofaru, Cordelia Schmid, and Martial Hebert, “Object recognition by integrating multiple image segmentations,” in *Proc. ECCV*, 2008.

[5] P. Salembier and L. Garrido, “Binary partition tree as an efficient representation for image processing, segmentation and information retrieval,” *IEEE Trans. on Image Processing*, vol. 9, no. 4, pp. 561–576, April 2000.

[6] V. Vilaplana, F. Marqués, and P. Salembier, “Binary partition trees for object detection,” *IEEE Trans. on Image Processing*, vol. 17, no. 11, pp. 2201–2216, November 2008.

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

[8] M. Leon, V. Vilaplana, A. Gasull, and F. Marques, “Caption text extraction for indexing pursposes using a hierarchical region-based image model,” *Proc. IEEE ICIP*, 2009.

[9] S. Maldonado et.Al., “Road-sign detection and recognition based on svm,” *IEEE Trans. on Int. Transportation Systems*, vol. 8, no. 2, pp. 264–278, June 2007.