FACE DETECTION AND SEGMENTATION ON A HIERARCHICAL IMAGE REPRESENTATION

Veronica Vilaplana, Ferran Marques

Technical University of Catalonia (UPC), Barcelona, Spain
{veronica, ferran}@gps.tsc.upc.edu

ABSTRACT

This paper presents a face segmentation technique that works on a hierarchical region-based representation of the image. The algorithm bases its analysis strategy on a reduced set of regions that represent the image content at different scales of resolution. For each region, a set of simple one-class classifiers that rely on different shape, color and texture attributes is evaluated. The outputs of the classifiers are combined into a final face likelihood. The proposed system has been tested on a large set of images, providing very good results both in detection rate and accuracy of the segmented faces.

1. INTRODUCTION

Face detection is a key issue for many applications [10]. In some applications, in addition to detecting the faces present in the scene, their accurate segmentation is useful: it may improve the algorithm performance (e.g. person recognition) or may even be required (e.g. 3D face modeling or color enhancement in the facial region).

Face detection is a classification problem [3]. In order to find face instances in an image, different areas or patterns in the image are analyzed. For each pattern, features are measured and used to decide whether the pattern is an instance of a face or not.

In any face detection system we can identify three basic elements: a description of the object being searched (Face Model), a procedure that simplifies the search by selecting some parts of the image where the search is to be performed (Selection of Candidates) and, finally, a mechanism to decide which of the candidates are face instances (Classification). The scheme of a general detector is depicted in Figure 1.

The Face Model is, in pattern recognition terms, the training or design cycle of a classifier [3]. It comprises the selection of a set of representative features, the proposal of a classifier model and the training and testing of the classifier. The cycle may be iterated, changing the features, the parameters or even the classifier model until a satisfactory solution is reached. The Selection of Candidates gathers the preprocessing (e.g. noise removal, pattern normalization, etc.) and the feature measurement on input patterns, whereas in the Classification block, the trained classifier assigns a label to the input candidate, based on the measured features.

In addition to the previous elements, we take into account other two fundamental parts in the system: the Image Model and the Application blocks. Since we are working with images, it is necessary to choose an image representation and to define the patterns in the image that will be analyzed (e.g. pixels, blocks or regions). This is the task of the Image Model block. The Application block makes explicit the idea that the use of a particular image model as well as the description of the face class are application-dependent issues. Usually a given face model is only valid in a particular context. Low or high resolution images, indoor or outdoor environments, simple or cluttered backgrounds are very different scenarios with varying complexities. They may require different strategies concerning the image model or the face model.

Most face detection algorithms work with a block-based image model, using a rectangular representation of patterns [10]. They scan the image at multiple scales and locations performing the analysis on each image sub-window. Among them, two of the state-of-the-art face detectors are [12] and [8], which are very fast and robust systems. The outputs of these detectors are rectangular areas around the main facial features, which may contain part of the background or may lack some skin areas. Therefore, to obtain the actual shape of the face, further processing is required.

In this paper we propose a face segmentation system that works on a hierarchical region-based image model. Regions are useful for several reasons: (i) they allow a first simplification of the analysis in terms of position and scale (the number of regions is lower than the number of blocks), (ii) they enable a robust feature estimation on image supports that are homogeneous, and (iii) face detection and segmentation are jointly performed; that is, we separate face from background and obtain the real contours of the face.

Figure 2 shows our approach to the problem. Details are given in the following sections. The image model is a hierarchical representation based on regions. We start building an initial partition of the image into regions that are homogeneous in color. Next, we propose, from this partition, a set of regions to analyze, through a hierarchy of regions (a binary tree). Concerning the face model, the system is built as a combination of one-class classifiers [9] based on visual attributes which are measured on the regions by a set of descriptors. The descriptors are organized in three groups (basic, shape and specific descriptors), according to their use in the other parts of the system. The selection of candidates analyzes the regions proposed by the tree, and performs a first simplification by discarding some of the regions and redefining the area of support of the remaining ones, if necessary. Then, a set of specific descriptors are computed and the features are input to the classifier for the final decision.

The system is a combination of classifiers, with one classifier for each descriptor, resulting in a very flexible structure. Therefore, it is very easy to select, for each application, the most useful set of descriptors and even the best combination for the classifiers.

This work has been partly supported by the EU project NoE MUSCLE FP6-507752, the grant TEC2004-01914 and the CENIT project HESPERIA of the Spanish Government.
2. IMAGE MODEL

For an accurate segmentation of the facial regions, we work with the Binary Partition Tree (BPT) \[7\]. The BPT proposes a hierarchy in terms of regions. The idea is to define a reduced set of regions at various scales of resolution that are representative of the perceptual image content. Instead of looking at all possible pixel locations and all possible scales, the detection algorithm bases its analysis strategy on this reduced set of regions.

The BPT representation is built using a region merging algorithm. Starting from an initial partition (that can be as fine as assuming each pixel is a region), the region merging algorithm proceeds iteratively by (1) computing a similarity measure (merging criterion) 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.

Since the criterion used to build the tree is generic (typically color homogeneity), it is unlikely that complex objects such as faces appear as single nodes in the tree. However, a node in the BPT can be a good estimate of an object, giving information about the position and scale at which the object can be found. To obtain a complete representation of the object the notion of extended node is introduced in Section 4.1. The node extension makes use of a fine partition (much finer than the initial partition) to ensure a sufficient accuracy in the representation. An example of the fine partition for the image in Figure 3 (a) is shown in Figure 3 (c). Therefore, the proposed image model is formed by a fine partition, the initial partition and the BPT. The complete hierarchical representation is illustrated in Figure 4. The precise procedure to build the region-based representation is detailed in [11].

Once the BPT has been created, the remaining steps of the system will directly work on its nodes. The number of nodes is much lower than the number of pixels. Moreover, the set of nodes spans all possible scales in terms of regions.

3. FACE MODEL

The face model is a characterization of the face class in terms of a set of visual descriptors. Taking into account the way we use the descriptors for the selection of candidates and for the final classification, we organize them into three groups: generic descriptors, a shape descriptor and specific descriptors.

Generic descriptors: They are descriptors associated to low-level (basic) visual attributes. Generic descriptors measure attributes which are common to any object. They are simple and relatively easy to measure, that is, with a low computational cost. Generic descriptors, which are measured on the regions associated to each node in the BPT, are used to simplify the search (see Subsection 4.1). We work with the following generic descriptors:

- **Color mean:** the mean value of the pixels within the region in each color component in the YCbCr color space.
- **Aspect Ratio and Oriented Aspect Ratio:** which measure the aspect ratio of the bounding box and the aspect ratio of the oriented bounding box of the region, respectively.
- **Complexity:** this descriptor is defined specifically for the BPT structure and measures the complexity of a node (region) in terms of the accumulated cost of the mergings performed to obtain the node.
- **Size:** the number of pixels that form the region.
- **Orientation:** which is computed using the central moments of the region.
- **Position:** the mass center of the region.

The last three features do not describe intrinsic attributes of faces but may be useful to guide the search in a particular scenario, since they measure properties related to size, position and orientation of the region in the image. Constraints on the values of these descriptors may be imposed by the application.

Shape descriptor: For the case of face detection, a simple shape description based on modeling the face as an ellipse is sufficient. We work with an elliptical mask that describes the general shape of a face, and allows to adapt the area of support of the regions proposed by the nodes in the tree to the shape of the object being searched. As will be discussed in Subsection 4.2 we use the nodes in the tree as markers (anchor points in terms of position an size) of possible face instances. The final face candidates are obtained using the markers and the shape descriptor.
Specific descriptors: They are descriptors related to attributes that are specific to faces. Generally, the selection of these attributes implies further knowledge about faces. They are usually more complex and costly to compute than generic descriptors but they will be evaluated on a very reduced set of regions. We work with the following set of specific descriptors:

- **Dominant Color**: the dominant color descriptor provides a compact and accurate description of the representative colors in a region. The descriptor for a region \( R \) is: 
  \[
  d_{dc}(R) = \left\{ (c_i, p_i) \right\}_{i=1}^{N},
  \]
  where \( N \) is the number of dominant colors. Each dominant color is given by a vector \( c_i \), and \( p_i \) is the fraction of pixels in the region corresponding to color \( c_i \), and \( \sum p_i = 1 \). We work with \( N = 8 \) colors.

- **Aspect Ratio and Oriented Aspect Ratio**: a function which is typically trained using sample data. Therefore, for each following set of specific descriptors:

  - **PCA distance in the feature space (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 and the face class is the Mahalanobis distance in the subspace (the distance between \( x \) and the sample mean \( \bar{x} \)).

  \[
  d_{difs}(x) = \sum_{i=1}^{M} \frac{y_i}{\lambda_i}
  \]
  where \( y_i \) is the projection of the mean normalized vector \( x - \bar{x} \) on the \( i \)-eigenvector and \( \lambda_i \) is the \( i \)-eigenvalue.

- **PCA distance from the feature space (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.

  \[
  d_{dffs}(x) = \|x - \bar{x}\|^2 - \sum_{i=1}^{M} y_i
  \]

Each descriptor is associated with a simple one-class classifier which is typically trained using sample data. Therefore, for each visual descriptor, a function \( f \) is 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 a face instance. The function \( f \) may also estimate a distance or resemblance of the region to the class. The learning of \( f \) can be approached with different techniques: density methods, boundary methods or reconstruction methods (see [9]).

Descriptors are then combined using likelihoods. For descriptors that provide distances \( d_{dffs}(x) \) instead of probabilities, the distances must first be transformed into likelihoods. This transformation can be performed by fitting sample descriptor values to some distribution or by mappings like 
\[
  \tilde{f}(x/a) = \frac{1}{K} \exp(-d_{dffs}(x)/\sigma_2),
\]
which models a Gaussian distribution around the model if \( d_{dffs}(x) \) is a squared Euclidean distance.

In this work, density methods are used to learn the feature models for color, Hauendorff distance, aspect ratios, symmetry and dffs 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 dffs and dffs. To characterize face color, the skin color distribution is modeled using a Gaussian distribution in the \( \{C_{p}, C_{s}\} \) space, and the most dominant color of each candidate is used to compute the likelihood. The distances defined by dffs and dffs are transformed into likelihoods by fitting them to the face class distributions obtained through the training data (a \( \chi^2 \) distribution for the dffs and a gamma distribution for the dffs).

Once the output values have been normalized to likelihoods, they are combined into a global likelihood. If the face model is defined by \( K \) descriptors, with decision functions \( f_k \), for \( k = 1, \ldots, K \), when a candidate region is evaluated \( K \) measurement vectors \( s_k \), \( k = 1, \ldots, K \) are obtained. Let \( x = (x_1, \ldots, x_K) \) be the set of measurements for the candidate. Typical combination rules are the weighted average \( f_{av}(x) = \sum_{k=1}^{K} w_k f_k(x_k) \), where \( \sum_{k=1}^{K} w_k = 1 \) and the product of likelihoods \( f_{prod}(x) = \prod_{k=1}^{K} f_k(x_k) \).

For the experiments presented in this work (see Section [6]), descriptors are combined by a product combination of estimated likelihoods.

4. **SELECTION OF CANDIDATES**

This part of the system performs two tasks: (i) a simplification of the search, by reducing the initial set of patterns (the nodes of the BPT) and (ii) the redefinition of the region of support of the remaining nodes.

The first step aims to eliminate as many nodes as possible by a fast analysis of the regions associated to the nodes. This is the **BPT filtering**. The goal is to analyze the patterns employing a set of generic descriptors computed for each node, eliminating those nodes that significantly differ from the characterization proposed by the face model.

The second step is the precise definition of the region of support for the remaining patterns. The objective is twofold: first, to ensure the required accuracy in the face representation by using the fine partition and, second, to solve possible problems derived from the segmentation process or from the BPT creation. This is the **node extension step**.

4.1 **BPT filtering**

At this stage, each node is analyzed to decide whether it may be part of an object instance. For all the nodes in the BPT, the set of generic descriptors defined in the face model is computed, and their values are compared with the characterization proposed by the face model.

Nodes not satisfying the criteria defined in the model are discarded. We call **active node** a node that is not discarded in this step. Therefore, generic descriptors allow a further simplification of the search space. Note that it is a hard selection step, where we eliminate patterns that are very different from faces in terms of the descriptors proposed by the model. Typically, patterns whose color is very different from skin color, which are too small...
to be analyzed or whose aspect ratio largely differ from that of a normal face are rejected.

Following the example presented in Figure 3 Figure 5 shows the result of the BPT filtering with mean color, size, aspect ratio and complexity descriptors. Square solid nodes represent the active nodes. The tree has 199 nodes, out of which 169 are filtered whereas 30 are still active (color filters 91 nodes, size 112, aspect ratio 86 and complexity 6).

Figure 5: Selection of candidates: simplification

4.2 Node extension

Shape information is used to improve the representation provided by the tree nodes. This is done by fitting a shape model of the object to the node, and then using this shape to modify the area of support of the node, by adding or removing small regions from the initial set of regions that define the node.

The shape fitting is performed with a shape matching technique based on distance transforms. If \( N \) is a binary image where zero-valued pixels represent the contour points of a node (of the region associated to a node), a distance transform on \( N \), \( DT_N \), is an image where each pixel value denotes the distance to the nearest contour point in \( N \). The matching is performed by transforming the reference shape model by a set of allowed geometrical transformations (typically translation, rotation and scale) and correlating the transformed template against the \( DT_N \) image. The similarity measure between the transformed shape and the node is:

\[
D(S,N) = \frac{1}{N_S} \sum_{x \in S} DT_N(x)
\]

(6)

where \( S \) is the contour of the transformed template \( \hat{S} \) and \( N_S \) is the number of contour points in \( S \). The goal is to search for the best matching, that is the parameters of the transformation that minimize the similarity measure \( D(S,N) \).

The advantage of matching with distance transforms is their ability to handle noisy or imperfect data. In our case, it allows filling gaps in the contour definition due to errors in the segmentation. Different distance transforms can be used. We work with binary distances obtained by dilating the contours in \( N \) (points that are within the dilated area around the contours receive a distance value of 1, whereas the other points have a distance value of 0). Binary distances allow partial matchings when the node is an incomplete representation of the object. In our work, the reference shape model is an ellipse. The allowed transformations for the reference shape are translation, rotation and scaling, and the set of parameters is calculated for each node, taking into account its width, height and position in the image.

An example of shape matching is presented in Figure 6. The best representation of the face in the BPT is the node shown in (b). The reference contour which is used as model for the human face shape is an ellipse. Figure (d) shows the result of the shape matching using a binary distance transform. Note that there is a region in the node (see the initial partition in (c)) that covers part of the face and the hair, and there is a missing region (the right eye). In spite of that, the reference contour is correctly located around the face.

The next step, the redefinition of the support area of the node, can be performed in many ways. One simple solution that uses only shape information consists in analyzing the degree of overlapping between the regions from the initial partition and the fitted face shape model \( \hat{S} \). Regions in the initial partition that are completely included or partially included but that do not grow too far from the shape \( S \) are included in the extended node. Regions that do not overlap \( \hat{S} \) are not included. For the remaining regions, the analysis is performed in terms of the fine partition, to improve the accuracy of the representation: fine partition regions that (do not) overlap with the shape are (removed from) included in the extended node. An example of this second step is shown in Figure 6 (e).

Figure 6: Node extension. Original image (a), best representation of the face in the BPT (b), initial partition (c), shape matching with an elliptical shape model (d) and extended node (e).

5. CLASSIFICATION

Here, candidates are evaluated and assigned a face or no-face label. Each specific descriptor is associated to a simple classifier that outputs, for each candidate, its likelihood of being a face instance. The outputs are combined into a global probability or face likelihood. Finally, the most likely candidates are selected as face instances.

For the previous example, Figure 7 shows the two candidates classified as faces.

Figure 7: Candidates classified as faces

6. RESULTS

In order to assess the quality of the proposed technique for both detection and segmentation, three different databases are used.

- XM2VTS images: 100 images from the XM2VTS database [5] have been selected and detection performance figures are obtained on this data set. This set has been manually segmented, generating 100 face objects, which are used as ground truth to analyze the accuracy of the proposed face segmentation.
- MPEG images: 100 images from the MPEG database have been selected and detection performance figures are obtained on this data set. This set has been manually segmented, generating 112 face objects, which are used as ground truth to analyze the accuracy of the proposed face segmentation.
- Mobile phone images: 136 images obtained with mobile phones in different complexity scenarios have been used for assessing the detection performance.

In Table 1 the results of the filtering step (see Section 4.1) are analyzed on the three databases. In it, the percentages of active (Act) and non-active (No Act) nodes after the filtering step are presented. As it can be seen, the use of generic descriptors largely reduces the amount of nodes to be further analyzed. Moreover, the percentage of nodes that are filtered by the different criteria are presented. Note that a node can be filtered by more than a criterion,
as it is typically the case for the color, size and aspect ratio criteria. The effect of the complexity criterion is different since it mainly acts on nodes in the top part of the tree (nodes of high complexity). The effect of the complexity criterion is different since it mainly acts on nodes in the top part of the tree (nodes of high complexity).

Table 1: BPT filtering results

<table>
<thead>
<tr>
<th></th>
<th>AcS</th>
<th>No AcS</th>
<th>Color</th>
<th>Size</th>
<th>AR</th>
<th>Comp</th>
</tr>
</thead>
<tbody>
<tr>
<td>XM2VTS</td>
<td>7.27</td>
<td>92.73</td>
<td>47.30</td>
<td>66.82</td>
<td>56.13</td>
<td>1.08</td>
</tr>
<tr>
<td>MPEG</td>
<td>16.03</td>
<td>87.97</td>
<td>35.17</td>
<td>52.58</td>
<td>27.34</td>
<td>1.47</td>
</tr>
</tbody>
</table>

Table 2: Detection and segmentation results

<table>
<thead>
<tr>
<th></th>
<th>Det</th>
<th>DE</th>
<th>Seg</th>
<th>SE</th>
<th>AcD</th>
<th>AcS</th>
<th>AcSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>XM2VTS</td>
<td>99.0</td>
<td>2.0</td>
<td>73.5</td>
<td>26.5</td>
<td>0.78</td>
<td>0.85</td>
<td>0.63</td>
</tr>
<tr>
<td>Mobile</td>
<td>91.2</td>
<td>8.8</td>
<td>85.5</td>
<td>14.5</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>MPEG</td>
<td>84.8</td>
<td>15.2</td>
<td>89.5</td>
<td>10.5</td>
<td>0.73</td>
<td>0.82</td>
<td>0.54</td>
</tr>
</tbody>
</table>

Figure 8: Examples of the (a) XM2VTS, (b) Mobile phone and (c) MPEG7 databases and associated segmentation results

Figure 9: Example of segmentation error

REFERENCES