X Congresso Brasileiro de Inteligência Computacional (CBIC'2011), 8 a 11 de Novembro de 2011, Fortaleza, Ceará

FACE SEGMENTATION BASED ON TEXTURE CLASSIFICATION AND HEURISTIC

V. R. S. Laboreiro, H. B. Chrisóstomo, T. P. de Araujo, J. Everardo Bessa Maia
State University of Ceará - Ceará - Brasil
{victor.laboreiro,bc.heitor,thelmo.dearaujo}@gmail.com, jmaia@uece.br

Abstract – Facial image segmentation is treated as a texture classification problem. We use supervised learning to classify image pixels into C classes, each class corresponding to a face part to be segmented. Four types of features are used: first order gray level parameters (2 features), textural measures (4 features), multi-scale features (1 feature) and moment invariant features (2 features), totaling nine features. A Radial Basis Function Neural Network (RBFNN) is used to classify image pixels into 11 regions (or classes): right and left eyes, right and left eyebrows, nose, mouth, right and left ears, face, hair, and background. Satisfactory qualitative and quantitative results were obtained and presented.

Keywords – Face segmentation, sum and difference histogram, radial basis function neural network, heuristic.

1. INTRODUCTION

Image segmentation is a central task for image analysis and pattern recognition [1]. It is a process of partitioning an image into multiple regions that are homogeneous with respect to one or more characteristics. Grayscale image segmentation is a task even more difficult since color information can not be used, but only the luminance information. As is known, color is one of the most significant low-level features that can be used to extract homogeneous regions that are most of the time related to objects or part of objects [2].

Most of the image segmentation methods can be broadly grouped into boundary-based or edge-based techniques, region based techniques and hybrid methods. They all rely on two basic pixel neighborhood properties: discontinuity and similarity. Pixel discontinuity gives rise to boundary-based methods, whereas pixel similarity gives rise to region-based methods. As from the viewpoint of clustering, these methods can be divided into supervised or unsupervised texture segmentation. In supervised algorithms, the object extraction procedure is controlled by human intervention, while unsupervised algorithms try to automate the decision about the number of object regions in the image and assign optimal parameter values for the segmentation algorithm. For an additional detailed survey of the different techniques, we direct the reader to the literature [1, 2].

All the above mentioned approaches may be applied to facial image segmentation. Nevertheless, specific knowledge about the application domain is useful to improve algorithm performance. One well-known way to improve facial image segmentation takes advantage of the locations of face elements, their proportions, the distances between them and uses templates to guide the algorithm through its tasks of face localization, segmentation, recognition, and identification. In this work a weak form of template is used in the last step of the algorithm to label the regions of interest.

Our goal in this research is not to extract the shapes of facial features, but to locate regions corresponding to facial features such as eyes, eyebrows, mouth, nose, and ears. Facial image segmentation is treated as a classification problem and solved by supervised machine learning techniques. More specifically, feature vectors are computed for each image pixel and a Radial Basis Function Neural Network (RBFNN) is used to classify the regions of interest. Pixel features consider both discontinuity –like lines and borders– and similarity features with neighbor pixels, therefore characterizing it as a hybrid approach –i.e., a mixture of boundary-based and region-based approaches. Part of the images are manually labeled to train the RBFNN and part to test it.

The rest of the paper is organized as follows: Section 2 presents the segmentation methodology in detail. Section 3 provides data description, illustrates the implementation procedure and analyzes the result obtained by the proposed approach. Finally, Section 4 concludes the work of this paper.

2. METHODOLOGY

2.1 FEATURE EXTRACTION

Rickard [3] analyzes 23 features divided into four categories: first order gray level parameters (3 features), textural measures (5 features), multi-scale features (8 features), and moment invariant features (7 features). From those, nine are considered to be the most relevant ones.

All features, except for the Gaussian derivative filter, are calculated for each pixel (x, y) using a $w \times w$ window W around it. We use $w = 8$.

Two first order gray parameters are computed for each pixel (x, y):

1. The gray level $I(x, y)$.

2. The gray level variance:
$$\sigma_{gl}^2 = \frac{1}{w^2-1} \sum_{(x,y)\in W} (I(x,y) - \mu_{gl})^2 \,,$$
where
$$\mu_{gl} = \frac{1}{w^2} \sum_{(x,y)\in W} I(x,y)$$
is the gray level mean for the window W.

The four next features are the textural ones and are computed by using sum (S) and difference (D) histograms, a statistical approximation to the gray-level co-occurrence matrix developed by Unser [4].

3. Mean, given by: $mean = \frac{1}{2}\sum_i S(i)$.
4. Contrast: $contrast = \sum_j j^2 D(j)$.
5. Homogeneity: $homogeneity = \sum_j \frac{1}{j^2+1} D(j)$.
6. Entropy:
$$-\sum_i S(i)\log S(i) - \sum_j D(j)\log D(j) \,.$$

Since we are using an 8×8 window to produce both histograms, i and j range from 0 to 7. Each histogram uses distance $d = 1$ and angles $\theta = 0°$, $45°$, $90°$, and $135°$ and is normalized by a factor of $(8-1)8 = 56$. The results are then averaged to obtain the sum and the difference histograms.

The multi-scale feature chosen is a squared gradient that uses first order Gaussian derivative filters. It is given by

7. $L_i L_i = L_x^2 + L_y^2$, where
$$L_x = I \otimes \frac{\partial G_\sigma}{\partial x} \quad \text{and} \quad L_y = I \otimes \frac{\partial G_\sigma}{\partial y} \,,$$
$\otimes$ represents the convolution with a 5×5 window, and $\sigma = 1$ is set in the Gaussian kernel.

Introduced by Hu [5] in 1962 for image analysis, moments are invariant under rotation, translation, and scaling. The regular moments of an image are given by
$$\eta_{pq} = \sum_x \sum_y x^p y^q I(x,y) \,,$$
but, to assure translation invariance, one needs to consider the central moments m_{pq}, defined by
$$m_{pq} = \sum_x \sum_y (x-x')^p (y-y')^q I(x,y) \,,$$
where
$$x' = \frac{\eta_{10}}{\eta_{00}} \quad \text{and} \quad y' = \frac{\eta_{01}}{\eta_{00}} \,.$$

For scale invariance, one normalizes the central moments:
$$\mu_{pq} = \frac{m_{pq}}{m_{00}^\gamma} \quad \text{with} \quad \gamma = \frac{p+q}{2} + 1 \,.$$

The moment invariant features chosen are:

8. $\phi_1 = \mu_{20} + \mu_{02}$, and
9. $\phi_3 = (\mu_{30} - 3\mu_{12})^2 + (3\mu_{21} - \mu_{03})^2$.

Recalling that all computations for the moments are done in an 8×8 window.

Using this set of features, Rickard [3] obtained good results on the segmentation of MRI brain images. This work aims to tune this set of features to the facial gray images segmentation. In order to do that, the gray scale face images are represented on the feature space for training a RBFNN.

2.2 RADIAL BASIS FUNCTION NEURAL NETWORK (RBFNN)

The RBFNN is a three-layered network in which the input nodes pass the input values directly to the internal nodes that formulate the hidden layer. The hidden layer performs crisp clustering using Gaussian basis functions at the nodes. The output layer performs a linear combination of the weighted activations (nonlinear responses) from the hidden layer. Therefore, RBFNNs are local approximators [6].

The overall structure in a RBFNN, assuming input dimensionality n, implements nonlinear mappings $f_p : \mathbb{R}^n \rightarrow \mathbb{R}$ that form a finite basis of nonlinear functions. When Gaussian functions are chosen to form the basis functions, the p-th output f_p can be expressed by

$$f_p() = \sum_{j=0}^{L} w_j G_j = \sum_{j=0}^{L} w_j \exp\left\{\frac{-\|-\mathbf{m}_j\|^2}{2\sigma_j^2}\right\}, \quad (1)$$

for $p = 1, \ldots, P$, where $\in \mathbb{R}^n$ is the input vector; w_j is the synaptic weight, with bias $w_0 = \theta$; $\mathbf{m}_j$ is the center and σ_j is the width of each Gaussian function G_j, with $G_0 = -1$; and L is the number of Gaussian functions.

In general, the RBFNN training is divided into two phases and requires designing the hidden layer RBFs and calculating the hidden-to-output layer weights. In order to design the RBF hidden layer in a classification problem, one allocates each RBF neuron to a local region of space associated with each cluster of training samples (localized representation) [6]. In this work, each $\mathbf{m}_j$ is the centroid of the j-th sample class.

Once all L centers are determined, each width parameter, σ_j, of the radial basis functions has to be estimated. The value of this parameter determines the shape of the radial basis functions. One of the simplest ways is to set the σ_j value equal to the Euclidean distance of the j-th center m_j from its nearest neighbor or to take into account the B nearest neighbors [7]. In this case:

$$\sigma_j = \frac{1}{\sqrt{B}} \sum_{k=1}^{B} \|\mathbf{m}_k - \mathbf{m}_j\|^2 .$$

In this work, the values of σ_j for $B = 1$ and $B = 4$ were used as bounds for the experimental determination of σ_j. The best results were obtained for $\sigma_j = 0.15$ and this value was used.

The hidden-to-output layer weights was obtained using the least square method. It was determined by a pseudo inverse or singular value decomposition method [6]. The least square solution for these weights satisfies

$$(A^T A)\omega = A^T \mathbf{b} ,$$

where A is a $N \times (M + 1)$ matrix, with element A_{ij} representing the output of the j-th input; N is the total number of training samples; $\mathbf{b}$ is the desired output vector; and ω is the connection weight vector.

Finally, in order to determine the number of neurons in the hidden layer the following strategy was used: each face image was partitioned in L regions of interest, generating L clusters (hidden classes) with a neuron then allocated to each group. Several values were tried and finally $L = 12$ was used. Thus the RBFNN design is complete. More elaborate RBFNN training techniques could be used, for example k-means followed with successive approximation, but we opted for simplicity and low computational cost.

2.3 POST-PROCESSING HEURISTIC

The post-processing heuristic completes the work of the RBFNN. It starts with a class filter, which consists of attributing to each pixel the majority class in a 9×9 window centered in that pixel. To evaluate the performance of the segmentation, each region is visually located and any point in the region is marked manually. Then, the algorithm calculates the center of mass of each segmented region of interest. In order to do this, the filtered image is scanned one time for each class to find the largest contiguous region in the vicinity of the marked point classified as that class. Once the region is determined, the coordinates of center of mass are computed.

3. RESULTS AND DISCUSSION

To submit the method to more stressful working conditions, it was decided to present the test results when targeting the RBFNN trained with only one image. Then, in order to evaluate our approach for facial image segmentation, ten images from FERET [8] repository were randomly chosen. These images are: 00213-940128-fa, 00248-940128-fb, 00291-940422-fb, 00585-940928-fa, 00567-940928-fa, 00592-940928-fa, 00666-941121-fa, 00659-941121-fa, 00701-941201-fa and 00734-941201-fb, named, respectively, I0 to I9. Since a supervised learning algorithm is used, a ground truth process must be applied. In this paper, this was manually generated. As such, decisions about borderline pixels were subjective and intentionally fuzzy. We believe this will improve future implementations of an interactive procedure for labeling interest regions on face images.

The evaluation procedure consisted on processing all 10 images to obtain 10 files, one for each image, that represent the images on the feature space (the transformed space). Then, the RBFNN was trained using only one image and the resulting model was used to classify the other 9 images.

Qualitative and quantitative results follow. The original color images were converted to gray scale by averaging the R, G, and B pixel values. Fig. 1 shows (bold entries in the tables) qualitative results for two of those images. Note that all 11 interest

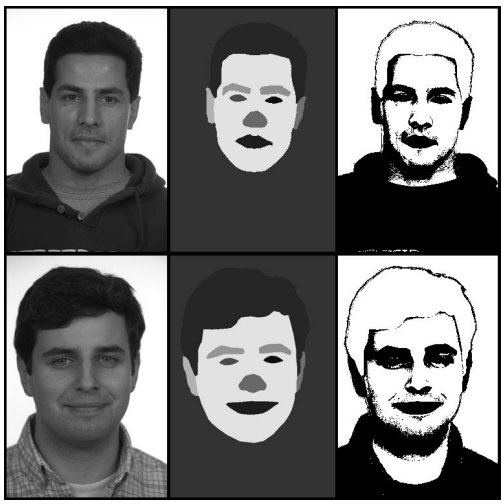

Figura 1: Face segmentation in the I0 (top) and I1 (bottom) images, with RBFN trained with I4 image.

regions are visually segmented from their surroundings: right and left eyes, right and left ears, right and left eyebrows, nose, mouth, cheeks, hair, and background.

Note yet that eyes, mouth, and hair are well segmented in all two images. This performance is met in the other images. Also, eyebrows and nose are just small blurs, but the algorithm got their location right. In general, borders are fuzzy.

Three indexes were used to measure performance: classification accuracy, peak signal-to-noise ratio (PSNR), and mass center deviation (MCD) between the center of mass of segmented regions obtained by the algorithm and the ones in the ground truth images. Both classification accuracy and PSNR are global measures for the algorithm in its first stage (before post-processing). On the other hand, MCD is a final measure of the segmentation process. It indicates how well the algorithm could localize each of the interest regions.

Segmentation accuracy is the ratio between the number of correctly classified pixels and the total number of pixels in the image. It is a crispy measure of classes hits and misses, while PSNR measures the relative error on pixel recovery.

Peak signal-to-noise ratio (PSNR) for two gray-scale images I and K, with dimensions $m \times n$ and 256 gray levels, is defined by:

$$PSNR = 10 \log_{10} \left(\frac{255^2}{MSE} \right),$$

where,

$$MSE = \frac{1}{mn} \sum_{i=1}^{m-1} \sum_{j=1}^{n-1} [I(i,j) - K(i,j)]^2$$

is the mean square error.

As for the MCD index, it is given by the distance (in pixels) between the centers of mass (MC) of the original region and its segmented counterpart respectively. For a region with N points, $MC = (\bar{x}, \bar{y}) = 1/N(\sum_i x_i, \sum_j y_j)$.

Table 1 presents the accuracy in all experiments. Values in the diagonal refer to the classification of the same image that was used to train the RBFNN. Each row refers the image used in the RBFNN training and each column refers to the tested image. Note that in many cases the NN classifies better other images than the one that was used in the training. The mean accuracy was 63,24%.

	I0	I1	I2	I3	I4	I5	I6	I7	I8	I9
I0	0.80	0.63	0.74	0.71	0.81	0.80	0.73	0.70	0.77	0.68
I1	0.58	0.70	0.63	0.70	0.79	0.65	0.28	0.34	0.32	0.42
I2	0.62	0.60	0.71	0.73	0.75	0.75	0.50	0.53	0.62	0.54
I3	0.70	0.62	0.71	0.77	0.83	0.75	0.39	0.46	0.39	0.47
I4	**0.75**	**0.55**	0.70	0.73	0.84	0.73	0.44	0.44	0.35	0.35
I5	0.74	0.55	0.70	0.75	0.78	0.77	0.28	0.39	0.32	0.32
I6	0.62	0.63	0.74	0.74	0.78	0.79	0.73	0.70	0.72	0.64
I7	0.26	0.43	0.39	0.48	0.36	0.45	0.66	0.70	0.65	0.58
I8	0.67	0.61	0.73	0.75	0.77	0.78	0.59	0.60	0.67	0.58
I9	0.75	0.69	0.70	0.66	0.79	0.78	0.70	0.67	0.76	0.72

Tabela 1: Accuracy on classification. Mean = 63,24%

Table 2 shows PSNR results. To compute PSNR, the gray scale values for the image after classification were recovered by setting each pixel on the tested image to the gray scale value of the corresponding region with the nearest (in Euclidean distance) feature vector. Note that the values range from 75.05 to 84.74, with mean value 80.92. Qualitatively, one may note that PSNR values starting from 75 are related to generated images with distinguishable regions. Interestingly, a chi-square test, with 1% significance level, applied to tables accuracy and PSNR indicated their independence, showing the relevance of their data.

	I0	I1	I2	I3	I4	I5	I6	I7	I8	I9
I0	84.0	81.8	83.3	83.5	84.0	84.7	75.3	79.2	77.2	76.6
I1	81.0	82.9	81.9	82.1	84.4	82.7	80.8	79.6	81.2	80.8
I2	82.8	82.4	83.6	84.0	84.2	84.8	75.2	78.4	76.8	76.0
I3	82.9	81.9	82.5	83.8	84.7	84.3	83.0	82.9	82.3	80.9
I4	**82.6**	**81.7**	82.1	83.8	84.6	84.4	81.8	82.2	82.5	80.7
I5	83.8	82.1	82.5	84.0	84.3	84.8	75.0	78.8	76.8	75.5
I6	77.4	76.6	79.5	80.0	77.3	80.1	83.1	82.6	83.2	80.5
I7	76.4	75.5	76.2	78.1	76.0	78.6	82.1	82.8	82.5	80.6
I8	77.1	76.6	79.3	79.9	77.0	79.9	81.8	81.9	83.3	81.8
I9	75.4	80.0	78.4	79.0	76.9	80.0	81.8	81.8	83.7	83.5

Tabela 2: Peak signal-to-noise ratio (PSNR).

Table 3 presents MCD best values between the mass centers for all experiments. As already mentioned, the algorithm does not automatically identifies regions of interest. To calculate the MCD regions of interest were outlined manually and then the algorithm calculated their centers of mass. Values range from 2.23 to 16.40, with mean 9.33. In this case, the chi-square test on Tables 2 and 3, with 1% significance level, rejected the independence hypothesis, and one may conclude there is a strong correlation between the accuracy and the quality of the final segmentation. The table values are $MCD = \sqrt{dx^2 + dy^2}$, where dx and dx are, respectivelly, the deviation on row end column.

	I0	I1	I2	I3	I4	I5	I6	I7	I8	I9
I0	11.4	2.2	13.1	10.0	15.2	12.0	2.2	14.4	13.4	10.0
I1	5.0	13.0	11.4	4.4	12.5	5.8	6.4	10.6	9.48	9.8
I2	9.2	10.7	11.1	13.0	6.0	11.7	15.5	7.0	5.8	15.6
I3	11.4	9.8	7.2	9.4	9.2	7.61	14.7	9.4	13.9	10.6
I4	**13.6**	**8.6**	10.0	5.6	13.8	7.8	7.8	7.0	5.0	8.9
I5	10.8	9.2	12.2	5.3	15.8	5.6	7.2	14.7	13.4	13.3
I6	5.3	6.4	7.0	9.4	6.3	6.0	9.8	4.1	9.2	12.2
I7	12.0	10.1	13.4	9.8	11.4	10.1	12.6	13.6	11.1	13.6
I8	11.7	16.4	10.6	8.4	13.6	5.8	9.0	12.5	12.0	13.4
I9	13.1	13.0	4.4	9.8	5.0	3.1	9.2	10.7	12.2	11.4

Tabela 3: Mean mass center deviation (MCD).

4. CONCLUSION

This work presented the results of segmentation experiments on frontal facial images using a RBFNN and a low computational cost set of features. Nine features were used to segment eleven interest regions in the human face. Qualitative and quantitative performance results for the used algorithm were presented and considered satisfactory, encouraging us to proceed further on this line of reasoning.

We are currently working on improving the algorithm, mainly in three directions: first, by adding new features (such as Gabor wavelets) and re-selecting them. Second, by testing other classifiers, besides RBFNN. And, finally, by improving the post-processing algorithm using morphological processing techniques.

REFERÊNCIAS

[1] W. Zhao, R. Chellappa, P. Phillips and A. Rosenfeld. "Face recognition: a literature survey". *ACM Comput. Surv.*, vol. 35(4), pp. 399–486, 2003.

[2] J. Freixenet, X. Munoz, D. Raba, J. Marti and X. Cuff. "Yet Another Survey on Image Segmentation: Region and Boundary Information Integration". *Proc. European Conf. Computer Vision*, pp. 408–422, 2002.

[3] H. E. Rickard. "Feature Selection for Self-Organizing Feature Map Neural Networks with applications in medical image segmentation". *B.S.E.E., University of Louisville*, 2001.

[4] M. Unser. "Sum and difference histograms for texture classification". *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 8, pp. 118–125, 1986.

[5] M. Hu. "Visual pattern recognition by moment invariants". *IRE Trans. Inf. Theor.*, vol. IT-8, pp. 179187, 1962.

[6] S. Haykin. "Neural Networks: A Comprehensive Foundation, 2e". *Prentice Hall*, 2004.

[7] J. Moody and C. Darken. "Fast learning in networks of locally-tuned processing units". *Neural Comput.*, vol. 1, pp. 281–294, 1989.

[8] "National Institute of Standars and Technology, the Color FERET database". *http://www.itl.nist.gov/iad/humanid/feret/*.