HUMAN FACE SEGMENTATION AND TRACKING USING CONNECTED OPERATORS AND PARTITION PROJECTION[1]

Ferran Marqués, Verónica Vilaplana, Anabel Buxes

Universitat Politècnica de Catalunya, Campus Nord – Mòdul D5
C/ Jordi Girona 1-3, Barcelona (08034), Spain
ferran@gps.tsc.upc.es

ABSTRACT

A new technique for segmenting and tracking human faces in video sequences is presented. The algorithm uses a connected operator to extract the connected component that more likely belongs to a face. Such a connected operator is implemented by means of a Binary Partition Tree. A set of connected regions (a node in the tree) is selected maximizing an estimation of the likelihood of being part of a face. Faces are tracked through the sequence based on the partition projection approach. A face and a non-face core component are obtained in the current image by projecting the previous partition. The technique has been successfully assessed using several test sequences from the MPEG-4 database (raw format) as well as from the MPEG-7 database (MPEG-1 format).

1. INTRODUCTION

Automatic face detection and tracking is a key issue for many applications [1]. Currently, there is an increasing interest in this area due to the activities carried out in the MPEG-4 and MPEG-7 standardization processes. MPEG-4 applications rise a new necessity since the analysis algorithm should, not only detect the position of the face, but really segment it obtaining its actual shape. In the MPEG-7 context, face detection, as a previous step for face and person recognition, will help developing tools for enable the user to access databases.

A common approach to face detection is that of view based eigenspaces [2, 4]. In previous works [8, 9], this approach has been extended to directly deal with regions. In this paper, we present a new technique that improves both the face segmentation and the face tracking steps. Improvements are related to a more efficient analysis of the Binary Partition Tree, which stores the set of connected components that are candidates to form the face, and to a more accurate partition projection and lower complexity face refinement technique, which increase the tracking algorithm performance.

2. FACE SEGMENTATION USING CONNECTED OPERATORS

The proposed technique tries to avoid working at pixel level and, as first step, segments the image into homogeneous regions. Color information is used to achieve more accurate contours. The face component will be obtained as a union of regions belonging to this initial partition. Thus, this partition has to ensure that it correctly represents the face contours. Towards this goal, a modified watershed approach has been used. Figure 1 shows an example of this type of initial partition.

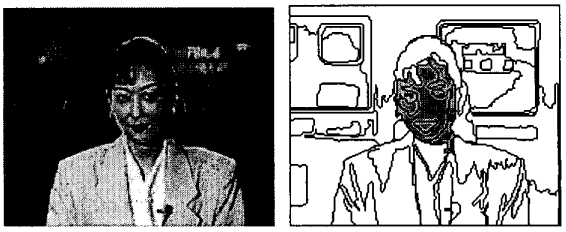

Figure 1. Original image (frame #0 of *Akiyo*) and example of initial partition where regions forming the face are highlighted

The watershed approach uses color and contour information. A scalar space is created by computing an Euclidean weighted distance in the (y, u, v) color space, where more relevance is given to the luminance component. A final scalar space combining contour information and the previous color measure is created to actually compute the watershed. In it, smooth contours are prioritized since they usually conform better to real objects.

The selection of the union of initial regions that forms the face could be carried out by computing, for each region and possible union of regions, an estimation of their likelihood to belong to a face. As it will be described in Section 2.2, this procedure is cumbersome and, therefore, a simplified technique has been chosen.

[1] *This work has been partially supported by the ACTS-057 project (VIDAS) of the European Union and by the grant* TIC98-0422 *of the Spanish Government.*

0-7803-5467-2/99/ $ 10.00 © 1999 IEEE

2.1 Building the Binary Partition Tree

The basic concept relies on the general merging strategy proposed in [3]. In it, a merging algorithm is defined by a merging order, a merging criterion and a region model, and the merging order and criterion are assumed to be independent. This merging strategy allows the implementation of segmentation algorithms and connected operators. An efficient way to implement some of these connected operators uses the concept of Binary Partition Tree (BPT) [6]. Given a partition, where the size of the regions can be as small as a single pixel, a region merging order is defined. This merging order is based on a similarity measure between regions and regions are merged by pairs. The merging sequence that is obtained is represented by a BPT. In it, leaves are related to the regions in the original partition, nodes are associated to the various regions created during the merging process and, therefore, links connect two merging nodes.

Once the merging order has been established, the BPT is analyzed and the merging criterion is assessed at each node. The merging order proposes a set of region unions to the merging criterion. Then, the merging criterion takes the final decision whether two regions have to be merged or not. This strategy is useful in a large set of applications [6] and, specifically, in the case of having a complex merging criterion that may imply a large computational load.

In the current application, the measure used in the merging order is chrominance similarity. Since faces contain regions that are homogeneous in chrominance, the merging order is computed using the region similarity in the (u, v) color components. The merging order is established up to only one region remains and the merging sequence is then analyzed. Figure 2 presents the BPT of the partition in Figure 1.

2.2 Analysis of the Binary Partition Tree

The distance proposed in [4] is used as merging criterion and, thus, computed at the nodes of the BPT. A face class (Ω) is created relying on a database of normalized face images (N-dimensional vectors). For each region, an N-dimensional vector representing the scaled version of the region x_w is constructed, in order to be able to detect faces of any size. The class membership of an image x_w is modeled as a unimodal gaussian density function:

$$P(x_w/\Omega) = \frac{\exp\left[-1/2(x_w-\bar{x})^T\Sigma^{-1}(x_w-\bar{x})\right]}{(2\pi)^{N/2}|\Sigma|^{1/2}}$$

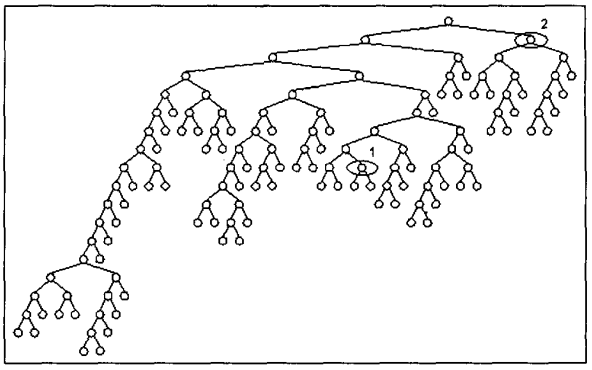

Figure 2. Binary Partition Tree created from the initial partition of Figure 1.

where the mean and covariance matrix are estimated using a training data set. The Mahalanobis distance is used as a sufficient statistic for characterizing this likelihood. A computationally tractable estimate of this distance, based on the M first principal components of the covariance matrix, is

$$\hat{d}(x_w/\Omega) = \sum_{i=1}^{M}\frac{y_i^2}{\lambda_i} + \frac{1}{\rho}\varepsilon^2(x_w/\Omega)$$

where y_i are the projections of x_i over the M principal components and λ_i are the M principal eigenvalues, with M<<N. Moreover, ρ is the average of the N-M remaining eigenvalues and $\varepsilon^2(.)$ is the residual reconstruction error:

$$\varepsilon^2(x_w/\Omega) = \sum_{i=M+1}^{N} y_i^2 = \|x_w-\bar{x}\| - \sum_{i=1}^{M} y_i^2$$

To compute and evaluate all possible scaled versions of a given region is not feasible. Therefore, the algorithm creates an auxiliary rectangular image and only two possible scaled versions are computed: horizontal and vertical normalization with respect to the images contained in the database.

Since different scaling is necessary for each region represented in the BPT, the computation of the distance at a given node cannot be obtained relying on the distances computed in its children nodes. Thus, to reduce the computational load of the BPT analysis, some nodes and even sub-trees are pruned. The first pruning step is based on the size under analysis. Only regions of a minimum size are assumed to be large enough to represent a face in the scene.

Moreover, and previous to assessing the given distance in a node, its color characteristics are analyzed. This way,

the BPT is pruned by comparing the average color components of each node with the average face color components. The comparison is not very restrictive in order to be robust against camera calibration and color saturation. Figure 2 presents the BPT of the partition in Figure 1. During the analysis step, the sub-tree associated to node 2 has been pruned. These regions represent the blue screen on the top-right corner of the image.

The analysis of the BPT shown in Figure 2 leads to the selection of the node 1. That is, node 1 is the region present in the BPT that maximizes the likelihood of being a part of a face. The region associated to this node is presented in Figure 3.a. Figure 3.b corresponds to the second candidate node (that leading to the second largest likelihood). Note that the selection of this node would have been correct as well. Figure 3.c presents the father node of the selected node in the BPT; that is, the candidate region obtained by merging the selected node with its closest neighbor in the (u, v) sense. Note that this node represents a region that contains areas not belonging to the face and, thus, the algorithm rejects it.

If the image contains several faces, the BPT analysis can provide with the core regions associated to them. Usually (see Figure 3), the best candidate regions belong to the same face in the scene. Therefore, the selected nodes have to be further analyzed to obtain unconnected regions representing the various faces. Candidate nodes can be selected either by fixing a maximum number of faces to be detected (if this information is known before hand) or by setting a threshold in the node distance. Figure 4 shows an example of multiple face detection.

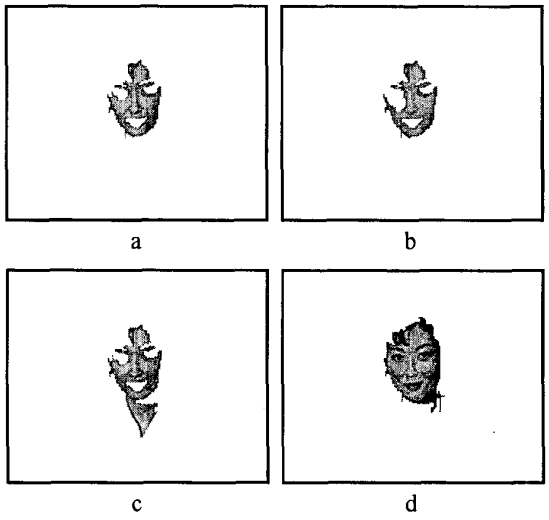

Figure 3. Example of the BPT analysis (a, b, c) and final face region (d) from frame #1 of the *Akiyo* sequence.

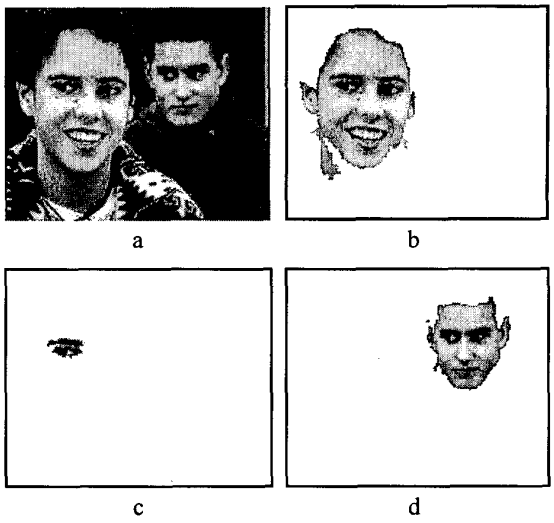

Figure 4. BPT analysis with several faces in the scene. Figure 4.a shows the original image. Figure 4.b and 4.c present two selected candidates, which belong to the same face. Figure 4.d shows the candidate region for the second face.

2.3 Face refinement

The initial estimate of the face may lack of some regions that form the face. The use of a merging process based on a chrominance criterion allows the simplification of the face segmentation procedure. However, it does not ensure that the optimum region (in the sense of the likelihood) is present as a node in the BPT. Nevertheless, once the core components have been detected, a refinement step can be applied to completely extract the face information, without largely increasing the computational load.

This refinement is performed in two steps. First, geometrical information about the type of region that is to be obtained is introduced. This way, a hole filling technique is applied in order to solve the problem of detecting dark, non skin-like areas (typically, eyes and mouth).

The second step is based on the same merging algorithm used for obtaining the core face component. Nevertheless, here the BPT is not used and the analysis is constrained to merging the detected face component with its neighbor regions. The face component usually contains a large area of the face and the necessary scaling for analyzing the different possible mergings can be therefore fixed. This allows for speeding up the algorithm since, in this case, the distances to be computed for the different possible unions can be computed recursively. Figure 3.d shows the final face region yield by the refinement algorithm.

3. FACE TRACKING BASED ON PARTITION PROJECTION

Once the face partition is obtained in the first image, it is used for tracking the face segment through the sequence. The face partition is not directly used for tracking purposes since its regions do not fulfill any fixed motion or spatial homogeneity. Instead, a second partition level is defined by re-segmenting the face partition [5] that guarantees the color homogeneity of each region (the so-called *texture partition*). The face in the current frame is obtained by tracking the texture partition using a partition projection approach [7]. In this application, the leaves of the BPT directly provide the texture partition [9].

3.1 Partition projection

The texture partition of the previous image is projected into the current frame to obtain the texture partition at the current image. The projection is done in two steps. First, the motion between the previous and current images is estimated and the previous texture partition is motion compensated. A backward block-matching algorithm is used in this step. Other motion estimation techniques have been tested, leading to similar results while increasing the computational time. Compensated regions are used as markers giving an estimate of the region positions in the current image. Second, motion compensated markers are fitted to a fine segmentation of the current image which contains its real boundaries. This partition is said to be fine because it contains a large number of regions; typically, more than 1000 for QCIF images. The fine partition is obtained using very strict color homogeneity criteria, which ensure the presence of all contours in the scene.

The fitting process involves geometrical, color and structural information. First, all regions from the fine partition that are totally covered by a motion compensated marker receive its label. This step provides with a first estimate of the position of previous regions in the current image. A second step that merges neighbor regions to those already labeled is then applied. In this step both geometrical as well as color information is used. The color information of the regions that are not totally covered is then analyzed. These regions are labeled if their distance to the compensate markers in the color space is small. The color distance that is used is that described in Section 2:

$$d(x,c)=\sqrt{\gamma\left(y_x-\bar{y}_c\right)^2+\frac{(1-\gamma)}{2}\left(\left(u_x-\bar{u}_c\right)^2+\left(v_x-\bar{v}_c\right)^2\right)}$$

where x represents the region being analyzed and c the compensate marker.

After the two previous steps, some parts of the current image may not have been assigned yet to any region from the previous partition. In the generic object tracking algorithm [5], these areas are labeled using a watershed algorithm. In this application, since the final goal is to obtain the face segment special mechanisms for assigning these areas are used. Regions from the fine partition can be assigned to the face segment if they are largely covered by compensated markers belonging to the face region in the previous image or if they totally surrounded by regions labeled as belonging to the face.

After this step, some uncertainty area may remain; that is, regions from the fine partition may have not been yet assigned. Relying on the classification carried out in the previous image, the current image is divided into three components after the fitting process: area that sure belongs to the face, area that sure does not belong to the face and uncertainty area. An example of this classification is presented in Figure 5.

3.2 Face refinement

A refinement process similar to that of Section 2.3 is used to obtain the final face. The union of each region forming the uncertainty area (see Figure 5.d) with the already detected face area (see Figure 5.b) is analyzed. A region is added when the union increases the likelihood of belonging to a face. This approach improves the generic tracking technique proposed in [5] since it includes semantic information in the tracking process. For instance, possible leakage of the face region to the neck zone is further prevented.

In order to speed up the algorithm, the scaling of the different region unions for computing the likelihood to belonging to a face is carried out only once. This is done by taking as size for the scaling the union of the face and uncertainty area. In the example in Figure 5, the final result is presented in Figure 5.e.

Figure 6 presents some results obtained during the tracking of the face present in the Foreman sequence. The selected frames are those close to time instants of rapid changes in the position of the person in order to assess the capability of the algorithm to handle such situations. Note that, in addition to the robustness in front to rapid changes, the algorithm is able as well to detect non-frontal views of faces as can be seen in Figure 6.a or Figure 6.e. However, the final results still present some problems due to the addition of regions that do not belong to the face to the selected region. Such problems are mainly due to problems on the initial segmentation that does not fully represent the face information as discussed in Section 2. This type of result can be seen in Figure 6.b or Figure 6.e.

4. CONCLUSIONS AND CURRENT WORK

The proposed technique for face segmentation and tracking has been successfully assessed on a large number of test sequences from the MPEG-4 database (raw format) as well as from the MPEG-7 database (MPEG-1 format).

In these experiments, the face class (Ω) has been formed using the Olivetti database, which only contains frontal images. Hence the relevance of being able to track the segmented faces in the case of non-frontal images. This is due to the fact of using a projection that provides, for each frame, a good first estimate of the face position.

The current work is mainly aiming at handling the case of false detection and, specially, that of no presence of faces in the scene. Since the face segmentation technique always yields that region with the higher probability of belonging to a face, candidate regions are always proposed (and therefore tracked) even in the absence of real faces. This is being solved by combining the results of intra-mode segmentation with those of the tracking. This combination allows the correction of false detection.

REFERENCES

[1] R. Chellappa, C. L. Wilson, S. Sirohey, "Human and machine recognition of faces: a survey", *Proceedings of the IEEE*, Volume 83, No. 5, pp. 705-740, May 1995.

[2] F. Davoine, et al., "On automatic face and facial features detection in video sequences, *International Workshop on Synthetic-Natural Hybrid coding and 3-D imaging*, pp. 196-199, Rhodes, Greece, 1997.

[3] L. Garrido, P. Salembier and D. García, "Extensive operators in partition lattices for image sequence analysis", *EURASIP Signal Processing*, Vol. 66, No. 2, pp. 157-180, April 1998.

[4] B. Moghaddam and A. Pentland, "Probabilistic visual learning for object representation", *IEEE Trans. PAMI*, Vol. 19, No. 7, pp. 696-710, July 1997.

[5] F. Marqués, J. Llach, "Tracking of generic objects for Video Object generation", *Proceedings of the ICIP 98, IEEE International Conference on Image Processing*, Chicago, 1998.

[6] P. Salembier and L. Garrido, "Binary partition tree as an efficient representation for filtering, segmentation and information retrieval", In *ICIP-98*, Chicago, 1998.

[7] F. Marqués, M. Pardàs, P. Salembier, "Coding-oriented segmentation of video sequences", *Video Coding: The second generation approach*, L. Torres and M. Kunt (Eds.), Kluwer Academic Publishers, pp. 79-124, 1996.

[8] V. Vilaplana, F. Marqués, "Face segmentation using connected operators", In *Mathematical Morphology and its Applications to Image and Signal Processing*, pp. 207-214, Amsterdam, 1998.

[9] V. Vilaplana, F. Marqués, P. Salembier, L. Garrido, "Region-based segmentation and tracking of human faces", *EUSIPCO-98*, volume I, pp 311-315, Rhodes, 1998.

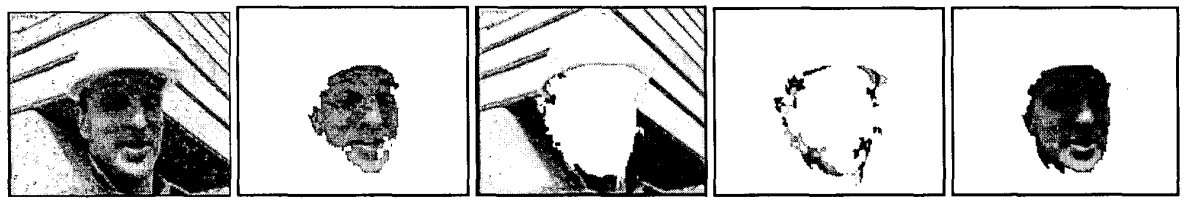

Figure 5. Example of tracking step of the face in the *Foreman* sequence. The images show: (a) the original frame #1 of the sequence, (b) the area selected as belonging to the face in the projection step, (c) the area selecting as non belonging to the face, (d) the regions dividing the area defined as possible face, (e) the final segmented face.

Figure 6. Example of the tracking results in the *Foreman* sequence: frames #15, #40, #89, #125 and #148 are presented.