A Fuzzy Clustering Approach for Face Recognition Based on Face Feature Lines and Eigenvectors

Mario I. Chacon M., Pablo Rivas P., and Graciela Ramirez A.

***Abstract*— This paper presents a new approach aimed to design a fuzzy face recognition system. Face feature lines, new features proposed in the paper, are incorporated in the feature vector used to design the patter recognition system. Face feature lines are considered as new features based on previous studies related to face recognition tasks on newborns. Besides the face feature lines the feature vector incorporates eigenvectors of the face image obtained with the Karhunen-Loeve transformation. The fuzzy face recognition system is based on the Gath-Gheva fuzzy clustering method and the Abonyi and Szeifert classification scheme. The performance of the face recognition system turned out to be 90% of correct classification tested on the ORL and Yale databases.**

***Index Terms*—Face Recognition, Face Feature Lines, Fuzzy Clustering.**

I. Introduction

Face recognition is one of the most interesting and challenging areas in computer vision and pattern recognition. Current face recognition systems have high recognition rates when face images are acquired in controlled conditions. However, robust face recognition systems are required in sophisticated security systems. Robustness must be translated into system tolerance to viewpoint, pose, illumination, and facial expression [1]-[16]. Two of the most important face recognition methods currently used are the eigenface and Fisherface methods. The eigenface method, or principal component analysis (PCA), is the most well known method for vector feature representation in face recognition [17]. PCA is a popular method in pattern recognition and communication theory that is quite often referred to as a Karhunen–Loeve transformation (KLT). The PCA approach exhibits optimality when it is applied to reduce the dimensionality of a feature vector [18]. The PCA method is used in this work to map an original feature vector to a new feature space. Besides, with the purpose of improving the classic methods for face recognition, fuzzy logic theory is incorporated in this work to generate a new face recognition method.

In this paper we describe the Hough-KLT algorithm for facial feature extraction and the Euclidean distance classifier approach in Section II. Section III describes the Gath-Gheva fuzzy clustering algorithm to generate fuzzy rules and the Abonyi and Szeifert algorithm as a fuzzy classifier. Finally the general conclusions of this work are presented in Section IV.

II. Hough-KLT Features for Face Recognition.

The different parts of the proposed Hough-KLT method are described in this section. The proposed method represents a novel approach by incorporating ideas from the visual perception point of view related to face recognition. From visual perception studies, it is known that some spatial face features like, mouth to nose distance, geometric shape between the mouth and the eyes, and face feature lines, are distinguishing characteristic. The features selected and incorporated in the proposed method correspond to face feature lines, FFL. Face features lines are prominent lines and can be extracted with the Hough transform from low resolution image faces, and are important features documented in newborn face recognition studies. One of the most interesting cases regarding the facial feature extraction process happens with newborns. Studies with newborns have shown that babies perceive a totally-fuzzy world in terms of vision. Their only tool to recognize faces are facial lines and circles [7] [8]. This suggests that the use of lines for face recognition is a theory also supported by the psychology and neurology regarding face perception.

A. Hough Transform.

The Hough transform is a useful transformation to detect geometric patterns in images, like lines, circles, and ellipses.

In the domain of the Hough transform, HT, any line is defined by the parametric equation

$$\rho = x\cos\theta + y\sin\theta \tag{1}$$

where x and y represent the coordinate of a pixel in the

Manuscript received June 30, 2006. 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 M., P. Rivas P., and G. Ramirez A. 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).

(Advance online publication: 15 August 2007)

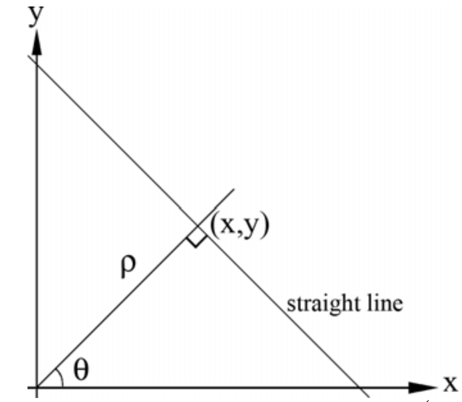


Fig. 1. Representation of a line with parameters (ρ, θ).

image, ρ is the distance of the line to the origin, and θ is the angle of the line with respect the horizontal axis, Fig. 1. In general practice, the HT algorithm requires a binary image as input, which represents the edges of the image [21]. In this work the edge detection algorithm used was the Canny operator [21].

Once the edges of the image are obtained, the HT is calculated, and the result is a representation of all the lines in the space ρ, and θ. Given this result we can extract the FFL by obtaining the maximum points from the result of the HT through ρ, and θ.

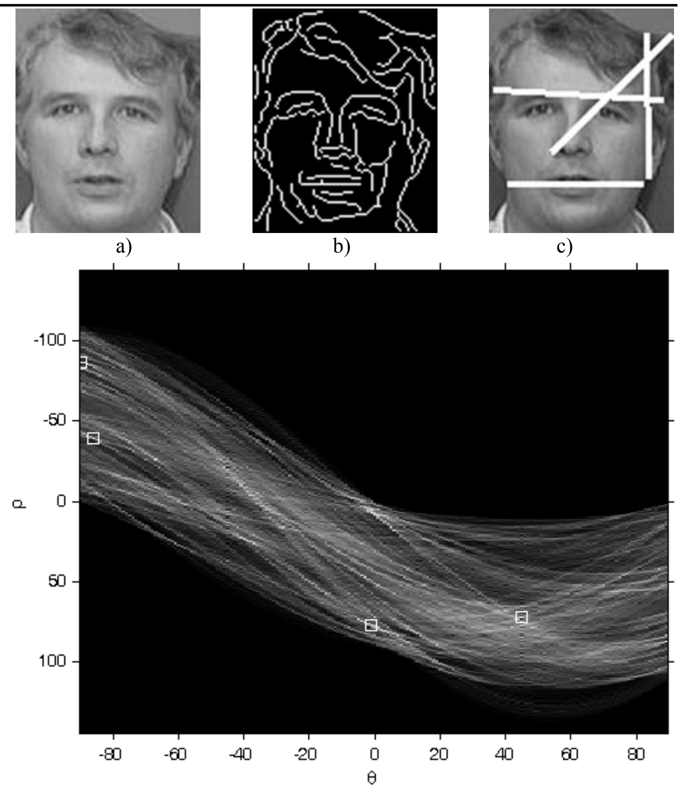
We consider that the four face feature lines may improve the performance of a face recognition system. This assumption is based on the experiments related to the newborns vision system. The information of these four FFL will be included as components of the feature vector which is defined in detail on further subsections.

In summary the FFL are obtained by applying the Canny operator to a gray scale image $I(x, y)$. This operator yields the edges of the image I_{BW}. Applying the HT we obtain the Hough *accumulator*, $\mathbf{P}_{Acumulator}$.

$\mathbf{P}_{Acumulator}$ contains the votes and the vectors, $\mathbf{P}_\theta$ and $\mathbf{P}_\rho$ with the values ρ and θ on which the matrix $\mathbf{P}_{Acumulator}$ was generated. In these terms a face feature line can be defined as a peak value in the accumulator as follows

$$\mathbf{N}_{Peaks}(\theta_i, \rho_i) = T_{HoughPeaks}(\mathbf{P}_{Acumulator}, i) \qquad (2)$$

where $\mathbf{N}_{Peaks}(\theta_i, \rho_i)$ is a matrix of $2 \times i$ elements where i denotes the number of peaks to extract. (θ_i, ρ_i) represents the coordinates in the space θ, ρ for the i *-th* peak value in $\mathbf{P}_{Acumulator}$. The points in the space domain of the image corresponding to a face feature line is defined by



d)

Fig. 2. Hough transform of a face: a) Original image, b) Face edges, c) Original image plus its four FFL, and d) Accumulator of the HT showing the 4 FFL.

$$\mathbf{R}_{Lines}(x_{1i}, y_{1i}, x_{2i}, y_{2i}) = T_{Lines}(I_{BW}, \mathbf{P}_\theta, \mathbf{P}_\rho, \mathbf{N}_{Peaks}) \quad (3)$$

where $\mathbf{R}_{Lines}(x_{1i}, y_{1i}, x_{2i}, y_{2i})$ denotes a matrix containing all the coordinates x_{1i}, y_{1i} where the i *-th* line begins and x_{2i}, y_{2i} where the line detected by the HT ends.

The result of apply the HT to a face to locate the four FFL is illustrated in the Fig. 2.

For the generation of the first part of the features vector from the coordinates of its four FFL, the following method was designed:

Step 1. Get the four maximum peak values with (2), for $i = 4$.

Step 2. Implement (3) and get the four characteristic lines coordinates, stored at $\mathbf{R}_{Lines}(x_{1i}, y_{1i}, x_{2i}, y_{2i}) \quad for \quad i = 1 \ldots 4$.

Step 3. Encode the coordinates information by taking the value of x_{1i} and add it to $\frac{y_{1i}}{1000}$, and include the result to l_{i_1}.

Step 4. Take the value of x_{2i} and add it to $\frac{y_{2i}}{1000}$, and include the result to l_{i_2}.

The feature vector can be defined as follows

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

$$\mathbf{z}_i = \begin{bmatrix} x_{11} + \frac{y_{11}}{1000}, x_{21} + \frac{y_{21}}{1000} \dots \\ x_{1i} + \frac{y_{1i}}{1000}, x_{2i} + \frac{y_{2i}}{1000} \end{bmatrix} \quad (4)$$

The $\mathbf{z}_i$ vector must be concatenated with the original image $I(x, y)$, in a canonical form (vector column) $\mathbf{i}_{xy}$, to construct the final feature vector

$$\mathbf{x}_{i+xy} = [\mathbf{z}_i \quad \mathbf{i}_{xy}] \quad (5)$$

The vector $\mathbf{z}_i$ is linked to the information of the original image in order to contribute and complement the face information representation before the transformation via KLT.

B. Principal Component Analysis

Principal Component Analysis, PCA, is a very widely used technique for dimensionality reduction. The objective of PCA is to transform the representation space X into a new space Y, in which the data are uncorrelated. The covariance matrix in this space is diagonal. The PCA method leads to find the new set of orthogonal axis to maximize the variance of the data. The final objective is dimensionality reduction of the feature vector [21].

The steps to compute the PCA are the following.

Step 1. The covariance matrix $\mathrm{Cov}_{\mathbf{X}}$ is calculated over the input vectors set $\mathbf{x}_i$ that corresponds to i facial images represented as vectors $\mathbf{x}$. The covariance is defined as

$$\mathrm{Cov}_{\mathbf{X}} = \frac{1}{n-1}\sum_{i=1}^{n}(\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^T \quad (6)$$

where $\bar{\mathbf{x}}$ denotes the mean of each variable of the vector $\mathbf{x}$, and n is the number of input vectors.

Step 2. The *n* eigenvalues of $\mathrm{Cov}_{\mathbf{X}}$ are extracted and defined as $\lambda_1, \lambda_2, \dots \lambda_n$, where $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n$.

Step 3. The *n* eigenvectors are $\Phi_1, \Phi_2, \dots \Phi_n$ and are associated to $\lambda_1, \lambda_2, \dots \lambda_n$.

Step 4. A transformation matrix, $\mathbf{W}_{PCA}$, is created $\mathbf{W}_{PCA} = [\Phi_1, \Phi_2, \dots \Phi_n]$.

Step 5. The new vectors $\mathbf{Y}$ are calculated using the following equation

$$\mathbf{Y} = \mathbf{W}_{PCA}^T \mathbf{X} \quad (7)$$

where T denotes the transpose of $\mathbf{W}_{PCA}$, and $\mathbf{X}$ denotes the matrix containing all the input vectors.

C. Karhunen-Loeve Transformation

The KLT is similar to the PCA [23], however in the KLT the each dimension of the input vectors $\mathbf{x}_i$ is normalized to the interval [0,1] before applying the PCA steps.

D. Hough-KLT Implementation

The vector $\mathbf{x}_{i+xy}$ as commented before, is composed by 8 coefficients $\mathbf{z}_i$, and by the original image $\mathbf{i}_{xy}$. All feature vectors are transformed with the KLT method. Face recognition can be achieved with the transformation matrix, $\mathbf{W}_{KLT}$, following the next steps.

Step 1. For an unknown facial image generate its $\mathbf{i}_{xy}$ representation.

Step 2. Compute the 8 $\mathbf{z}_i$ elements with (4).

Step 3. Generate $\mathbf{x}_{i+xy}$ with (5).

Step 4. Compute $\hat{\mathbf{x}}_{i+xy} = \mathbf{W}_{KLT}^T \mathbf{x}_{i+xy}$ with (7).

Step 5. Assign the facial image I_{face} to the class C_j If

$$D(\hat{\mathbf{x}}_{i+xy}, \hat{\mathbf{x}}_j) = \left\|\hat{\mathbf{x}}_{i+xy} - \hat{\mathbf{x}}_j\right\| < \left\|\hat{\mathbf{x}}_{i+xy} - \hat{\mathbf{x}}_k\right\| \quad \text{for all } j, k \; j \neq k \quad (9)$$

where $\hat{\mathbf{x}}_i$ represents the transformed feature vectors of the training faces.

This classifier together with the ten fold cross validation method was tested on the ORL database. The classifier had a performance of 91% of correct classification using 25 eigenvectors. If the FFL are included in the feature vector the performance increases to 95%. Also this classifier together with the ten fold cross validation method was tested on the Yale database. The classifier had a performance of 88% of correct classification using 25 eigenvectors. If the FFL are included in the feature vector the performance increases to 90%. These results are shown in Table I.

TABLE I
CLASSIFIER PERFROMANCES ON THE ORL AND YALE DATA BASES

	ORL		YALE	
CLASSIFIER	No FFL	FFL	No FFL	FFL
EUCLIDEAN	91%	95%	88	90

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) [23][24]. ORL has 40 different individuals. Fig. 3 presents an example of the ORL database.

The Yale database contains images of faces in a variety of conditions included with-without glasses, and variations on illumination and expression [23]. Fig. 4 illustrates two samples of two persons under the conditions described above.

III. Feature Analysis

The features described in section II, Hough-KLT, have shown a good performance with the Euclidean distance classifier. However, it is necessary to analyze the classification power of these features in order to assure that the performance will depend only on the type of the classifier.

This analysis can be performed with some evaluation metrics presented in [25]. These metrics are:

- **Uncorrelation**. This means, that features should not be dependant from each other, in order to provide discriminant information.
- **Reliability**. The objects of the same class should be the less sparse as possible.
- **Discriminant capacity.** It means that the classes α_j should be the more separated as possible.
- **Computing time**. This can be interpreted in different ways depending of the operating system. But here, computing time refers to the time consumed to compute the feature extraction process.

This last issue is the simplest to evaluate. Now we present the results obtained on every metric evaluated.

Fig. 3. Sample faces of the ORL database.

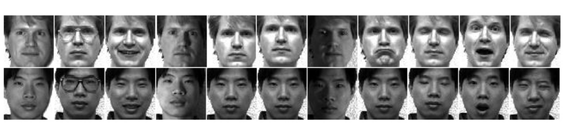


Fig. 4. Sample faces of the YALE database.

A. Uncorrelation

First, the uncorrelation is measured with the possible pairs of feature combinations, class to class, based on the covariance matrix:

$$C_i = \begin{pmatrix} C_{11} & \cdots & C_{1n} \\ \vdots & \ddots & \vdots \\ C_{n1} & \cdots & C_{nn} \end{pmatrix} \alpha_j \qquad (10)$$

The correlation coefficient of two generic features x_i, x_j is defined as follows:

$$r_{ij} = \frac{C_{ij}}{\sqrt{C_{ii}}\sqrt{C_{jj}}} \qquad -1 \le r_{ij} \le +1 \qquad (11)$$

These variables will be more independent to each other when the correlation coefficient is close to cero. The mean correlation coefficient is given by:

$$\overline{r_{ij}} = \frac{1}{nm}\sum_{i=1}^{n}\sum_{j=1}^{m} r_{ij} \qquad (12)$$

For the evaluation of this metric, we have utilized 10 samples per subject on the ORL and YALE database, for a total of 100 available samples. These samples were processed in order to extract their features with the Hough-KLT method. The Table II shows the results of the mean correlation coefficient of the feature vectors extracted from the samples. It can be noticed from Table II that the feature Hough-KLT presents the lowest $\overline{r_{ij}}$ compared with the KLT and eigenfaces.

TABLE II
Mean Correlation Coefficients $\overline{r_{ij}}$

	$\overline{r_{ij}}$ over Database	
Feature	ORL	YALE
KLT (no FFL)	0.01102	0.01466
Hough-KLT (with FFL)	0.01099	0.01464
Eigenfaces	0.09548	0.15247

B. Reliability and Discriminant Capacity

Regarding reliability, it should be evaluated in conjunction with the discriminant capacity. This is because a feature can have a high variance (meaning less reliability in the sense of sparse data within a class) but at the same time a highly discriminant feature. The quantification of the discriminant capacity should not be limited to the distance between classes, which can be measured as the Euclidean distance between each class mean. It is necessary to include the intern sparse of the classes. A useful tool to evaluate this is the Fisher's ratio [25].

The Fisher's criterion computes the separation between classes and the inner reliability of the classes at the same time. A feature should be more discriminant when the Fisher's ratio is higher. Good features must have high class mean values as well as high reliability.

The generalized Fisher ratio is denoted by:

$$F = \frac{\frac{1}{N}\sum_{j=1}^{N}\left(m_j - \overline{m}\right)^2}{\frac{1}{NP}\sum_{j=1}^{N}\sum_{i=1}^{P}\left(x_{ji} - m_j\right)^2} \qquad (13)$$

where

$$\overline{m} = \frac{1}{N}\sum_{j=1}^{N} m_j \qquad (14)$$

is the mean of the means. Table III summarizes the Fisher's ratio for the KLT, Hough-KLT and eigenfaces features computed over the ORL and Yale data based. As in the case of the mean correlation coefficients the Hough-KLT feature is better than the eigenfaces feature.

TABLE III
FISHER'S RATIO FOR THE CLASSES FOR THE ORL AND YALE DATABASES

	F OVER DATABASE	
FEATURE EXTRACTION METHOD	ORL	YALE
KLT (NO FFL)	8.12128	8.34377
HOUGH-KLT (WITH FFL)	8.22604	8.01492
EIGENFACES	5.65145	5.70566

C. Computing Time

The measure of the computing time is a very simple task. We only have to take the time right before and after the feature extraction process. Table IV shows the computation time to obtain each feature. The measures were taken on a PC Pentium 4 running at 2.4GHz with 512Mbytes of RAM. The implementation of the algorithms was done in the MATLAB language. As shown, the highest time corresponds to the Hough-KLT method.

In summary, the previous metrics indicate that the Hough-KLT represents a better choice than the KLT alone and the eigenfaces.

TABLE IV
COMPUTING TIME OF THE FEATURE EXTRACTION METHODS

FEATURE EXTRACTION METHOD	TIME IN SEC.
KLT (NO FFL)	0.0775
HOUGH-KLT (WITH FFL)	0.4345
EIGENFACES	0.0714

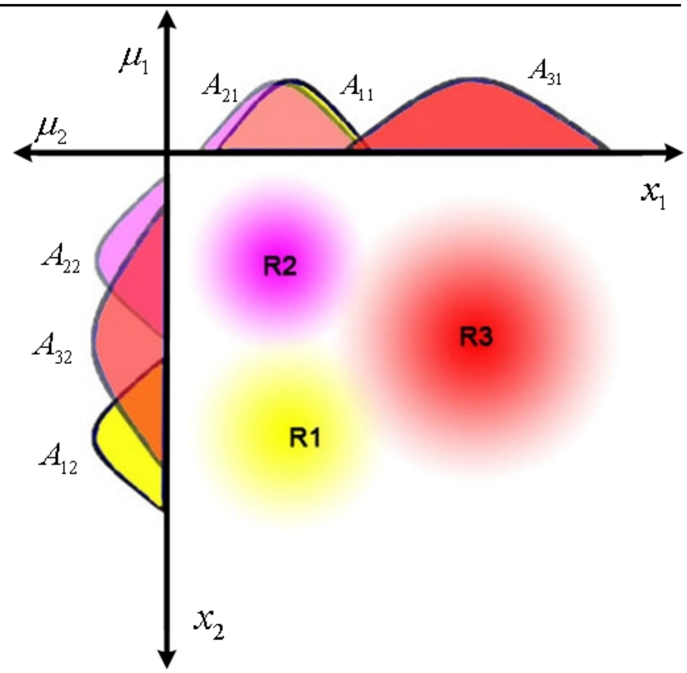


Fig. 5. Three fuzzy sets. The antecedents of fuzzy rules are created projecting the clusters to the axes of the input space.

IV. FUZZY HOUGH-KLT FOR FACE RECOGNITION

Commonly the membership functions of a fuzzy system are designed according to the experience of an expert who knows the behavior of a process. Fuzzy clustering in the input-output space is a technique widely used to create the membership functions of a fuzzy system [22] [26]. Applying the clustering techniques we can also obtain fuzzy sets that are utilized to model the antecedents of the rules in fuzzy systems. This is realized with the projection of the fuzzy sets as shown in Fig. 5.

In order to generate the fuzzy rules a clustering algorithm is applied to the input data. Once the clusters are generated, the membership functions of the fuzzy system are created by projecting the clusters to the axes of the input data. These membership functions can be used to define the antecedents of the fuzzy rules.

As an example, in Fig. 5 for the R3 cluster, we can identify the following rule

$$if\ x_1\ is\ A_{31}\ then\ y\ is\ R3 \qquad (15)$$

where x_1 is the input linguistic variable, A_{31} is a fuzzy set defined for the input linguistic variable x_1, y is the output linguistic variable, and $R3$ denotes the desired output cluster.

Using (15) we can identify the R3 cluster. However the R1 and R2 clusters are not identified in the terms of the input variable x_1. For this situation, x_2 is utilized to identify the clusters. The rules of the fuzzy system for R1 and R2 are defined as follows:

$$if\ x_1\ is\ A_{11}\ OR\ x_1\ is\ A_{21}\ AND\ x_2\ is\ A_{22}\ then\ y\ is\ R2 \quad (16)$$

$$if\ x_1\ is\ A_{21}\ OR\ x_1\ is\ A_{11}\ AND\ x_2\ is\ A_{12}\ then\ y\ is\ R1 \quad (17)$$

A. Fuzzy Clustering

The fuzzy c-means (FCM) algorithm has successfully been applied to a wide variety of clustering problems. Keller and Bezdek *et al.* [28-30] have successfully presented new approaches for the FCM. These approaches were called fuzzy-possibilistic c-means, FPCM, and possibilistic-fuzzy c-means, PFCM [28]. They have reported that one of the major contributions is that these algorithms overcomes the noise sensitivity of FCM. However, these algorithms work better for unlabeled data, and for this project we need an algorithm which can utilize the labels of the data for a better performance.

One of the most widely used algorithms for fuzzy clustering is Gath-Geva (GG) [22] [26] [27]. In the following subsection a brief description of GG algorithm to generate the corresponding antecedent part of a fuzzy rule system is presented.

1) Gath-Gheva algorithm

The Gath-Gheva algorithm GG, is a fuzzy clustering algorithm. One of the advantages of the GG algorithm is that it can utilize the label of the data to create fuzzy clusters in order to construct the antecedents of a fuzzy inference system [31].

The objective of clustering is to partition the data $\hat{\mathbf{x}}$ into C clusters. This means, each observation consists of the input $\hat{\mathbf{x}}_k$, and the output $\mathbf{y}_k$, grouped into a row vector. The fuzzy partition is represented by the $\mathbf{U}=[\mu_{i,k}]_{cxN}$ matrix, where the $\mu_{i,k}$ element of the matrix represents the degree of membership of $\hat{\mathbf{x}}_k$ to the class $i = 1,\ldots c$.

Clustering is based on the minimization of the sum of the weight squared distances between the data points $\hat{\mathbf{x}}_k$, and the centroids of the clusters v_i, $D^2_{i,k}$, defined by

$$J(Z,U,\eta)=\sum_{i=1}^{c}\sum_{k=1}^{N}\left(\mu_{i,k}\right)^m D^2_{i,k}\left(\hat{\mathbf{x}}_k, v_i\right) \quad (18)$$

where η contains the set of all the centroids: $v_1, \ldots, v_c$ and m is a fuzzy weighting exponent that determines the fuzziness of the resulting clusters.

Abonyi and Szeifert [31] presented a new distance measure D. This distance measure consists of two terms. The first term is based on the GG algorithm for non-supervised clustering. The second is based on the probability that the r_ith cluster describes the density of the class of the *k-th* data, $p(c_j = y_k | r_i)$. The second term, allows the use of class labels, and is defined as the consequent probability. The equation for D is denoted as

$$\frac{1}{D^2_{i,k}\left(\hat{\mathbf{x}}_k, r_i\right)} = \underbrace{P(r_i)\prod_{j=1}^{n}\exp\left(-\frac{1}{2}\frac{\left(\hat{\mathbf{x}}_{j,k}-v_{i,j}\right)^2}{\sigma^2_{i,j}}\right)}_{GG}\ p\left(c_j=\mathbf{y}_k | r_i\right) \quad (19)$$

where $\hat{\mathbf{x}}_k$ is the input vector, $\mathbf{v}_i$ denotes the centroid of the *i-th* cluster, σ_i is the standard deviation of the Gaussian membership function that is created according to the diagonal of the covariance matrix of the centers $\mathbf{v}$, $\mathbf{y}_k$ is the class of the input data. $P(r_i)$ denotes the a priori probability of the data in c_j.

The steps for the clustering process are the following. Given a set of data $\hat{\mathbf{x}}$ specify C, and choose a termination tolerance ϵ>0. Initialize the $\mathbf{U}=[\mu_{i,k}]_{cxN}$ partition matrix randomly, where $\mu_{i,k}$ denotes the membership that the $\hat{\mathbf{x}}_k$ data is generated by the *i*-th cluster.

Repeat until the termination tolerance ϵ is met

Step 1. Calculate the parameters of the clusters

Calculate the centers and standard deviation of the Gaussian membership functions.

$$\mathbf{v}_i^{(l)} = \frac{\sum_{k=1}^{N}\left(\mu_{i,k}^{(l-1)}\right)^m \mathbf{x}_k}{\sum_{k=1}^{N}\left(\mu_{i,k}^{(l-1)}\right)^m},$$

$$\sigma^{2(l)}_{i,j} = \frac{\sum_{k=1}^{N}\left(\mu_{i,k}^{(l-1)}\right)^m (\hat{\mathbf{x}}_{j,k}-v_{j,k})^2}{\sum_{k=1}^{N}\left(\mu_{i,k}^{(l-1)}\right)^m} \quad (20)$$

Estimate the consequent probability parameters,

$$p(c_i = \mathbf{y}_k | r_j) = \frac{\sum_{k|y_k=c_i}(\mu_{j,k}^{(l-1)})^m}{\sum_{k=1}^{N}(\mu_{j,k}^{(l-1)})^m} \quad (21)$$

Compute the a priori probability of the cluster and the weight (impact) of the rules:

$$P(r_i) = \frac{1}{N}\sum_{k=1}^{N}(\mu_{i,k}^{(l)})^m, \quad (22)$$

$$\mathbf{w}_i = P(r_i)\prod_{j=1}^{n}\frac{1}{\sqrt{2\pi\sigma_{i,j}^2}}$$

Step 2. Compute the distance $D_{i,k}^2(\hat{\mathbf{x}}_k, r_i)$ with (19)

Step 3. Update the partition matrix

$$\mu_{i,k}^{(l)} = \frac{1}{\sum_{j=1}^{c}\left(D_{i,k}\left(\hat{\mathbf{x}}_k, r_i\right)/D_{j,k}\left(\hat{\mathbf{x}}_k, r_i\right)\right)^{2/(m-1)}}, \quad (23)$$

$$1 \le i \le c, \quad 1 \le k \le N$$

$$\text{until } \left\|\mathbf{U}^{(l)} - \mathbf{U}^{(l-1)}\right\| < \varepsilon \quad (24)$$

B. Abonys and Szeifert fuzzy classifier

The classifier proposed by Abonyi and Szeifert [31] defines the consequent of the fuzzy rule as the probabilities of the given rule to represent the $c_1,\ldots,c_c$ classes:

$$r_i: \ \textbf{If } x_1 \textit{ is } A_{i,1}(\hat{\mathbf{x}}_{1,k}) \ \textbf{ and} \ldots x_n \textit{ is } A_{i,n}(\hat{\mathbf{x}}_{n,k}) \quad (25)$$
$$\textbf{then } \hat{y} = c_i \textit{ with } p(c_1 | r_i) \quad [w_i]$$

Similarly to Takagi-Sugeno fuzzy models [32], the rules of the fuzzy model are aggregated with the normalized fuzzy mean formula. The label of the class that has the highest activation determines the output of the classifier:

$$\hat{y}_k = \arg\max_{1\le i\le c} \frac{\sum_{l=1}^{R}\beta_l(x_k)P(c_i | r_l)}{\sum_{i=1}^{R}\beta_l(\hat{\mathbf{x}}_k)} \quad (26)$$

where

$$\beta_i(\hat{\mathbf{x}}_k) = w_i A_i(\hat{\mathbf{x}}_k) = w_i \exp\left(-\frac{1}{2}\left(\hat{\mathbf{x}}_k - \mathbf{v}_i\right)^T (\mathbf{F}_i)^{-1}\left(\hat{\mathbf{x}}_k - \mathbf{v}_i\right)\right) \quad (27)$$

$$w_i = \frac{P(r_i)}{|2\pi\mathbf{F}_i|^{n/2}} \quad (28)$$

where $\mathbf{v}_i = \left[v_{1,i},\ldots,v_{n,i}\right]^T$ denotes the center of the Gaussian functions, and $\mathbf{F}_i$ is equal to the diagonal of the matrix that contains the variances $\sigma_{i,j}^2$. Equation (27) defines how the membership functions $A_i(\hat{\mathbf{x}}_k)$ are created. These functions are generated by projecting the data of the created clusters, its centroids v_i, with the diagonal of the matrix containing the variance of the cluster, $\sigma_{j,i}^2$. The centroid of the i-th cluster will be the same as the center of the i-th Gaussian function; therefore, the number of clusters is the same as the number of functions.

C. Clustering validity.

There exist some validity techniques that offer a measure of clustering validity. This validation evaluates the quality of the clusters. In this work we utilize a validity function, *S*, proposed by Xie and Beni [33] that basically is designed to measure the overall average compactness and separation of the fuzzy partition.

1) Validity Function S.

The *S* function is defined as the relation between compactness and separation. The compactness is defined as the relation of the total shape area of the cluster between the minimum rectangle composing the cluster shape. A smaller *S* value denotes a partition in which the clusters are well compact and separated. Hence, we need to find a fuzzy partition with a small *S* value, as follows

$$S = \frac{\sum_{i=1}^{C}\sum_{j=1}^{N}\mu_{ij}^2\left\|v_i - x_j\right\|^2}{N\min_{i,j}\left\|v_i - v_j\right\|^2} \quad (29)$$

where *N* denotes the number of input vectors.

D. Scheme of the fuzzy model based on the Abonyi and Szeifert GG algorithm

The fuzzy classifier can be summarized with the following steps.

Step 1. Given a set of data $\hat{\mathbf{x}}_k$ where $\hat{\mathbf{x}}_k = \left[\mathbf{x}_k^T \ \mathbf{y}_k\right]$, $\mathbf{x}_k$ denotes the vector to be classified, and $\mathbf{y}_k$ denotes the class corresponding to $\mathbf{x}_k$. Use the GG clustering algorithm with (20 - 24), in order to obtain $\mathbf{U} = \left[\mu_{i,k}\right]_{cxN}$.

Step 2. Once we have the fuzzy matrix **U** we can start the design of the fuzzy model, similar to the Takagi-Sugeno, where the rules of the fuzzy model are added using the formula for a normalized fuzzy mean. First we calculate the activation grade of the rules β with (25). Then we compute the output of the classifier with (26). The membership functions are evaluated with (27) and the rule weight is computed with (28).

E. Testing the Abonyi and Szeifert GG algorithm

The proposed method that combines the FFL, the GG clustering algorithm and the Abonyi-Szeifert classifier was

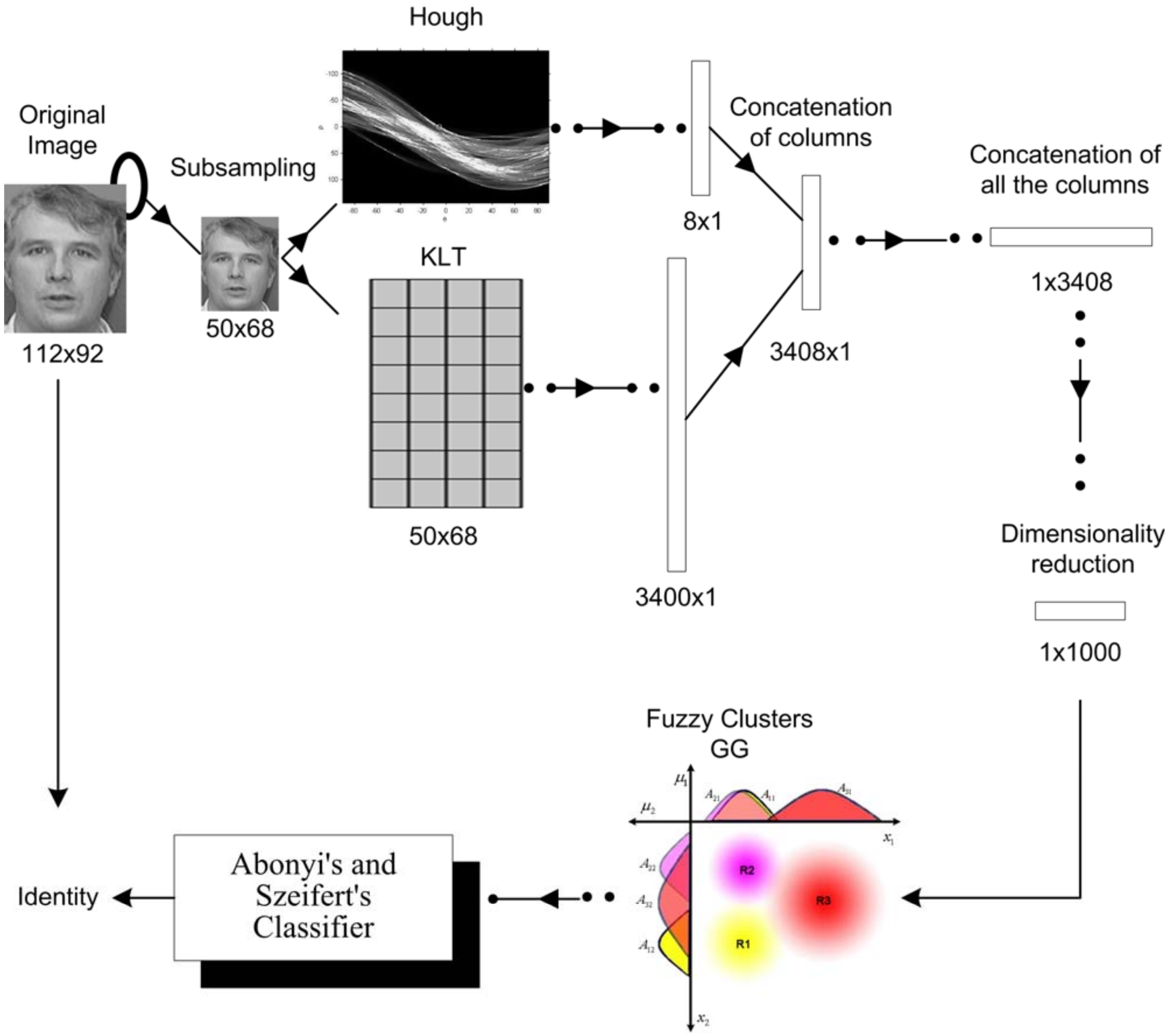


Fig. 6. General work scheme for Fuzzy Clustering using GG.

tested with the ORL database. In the design stage, 8 out of 10 faces of 40 persons were used in the GG clustering algorithm. During the verification the other two faces of each individual was submitted to the face recognition system. The system had a performance of 90% of correct classification. When the FFL are removed from the feature vector the performance drops to 88.5%. Also the proposed method was tested with the Yale database. In the design stage, 8 out of 10 faces of 10 persons were used. During the verification the other two faces of each individual was submitted to the face recognition system. The system had a performance of 89% of correct classification. When the FFL are removed from the feature vector the performance drops to 85%.

V. Conclusion

This paper described a new approach for face recognition. The method proposed a new feature called face feature lines. Incorporation of these features is based on studies related to face recognition on newborns. The feature vector used to achieve face recognition is a combination of the face feature lines and eigenvectors computed by the KLT method.

Two face recognitions systems were developed. One based on the Euclidean distance of the feature vector in the KLT space, and other based on fuzzy clustering and classification. The Euclidean distance classifier showed a correct classification performance of 91% and 88% using 25 eigenvectors over ORL and Yale databases respectively; and 95% and 90% if the face feature lines are included over ORL and Yale databases respectively. The fuzzy clustering and classification systems, illustrated in Fig. 6, had a 88.5% and 86% of correct classification using a feature vector with 25 eigenvectors over ORL and Yale databases respectively; and 90% of correct classification for both ORL and YALE databases if the face feature lines are included in the feature vector. The Table V and Table VI summarizes these results.

TABLE V
CLASSIFIERS PERFROMANCES ON THE ORL DATA BASE

CLASSIFIER	No FFL	FFL
EUCLIDEAN	91%	95%
FUZZY CLUSTERING	88.5%	90%

TABLE VI
CLASSIFIERS PERFROMANCES ON THE YALE DATA BASE

CLASSIFIER	No FFL	FFL
EUCLIDEAN	88%	90%
FUZZY CLUSTERING	86%	90%

From Table V and VI we can notice that the FFL provide more discriminative power to the feature vector since the performance of the classifiers are better with the FFL. Although the Euclidean classifier has a better performance the fuzzy method has the advantage that is less computational expensive.

That is, in the Euclidean classifier the unknown sample is compared against all samples of all individuals in the data base, in this work 320. In the fuzzy classifier the unknown sample is only compared with the representation, the cluster, of each individual, in this case 40.

Cluster validity on the clusters generated by the GG algorithm was evaluated with the S validity function. Table VII and Table VIII illustrates how the cluster validity function accomplishes its best value when the number of cluster is equal to the number of actual classes, in this case 40 for ORL and 10 for YALE. We can also observe from Table VII that the highest performance is accomplished when S is equal to 40, as it was expected, and from Table VIII that the highest performance is accomplished when S is equal to 10, as it was expected too.

TABLE VII
CLUSTER VALIDITY AND CLASSIFIER PERFROMANCES ON ORL

	C	S	ERRORS	% PERFORMANCE
EXPERIMENT 1	38	51.98	10	75
EXPERIMENT 2	39	33.95	7	82.5
EXPERIMENT 3	40	1.135	4	90
EXPERIMENT 4	41	1.354	6	85
EXPERIMENT 5	42	19.24	13	67.5

TABLE VIII
CLUSTER VALIDITY AND CLASSIFIER PERFROMANCES ON YALE

	C	S	ERRORS	% PERFORMANCE
EXPERIMENT 1	8	15.98	7	87.5
EXPERIMENT 2	9	17.95	6	88
EXPERIMENT 3	10	5.5	2	90
EXPERIMENT 4	11	8.34	5	87
EXPERIMENT 5	12	13.81	10	86

ACKNOWLEDGMENT

The authors appreciate the support of COSNET, and SEP-DGEST for the support of this research under grant 445.05-P. Also CONACyT for the support of this research under grant #193324.

REFERENCES

[1] X. Lu, and A. K. Jain, “Deformation Analysis for 3D Face Matching”, Proceedings of the Seventh IEEE Workshop on Applications of Computer Vision, 2005.
[2] H. Cevikalp, M. Neamtu, M. Wilkes, A. Barkana, “Discriminative Common Vectors for Face Recognition”, IEEE Transactions on Pattern Analysis and Machinery Intelligence, Vol. 27, No. 1, Jan 2005.
[3] 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.
[4] T. Kim, J. Kittler, “Locally Linear Discriminant Analysis for Multimodally Distributed Classes for Face Recognition with a Single Model Image”, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 27, No. 3, Marzo 2005.
[5] P. C. Yuen, D. Q. Dai, G. C. Feng, “Wavelet-based PCA for Human Face Recognition”, Proceeding of IEEE Southwest Symposium on Image Analysis and Interpretation, Abril 1998, pp.223 - 228.
[6] R. Gross, I. Matthews, and S. Baker, "Appearance-Based Face Recognition and Light-Fields," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 26, No. 4, April, 2004, pp. 449 - 465.
[7] B. Bower, “Faces of Perception”, *Science News*, Vol. 160, No. 1. Jul. 2001. pp. 10.
[8] A. Slater, P. C. Quinn, “Face recognition in the newborn infant”, *Infant and Child Development*, Vol. 10, Mar. 2001, pp. 21.
[9] B. Moghaddam, A. Pentland, “Probabilistic Matching for Face Recognition”, IEEE Southwest Symposium on Image Analysis and Interpretation, April 1998. pp. 186 – 191.
[10] M. Zhang, J. Fulcher, “Face Recognition Using Artificial Neural Network Group-Based Adaptive Tolerance (GAT) Trees”, IEEE Transactions on Neural Networks, Vol. 7, No. 3, Mayo 1996. pp. 555 – 567.
[11] 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, Enero 2004. pp. 166 – 177.
[12] W. Zheng, J. Lai, P. C. Yuen, “GA-Fisher: A New LDA-Based Face Recognition Algorithm With Selection of Principal Components”, IEEE Transactions on Systems, Man, and Cybernetics—Part B: Cybernetics, Vol. 35, No. 5, Octubre 2005. pp. 1065 – 1078.
[13] Z. Li and X. Tang, “Bayesian Face Recognition Using Support Vector Machine and Face Clustering”, Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’04), 2004.
[14] R. Gross, I. Matthews, and S. Baker, “Fisher Light-Fields for Face Recognition Across Pose and Illumination”, 2002.
[15] M. A. Turk, “Interactive-Time Vision: Face Recognition as a Visual Behavior”, Ph.D. dissertation, Media Arts and Sciences Section, School of Architecture and Planning, Massachusetts Institute of Technology, Sep 1991.
[16] X. Xie, and K. Kam, “Face Recognition Under Varying Illumination Based on a 2D Face Shape Model”, Pattern Recognition. Vol. 38, 2005, pp. 221-230.
[17] K. Kwak, and W. Pedryez, “Face Recognition Using a Fuzzy Fisherface Classifier”, Pattern Recognition. Vol. 38, 2005, pp. 1717-1732.
[18] M. Turk and A. Pentland, “Face recognition using eigenfaces,” in Proc. IEEE Conf. Computer Vision Pattern Recognition, 1991, pp. 586 – 591.
[19] D. Liu, K. Lam, and L. Shen, “Illumination Invariant Face Recognition”, Pattern Recognition. Vol. 38, 2005, pp. 1705-1716.
[20] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, “Eigenfaces vs. Fisherfaces: Recognition using class specific linear projection,” *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 19, July 1997, pp. 711–720.
[21] R. C. Gonzalez, R. E. Woods, S. L. Eddins, “*Digital image processing using Matlab*”, New Jersey, Pearson Prentice Hall, 2004
[22] .G. M. J. Ramírez, “Estudio de métodos de reconocimiento de patrones basado en redes neurales y lógica difusa”, M.S. dissertation, Grad. Studies & Research Div., Chihuahua Institute of Technology, Chihuahua, Mexico, 2005.
[23] S. Z. Li and A. K. Jain, Handbook of face recognition, Springer, USA, 2004.
[24] “Olivetti Research Lab (ORL) Database”, On-Line: http://www.uk.research.att.com/facedatabase.html

[25] [1] D. M. Gómez-Allende, "*Reconocimiento de Formas y Visión Artificial*", Addison Wesley, USA, 1993.

[26] J. Abonyi, R. Babuska, F. Szeifert, "*Modified Gath-Geva fuzzy clustering for identification of Takagi-Sugeno fuzzy models"*, IEEE Transactions on Systems, Man and Cybernetics, Part B, Vol. 32, Issue: 5, Oct. 2002, pp:612 – 621.

[27] M. Setnes, R. Babuska, U. Kaymak, and H.R Van Nauta Lemke, "*Similarity measures in fuzzy rule base simplification",* IEEE Transactions on Systems, Man and Cybernetics, Part B, Vol. 28, Issue: 3 , June 1998, pp:376 – 386.

[28] N. R. Pal, K. Pal, J. M. Keller, and J. C. Bezdek, "A Possibilistic Fuzzy c-Means Clustering Algorithm", IEEE Transactions on Fuzzy Systems, Vol. 13, No. 4, August 2005. pp. 517-530.

[29] R. J. Hathaway, and J. C. Bezdek, "Fuzzy c-Means Clustering of Incomplete Data", IEEE Transactions on Systems, Man, and Cybernetics—Part B: Cybernetics, Vol. 31, No. 5, October 2001. pp. 735-744.

[30] R. J. Hathaway, J. C. Bezdek, and Y. Hu, "Generalized Fuzzy c-Means Clustering Strategies Using Lp Norm Distances", IEEE Transactions on Fuzzy Systems, Vol. 8, No. 5, October 2000. pp. 576-582.

[31] J. Abonyi and F. Szeifert, *"Supervised fuzzy clustering for the identification of fuzzy classifiers"*, Pattern Recognition Letters, Vol. 24, iss(14), October 2003, pp:2195-2207(13).

[32] Takagi T., Sugeno M. (1985) Fuzzy identification of systems and its application to modeling and control. IEEE Trans. SMC 15, 116–132.

[33] X.L.Xie and G.Beni, "*A validity measure for fuzzy clustering"*, IEEE Transaction Pattern Anal. Machine Intelligence, 13(8), 1991, pp. 841-847.

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 Aerospace and Electronic Systems, Broadcast Technology, Circuits and Systems, Communications, Computational Intelligence, Computer, Microwave Theory and Techniques, and Signal Processing, Industrial and Applied Mathematics. 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. Eng. Rivas Perea has received several awards in concern to the creativity and to the academic areas.