OBJECT RECOGNITION BASED ON BINARY PARTITION TREES

Oreste Salerno, Montse Pardàs, Verónica Vilaplana and Ferran Marqués
Dept. Teoria del Senyal i Comunicacions, Universitat Politécnica de Catalunya
UPC-Campus Nord, C/ Jordi Girona, 1-3, 08034 Barcelona
e-mail: montse@gps.tsc.upc.es

ABSTRACT

This paper presents an object recognition method that exploits the representation of the images obtained by means of a Binary Partition Tree (BPT). The shape matching technique in which it is based was first presented in [1]. This method compares a transformed version of an object shape model (reference contour) to the contours of a partition of the image. The comparison is based on a distance map that measures the euclidean distance between any point in the image to the partition contours. In [1], this algorithm was applied using a colour-based segmentation of the image and a full-search was performed to find the best match between the searched object and the contours of this segmentation. Here, the information of the Binary Partition Tree is used both to obtain the segmentation and to guide and reduce the search for the optimum match between the shape and the objects of the image.

1. INTRODUCTION

Shape recognition is an important problem in image processing with a large number of applications, such as model based recognition, depth from stereo, image database categorization and tracking [2]. Typical approaches to object matching rely on the detection of the most relevant contours in an image and their subsequent comparison to an object shape model (*reference contour*). That is, it is decided whether the reference contour is correctly represented by the extracted contours, up to some scaling or 2D rigid transformation and some permitted level of noise.

In [1], we proposed to obtain the contours present in the scene using a homogeneity-based technique; that is, a segmentation approach. The segmentation looks for the areas in the image where elements are homogeneous with respect to a given criterion. This way, global and complex criteria can be easily introduced leading to more reliable contours [3]. The approach presented in [1] is improved in this paper by using a Binary Partition Tree [4] to obtain the partition and to guide and reduce the search for the optimum match between the shape and the objects of the image.

Transitions in the scene (image contours) are represented in the image partition. The object matching technique proposed in [1] compares transformed versions of the reference contour to the contours in the partition. The comparison is based on a distance map that measures the euclidean distance between any point in the image and the partition contours. In [1], the candidates for the central position of the matching object were considered to be all the pixels of the image. The main problem of this approach is that the best matching can correspond to pieces of contours of different objects which do not compose a correct object. To use as candidates only the central position of the regions of the partition is not a good solution either, because the objects that we look for may be composed of different regions of the partition. The nodes of the Binary Partition Tree provide a solution to this problem. The search will be conducted only on the positions defined by the possible merging of regions that appear as nodes of the Binary Partition Tree. This kind of search will also make straightforward the creation of multiple hypotheses as a result of the algorithm.

The structure of this paper is as follows. After this introduction, Section 2 discusses the main steps of the matching algorithm; namely, Binary Partition Tree creation, generation of the image partition, distance map computation, reference contour generation, cost computation and search using the Binary Partition Tree. In Section 3, some examples are given to show the improvements of the algorithm with respect to the previously published work. Finally, Section 4 gives some conclusions.

2. MATCHING ALGORITHM

The proposed object matching algorithm involves the search of a transformation of the reference contour that minimizes a given cost function. In our case, the cost function to be mimized relies on the use of a distance map. The distance map stores the information about the euclidean distance between any point in the image and the contours in the partition. This partition is generated using a Binary Partition Tree. The pixels defining the transformed contour are evaluated over the distance map. The cost is computed by adding these values and normalizing them with respect to the size (number of pixels) of the transformed contour. This cost is computed not for every pixel of the image, as in [1], but only in a selected area around the center of the regions corresponding to the nodes of the BPT. The position and transformation of the contour leading to the minimum cost is selected.

2.1 Binary Partition Tree

Object matching aims at detecting the presence and position in an image of a given object, modeled by the reference contour. An object is an entity with semantic meaning that is commonly associated to a set of connected regions whose contours are defined in the original image. As a result, it should be possible to define an object by selecting a set of regions from a correctly segmented image. This concept leads to the use of color and texture homogeneity criteria in the segmentation process. Many segmentation procedures could be used for this aim. However, the usage of the Binary Partition Tree for creating this segmentation, will at the same time provide us with a suitable data structure to perform the search in the object matching procedure.

A Binary Partition Tree is a structured and compact representation of the regions that can be obtained from an image. Several approaches can be used to create this tree. We have used a segmentation that follows a bottom-up approach. The algorithm first constructs the Region Adjacency Graph of an initial partition. In our case, the initial partition is the partition of flat zones. Using a region based segmentation, the Binary Partition Tree is created by keeping track of the regions that are merged at each iteration until one region is obtained. That is, for each pair of neighboring regions a homogeneity measure is assessed. The algorithm then starts merging the pair of neighbors whose distance is lowest. The process is iterated until one region is obtained. An example is shown in Figure 1. In order to be able to show the correspondence of the nodes with the image regions, we have taken an initial partition of only 10 regions, as shown in Figure 1 (a). Thus, the leaves of the tree in Figure 1 (b) represent the regions of this initial partition. The remaining nodes of the tree represent regions that are obtained by merging the regions represented by the two children of the node.

The Binary Partition Tree should be created in such a way that the most meaningful regions are represented in its nodes. In our case, the object recognition technique that we are designing will usually search for objects that are homogeneous in color or texture. This leads us to generate a BPT where the homogeneity criterion is the color similarity.

The merging algorithm we are using is a region growing approach [4] based on an euclidean weighted distance in the *(y, u, v)* color space, where more relevance is given to the luminance component.
The merging order between pairs of neighbor regions is based on this distance. Regions (r_1, r_2) are compared using the mean values of their *(y, u, v)* components.

(a) (b)

Figure 1. Example of initial partition (a) and the Binary Partition Tree created from this initial partition (b).

2.2 Partition selection

Once the Binary Partition Tree has been constructed, a partition within this tree is selected. This partition will be used for computing the Distance map in the next step. The segmentation consists on merging the most similar regions until a termination criterion is reached. Because of the construction of the BPT, this segmentation is computed by deactivating nodes following the merging sequence until the stopping criterion is met.

The stopping criterion is based on the PSNR achieved when representing the original image by the partition with all regions filled with their mean values. The PSNR is tipycally set to 22dBs.

2.3 Distance map generation

As above commented, the distance map represents the euclidean distance between any point in the image and the contours in the selected partition. Several approaches can be followed to estimate the euclidean distance [5]. We have computed the distance map d(x,y) for every pixel with coordinates (x,y) following the morphological approach [6]. More details are given in [1].
In Figure 2(a) we show a representation of the distance map using as partition the one used as initial partition in the example of Figure 1, that is Figure 1 (a).

2.4 Reference contour generation

Given a reference contour modeling the desired object, its various transformations have to be computed to look for the one leading to the best match with the map distance information. The transformation parameter space is sampled and quantized so that the set of possible solutions is reduced.

As explained in [1], two different types of contour representation are possible: parametric and non-parametric ones. Both types of contour representation can be used within this BPT framework

2.5 Cost computation

Once a set of parameters defining a transformation has been fixed, the transformed reference contour is computed (C_T) and so is its cost:

$$J(C_T) = \frac{1}{\|C_T\|} \sum_{(x,y)\in C_T} d(x,y) \quad (1)$$

That is, the cost is estimated by adding the distance values of those pixels defining the transformed contour and normalizing them with respect to its size. The transformed contour within the search space leading to the minimum cost will be selected. In the next sub-section we will describe how this search space is selected using the BPT.

2.6 Search space using the BPT

Our objective is to find, using the partition of the image, a set of regions which best match a given shape. In [1], the cost function (1) was computed using all the pixels of the image as central position for the transformed reference contours in order to find the best matching regions. This full search for the central position of the matched object has several disadvantages. First, it has a high computational load. Second, in the presence of a complicated background, as many contours can appear in the partition, the selected contours may not correspond to an object. Because no texture information is considered for the cost computation, some object matchings are considered which do not correspond to real objects. They may be composed of pieces of contours belonging to different objects or be a group of neighboring regions which do not group into an object. To avoid these problems, we can take advantage of the information contained in the BPT.

For the construction of the search space we will use the BPT that is generated taking as leaves the regions of our partition (as defined in section 2.2). The nodes of this BPT (including the leaves) will be used as references for the positioning of the transformed reference contours. As the BPT is created by merging neighboring regions using an homogeneity criterion which should define the objects of the image, its nodes are good candidates in the object matching procedure.

For instance, following the example of Figure 1, if we consider Figure 1(a) the partition of our image, then the search will be conducted in all the nodes of the BPT represented in Figure 1(b).

In order to find the best match, the following steps are performed to compute the matching cost of each node.

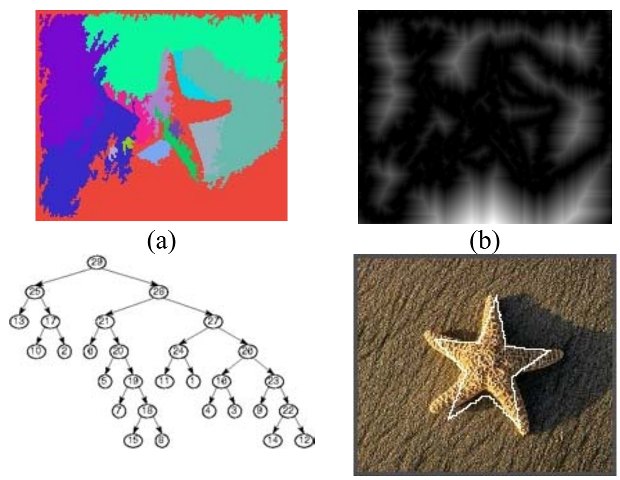
a) Define the search area around the mass center of this node
b) Define the transformation parameters for the reference contour appropriate for the node size. Define also the rotation parameters range.
c) Compute the cost (1) for every candidate center position (defined in 1(a)) and for every transformation parameters (defined in 1(b))

After doing this procedure for every node, the one which has obtained a smaller cost is selected, together with the positioning and transformation parameters that lead to this cost.

In Figure 2 (b), the result of the best matching to an ellipse for the example of Figure 1 is shown. Observe that this ellipse corresponds to the node 11 in Figure 1 (b), which is the merging of regions 6 and 4. Another example is shown in Figure 3 for a different shape, in this case, a star. The selected node is number 21.

(a) (b)

Figure 2. (a) Distance map for the partition in Figure 1 (a) (b) Matching result for the same partition.



(a) (b)

(c) (d)

Figure 3. (a) Partition for "Starfish" (b) Distance function (c) Binary Partition Tree (d) Best match result

3. RESULTS

In this section, we will present two examples of object location which show the advantage of the usage of the BPT in the search strategy. In the first place, we will show how the usage of the BPT allows introducing another restriction, as the color homogeneity, in the selection of the region. Second, we will show how the BPT can be used for generating a result with multiple hypotheses which can be useful in many applications.

3.1 Comparison with a full search strategy

Figure 4 shows how the BPT search strategy can improve the results of the full search strategy. In Figure 4(a) the partition of the image is shown. Since this is a very textured image, the initial partition is composed of 62 regions. A full search of the best matching ellipse produces the result of Figure 4(b). As it can be seen, the regions which this matching defines are not homogeneous and do not compose an object of the image. When the BPT based search is applied, the node shown in Figure 4(c) is obtained as the best match. The final result is shown in Figure 4(d).

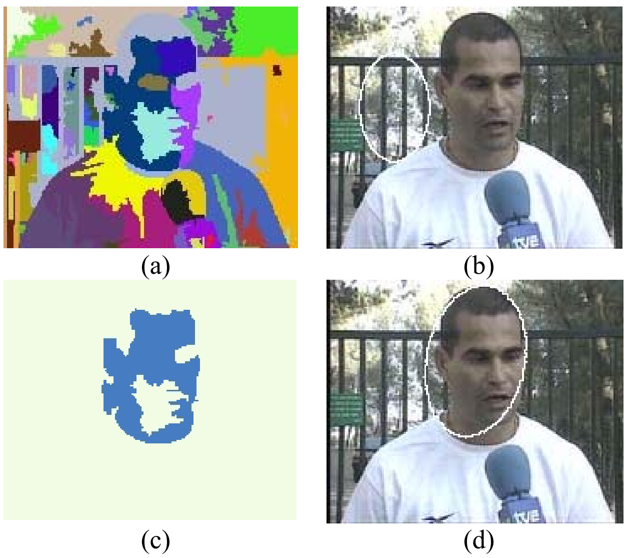


Figure 4. (a) Partition (b) Result using a full search (c) Representation of the node selected (d) Final result

3.2 Multiple hypotheses generation

Using the BPT search strategy, it is straightforward to generate a multiple hypotheses result. In Figure 5, two examples are shown. If a full search is used, the best hypotheses correspond to slightly different positions with slightly different transformation parameters for the reference contour. Instead, by using the BPT search, the selection of the N nodes which produce the minimum cost will provide the N best matchings. We only need to take into account that a parent node can produce the same result as a child node if one of the children nodes is inside the other one. Thus, the multiple hypotheses generated in this situation have to be discarded.

These multiple hypotheses can be useful as input for a more complex classification system which takes into account higher level information.

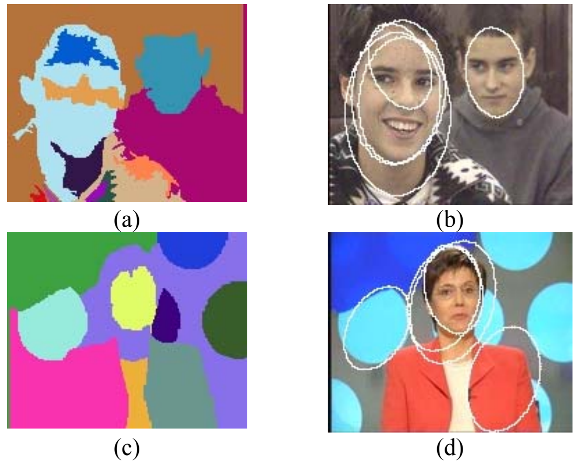


Figure 5. (a), (c): Partition. (b), (d): Result of the 5 best hypotheses using the BPT

4. CONCLUSIONS

In this paper, we have presented an approach for object matching that makes use of the partition information obtained using Binary Partition Trees. The Binary Partition Tree has been used in the search strategy for the best match in order to overcome some of the problems encountered by a full search strategy.

The tests performed confirm that the use of the BPT is useful to avoid finding matches with sets of neighboring regions which do not correspond to objects. This approach has also shown to be useful for generating multiple hypotheses results.

ACKNOWLEDGMENTS

This work has been partly supported by the IST program of the EU IST-2000.32795 SCHEMA and grant CICYT TIC2001-0996 of the Spanish Government.

5. REFERENCES

[1] F. Marqués, M. Pardàs, R. Morros, "Object matching based on partition information", proceedings of IEEE Int. Conf. on Image Processing, ICIP 2002, Vol. 2 pp. 829-832, Sept. 2002.

[2] A. K. Jain, Y. Zhong and S. Lakshmanan. "Object matching using deformable templates". *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 10(3):267-27, March 1996.

[3] P. Salembier and F. Marqués. "Region-based representations of image and video: Segmentation tools for multimedia services". *IEEE Trans. on Circuits and Systems for Video Technology*, 9(8):1147-1169, December 1999.

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

[5] O. Cuisenaire, B. Macq. "Fast Euclidean distance transformations by propagation using multiple neighbourhoods", *Computer Vision and Image Understanding*, 76(2):163-172, Nov. 1999.

[6] P. Soille, *Morphological Image Analysis : Principles and Applications*, Springer Verlag, June 1999.