DCDIS A Supplement, Advances in Neural Networks, Vol. 14(S1) 232--241

Fusion of Fuzzy FFL-KLT and PCNN Features on the Face Recognition Problem

M. I. Chacón M., P. Rivas P.

Abstract-- The problems related to face recognition has been investigated from different points of view. Artificial Neural Networks, ANN, is not the exception. This paper presents a novel approach for face recognition based on three features, first the Pulse Coupled Neural Network Features; second, the Face Feature Lines, FFL, and third, the Karhunen Loève transformation, KLT. The facial features are inputs used in a neurofuzzy system that involves their fuzzyfication. These fuzzy inputs are fed to a RBF neural network with a variable architecture on the first layer. The membership functions used to fuzzify the facial features are created according to the features automatically. The system performs well with ORL and YALE face databases, reaching recognition rates comparable with the current face recognition systems.

Keywords: Pulse Coupled Neural Network, PCNN, Face Feature Lines, FFL, Karhunen Loève transformation, KLT, Face Recognition, RBF, Neurofuzzy Systems.

I. Introduction

NOWADAYS we live in a world in which security problems are increasing and security systems became important. Biometric systems are a very useful tool for person indentification/recognition. The biometric systems such as fingerprint systems and face recognition systems are well known by the media.

Face recognition is an important task as a biometric tool and can be used in a wide range of applications such as identity authentication, access control, and surveillance. If we compare the face recognition problem to classical pattern recognition problems such as optical character recognition (OCR), face recognition becomes very difficult if the amount of individuals (classes) increases, or if only a few images (samples) per person are available. The changes in facial images present great challenges, and a face recognition system must be robust with respect to the many variabilities of face images such as pose, illumination, and facial expression conditions, also known as PIE.

The authors want to thank to SEP-DGEST for the support of this research under grant 445.05-P and to CONACyT for the support of this research under grant #:193324.

M. I. Chacon, and P. Rivas are with the Digital Signal Processing & Vision Laboratory, Chihuahua Institute of Technology, Ave. Tecnológico # 2909 , C.P. 31310 Chihuahua, Chih. México. (Phone: +(52) 614-413-7474; fax: +(52) 614-413-5187 e-mail: mchacon@itchihuahua.edu.mx, pablorp80@ieee.org).

There are different approaches aimed to solve the face recognition problem. Some of these approaches utilize the DCT transform [1]-[3]. Other authors utilize wavelet techniques for feature extraction [4]-[5]. The subspace approach for face recognition is also well utilized for face representation since this method has shown to be fast and successful in face recognition applications [6]-[13]. There have been explored different approaches for classification, such as artificial neural networks, ANN, and distance based classifiers [14]-[19]; as well as probabilistic approaches [20]. However, the face recognition problem is still an open problem even there are more than 30 years of research in this field. Neural networks needs to be explored for the face recognition problem since new ANN designs and architectures have been proposed. Therefore, more research is needed applying these novel ANN architectures and designs, and to combine them to the existing approaches.

The Pulse Coupled Neural Network, PCNN, is a relative new ANN model with a great potential in the area of image processing. A PCNN, is a model derived from a neural mammal model [21]-[24]. Current research with PCNN documents how the PCNN can be used to perform important image processing task; edge detection, segmentation, feature extraction, and image filtering [22]-[30]. Because of this kind of performance the PCNN is considered a good preprocessing element.

In this paper we consider the PCNN as an illumination-robust facial feature extraction method, which combined with other feature extraction methods like the Karhunen-Loève Transform, KLT and the Hough Transform, HT, presents a very complete facial feature extraction method.

It is well known that the distance-based classifiers are the most widely used classifiers for face recognition applications. This may be due to the easy implementation of the method. Other alternatives as fuzzy approaches for classification have proven to be a good choice for face recognition [31]. However, neurofuzzy systems for face recognition needs more research in combining these systems with novel feature extraction methods.

In this paper we propose a novel neurofuzzy scheme for face recognition. The neurofuzzy system is composed of an RBF neural network with fuzzyfied inputs. The system tuned out to perform better than current face recognition systems, overcoming the illumination problem with the PCNN approach.

This paper is organized as follows: Section II introduces the FFL-KLT method for facial feature extraction. The PCNN model is outlined in Section III. Section IV shows the complete feature extraction method. The neurofuzzy classification method is presented in Section V. In Section VI the stages of the proposed system are described as well as the experiments. The results of the experiments are reported in Section VII. The conclusions are pointed in Section VIII.

II. Face Feature Lines and KLT

From perception studies it is noted that some facial features in the space domain like, nose to mouth distance, or geometric shapes like the eyes to mouth shape, are discriminative features between different human been.

In other related research [31] we have proposed a new spatial feature named face feature lines, FFL. FFL are prominent lines in low resolution face images, and can be extracted using the Hough Transform, HT. These features are important as reported in studies with new born regarding face recognition [32]-[33].

A. Face Feature Vector Generation

The detailed explanation of the FFL method can be found in [31]. But in general, the FFL feature vector can be defined as follows

$$\mathbf{z}_i = [l_{i_1} \quad l_{i_2}] \tag{1}$$

where $\mathbf{z}_i$ is a vector of $i \times 2$ columns and it contains the information of the FFL, l_{i_1} and l_{i_2} represents the information of the beginning and ending of the *i*-th FFL respectively.

B. Karhunen-Loève Transform

The Karhunen-Loève Transform, KLT [34], is also known as principal component analysis. As a difference from PCA, with KLT the input vectors are not zero mean. The KLT transformation matrix $\mathbf{w}$ can be computed obtaining the eigenvectors associated to the eigenvalues of the covariance matrix of the training data. To transform an image to the KLT domain, the original image $I(x, y)$ needs to be transformed to a canonical form (vector column) $\mathbf{i}_{xy} = T_{can}\left(I(x, y)\right)$. Where $T_{can}(\cdot)$ is the operator for canonical transformation. Then we can project the image to the KLT domain using

$$\hat{\mathbf{i}}_{xy} = \mathbf{w}_{KLT}^T \mathbf{i}_{xy} \tag{2}$$

where $\hat{\mathbf{i}}_{xy}$ denotes the transformed image $\mathbf{i}_{xy}$, and $\mathbf{w}_{KLT}^T$ denotes the transposed transformation matrix $\mathbf{w}_{KLT}$.

III. Pulse Coupled Neural Networks for Facial Feature Extraction

In this section we present a novel approach to define face features using the pulsed images generated by a Pulse Coupled Neural Network, PCNN, architecture.

A. The PCNN Model

The Pulse Coupled Neural Network, PCNN, is a model derived from a neural mammal model. The basic model of a neuron element of a PCNN has three main modules: the dendrite tree, the linking and the pulse generator [21]. The dendrite tree includes two special regions of the neuron element, the linking and the feeding. Neighborhood information is incorporated through the linking. The input signal information is obtained through the feeding. The pulse generator module compares the internal activity, linking plus feeding activity, with a dynamic threshold to decide if the neuron element fires or not. Fig.1 illustrates the basic model of the PCNN. The PCNN mathematical definition is given by (3) to (7).

$$F(t) = G_{Feed} e^{-\alpha_F \Delta t} F(t-1) + S + Y(t-1) * W \tag{3}$$

Equation (3) corresponds to the feeding region of the neural element, where G_{Feed} is the feed gain, S is the input image, $\alpha_{F\Delta t}$ is the time constant of the leakage filter of the feeding region, $Y(t)$ is the neuron output at time *t,* and W is the feeding kernel. The outputs $Y(t)$ of the PCNN can be observed as output images called pulsed images of the PCNN.

Equation (4) describes the linking activity. Here G_{Link} is the linking gain, $\alpha_{L\Delta t}$ is the time constant of the leakage filter of the linking region, and M is the linking kernel.

$$L(t) = G_{Link} e^{-\alpha_L \Delta t} L(t-1) + Y(t-1) * M \tag{4}$$

Equation (5) corresponds to the internal activity of the neuron element. The internal activity depends on the linking and feeding activity. In (5) β is the linking coefficient. β defines the amount of modulation of the feeding due to the linking activity.

$$U(t) = F(t)\left[1 + \beta L(t)\right] \tag{5}$$

The dynamic threshold is implemented by (6), where α_θ is the time constant of the leakage filter of the threshold and V is the threshold gain.

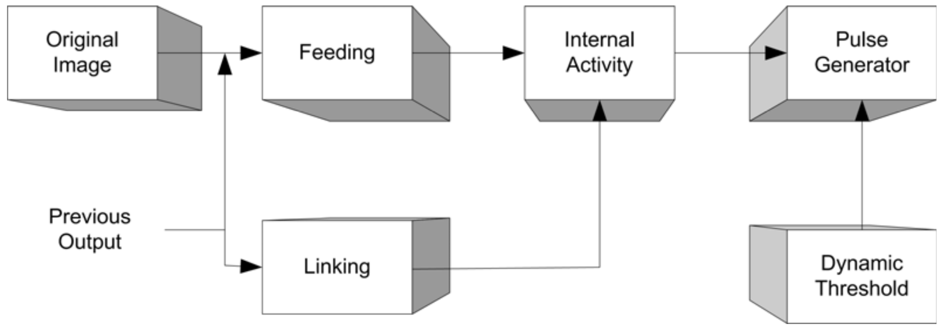

Fig. 1. Basic model of the PCNN.

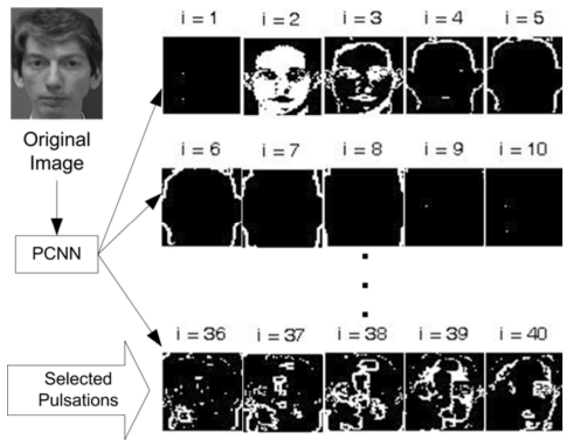

Fig. 2. Images pulsed by a PCNN when a face image is the input. Pulsations 36 to 40 are selected as features.

$$\theta(t) = e^{-\frac{1}{\alpha_\theta}} \theta(t-1) + VY(t) \tag{6}$$

Finally the output of the neuron is defined by (7). In the case of an image processing task, each pixel is related to a neural element. For detailed information on how a PCNN works consult [21].

$$Y(t) = \begin{cases} 1 \text{ if } U(t) \rangle \theta(t) \\ 0 \text{ otherwise} \end{cases} \tag{7}$$

B. PCNN Initial Parameters

For this particular project, the initial parameters of the PCNN are shown in Table 1. The kernels W and M are 3×3 average kernels, because average kernels reinforce grouping.

C. PCNN Facial Feature Extraction

A pulsed image is the result of applying the PCNN to a given image. Now, if a gray-scale facial image is given to the inputs of the PCNN, we obtain pulsed images similar to the ones in Fig. 2 where it is shown that the pulsations of the faces changes across the time. The selected pulses are the 36 to 40. These pulsations are selected because the content of facial information is useful to construct a feature vector, even when the illumination is not controlled, this means that this features are illumination-robust and adds discriminant information to the system. As in Fig. 2 the pulses 36 to 40 have more content rather than the first 10 pulsations.

TABLE I
PCNN INITIAL PARAMETERS

For 50 iterations
$\beta = 1.0$
$G_{Feed} = 0.1$
$\alpha_F = 0.1$
$G_{Link} = 1$
$\alpha_L = 0.1$
$\alpha_\theta = 5$
$V = 5$

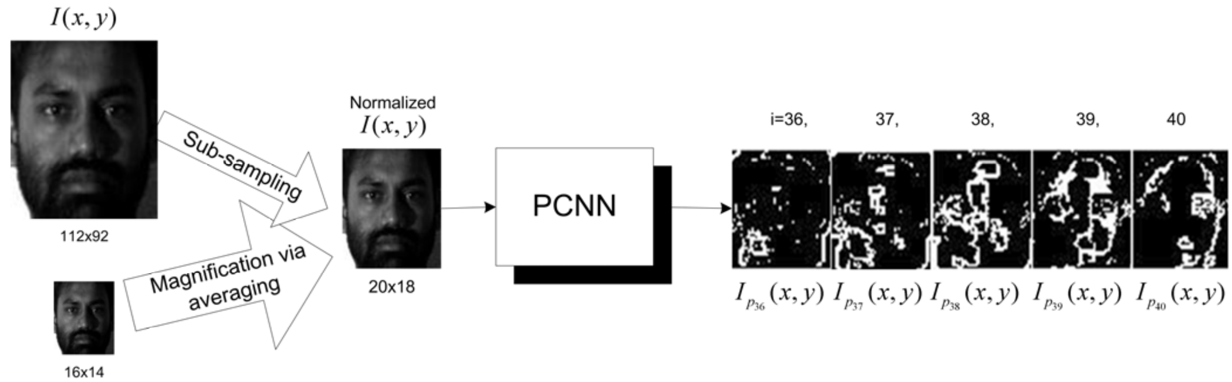

Fig. 3. Reduction via sub-sampling, and magnification via averaging, from two different image sizes.

D. PCNN Feature Vector

The original image $I(x,y)$ is preprocessed by the PCNN. The PCNN pulses and generates an output image $I_p(x,y)$. The pulsed images from the PCNN are collected to generate a feature vector. The feature vector $\mathbf{T}_{pcnn}$ (8) depends on the original image size, providing particularly a big size. This is why we select only 5 pulsed images as said before.

$$\mathbf{T}_{pcnn} = \left[T_{can}\left(I_{p_{36}}(x,y)\right) \quad \ldots \quad T_{can}\left(I_{p_{40}}(x,y)\right) \right] \tag{8}$$

E. PCNN Feature Vector Dimensionality Reduction

In order to reduce the dimensionality of the feature vector $\mathbf{T}_{pcnn}$, we need to implement an image size normalization. This process is performed before presenting the image to the PCNN in order to save processing time; otherwise the PCNN will be consuming extensive time.

The original face image is reduced or magnified to the static size of 20×18 pixels no matter what the original size is, an example of this situation is illustrated in Fig. 3. This amount of reduction is performed only for this particular component of the feature vector. Thus the dimension of $\mathbf{T}_{pcnn}$ is 1800 because $xy \times 5 = \overset{x}{20} \times \overset{y}{18} \times 5 = 1800$.

IV. Generation of the Complete Feature Vector

The proposed feature extraction method can be viewed as a whole by combining the previously described facial feature extraction methods: FFL, KLT, and PCNN.

The feature extraction method proposed involves the computation of the transformation matrix $\mathbf{w}$ of the KLT method. The KLT method needs the complete training set of facial images. The training stage of the system will be pointed in further sections. At this point we will assume that the transformation matrix is already computed and stored at $\mathbf{w}$.

So, for a new facial image $I(x,y)$ we need to compute:

First, generate $\mathbf{z}_i$ with (1) for $i = 4$ [31],

Second, using the KLT obtain, $\hat{\mathbf{i}}_{xy}$ with (2)

Third, apply the PCNN to generate $\mathbf{T}_{pcnn}$ with (8).

The complete feature vector of a given image can be then defined as

$$\mathbf{d} = \left[\mathbf{z}_{i=4} \quad \hat{\mathbf{i}}_{xy} \quad \mathbf{T}_{pcnn} \right] \tag{9}$$

where $\mathbf{d}$ denotes the concatenation of all the features extracted from an $I(x,y)$ input facial image.

An example of the complete feature extraction process is shown graphically in Fig. 4. As expected, $\mathbf{d}$ is a high dimensional feature vector. Once we have defined the feature extraction method, we need to introduce the classification scheme. In the following section the classification scheme based on a neurofuzzy system is described.

V. Neural-Fuzzy Classifier

It is found that the Radial Basis Functions, RBF network, is equivalent to a fuzzy system in terms of functionality under certain conditions [35]. These functional equivalencies allow us to combine the advantages of ANN and, Fuzzy Logic, FL, to create a neural-fuzzy system. Considering this, a neurofuzzy network is created based on a RBF network.

A. Radial Basis Functions Neural Network

The network utilized is a two-layer probabilistic RBF. When an input is presented to the network, the first layer obtains the distance between the input vectors and the training vectors and produces a new vector which elements describe how near is the input regarding the training vectors. The second layer adds the contribution to each class and produces

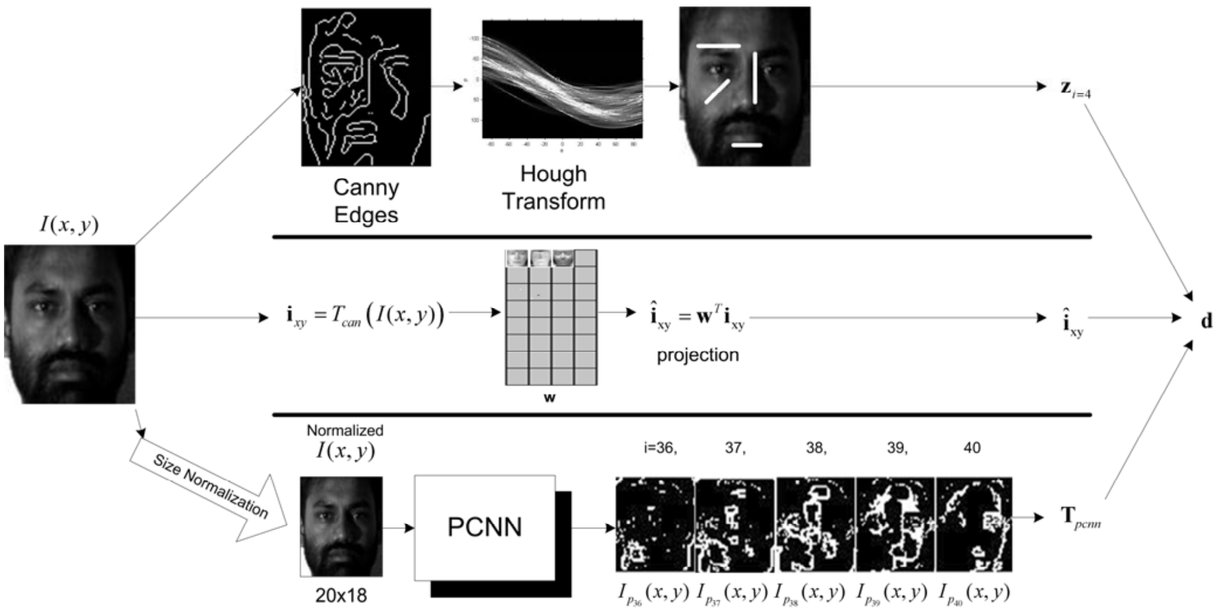

Fig. 4. Feature extraction method. First the Face Feature Lines, FFL, are calculated and placed in $\mathbf{z}_{i=4}$; second, uncorrelated features are extracted with KLT method and placed in $\hat{\mathbf{i}}_{xy}$; third, the PCNN approach is computed and the pulsed images are stored in $\mathbf{T}_{pcnn}$. All the features are concatenated into the vector $\mathbf{d}$.

a new probability vector; the nearest class to the input vector has the highest probability to represent it. Finally the output of the second layer selects the class with the highest probability value and put 1 for that class and 0 for the other classes. In other words this layer performs a competence between the classes and the winner represents the input. The activation function of the first layer is Gaussian, and produces a maximum value of 1 when the distance between the input and the test vector is 0. As the distance between the input and the test vector increases, the output of the activation function decreases. In the training process every single neuron of the first layer represents a training sample with tolerance depending on the variance of the Gaussian function. Therefore the number of neurons in the first layer is the same as the training samples utilized, and the number of neurons on the output layer corresponds to the number of classes. In this paper we propose a system to recognize 10 people, therefore, we have 10 classes. Assuming that we have 8 samples per subject (class), we will have 80 samples to train the RBF. Indeed, we have an architecture of 80 neurons in the first layer, and 10 neurons in the output layer, as shown in Fig. 5.

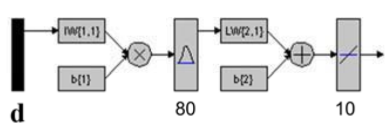

Fig. 5. Example of a Radial Basis Functions Neural Network architecture.

A fuzzy neural network has the same basic architecture of an artificial neural network, except that some of its elements are fuzzyfied [36]. There are some ways to fuzzificate a neural network. In this paper, some of the inputs of the RBF network are fuzzyfied to create a neurofuzzy network.

B. Neurofuzzy Network

The design of the network as previously mentioned, is based on a Probabilistic RBF with two layers.

We have decided to fuzzy the feature vector elements corresponding to the FFL, $\mathbf{z}_{i=4}$, and the KLT, $\hat{\mathbf{i}}_{xy}$. These features are fuzzyfied through membership functions. This membership functions are created according to the upper and lower limits of each feature. The algorithm for the automatic generation of the membership functions is shown in Fig. 6. Basically the algorithm first calculates the upper $\mathbf{vl}(s,2)$ and lower limits $\mathbf{vl}(s,1)$ for all the classes for a specific feature. Then, the lower $mfll$ and upper $mful$ limits of the current feature are computed as well as their indexes $inll$ and $inul$ in the vector $\mathbf{vl}$ respectively. The z-type membership functions, $zmf(\cdot)$ are created if the current feature index corresponds to the index of the lower limit, $inll$. If the current feature index corresponds to the index of the upper limit, $inul$ then the s-type membership functions, $smf(\cdot)$ are created. Otherwise, pi-type membership functions $pimf(\cdot)$ are created. The

$nfeatures$ = number of features ($sizeof(\mathbf{z}_{i=4}) + sizeof(\hat{\mathbf{i}}_{xy})$)

$nfaces$ = samples per subject ; $nclasses$ = number of subjects (classes)

for $f = 1;\ f = nfeatures;\ f{+}{+}$

  for $s = 1;\ s = nclasses;\ s{+}{+}$ // Buckle to find the limits of the features for each class

    $\mathbf{vl}(s,1) = \min\left(\mathbf{d}\left(f, (s \times nfaces) - (nfaces - 1) : (s \times nfaces)\right)\right)$

    $\mathbf{vl}(s,2) = \max\left(\mathbf{d}\left(f, (s \times nfaces) - (nfaces - 1) : (s \times nfaces)\right)\right)$

  end

  $\left[mfll, inll\right] = \min\left(\mathbf{vl}(:,1)\right)$ // Calculates the lower limit for the current feature

  $\left[mful, inul\right] = \max\left(\mathbf{vl}(:,2)\right)$ // Calculates the upper limit for the current feature

  for $s = 1;\ s = nclasses;\ s{+}{+}$

    for $ss = 1;\ ss = nfaces;\ ss{+}{+}$

      if $\left(s == inll\right)$ then // If the current class represents the lower values, create a Z membership function

        $\mathbf{fmf}(s,:) = zmf\left(0 : mful, \left[\mathbf{vl}(s,1) \quad \mathbf{vl}(s,2)\right]\right)$

      else if $\left(s == inul\right)$ then // If the current class represents the higher values, create a S membership function

        $\mathbf{fmf}(s,:) = smf\left(0 : mful, \left[\mathbf{vl}(s,1) \quad \mathbf{vl}(s,2)\right]\right)$

      else // If the current class represents the middle values, create a Pi membership function

$$\mathbf{fmf}(s,:) = pimf\left(0 : mful, \begin{bmatrix} \mathbf{vl}(s,1) \ \ldots \\ \left(\dfrac{\left(\mathbf{vl}(s,1) + \mathbf{vl}(s,2)\right)}{2}\right) - 0.5 \ \ldots \\ \left(\dfrac{\left(\mathbf{vl}(s,1) + \mathbf{vl}(s,2)\right)}{2}\right) + 0.5 \ \ldots \\ \mathbf{vl}(s,2) \end{bmatrix}\right)$$

      end if

    end

  end

  $\mathbf{allmf}\left(f,:,1 : (mful + 1)\right) = \mathbf{fmf}$ // Store the membership functions of all the classes for the current feature

  $\mathbf{fmf}$ = null

end

Fig. 6. Pseudo-code algorithm for the automatic generation of the membership functions.

membership functions for the feature are stored in $\mathbf{fmf}(s,:)$ where s is the corresponding class. At the end of the buckle, all the membership functions for all the features are stored in $\mathbf{allmf}$.

An illustration of the algorithm Fig. 7 shows one of the membership functions created for the 1st feature using 3 faces.

Once we have all the membership functions in the matrix $\mathbf{allmf}$, then all the selected features (FFL and KTL) of the input vectors are fuzzyfied with the membership functions just created.

C. Neurofuzzy System Summary

At this point the neurofuzzy system based on an RBF neural network has been described. The architecture of the RBF network varies with the amount of training samples. The FFL and KLT features (inputs of the RBF as well as the PCNN features) are fuzzyfied before the input layer of the RBF.

An algorithm for automatic membership functions generation fitting the class data for each feature has also been explained.

In the following section we present the stages of the proposed system, linking the feature extraction method to the classification scheme.

VI. System Integration and Experimentation

In this paper we have proposed a feature extraction

Sample	Class	$\mathbf{z}_{i=4}$							$\hat{\mathbf{i}}_{xy}$								...
1	1	55.04	56.014	56.035	8.003	45.002	2.05	51.049	65	63	53	49	59	68	70	73	
2	1	11.058	7.004	37.002	28.002	35.058	14.002	38.002	44	51	66	69	50	36	38	42	
3	1	50.018	2.011	2.035	8.003	36.002	2.011	2.03	63	52	40	34	30	25	26	31	
4	2	49.002	55.038	55.044	56.021	56.035	4.051	39.059	56	74	51	24	40	56	51	51	
5	2	49.002	12.003	53.003	5.044	57.048	7.049	53.003	44	47	48	34	54	63	47	85	...
6	2	57.029	5.022	51.027	22.043	57.042	6.025	51.026	44	46	52	51	45	46	48	43	
7	3	11.054	6.035	9.007	49.053	53.024	8.021	50.018	43	41	42	43	39	32	27	26	
8	3	10.004	9.009	12.059	9.013	53.024	49.006	57.043	45	46	48	48	44	40	40	43	
9	3	57.027	2.03	57.028	20.004	49.006	5.048	51.048	31	33	34	30	25	23	27	31	
10	4	53.014	3.02	57.023	2.022	57.022	2.023	57.03	37	36	36	36	36	36	35	35	

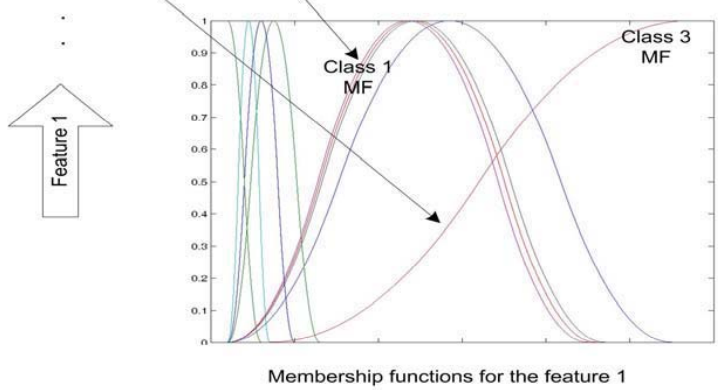

Fig. 7. Illustration of the algorithm proposed to compute membership functions according to the features.

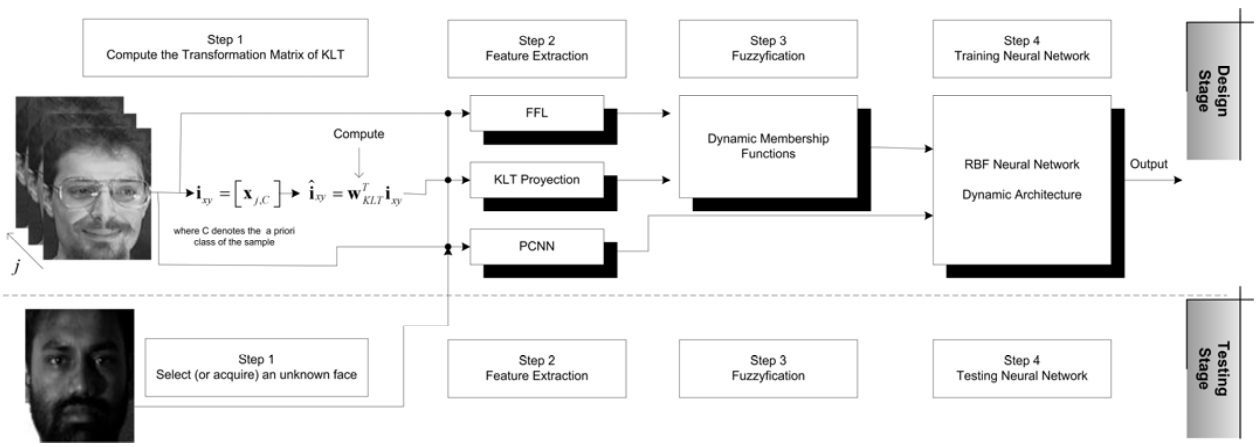

Fig. 8. Stages of the proposed system. This figure shows the similarity of the Design and Tesging stages, and ilustrates the overall process.

method, and a classification scheme for face recognition. These schemes involve different stages. For comprehension purposes we have divided the complete system in two main stages: Design, and Testing. These stages are described in the following subsection.

A. Design Stage

Fig. 8 shows the stages of the overall system. The first step involves the selection of *j* design (training) samples, $\mathbf{x}_{j,C}$, and *i* testing (probe) samples, $\mathbf{x}_i$. Where *C* is the a priori known class for the sample. The design of the system follows the next steps:

Step 1. Compute the transformation matrix $\mathbf{w}_{KLT}$ given the design samples, $\mathbf{x}_{j,C}$, according to (2).

Step 2. Extract the features of the FFL following (1) and compute $\mathbf{z}_{i=4}$. Project the design samples $\mathbf{x}_{j,C}$ using (2) and obtain $\hat{\mathbf{i}}_{xy}$. Let the PCNN pulse in 50 iterations and compute (8) and obtain $\mathbf{T}_{pcnn}$. Finally, construct $\mathbf{d}$ with (9).

Step 3. According to the features extracted, the membership functions are created for $\mathbf{z}_{i=4}$ and $\hat{\mathbf{i}}_{xy}$, using the algorithm described in Fig. 6. Then these features are fuzzyfied according to the membership functions created.

Step 4. Design the architecture of the RBF network, creating the number of inputs according to the size of $\mathbf{d}$. Create *j* neurons at the first layer. Create *C* neurons at the output layer. Finally train the network with the features contained in $\mathbf{d}$.

B. Testing Stage

The testing stage has the following steps:

Step 1. Select *i* unknown faces for testing, $\mathbf{x}_i$.

Step 2. Extract the features of the FFL following (1) and compute $\mathbf{z}_{i=4}$. Project the testing samples $\mathbf{x}_i$ using (2) and obtain $\hat{\mathbf{i}}_{xy}$. Let the PCNN pulse in 50 iterations and compute (8) and obtain $\mathbf{T}_{pcnn}$. Finally, construct $\mathbf{d}$ with (9).

Step 3. According to the membership functions created during the design stage, fuzzify the features $\mathbf{z}_{i=4}$ and $\hat{\mathbf{i}}_{xy}$.

Step 4. Test the RBF network with the testing feature vectors $\mathbf{d}$.

C. Experimentation

Experimentation was performed over the well known face databases ORL and YALE.

The face database “Olivetti Research Laboratory” (ORL), was collected between 1992 and 1994, it has slight variations on pose, illumination, facial expression (eyes open/closed, smiling/not-smiling) and facial details (glasses/no-glasses) [34], [37]. The ORL database has 40 different subjects, for experimental purposes 10 samples per subject were used. Fig. 9 presents an example of the ORL database.

Fig. 9. Sample faces of the ORL database.

The Yale face database contains images of subjects in a variety of conditions included with/without glasses, illumination and expression variations [34]. We have utilized 10 subjects of this database and 10 samples per subject. In Fig. 10 illustrates samples of two different subjects under the conditions described above.

The experiments were designed using two options: with and without the PCNN features. These experiments let us to observe if the PCNN features improve the performance of the face recognition system. The experiments consist on vary the amount of samples for training from 1 to 8 samples per subject. 10 classes were constructed according to the number of subjects to recognize. The samples picked for training and for testing were selected randomly. In the following section we present the results of these experiments.

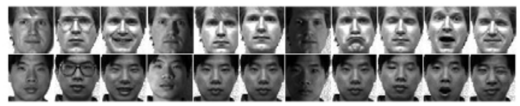

Fig. 10. Sample faces of the YALE database.

VII. Results

Table II shows the results for testing on the ORL database without the PCNN features. The highest performance obtained reaches 98% of correct recognition. Results in the YALE database without the PCNN features report a performance of 78% as presented on Table III. Table IV shows the results for testing on the ORL database with PCNN features, which are the same as Table II but better on 2 training samples (TS) for the subject #8 (S8) reaching 95%. For the YALE database with the PCNN features the results were better than without the PCNN features reaching 81% as shown in Table V. In the Fig 11 is shown the performance graphic of the experiments on the two databases, ORL and YALE with/without PCNN features.

It is shown that the algorithm performs better on ORL database because of the less variation of the face samples regarding lighting conditions.

VIII. Conclusions

We have presented a neurofuzzy scheme for face recognition based on the PCNN feature extraction scheme, and also in the aspects of face perception through the Hough transform and Karhunen-Loève transform, called Hough-KLT. The neurofuzzy algorithm was constructed extracting facial features via the PCNN, fuzzyfing the results of the Hough transformation and then the results are concatenated with the results of KLT. These fuzzy features are the inputs of an RBF neural network as the final classifier reaching 98% of recognition rates on the ORL database and 81% on the YALE database. The results are comparable to PCA, LDA, FLDA methods. Fig 11 shows a graphical comparison of the four experiments realized over the ORL and YALE face databases. As can be shown the system performs better on the ORL and the YALE databases when the PCNN features are

TABLE II

RECOGNITION RATE WITHOUT THE PCNN ON ORL

	TRAINING SAMPLES							
SUBJECT #	TS1	TS2	TS3	TS4	TS5	TS6	TS7	TS8
s1	0.8	0.9	0.9	0.9	0.9	0.9	0.9	0.9
s2	1	1	1	1	1	1	1	1
s3	1	1	1	1	1	1	1	1
s4	1	1	1	1	1	1	1	1
s5	1	1	1	1	1	1	1	1
s6	0.8	0.9	0.9	0.9	0.9	0.9	0.9	0.9
s7	1	1	1	1	1	1	1	1
s8	0.8	0.7	0.8	0.9	0.9	1	1	1
s9	0.9	0.9	1	1	1	1	1	1
s10	0.9	1	1	1	1	1	1	1
TOTAL	**0.92**	**0.94**	**0.96**	**0.97**	**0.97**	**0.98**	**0.98**	**0.98**

Where S refers to the subject, and TS means Training Sample.

TABLE III

RECOGNITION RATE WITHOUT THE PCNN ON YALE

	TRAINING SAMPLES							
SUBJECT #	TS1	TS2	TS3	TS4	TS5	TS6	TS7	TS8
s1	0.2	0.4	0.7	0.7	0.8	0.8	0.8	0.8
s2	0.9	0.9	0.9	1	1	1	1	1
s3	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9
s4	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.3
s5	0.3	0.9	0.9	0.9	0.9	1	1	1
s6	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9
s7	0.4	0.4	0.4	0.3	0.3	0.3	0.5	0.6
s8	0.6	0.8	0.9	0.9	0.9	0.9	0.9	0.9
s9	0.5	0.6	0.6	0.7	0.7	0.8	0.7	0.6
s10	0.6	0.7	0.8	0.8	0.8	0.8	0.8	0.8
TOTAL	**0.55**	**0.67**	**0.72**	**0.73**	**0.74**	**0.76**	**0.77**	**0.78**

included, the performance obtained in the YALE database may indicate that the features generated with the PCNN makes the face recognition system more robust.

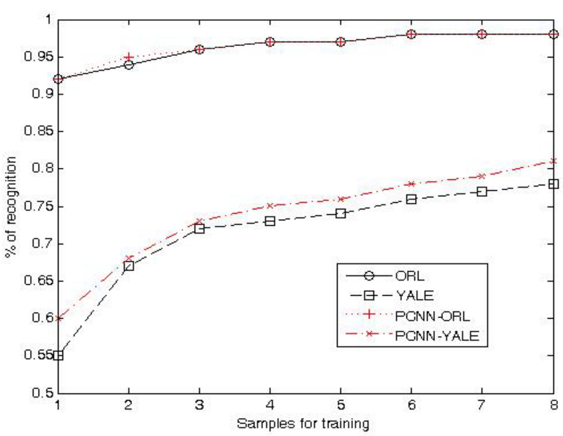

Fig. 11. Comparison of the experiments realized on ORL and YALE face databases with/without PCNN.

According to the reported results we may conclude that the PCNN and the feature fuzzyfication fusion represent a good alternative in the face recognition problem.

TABLE IV

RECOGNITION RATE WITH THE PCNN ON ORL

	TRAINING SAMPLES							
SUBJECT #	TS1	TS2	TS3	TS4	TS5	TS6	TS7	TS8
s1	0.8	0.9	0.9	0.9	0.9	0.9	0.9	0.9
s2	1	1	1	1	1	1	1	1
s3	1	1	1	1	1	1	1	1
s4	1	1	1	1	1	1	1	1
s5	1	1	1	1	1	1	1	1
s6	0.8	0.9	0.9	0.9	0.9	0.9	0.9	0.9
s7	1	1	1	1	1	1	1	1
s8	0.8	0.8	0.8	0.9	0.9	1	1	1
s9	0.9	0.9	1	1	1	1	1	1
s10	0.9	1	1	1	1	1	1	1
TOTAL	**0.92**	**0.95**	**0.96**	**0.97**	**0.97**	**0.98**	**0.98**	**0.98**

TABLE V

RECOGNITION RATE WITH THE PCNN ON YALE

	TRAINING SAMPLES							
SUBJECT #	TS1	TS2	TS3	TS4	TS5	TS6	TS7	TS8
s1	0.3	0.4	0.7	0.7	0.8	0.8	0.8	0.8
s2	0.9	0.9	0.9	1	1	1	1	1
s3	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9
s4	0.3	0.3	0.3	0.3	0.3	0.3	0.3	0.4
s5	0.6	0.9	0.9	0.9	0.9	1	1	1
s6	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9
s7	0.4	0.4	0.4	0.4	0.4	0.4	0.5	0.6
s8	0.6	0.8	0.9	0.9	0.9	0.9	0.9	0.9
s9	0.5	0.6	0.6	0.7	0.7	0.8	0.8	0.8
s10	0.6	0.7	0.8	0.8	0.8	0.8	0.8	0.8
TOTAL	**0.6**	**0.68**	**0.73**	**0.75**	**0.76**	**0.78**	**0.79**	**0.81**

IX. ACKNOWLEDGMENT

The authors gratefully acknowledge the support provided by SEP-DGEST and CONACyT #193324, for this research under grant 445.05-P.

X. REFERENCES

[1] M. J. Er, W. Chen, S. Wu, "High-Speed Face Recognition Based on Discrete Cosine Transform and RBF Neural Networks", IEEE Transactions on Neural Networks, Vol. 16, No. 3, May 2005.

[2] Z. M. Hafed, M. D. Levine, "Face Recognition Using the Discrete Cosine Transform" International Journal of Computer Vision, Vol. 43, No. 3, 2001. pp. 167–188.

[3] W. L. Scott, "Block-Level Discrete Cosine Transform Coefficients For Autonomic Face Recognition", Ph.D. dissertation, Graduate Faculty of the Louisiana State University and Agricultural and Mechanical College, May 2003.

[4] B. Zhang, H. Zhang, S. S. Ge, "Face Recognition by Applying Wavelet Subband Representation and Kernel Associative Memory", IEEE Transactions on Neural Networks, Vol. 15, No. 1, January 2004. pp. 166 – 177.

[5] E. Hjelmås, "Feature-Based Face Recognition", NOBIM Proceedings (Norwegian Image Processing and Pattern Recognition Conference), 2000.

[6] H. Kong, X. Li, L. Wang, E. K. Teoh, J. Wang, R. Venkateswarlu, "Generalized 2D Principal Component Analysis", IEEE International Joint Conference on Neural Networks (IJCNN), Montreal, Canada, August 2005.

[7] W. Li and C. Zhou, "Face Recognition Using Feature Combination and Improved LDA", IEEE Transactions on Systems, Man, and Cybernetics, 2004. pp. 1292 – 1296.

[8] W. Mio, Q. Zhang, X. Liu, "Nonlinearity and optimal component analysis", IEEE International Joint Conference on Neural Networks (IJCNN), Montreal, Canada, August 2005.

[9] C. Liao and J. Chien, "Nonsingular Discriminant Feature Extraction for Face Recognition", IEEE International Conference on Acoustics Speech and Signal Processing, 2005, pp. II 957 – II 960.

[10] J. Yang, D. Zhang, X. Young, and J. Yang, "Two-dimensional Discriminant Transform for Face Recognition", Pattern Recognition. Vol. 38, 2005, pp. 1125-1129.

[11] H. Xiong, M. N. S. Swamy, and M. O. Ahmad, "Two-dimensional FLD for Face Recognition", Pattern Recognition. Vol. 38, 2005, pp. 1121-1124.

[12] K. Kwak, and W. Pedryez, "Face Recognition Using a Fuzzy Fisherface Classifier", Pattern Recognition. Vol. 38, 2005, pp. 1717-1732.

[13] D. Xu, S. Yan, L. Zhang, M. Li, W. Ma, Z. Liu, and H. Zhang, "Parallel Image Matrix Compression for Face Recognition", Proceedings of the 11th International Multimedia Modelling Conference (MMM'05), 2005.

[14] I. Bax, G. Heidemann, H. Ritter, "Face Detection and Identification Using a Hierarchical Feed-forward Recognition Architecture", IEEE International Joint Conference on Neural Networks (IJCNN), Montreal, Canada, August 2005.

[15] X. Lu, Y. Wangy, and A. K. Jain, "Combining Classifiers For Face Recognition", 2003.

[16] M. Vatsa, R. Singh and P. Gupta, "Face Recognition Using Multiple Recognizers", IEEE Transactions on Systems, Man, and Cybernetics, 2004. pp. 2186 – 2190.

[17] M. T. Harandi, M. N. Ahmadabadi, and B. N. Araabi, "Face Recognition Using Reinforcement Learning", IEEE International Conference on Image Processing, 2004, pp. 2709-2712.

[18] W. Zheng, C. Zou, and L. Zhao, "Face Recognition Using Two Novel Nearest Neighbor Classifiers", IEEE International Conference on Acoustics Speech and Signal Processing, 2004, pp. V 725 – V 728.

[19] E. Hjelmås and J. Wroldsen, "Face Recognition with Multi-Layer Perceptrons", NOBIM Proceedings (Norwegian Image Processing and Pattern Recognition Conference), 1997.

[20] C. Hoi, and M. R. Lyu, "Robust Face Recognition Using Minimax Probability Machine", IEEE International Conference on Multimedia and Expo (ICME), 2004, pp. 1175-1178.

[21] T. Lindbland and J. Kinser, "Image Processing Using Pulse-Coupled Neural Networks," Spring, 1998.

[22] J. L Johnson and M. Lou Padgett , "PCNN Models and Applications", IEEE Transactions on Neural Networks, Vol. 10 No. 3, May 1999.

[23] J. Johnson, "Pulse-Coupled Neural Nets: Translation, Rotation, Scale, Distortion, and In-tensity Signal Invariance for Images, "Applied Optics, vol. 30, no. 26, pp. 6239-6253, 1994.

[24] G. Kuntimad and H. Ranganath, "Perfect Image Segmentation Using Pulse Coupled Neu-ral, Networks," IEEE Transactions on Neural Networks, vol.10, no. 3, pp. 591-598. 1999.

[25] P. Keller and J. Johnson, "Pulse-Coupled Neural Network for Medical Analysis," Proceed-ings of SPIE Applications and science of Computation Intelligence II, pp. 444- 451, 1999.

[26] H. S. Ranganath and G. Kuntimad , "Object Detection Using Pulse Coupled Neural Net-works," IEEE Transactions on Neural Networks, vol.10, no. 3, May 1999.

[27] H. Rughooputh and S. Rughooputh, "Spectral Recognition Using a Modified Eckhorn Neu-ral Network Model," Image and Vision Computing, pp. 1101-1103, 2000.

[28] H. Ranganath and G. Kuntimad, "Image Segmentation Using Pulse Coupled Neural Net-works," Proceedings of the IEEE, pp. 285-1290, 1994.

[29] M.I. Chacón and A. Zimmeman, "PCNNP: A Pulse-Coupled Neural Network Processor," Proceedings of the IEEE IJCNN 2002, pp. 1581-1585, May 2002.

[30] M.I. Chacón and A. Zimmeman, "Image Processing Using the PCNN Time Matrix as a Se-lective Filter," Proceedings of the IEEE ICIP, pp. 1195-1199, September 2003.

[31] M. I. Chacon, P. Rivas, G. ramirez,"A Fuzzy Clustering Approach for Face Recognition Based on Face Feature Lines and Eigenvectors," Proceeding of the International Seminar on Computational Intelligence 2006, October 2006.

[32] B. Bower, "Faces of Perception", Science News, Vol. 160, No. 1. Jul. 2001. pp. 10.

[33] A. Slater, P. C. Quinn, "Face recognition in the newborn infant", Infant and Child Development, Vol. 10, Mar. 2001, pp. 21.

[34] S. Z. Li, A. K. Jain, Handbook of face recognition, Springer, USA, 2004.

[35] Y. Jim, W. v. Seelen and B. Sendhof, "Extracting Interpretable fuzzy rules from RBR neural networks". Institut Fur Neuroenformaski. Ruhr-Universitat Bochum, Internal Report, January 2000.

[36] C. T. Lin and C. S. G. Lee, Neural Fuzzy Systems: A Neuro Fuzzy Synergism To Intelligent Systems, Prentice Hall, 1995

[37] "Olivetti Research Lab (ORL) Database", On-Line: http://www.uk.research.att.com/facedatabase.html

XI. Biographies

Mario I. Chacón Murguía, received the BS in Electrical Engineering and the MS in Computer Engineering degree from the Chihuahua Institute of Technology, Chihuahua Mexico in 1982 and 1985 respectively and his Ph.D. in Electrical Engineering from New Mexico State University, USA, in 1998. He has made research for the industry and other universities. Dr. Chacón has several scientific and technical international publications. He is a Senior member of the IEEE, Member of the Mexican National System of Research, member of the IEEE Computational Intelligence Society, and member of the Pattern Recognition Society. His current research interests include Computer Vision, Digital Signal Processing (DSP), Pattern Recognition, Neural Networks and Fuzzy Logic applications to DSP, and Digital Systems.

Pablo Rivas Perea (S'02) This author became a Student Member (S) of IEEE in 2002. His birthplace is Guaymas, Sonora, México; on August 4 1980. He received the BS in Computer Systems Engineering degree from Nogales Institute of Technology, Sonora Mexico in 2003. Actually he is pursuing the MS in Electrical Engineering at Chihuahua Institute of Technology, Chihuahua Mexico since 2004.

He has worked on the manufacturing industry for 3 years as a Product Engineer. He also worked as a Software Engineer for 4 years. He has national and international publications on the fields of computer science and electrical engineering. His current research interests are Signal Processing, Industrial and Applied Mathematics, Soft Computing, and Medical Imaging. He is the IEEE Chihuahua Student Branch chair.

Mr. Rivas Perea is a student member of the Society of Industrial and Applied Mathematics, SIAM. He is also a student member of the Hispanic-American Fuzzy System Association, HAFSA. He has received several awards in concern to the creativity and to the academic areas.