OBJECT MATCHING BASED ON PARTITION INFORMATION

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

ABSTRACT

This paper presents a new technique for object matching that exploits the information about transitions in the image obtained by means of a segmentation approach. Object matching is performed by comparing a transformed version of an object shape model (reference contour) to the contours in the image partition. The comparison is based on a distance map that measures the euclidean distance between any point in the image to the partition contours. Examples using parametric and non-parametric reference contours are provided to assess the quality of the proposed technique.

1. INTRODUCTION

Object matching 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 [1]. 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.

Object matching approaches usually extract the contours in the image using a transition-based technique (i.e.: gradient detection). This type of contour detection technique are commonly based on local decisions that may overlook some contour information. This is due to the fact that even abrupt transitions in the scene may be represented by local smooth transitions in the image data and thus contours may not be easy to localize.

In this paper, we propose obtaining 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 [2].

Transitions in the scene (image contours) are represented in the image partition. The proposed object matching technique compares transformed versions of the reference contour to the contours in the partition. The accepted transformations depend on whether the reference contour is defined as a parametric or a non-parametric curve. The comparison is based on a distance map that measures the euclidean distance between any point in the image and the partition contours.

The structure of this paper is as follows. After this brief introduction, Section 2 discusses the main steps of the matching algorithm; namely, image segmentation, distance map computation, reference contour generation and cost computation. In Section 3, some examples are given of both parametric and non-parametric reference contours. Finally, Section 4 gives some conclusions and comments on the work we are currently developing.

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. The pixels defining the transformed contour are evaluated over this 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. The transformation leading to the minimum cost is selected.

2.1. Image segmentation

An object matching process 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 which 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. Furthermore, contour features may be introduced to avoid noisy contours and to obtain regions conforming to the shape of natural objects.

In this work, we want to use a generic segmentation in order to be able to deal with any type of object. Therefore, we are using homogeneous criteria relying on basic features such as color. This way, the object will very likely be defined by the union of a set of color homogeneous regions. That is, the transformed reference contour will most likely match to a set of contours representing not only a single region but the union of several regions in the partition.

The segmentation algorithm we are using is a region growing approach [3] based on an euclidean weighted distance in the *(y, u, v)* color space, where more relevance is given to the luminance component:

$$d(r_1,r_2)=\sqrt{\gamma(\bar{y}_1-\bar{y}_2)^2+\frac{(1-\gamma)}{2}\left((\bar{u}_1-\bar{u}_2)^2+(\bar{v}_1-\bar{v}_2)^2\right)}$$

The merging order between pairs of neighbor regions is based on the previous distance. Regions (r_1, r_2) are compared using the mean values of their *(y, u, v)* components. 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 weight γ is tipycally set to 0.5 and the PSNR to 35 dBs.

2.2. Distance map generation

As above commented, the distance map represents the euclidean distance between any point in the image and the contours in the partition. Several approaches can be followed to estimate the euclidean distance [4]. In this work, the morphological approach has been followed [5]; that is, the euclidean distance is estimated carrying out successive morphological dilations of the partition contour set (Q):

$$d(Q,x)=n \Leftrightarrow x\in Q\oplus B_n \cap x\notin Q\oplus B_{n-1}$$

where x is the point under analysis and B represents the structuring element. In this work, we have used a 3x3 square structuring element.

The fact of using a partition distance map for computing the matching cost presents several advantages with respect to directly using the contour information. Distance functions are continous and smooth. Therefore, areas with low gradient values in the original image which may have leaded to errors in the partition (absence of contours) receive the distance value related to the closest contour in the partition. Such distance values are usually low and, therefore, gaps in the contour definition are correctly filled.

2.3. 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. Nevertheless, a full search within this reduced space has been adopted.

In this work, we have dealt with two different types of contour representation: parametric and non-parametric ones. In the case of parametric contours, the transformed contour is re-computed for each set of transformation parameters. On the contrary, in the case of non-parametric contours, the points representing the transformed contour before applying the uniform translation parameters (e.g.: after rotation and scaling) are stored. For each point, the off-set with respect to the center of the (partially) transformed contour is calculated and used for computing the matching cost for each permitted translation. This allows reducing the computational load of the algorithm.

2.4. 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:

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

That is, the cost is estimated by adding the distance values of those pixels defining the tranformed contour and normalizing them with respect to its size. The transformed contour leading to the minimum cost is selected.

In order to avoid local minima of the cost function, only transformed contours totally included in the image are taken into account.

3. RESULTS

In this section, two different examples of object location using the proposed contour matching are presented. The main difference between both cases is the representation of the reference contour either as a parametric or a non-parametric curve. In the case of having a parametric curve, the set of possible transformations to be applied to the reference contour is determined by the curve parametrization itself. On the contrary, if a non-parametric curve is used, the allowed transformations are only rotation, scaling and translation.

3.1. Parametric curve

In this example, we want to locate a human face in the scene. The reference contour which is used as model for the human face shape is an ellipsis (thus, a parametric curve). In both examples, the partition is presented in color to better allow its visual analysis. As it can be seen in both cases, no single region in the partitions correctly represents the face shape. Moreover, both faces contained a region that, due to the lack of contrast in the neck area, covers part of the face and the neck (even the pullover in the case of Figure 1). In spite of that, both distance maps present rather low values all around the face due to the smooth nature of distance functions which allows the correct location of the reference contour. Note that, since in this case reference contours are parametrized, the aspect ratios of the final selected elipses accomodate to the face shapes.

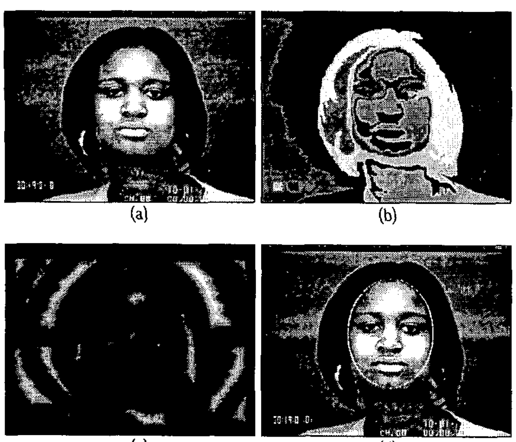

Fig. 1 Example of parametric contour matching: (a) original image, (b) image partition, (c) distance map and (d) final result.

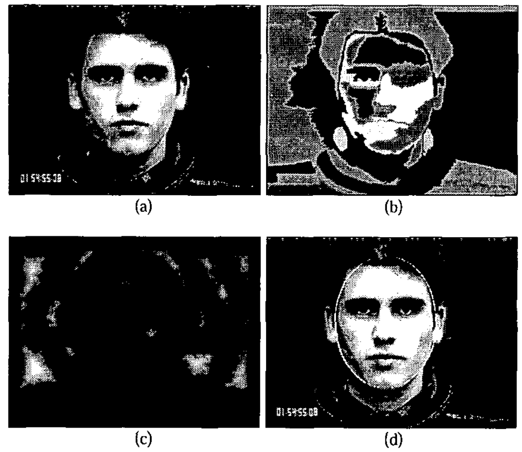

Fig. 2 Example of parametric contour matching: (a) original image, (b) image partition, (c) distance map and (d) final result.

3.2. Non-parametric curve

In this example, we want to locate a starfish in the scene. The reference contour which is used as model for the starfish shape is a fixed five-points star (thus, a non-parametric curve). Here, partitions are not presented in color since, given the type of images, they contain very few regions. In this example, the reference contour (see Figure 3.d) does not correspond very closely to the object shape. In addition, the presence of very textured areas in the images leads to regions in the partition that do not perfectly conform the object shape.

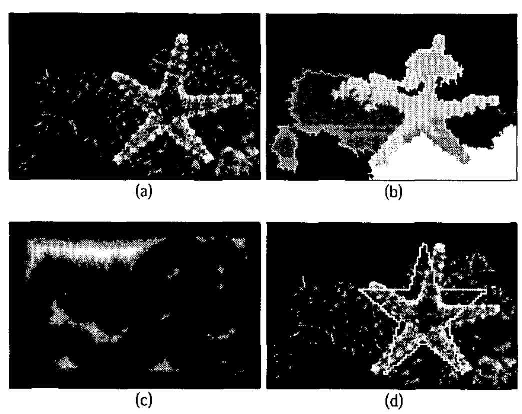

Fig. 3 Example of non-parametric contour matching: (a) original image, (b) image partition, (c) distance map and (d) final result.

In spite of the previous comments, the object matching algorithm correctly locates the model. It has to be commented that in Figure 3, even with the differences between the object and model shapes, the algorithm has correctly located the starfish although it has not been able to estimated its rotation.

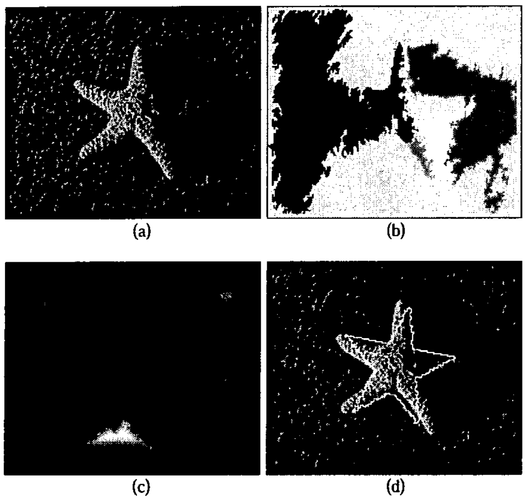

Fig. 4 Example of non-parametric contour matching: (a) original image, (b) image partition, (c) distance map and (d) final result.

In both cases, it has to be noticed that, although using a pure color based segmentation, the resulting partition has enough quality to allow detecting the object. In these cases, contour detection algorithms may detect a large amount of contours in the textured areas that would complicate the object detection process.

4. CONCLUSIONS AND CURRENT WORK

In this paper, we have presented a new approach for object matching that makes use of the partition information obtained using a color-based segmentation approach.

The use of a segmentation for computing the contours in the scene permits a better estimation of the contour positions. Segmentation processes enable a more global analysis of the image than typical contour extraction techniques. It has been shown that, although relying on a simple color criterion, the obtained partitions can extract the most relevant contours even in the presence of very textured objects (see Figures 3 and 4). Such textured objects would lead to a very dense contour map in the case of applying a contour detector.

Furthermore, it has been seen that the use of distance functions allows smoothing the contour information stored in the image partition. This allows handling cases of not perfect match and interpolating the contour information so that a sort of gap filling is performed.

The proposed technique has shown to be robust in front of location, rotation and moderate scale changes of the reference contour. Local minima of the cost function can be avoided if some a priori knowledge of the range of possible size variations is applied.

Currently, we are improving the previous technique by including multiresolution analysis into the process. This is done in two different ways. First, the parameter space is quantized and analyzed from coarse to fine resolutions, which allows concentrating the computational effort in the subspace(s) containing the most promising solutions. Second, the image partition is filtered at different levels using a size criterion. This enables the removal of small regions in comparison with the size of the reference object being considered, which reduces the amount of possible local minima.

ACKNOWLEDGMENTS

This work has been partly supported by the CEE project InterFace (IST-1999-10036) and by the grant CICYT TIC2001-0996 of the Spanish Government.

5. REFERENCES

[1] 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.

[2] 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.

[3] 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.

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

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