

# Improving Spectroface using Pre-processing and Voting

Ricardo Santos

Dept. Informatics, University of Beira Interior,  
Portugal

Email: ricardo\_psantos@hotmail.com

Luís A. Alexandre

Dept. Informatics, University of Beira Interior and  
Instituto de Telecomunicações  
Covilhã, Portugal

**Abstract**—In this paper we present pre-processing steps and a voting scheme that improve the effectiveness of the spectroface approach. It consists on a series of pre-processing steps prior to spectroface together with a texture feature that are used independently. The classifier output for each of the 13 features is fused using a majority voting scheme coupled with rules for ties and strong features. Yale (15 subjects), Olivetti (40 subjects) and Notre Dame (487 subjects) face databases are selected to evaluate the proposed method yielding 97.33, 85.28 and 75.91% accuracy, respectively, using only one training image per subject.

## I. INTRODUCTION

Modern society has high mobility booth for humans and for information. This forced entities to increase security and deploy methods like identification and authentication, entrance control in buildings, access control, etc.

Most classic control methods require the user to be cooperative or remember a password or PIN, to have a card or some kind human action in the process of identification or authentication. Fingerprints are viable and non-intrusive, but force subjects to cooperate with the system, and so are not applicable to non-consenting people. Iris and retina are also viable, but retina scans are also very intrusive and force people to be highly cooperative. Iris recognition is accurate, but expensive to implement.

Nowadays face recognition has reached an identification rate greater than 90% for larger databases with well controlled pose and illumination conditions. Even if its not as accurate has finger, iris or retina scans, it can be used to recognize subjects with little or no cooperation, and this makes face recognition more appealing.

So, face recognition presents a good compromise between what is socially acceptable and balances security and privacy well. The required data can be easily obtained and readily available and can even be used without the subjects knowledge, which are other advantages of face recognition.

## II. SPECTROFACE

Holistic approaches or face-based approaches try to capture and define a face as a whole: a two-dimensional pattern with intensity variations. Under this family of approaches is Spectroface. Spectroface is a holistic face representation method using wavelet transform and Fourier transform. The method was proposed in [1] and evaluated in several articles as [2], [3]. Like ICA [4] and Eigenfaces [5], [6], Spectroface is a reference for holistic approaches.

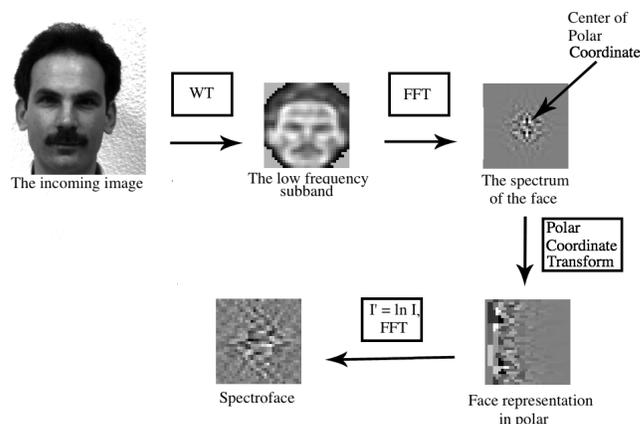


Figure 1. Spectroface general scheme.

The method combines the wavelet transform and the Fourier transform. The faces are decomposed using wavelet transform to filter high frequencies and keep low frequencies that are less sensitive to facial expressions. It also reduces noise such as illumination artifacts and, at the same time, reduces image resolution and therefore computational load.

In [2], Spectroface is matched against other holistic methods, Standard Eigenface [5], [6], in five face recognition problems - 3D pose, facial expressions, nonuniform illumination, translation, and scaling. It is shown that the Spectroface method outperforms the Eigenface method in the 3D pose, facial expressions, nonuniform illumination and translation problem. In the scaling problem the difference is minimal.

The computational load of Spectroface can be considered low,  $O(d \cdot \log d)$ , where  $d$  is the resolution of the image (number of pixels), as shown in [1].

The general scheme of Spectroface is as follows and is resumed in Fig. 1. First the low-frequency band is extracted from the image using a wavelet transform. Then, a Fourier transform and a transform from the Cartesian coordinate system to Polar coordinate system, are applied. Finally a  $\log()$  function is applied to the image. The result is stored in form of vector to be matched against new subjects by a simple Euclidean distance measure.

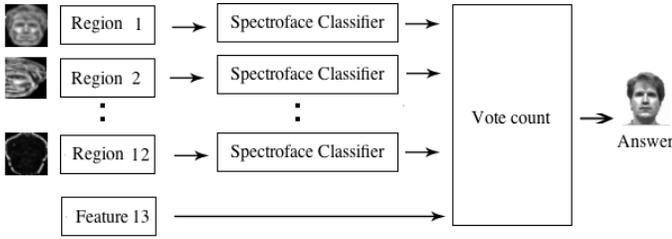


Figure 2. General scheme of this method.

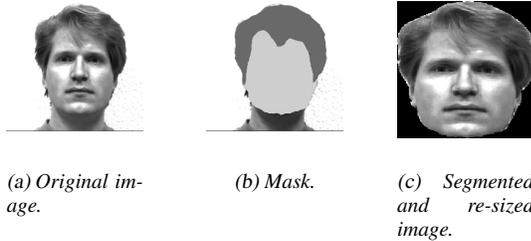


Figure 3. Example of a segmented face after applying a mask.

### III. PRE-PROCESSING

In this work we present a method to improve the performance in holistic-based approaches with only one image for training, in particular to Spectroface. If we have only one training image there isn't much to do in the classifier since each class is represented by only one point in feature space, so we have to concentrate on the only train image. So we extract from the train image as much information as possible in form of features. These features are then fed into a  $k$ -nearest neighbor classifier and the result is combined using voting.

The general scheme of this approach resumed in Fig. 2. As can be seen, each feature (obtained from a segmented zone or part of the face) will count with one vote after converted to the Spectroface representation. Feature 13 is a texture feature applied to the full face and is going to be discussed further in Sec. IV.

This system assumes, as many other face recognition algorithms [7], [8], [9] do, that the face region has been segmented as show in Fig. 3.

Pre-processing to extract the features include four basic steps:

- 1) Image resize.
- 2) Image normalization by sliding window.
- 3) Filter application.
- 4) Intensity normalization.

#### A. Step 1: Image resize

All images are cropped and re-sized to dimension  $128 \times 128$ . Several other dimensions were tested like  $32 \times 32$ ,  $64 \times 64$ ,  $128 \times 128$  and  $256 \times 256$ . Tests showed that  $128 \times 128$  gave the best performance.

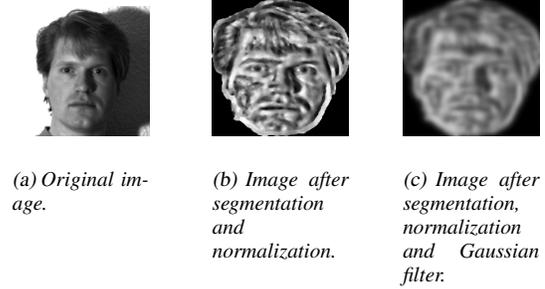


Figure 4. Normalization effect on shadows.

#### B. Step 2: Image normalization by sliding window

This step is crucial, because it smooths strong shades and highlights differences between subjects in the database. Jumping this step will largely reduce system's accuracy as can be seen in table I.

We use a simple type of normalization that turns out to be very effective against shadows, as we can see in Fig. 4a before normalization and Fig.4b after normalization. For each pixel in the column  $j$  and line  $k$  from the image  $I$ ,  $I_{jk}$ , we normalize  $I_{jk}$  taking the maximum and minimum pixel intensity values of window with dimension  $17 \times 17$  surrounding the pixel  $I_{jk}$ . This step highlights face details and enhances shadow zone brightness, as shown in Fig.4. Several pairs of sliding window dimensions with input image dimensions were tested. After testing sliding windows from  $3 \times 3$  up to  $25 \times 25$ , and input images from  $32 \times 32$  up to  $256 \times 256$ , it was concluded that the pairs  $17 \times 17$  for the sliding window with input images with size  $128 \times 128$  or  $256 \times 256$  gave the best results. For higher resolution input images (above  $256 \times 256$ ), resizing the image to  $256 \times 256$  or  $128 \times 128$  is one way to ensure the appropriated sliding windows dimension is  $17 \times 17$ . This normalization is not applied in region 5 and region 6 because what was intended to capture here is shape and contours.

#### C. Step 3: Filter application

Depending on what properties of the image we want to highlight or smooth, a filter is applied. To smooth a Gaussian blur filter is used (regions 1, 10, 11 and 12) or a median blur filter (regions 2, 3, 4, 7, 8 and 9). To highlight some shapes an unsharp filter is used in 2 regions, subtracting a Gaussian blurred image from the original image (region 5) or a median blurred image (region 6). After the unsharp filter, we keep the high frequencies and to make then more visible the image is dilated with a size 9 window.

#### D. Step 4: Intensity normalization

Before the images are delivered to the classifier they are normalized between 0 and 255. This step could be seen as less important, but the final system accuracy decreases if it is not performed.

#### IV. FEATURE DESCRIPTION

Since we use only one image for training and we have the face segmented, we generate the Spectroface representation for several regions of the image. The regions used to produce the features can be seen in Fig. 5 for one example image and are described in more detail in table V.

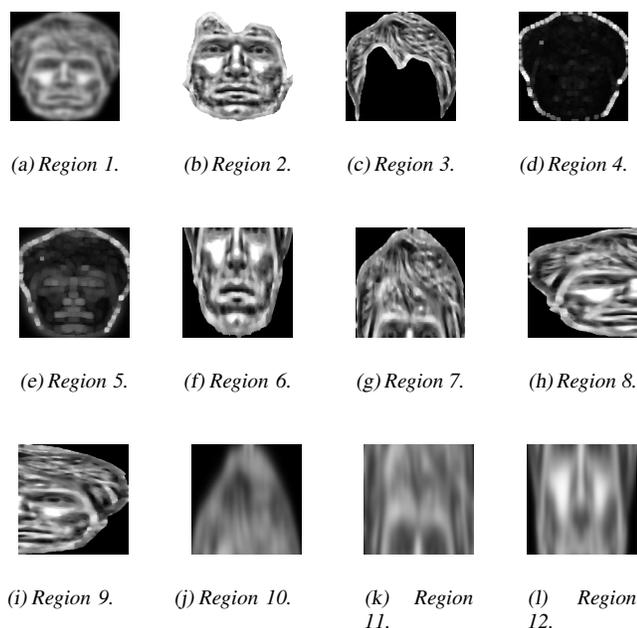


Figure 5. First 12 features used are obtained by applying the spectroface algorithm to the above face regions (shown here after the pre-processing). The last feature is obtained by applying the LBP texture feature algorithm to region 1.

From this table we can see that region 1 is constructed with the segmented face and hair and the non-segmented pixels (not face and not hair) were filled with zero. Then the image was re-sized to  $128 \times 128$  (pre-processing step 1) and latter normalized with a sliding window of size 17 (pre-processing step 2). In the end, a Gaussian blur with window size 5 was applied (pre-processing step 3). Region 4 is constructed like region 1, with difference in pre-processing step 3, in which an unsharp median blur with size 5 is applied and then the image is dilated with a window size 9. The pre-processing done for the remaining regions is likewise described in table V.

In [10] a method is proposed for texture extraction from images. In this case the texture features are estimated using a  $3 \times 3$  neighborhood. The value of each pixel in the neighborhood is compared with the value of the center pixel. The LBP value for this neighborhood is obtained by summing the binomial coefficients shown in figure 6 that correspond to the position of the pixels with value larger than the value of the center pixel. A histogram of these LBP values is built for the image.

A contrast measure can also be used to build a two-dimensional histogram by combining the information of LBP with the contrast information. The contrast is the difference

1	2	4
8		16
32	64	128

Figure 6. LBP binomial mask.

between the average value of the pixels that have a value larger than the center pixel and the average value of the pixels that have smaller value than the center pixel. These values are placed on a histogram together with the LBP values making a bi-dimensional histogram.

In this paper we used 64 bins for the LBP histogram and 8 bins for the contrast. These values were obtained by experimentation. These features were produced on non-overlapping windows of size  $32 \times 27$  (Yale) and  $23 \times 16$  (ORL).

#### V. VOTING SCHEME

Each feature gives one vote on one subject, then the votes are counted and the subject with more votes wins (majority vote).

The first set of rules state that in case of a tie, if feature 1 (the most strong) voted in one of the tie subjects, then that is the choice, else the choice is the first subject that appears in the database and has a maximum vote.

The second set of rules dealing with good accuracy features increase voting if pairs of strong features match the same subject.

There are 3 rules of this type for Yale and ORL database:

- 1) If feature 6 and feature 7 give the same match then the subject vote is increased by 2 votes.
- 2) If feature 8 and feature 9 give the same match then the subject vote is increased by 2 votes.
- 3) If feature 1 and feature 5 give the same match then the subject vote is increased by 2 votes.

The above rules were used in experiment 1 and 2 for Yale and ORL database.

In the case of Notre Dame database (and for all databases in experiment 3) the third rule was changed, and feature 1 vote was replace by feature 2 vote because feature 1 included segmented-face + segmented-hair, and the images used from this database have no hair.

#### VI. EXPERIMENTS

##### A. Databases

To evaluate the proposed system accuracy we selected three standard databases, the Yale University database, the Olivetti Research Laboratory database (ORL) and the University of Notre Dame Face Database - Collection B.

Because there are scenarios like passport verification or identity card verification that could dispose only one image per person we choose to use only one training image per person for each database. Note that this setting is much more challenging than the more traditional one of using 3, 5 or more training set images per subject.



Figure 7. Example of Yale database images.



Figure 8. Example of ORL database images.



Figure 9. Example of Notre Dame database images.

The Yale database contains 165 gray-scale images of 15 subjects. There are 11 images per subject, one per different facial expression or configuration: center-light, with glasses, happy, left-light, with no glasses, normal, right-light, sad, sleepy, surprised, and wink. See Fig. 7.

The Olivetti Face Database contains 400 gray-scale images of 40 subjects. There are 10 images per subject that represent various expressions, occlusions (by glasses), scale and orientation variation up to about 20 degrees. See Fig. 8.

In Yale database we choose the .nor image of each subject (15 images) for train and leave the rest 145 images for test. As for ORL database, we choose the images in Appendix A for train (40 images) and leave the remaining 360 images for test.

The University of Notre Dame Face Database [11] - Collection B has 33,287 frontal face images from 487 human subjects. Each image is captured under different lighting and expression conditions. The number of images per subject varies from 4 to 227, so we used the first image of every subject for training and the next 3 images to test. The images used in this database have already been segmented and without hair (face only). So, all features that were using some kind of feature with hair could not be used, leaving 10 of the 13 original features. So, in Notre Dame database we choose the first subject image for train (487 images) and the following three per subject (1461 images) for test.

## B. Results

There are three different experiments. The first experiment results are based in the voting of the individual features and applying the rules discussed in Sec. IV. We can see the accuracy of each one of the features in Table I.

The second experiment combined the following features by a majority vote: Feature1, Feature2, Feature3, Feature8 and Feature13. We came to this combination by experimentation.

The last experiment was performed to compare the system on the three databases with the same parameters. So in this experiment we used 9 instead of the 12 features, the same nine features used to test the Notre Dame database, and use the second combination of sub-rules as discussed in V.

In Table VI-B we can see the results for the three experiments.

Region number	Yale DB	Yale (*) DB	ORL DB	ORL(*) DB	Notre Dame DB
1	86.0%	68.0%	70.3%	26.1%	Not Def.
2	76.0%	63.3%	55.3%	35.6%	61.9%
3	83.3%	73.3%	49.8%	46.9%	Not Def.
4	81.3%	70.0%	56.4%	26.7%	73.5%
5	80.7%	58.0%	63.6%	12.5%	39.9%
6	78.7%	78.7%	60.0%	60.0%	40.5%
7	70.0%	70.0%	64.2%	64.2%	51.9%
8	91.3%	70.7%	54.8%	31.7%	51.1%
9	94.0%	72.0%	58.6%	26.9%	59.9%
10	82.7%	69.3%	42.2%	15.6%	50.4%
11	65.3%	56.0%	36.7%	25.0%	43.2%
12	55.3%	44.0%	33.9%	16.9%	Not Def.
13	64.0%	Not Tested	83.1%	Not Tested	Not Def.

Table I  
INDIVIDUAL FEATURES ACCURACY. (\*) ACCURACY WITHOUT THE SLIDING WINDOW NORMALIZATION IN PREPROCESSING STEP 2.

Method	Yale DB	ORL DB	Notre Dame DB
Experience 1	97.33%	83.06%	Not Tested
Experience 2	97.33%	85.28%	Not Tested
Experience 3	98.67%	76.11%	75.91%

Table II  
RESULTS OF THE EXPERIENCES: RECOGNITION ACCURACIES.

## C. Comparisons with other existing systems

In [12] a method is proposed based on Spectroface and Singular Value Decomposition. Spectroface is applied on the train image and the result is projected on a uniform eigenspace that is obtained for SVD. The result coefficient vector is stored on the train database for future recognitions.

In [3] the author uses a combining classifier approach for improving recognition performance. A combination of local and global features is proposed using the normalization of the classifier output, selection of classifier(s) for recognition and the weighting of each classifier. Four popular face recognition methods, namely, eigenface, spectroface, independent component analysis (ICA) and Gabor jet are selected for combination. The author uses two basic transforms for scale normalization, LENM : Linear-exponential normalization method and DEGNM : Distribution-weighted Gaussian normalization method.

Table III presents these results and also the ones obtained with our approach. As we can see, all three experiments on Yale outperformed other approaches. In ORL database experiment 2 has the best result with 85.28% of correct

matches. However in ORL database experiment 1 took the second place, loosing by less than 2% for the FR using local and global features LENM [3] approach, but winning in 4% to the same approach in Yale database.

Method	Yale DB	ORL DB
Experiment 1	97.33%	83.06%
Experiment 2	97.33%	<b>85.28%</b>
Experiment 3	<b>98.67%</b>	76.11%
Spectroface [1]	91.33%	81.94%
Septroface and SVD [12]	94.67%	82.50%
FR using local and global features DWGNM[3]	94.16%	82.21%
FR using local and global features LENM[3]	93.33%	85.00%

Table III  
RESULT COMPARISON WITH OTHER PUBLISHED METHODS, ON YALE AND ORL DATABASES.

## VII. CONCLUSIONS

We presented a simple and fast but effective face recognition algorithm and demonstrate its effectiveness in well known databases and with superior performance when compared with other spectroface-based algorithms. This paper successfully combines different features from each individual image.

Normalization has a key role since it deals with shadows and highlights facial features successfully. The other main step is the combination of features in the proper way. There are also interesting findings like the importance of contours feature, and the fact that only the right face or left face alone on Yale database yield matches above 90% and suggesting that if we have symmetry or aligned faces we might only use half of the face and so half of computer load.

The drawback in the Notre Dame database maybe due to the fact that the images were already face-segmented with an elliptic shape and 2 of the features used (numbers four and five) had a strong relationship with the shape of the face as we can see in Fig. 5, and since all the segmented images had the same face shape, much information was lost.

## APPENDIX A

In Table IV we list the name of the images used for training in the ORL database.

<b>Subject</b>	1	2	3	4	5	6	7	8	9	10
<b>Image</b>	5	3	9	7	9	6	3	3	9	8
<b>Subject</b>	11	12	13	14	15	16	17	18	19	20
<b>Image</b>	1	2	10	4	2	2	7	8	4	2
<b>Subject</b>	21	22	23	24	25	26	27	28	29	30
<b>Image</b>	10	1	10	1	3	1	1	5	5	6
<b>Subject</b>	31	32	33	34	35	36	37	38	39	40
<b>Image</b>	4	6	2	6	6	3	2	5	5	8

Table IV  
TRAIN IMAGES FOR ORL DATABASE.

Table V presents a more detailed description of the features used.

Region	Segmtd zone	Non segmented fill	Gauss blur	Median blur	Norm. window size	Dilate size
1	Face+Hair	0	5	-	17	0
2	Face	X(*)	-	5	17	0
3	Hair	0	-	5	17	0
4	Face + Hair + Sharp 1	0	-	5 (**)	17	9
5	Face+Hair + Sharp 2	0	11 (**)	-	17	9
6	Face+Hair +Inf Part	0	-	5	17	0
7	Face+Hair +Sup Part	0	-	5	17	0
8	Face+Hair +Left Part	0	-	5	17	0
9	Face+Hair +Right Part	0	-	5	17	0
10	Face+Hair Strip 1	0	5	-	17	0
11	Face+Hair Strip 2	0	5	-	17	0
12	Face+Hair Strip 3	0	5	-	17	0
13	LBP	ND.	ND.	ND.	ND.	ND.

Table V  
THE PARAMETERS USED TO GENERATE THE FEATURES. (\*) 255 IN YALE AND 0 IN ORL (\*\*) IN THIS CASE USE THE UNSHARP FILTER

## ACKNOWLEDGMENTS

This work was partially supported by the “FCT – Fundação para a Ciência e Tecnologia” project number PTDC/EIA/69106/2006, financed by FCT and FEDER.

## REFERENCES

- [1] “Face recognition using holistic Fourier invariant features,” *Pattern Recognition*, vol. 34, no. 1, pp. 95 – 109, 2001.
- [2] T. I. El-Arief, K. A. Nagaty, and A. S. El-Sayed, “Eigenface vs. spectroface: a comparison on the face recognition problems,” in *SPPR’07: Proceedings of the Fourth conference on IASTED International Conference*. Anaheim, CA, USA: ACTA Press, 2007, pp. 321–327.
- [3] J. Huang, P. C. Yuen, J. H. Lai, and C.-h. Li, “Face recognition using local and global features,” *EURASIP J. Appl. Signal Process.*, vol. 2004, pp. 530–541, 2004.
- [4] M. Bartlett, J. Movellan, and T. Sejnowski, “Face recognition by independent component analysis,” *Neural Networks, IEEE Transactions on*, vol. 13, no. 6, pp. 1450–1464, Nov 2002.
- [5] M. Turk and A. Pentland, “Eigenfaces for recognition,” *J. Cognitive Neuroscience*, vol. 3, no. 1, pp. 71–86, 1991.
- [6] —, “Face recognition using eigenfaces,” Jun 1991, pp. 586–591.
- [7] Y. Yacoob and L. Davis, “Detection and analysis of hair,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 28, no. 7, pp. 1164–1169, July 2006.
- [8] C. Rousset and P. Coulon, “Frequent and color analysis for hair mask segmentation,” in *Image Processing, 2008. ICIP 2008. 15th IEEE International Conference on*, Oct. 2008, pp. 2276–2279.
- [9] K. chih Lee, D. Anguelov, B. Sumengen, and S. Gokturk, “Markov random field models for hair and face segmentation,” in *Automatic Face and Gesture Recognition, 2008. FG ’08. 8th IEEE International Conference on*, Sept. 2008, pp. 1–6.
- [10] T. Ojala and M. Pietikainen, “Unsupervised texture segmentation using feature distributions,” *Pattern Recognition*, vol. 32, pp. 477–486, 1999.
- [11] P. Flynn, K. Bowyer, and J. Phillips, “Assessment of time dependency in face recognition: an initial study,” in *4th International Conference on Audio and Video based Biometric Person Authentication*, 2003, pp. 44–51.
- [12] J.-Z. He, M.-H. Du, S.-W. Pei, and Q. Wan, “Face recognition based on spectroface and uniform eigen-space SVD for one training image per person,” vol. 8, Aug. 2005, pp. 4842–4845 Vol. 8.