Translation, Rotation and Scale Invariant Object Recognition

L. A. Torres-Méndez[1], J. C. Ruiz-Suárez[2] and L. E. Sucar[1]

[1]Instituto Tecnológico y de Estudios Superiores de Monterrey
Campus-Morelos
Apdo. Postal 99-C, Cuernavaca Morelos
62050, México.

[2]Departamento de Física Aplicada, CINVESTAV-IPN
Unidad Mérida, A. P. 73 Cordemex, 97310 Mérida
Yucatán, México.

Abstract

A method for object recognition invariant under translation, rotation and scaling is addressed. The first step of the method (preprocessing) takes into account the invariant properties of the normalized moment of inertia and a novel coding that extracts topological object characteristics. The second step (recognition) is achieved by using a Holographic Nearest Neighbor algorithm (HNN), where vectors obtained in the preprocessing step are used as inputs to it. The algorithm is tested in character recognition, using the 26 upper case letters of the alphabet. Only four different orientations and one size (for each letter) were used for training. Recognition was tested with 17 different sizes and 14 rotations. The results are encouraging since we achieved 98% correct recognition. Tolerance to *manual* and random noise was tested. Results for character recognition in *real* images of car plates are also presented.

Keywords - Invariant object recognition, character recognition, holographic nearest neighbor.

1 Introduction

Invariant object recognition (IOR), whose aim is to identify an object independently of its position (translated or rotated) and size (larger or smaller), has been the object of an intense and thorough study. In the last several years, an increasing number of research groups have proposed a great variety of IOR methods. Among them, we can find a number of optical techniques [16, 6], boundary-based analysis via Fourier descriptors [8, 11], neural networks models [1, 5, 7, 18, 17], invariant moments [12, 10, 2], and genetic algorithms [13]. However, most of these methods are too computationally expensive or are not invariant under the 3 types of transformations: scaling, rotation and translation.

A number of IOR methods have been proposed in the literature. These can be classified as: optical techniques [16, 6], boundary-based analysis via Fourier descriptors [8, 11], neural networks models [1, 5, 7, 18, 17], invariant moments [12, 10, 2], and genetic algorithms [13].

It is important to mention recent IOR research based on optical techniques such as composite harmonic filters [16] or STIR invariant transformations [4]. The former filters involve the Mellin radial harmonics for scale invariance [15], the logarithmic harmonics for projection invariance [14], and the circular harmonics for rotation invariance [9]. Fang and Hausler [4] introduced a new class of transforms that achieve scale, translation and in-plane rotation (STIR) invariance, simultaneously. In their approach, an intensity function $S(x, y)$ is mapped into a one-dimensional frequency spectrum function. Later, Ghahramani and Patterson [6] proposed a higher-dimensional version of the STIR invariant transforms in conjunction with an orthonormalization technique in an optical neural network resonator. Computer simulations show that these type of techniques perform well and have excellent noise tolerance. However, the major disadvantage is their heavy computational requirements.

Boundary-based analysis using discrete Fourier transforms has been proposed as alternative in IOR [8, 11]. Algorithms based on this kind of analysis are called Fourier descriptors and basically, invariance is obtained by normalizing the frequency representation of the image shape. This is easily done via the discrete Fourier transform properties but only on uniform contours of the shape.

Madaline structures for translation-invariant recognition [1], the self-organized neocognitron [5], and high-order neural networks [7, 18, 17], are examples of IOR neural based methods. The self-organized neocognitron is a further extension of the cognitron, originally proposed by Fukushima in 1975 [5]. This learning machine has the ability to learn with no teacher and obtains, when learning is completed, a structure similar to the hierarchical model of the visual nervous system. Although the work of Fukushima is a major advance in the understanding of visual processing in our brain, from an engineering point of view its major drawback is that it is unable to cope with large translations and rotations in the image. Furthermore, the number of cells in this model increases almost linearly with the number of objects to be recognized, making the training process very slow.

High Order Networks (HON) have been utilized recently for invariant recognition [7, 18]. In this type of model, one has to encode in the values of the synaptic weights the properties of invariance. In other words, the known relations between pixels of the images are used and the invariance is directly constructed in the network. A third order network has been proposed [17], in which combinations of triplets of image pixels are used as invariant relations. The triplets form triangles representing similar angles (α, β, γ) in any transformation of the same image. The weights are restricted in such way that all the combinations of three pixels defining similar triangles are connected to the output with the same weight. The number of combinations of possible triplets increases in a nonlinear proportion to the number of input data, this being the main disadvantage of this approach.

IOR based on moments and invariant functions of moments, is another popular invariant recognition scheme. In 1962 Hu [10], introducing nonlinear combinations of regular moments derived a set of seven composed moments with translation, scaling and rotation invariant properties. However, the moments proposed by Hu do not posses orthogonal properties making reconstruction of the input image computationally difficult. To overcome this problem, Teague [20] suggested orthogonal moments based on the general properties of orthogonal polynomials. In general, it has been shown by Teague and other researchers [21], that in terms of information redundancy, orthogonal moments (Legendre, Zernike and pseudo-Zernike) perform better than any other type of moments. In terms of overall performance, Zernike and pseudo-Zernike moments outperform the others [12]. But in general, the main disadvantage of using these methods is that the moment computation is too computationally intensive.

A genetic classifier system, able to correctly classify all the letters of the alphabet has been proposed by McAulay et al. [13]. This classifier system has only scaling and translation invariant properties and some robustness against certain distortions and noise. Finding an efficient mapping of the 2-D image into the classifier system rules is one of the main difficulties of this approach. Watanabe [23] proposed a direct coding using four strings representing projected views in 4 directions: horizontal, vertical, ordinary diagonal, and auxiliary diagonal. These strings are formed considering the number of runs of blacks in each row, in the corresponding direction string, and are compressed to show only variations. This coding is an efficient way to extract topological characteristics but it is only invariant to scaling and translation, not to rotation.

In this contribution we report a simple method for object recognition that achieves excellent invariance under translation, rotation and scaling. The method has two steps: preprocessing and recognition. The first takes into account the moment of inertia of the object and a novel coding that extracts topological object characteristics. The second step is done by using a Holographic Nearest Neighbor algorithm (HNN), where vectors obtained in the preprocessing stage are used as inputs to it. Learning and recall with the HNN algorithm is extremely fast. Initially we consider two dimensional (2D) binary images and tested our algorithm for invariant character recognition. The method could be easily extended for multilevel images and we present some results in recognition of characters in real images (grey scale) of car plates.

In section 2 we will describe the preprocessing stage of our model. In section 3 we introduce the Holographic Nearest Neighbor algorithm and discuss the way this is used in IOR. In section 4 we will present results that show the ability of our model to recognize the 26 letters of the alphabet regardless of size and position (translation and rotation). In section 5 we briefly describe related work in IOR. Conclusions and future work will be the object of last section.

2 Preprocessing

In invariant pattern recognition models, preprocessing is defined as the extraction of appropriate invariant features that are then used for recognition by a classification system. The invariant features in our work are real numbers that are fed as vectors to the classification system. Figure 1 illustrates the way that strings are created.

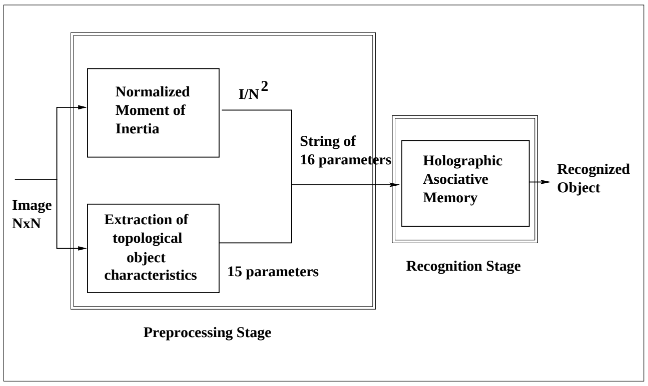

Figure 1: General diagram showing the steps of the Invariant Object Recognition.

The moment of inertia of the image is first calculated. In general, the moment of inertia quantifies the inertia of a rotating object considering its mass distribution. The moment of inertia is normally calculated by dividing the object into N small pieces of mass m_1, m_2, ... m_N. Each piece is at a distance, r_1, r_2, ..., r_N, from the axis of rotation. The moment of inertia of the object is:

$$I = m_1 r_1^2 + m_2 r_2^2 + ... + m_N r_N^2$$

The moment of inertia depends on the position of the axis of rotation and on the shape and mass of the rotating object and is invariant under translation and rotation. A bidimensional image is not an object with mass, but we can represent it by a continuous function f(x,y), where each pixel of the image can be considered as a particle with mass equal to the value of the intensity of the pixel. For binary images, the moment of inertia with respect to the image centroid (central moment of inertia) is:

$$\mathbf{I} = \sum_{i=1}^{N} d_i^2 = \sum_{i=1}^{N} ((x_i - C_x)^2 + (y_i - C_y)^2) \qquad (1)$$

where C_x, C_y are the centroid coordinates, x_i, y_i the image pixel coordinates of the object and N the total number of pixels.

Translation and rotation invariance is achieved by calculating the central moment of inertia. On the other hand, by dividing $\mathbf{I}$ by N^2 (we will name it $\mathbf{I_N}$), scaling invariance is achieved. Is was found empirically that dividing by N^2 gives better results (in recognition) that dividing just by N [22]. It is also worth remarking that due to the finite resolution of any digitized image, a rotated object may not conserve intact the number of pixels, so $\mathbf{I}$ may vary. Using $\mathbf{I_N}$ reduces this problem, too.

The possibility that two or more different images have the same, or very close, $\mathbf{I_N}$ may generate a real problem for classification. To circumvent this problem, we generalize the idea of Watanabe [23], and propose a simple heuristic method able to extract invariant topological characteristics. This is based on the fact that the circle is the only geometrical shape that is naturally and perfectly invariant to rotation (in 2D). The first part of the heuristic considers the number of intensity pixel changes in a circle of some radius inside the object as it crosses it. This simple coding scheme extracts the topological characteristics of the object regardless its position, orientation, and size. Moreover, to obtain a more robust representation we use several proportionally arranged circles over each object. However, in some cases two different objects could have the same or very similar radial coding (for example, letters **M** and **N**). In the second part of the heuristic we also take into account the difference in size of the two largest arcs (for each circle) that are not part of the object. For achieving size normalization, we divide this difference by the total number of pixels in the circle.

The methodology to obtain the radial coding of a binary 2D object con be summarized as follows:

1. Obtain the centroid of the object.

2. Generate K equidistant concentric circles, C_i, around the centroid.

3. For each circle, count the number of intensity changes (0 to 1 or 1 to 0) that occur in the image, this is R_i.

4. For each circle, obtain the two largest arcs that are not part of the object (we assume a known value for object and background). Measure each arc by counting the number of pixels, obtain the difference and divide by the size of the circle. This is: $D_i = (d_1 - d_2)/d_C$, where d_1 is the length of the largest arc, d_2 is the length of the second largest arc, and d_C is the length of the circle.

The radial coding can be represented by the following vector:

$$R_1, R_2, ..., R_K, D_1, D_2, ..., D_K,$$

considering K circles. R_i is positive integer and D_i is a real value in [0,1]. Figure 2 shows this coding scheme for different sizes and orientations of letter **E**.

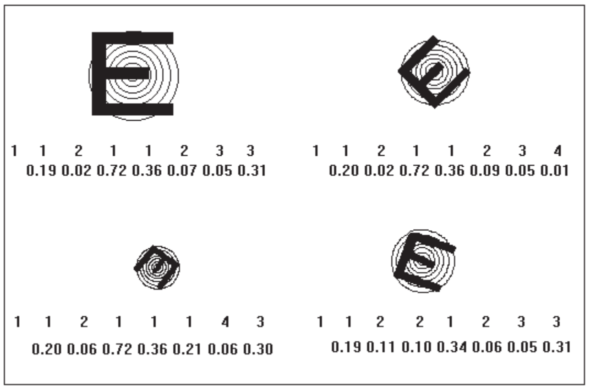

Figure 2: Extraction of topological characteristics of different rotations and sizes of letter E. The first line indicates the number of intensity pixel changes in each one of the 8 circles, and the second line represents the normalized differences over the largest 7 circles.

In summary, we define 3 sets of invariant features: (i) normalized central moment of inertia, (ii) radial coding, and (iii) differential radial coding; to be obtained in the preprocessing stage. All these features are invariant to translation, rotation and scaling; and together provide a robust feature set for the recognition or classification stage.

3 Holographic Nearest Neighbor

The recognition stage uses a Holographic Nearest Neighbor (HNN) algorithm that is based on the principles of the Holographic Associative Memory (HAM) [19]. The main motivations for using this technique are:

- Unlike other neural network architectures, learning and recall with the HAM is *very* fast (see next section).
- The HNN algorithm has, in general, a better performance (recognition rate) that a simple nearest neighbor technique.

As in optical holography, the Holographic Nearest Neighbor algorithm bases its operation on the principle of unfolding information of different phases in a single plane, see [19].

The external data field is represented by the stimulus-response set $\mathbf{S}$:

$$\mathbf{S} = (s_1, s_2, ...s_M, s_{M+1}),$$

where s_i are scalars defining the stimulus-response field, M is the number of input variables and s_{M+1} is the associated response.

Each input real variable is mapped to polar variables by means of the sigmoidal relationship:

$$\theta_i = 2\pi(1 + e^{(\mu - s_i)/\sigma})^{-1} \tag{2}$$

where μ and σ are the mean and standard deviation for each of these variables. Equation 2 maps each s_i to phase values between 0 and 2π.

The HNN algorithm is similar to the well known Nearest Neighbor algorithm (NN) [12]. This later algorithm is based on the fact that the minimum Euclidean distance between the input and each training vector gives the best answer for each class. Moreover, to prevent the domination of a subgroup of features, the NN algorithm normalizes these features. The normalization consists of subtracting from each variable the mean and dividing the result by the standard deviation of the corresponding class.

In our model, we calculate the mean (μ) and standard deviation (σ) for each variable on the complete set of training vectors (all classes). Furthermore, once μ and σ are evaluated, the real components of each vector are mapped to a phase domain. Thus, we end up with q phase vectors of dimension M. When a new pattern is presented to the HNN, it decides which is the best match by finding the minimum distance between that new pattern and training phases for each variable. More clearly, among the q phase vectors we find the minimum of $\sqrt{\sum_{i=1}^{M}(\theta_{exp}^{i} - \theta_{t}^{i})^2}$.

Working with phases instead of real numbers, and calculating the mean (μ) and standard deviation (σ) for each variable on the complete set of training vectors, makes the HNN algorithm slightly superior to the normal NN algorithm [22].

4 Experimental Study and Results

The algorithm was tested in character recognition using the 26 upper case letters of the alphabet. Four different orientations and only one size were used for training. Recognition was tested with 17 different sizes and 14 rotations for each size.

4.1 Learning

In order to obtain an increased noise tolerance we consider, during the learning stage, four different orientations for each character. Thus, we have only 104 input patterns (26×4) for the learning process. Each input vector is formed by 17 positive numbers ($\mathbf{M} = 16$), using in this case 8 circles for the radial coding. The first one is the normalized central moment of inertia ($\mathbf{I}_N$), the next eight are the number of intensity pixel changes when the 8 circles are intersected by the letter ($R_0 ... R_7$), the next seven are the normalized differences of the largest seven circles ($D_0 ... D_6$) and finally, the last number is the letter identifier. As an example, Table 1 shows the training vectors for letters **A, F, M and T**.

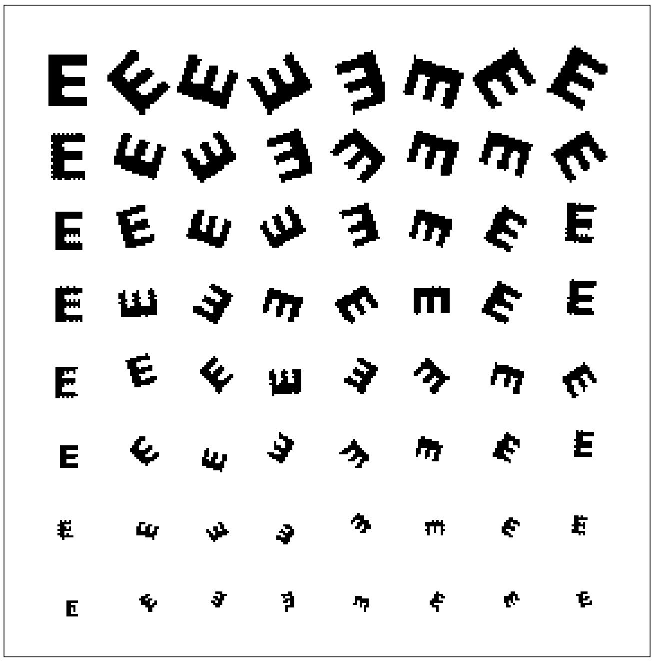

Figure 3: Some testing images of letter E.

4.2 Recognition

We generated 238 experimental images (17 different sizes and 14 different rotations) for each one of the 26 letters. The largest letter has 100 x 100 pixels, the smallest 20 x 20 pixels. Figure 3 shows few of the generated images of letter **E**.

During recognition, each one of the 238 images for each one of the 26 characters, is presented to the HNN. It is important to mention that most of the characters used to test the method present certain degree of noise or deformation. This noise is intrinsically produced during the transformation of the letters to other sizes and orientations. Moreover, we also add different amounts of noise to the letters to be recognized. The noise was applied in *manual* form (Fig. 4) or randomly (Fig. 5). The *manual noise* simulates the effect of deformations in the object boundary, although it is difficult to measure the exact degree of deformation. The random noise is generated by changing the value of pixels inside the object in a random (uniform) manner, the percentage of altered pixels is varied from 10% to 90%.

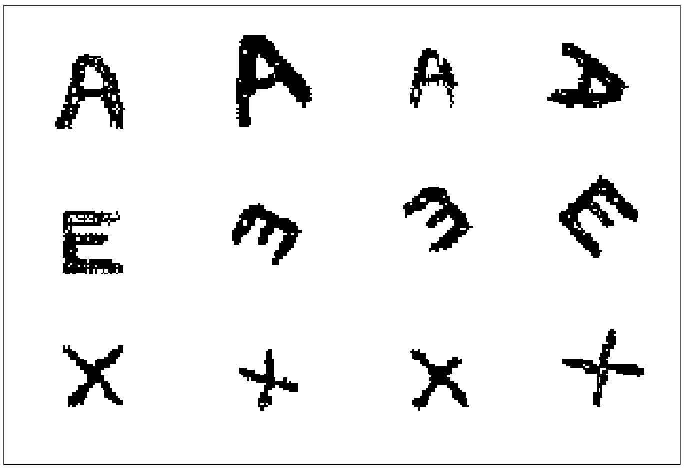

Figure 4: Letters A, E, and X with some degrees of noise recognized with 100% accuracy.

4.3 Results

The obtained results are shown in table 2. As we can see, invariant object recognition is obtained with almost 100% accuracy on images with sizes between 100x100 and 45x45 pixels. The performance of the model decreases slightly for smaller letters. However, this problem could be solved by adding a second group of training vectors for letter sizes between 30x30 and 20x20 pixels. We used another 104 training patterns for small letters and the performance for this range (25x25–20x20) improves [22].

Noisy images (see Figs. 4 and 5) are also correctly recognized by our model, indicating the robustness of it. An accuracy of more than 98% is obtained with images having up to 60% of random noise.

As it was mentioned before, both preprocessing and recognition are computationally efficient. The system was tested in an IBM RS6000-SP2 workstation with the following results. The mean time for the preprocessing stage, including the moment of inertia and the radial coding, is 0.105462 seconds, with a standard deviation of 0.0187. The mean time for recognition using the HNN algorithm is 0.108846 seconds, with a standard deviation of 0.03449. So the total mean time for IOR in our experiments is approx. 0.2 seconds and is very stable. The training time for the HNN is the same as the recognition time for each pattern used for training, that is approx. 0.1 seconds times the number of training patterns.

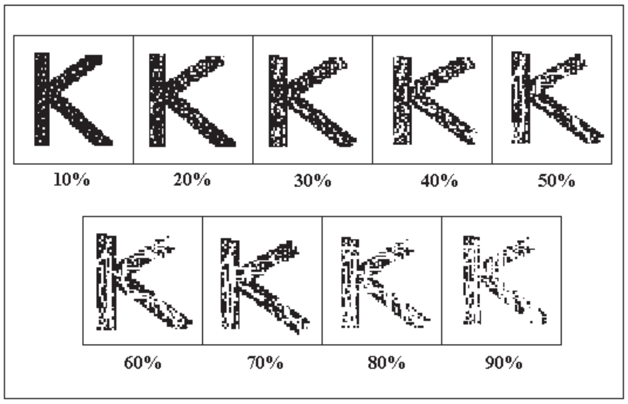

Figure 5: Different percentages of random noise applied to letter K.

4.4 Multilevel Images

The methodology for IOR can be extended from binary images to multilevel images. For this, before the IOR method is applied, the image is made binary by a technique based on the intensity histogram of the image. Assuming a single object in the image, the intensity histogram will usually be bimodal, with one peak corresponding to the object and other to the background. So we apply a simple histogram–based segmentation (binarization) using a dividing threshold at the following intensity value [3]:

$$Th = m - k \times (max - min)$$

where Th is the threshold, m is the median of the intensity values, max is the maximum and min is the minimum. k is a constant that is obtained empirically.

5 Conclusions

In the present work, a novel method to recognize 2-dimensional objects invariant to size, translation and rotation, is reported. This method takes advantage of the properties of the normalized central moment of inertia and a coding scheme which extracts invariant topological characteristics of two-dimensional objects. The method is easy to understand and implement, and compared to other methods, computer requirements are negligible.

Recognition is made by means of a holographic nearest neighbor algorithm. In the work reported here, only 104 patterns representing 26 letters are used for training. The HNN is able to correctly recognize, regardless of its position, orientation, and size, any upper case letter with 17 different sizes and 14 different rotations, including some noise added randomly or *manually.* The method was extended to handle multilevel images and applied for character recognition in images of car plates.

In summary, we have described a IOR model which is robust, computationally fast and easy to implement.

Future work will be done in testing the model with different 2D objects, and in using other classification techniques in the recognition stage.

References

[1] M. Caudill, Neural Networks Primer Part II, AI Expert-Neural Networks Premier, 9-15, (1994).

[2] Cho-Huak Teh and R. T. Chin, On Image Analysis by the Methods of the Moments, IEEE Electrical and Computer Engineering Department, University of Wisconsin, 556-561, (1988).

[3] E.B. Davies, Machine Vision, Academic Press, 1990.

[4] M. Fang and G. Hausler, Class of transforms invariant under shift, rotation, and scaling, Appl. Opt. **29,** 704-708 (1990).

[5] K. Fukushima, Neocognitron: A Self-organizing Neural Network Model for a Mechanism of Pattern Recognition Unaffected by Shift in Position, Biol. Cybernetics **36,** 193-202 (1980).

[6] E. Ghahramani, and L. R. B. Patterson, Scale, translation, and rotation invariant orthonormalized optical/optoelectronic neural networks, Appl. Opt. **32,** 7225-7232 (1993).

[7] G. L. Giles and T. Maxwell, Learning, invariances, and generalization in high-order neural networks, Appl. Opt. **26,** 4972-4978 (1987).

[8] R. Gonzalez and P. Wintz, Digital Image Processing, 2nd. Ed., Addison-Wesley Publishing Co., 404-411, 1987.

[9] Y. N. Hsu, and H. H. Arsenault, Optical pattern recognition using the circular harmonic expansion, Appl. Opt. **21,** 4016-4019 (1982).

[10] M. K. Hu, Visual Pattern Recognition by Moment Invariants, IRE Trans. Inform. Theory **8,** 179-187 (1962).

[11] H. Kauppinen, T. Seppänen, and M. Pietikäinen, An Experimental Comparison of autoregressive and Fourier-Based Descriptors in 2D Shape Classification, IEEE Trans. Pattern Anal. Machine Intell, **17,** 210-207, (1995).

[12] A. Khotanzad and Yaw Hua Hong, Invariant Image Recognition by Zernike Moments, IEEE Trans. Pattern Anal. Machine Intell, **12,** 489-497, (1990).

[13] A. D. McAulay and J. C. Oh, Improving Learning of Genetic Rule-Based Classifier Systems, **24,** 152-159 (1994).

[14] D. Mendlovic, N. Konforti, and E. Marom, Shift and projection invariant pattern recognition using logarithmic harmonics, Appl. Opt. **29,** 4784-4789 (1990).

[15] D. Mendlovic, E. Marom, and N. Konforti, Shift and scale invariant pattern recognition using Mellin radial harmonics, Opt. Commun. **67,** 172-176 (1988).

[16] D. Mendlovic, Z. Zalevsky, I. Kiryuschev, and G. Lebreton, Composite harmonic filters for scale-, projection-, and shift-invariant pattern recognition, Appl. Opt. **34,** 310-316 (1995).

[17] S. J. Perantonis and P. J. G. Lisboa, Translation, Rotation, and Scale Invariant Pattern Recognition by High-Order Neural Networks and Moment Classifiers, IEEE Trans. Neural Networks **3,** 241-251 (1992).

[18] L. Spirkovska and M. B. Reid, Coarse-coded Higher-order Neural Networks for PSRI Object Recognition, IEEE Trans. Neural Networks **4,** 276-283 (1993).

[19] Sutherland J. G., The Holographic neural method, in Fuzzy, Holographic and Parallel Intelligence, John Wiley and Sons, New York, 1992.

[20] M. R. Teague, Image Analysis via the General Theory of Moments, J. Opt. Soc. Amer. **70,** 920-930 (1980).

[21] C. H. Teh and R. T. Chin, On image analysis by the method of moments, IEEE Trans. Pattern Anal. Machine Intell. **10,** 496-513 (1988).

[22] Torres-Mendez L. A., Invariant 2-D Object Recognition, M. Sc. Thesis, ITESM-Campus Morelos, Mexico, 1995 (in Spanish).

[23] S. Watanabe, Pattern Recognition (Human and Mechanical). New York, Wiley, 1985.

I/N^2	Radial coding								Normalized differences							Id
LETTER A																
0.34	1	1	2	2	1	3	3	2	0.43	0.05	0.26	0.31	0.09	0.00	0.29	65
0.40	1	1	2	2	1	3	3	2	0.39	0.07	0.23	0.31	0.11	0.02	0.42	65
0.36	1	1	2	2	1	3	3	2	0.42	0.06	0.26	0.31	0.09	0.00	0.43	65
0.35	1	1	2	2	1	3	3	2	0.41	0.09	0.25	0.31	0.11	0.01	0.43	65
LETTER F																
0.34	0	1	2	2	2	2	1	1	0.57	0.16	0.04	0.10	0.19	0.66	0.70	70
0.40	0	1	2	2	3	4	2	2	0.49	0.11	0.05	0.11	0.01	0.10	0.05	70
0.37	0	1	2	2	3	4	2	2	0.51	0.15	0.06	0.11	0.00	0.06	0.03	70
0.36	0	1	2	2	3	4	2	2	0.48	0.11	0.05	0.11	0.00	0.18	0.10	70
LETTER M																
0.27	0	1	1	2	3	3	2	1	0.28	0.21	0.11	0.03	0.04	0.04	0.05	77
0.31	0	1	1	1	3	3	2	4	0.28	0.22	0.20	0.05	0.04	0.04	0.01	77
0.29	0	1	1	1	3	3	2	4	0.28	0.22	0.19	0.04	0.04	0.04	0.01	77
0.28	0	1	1	1	3	3	2	4	0.29	0.21	0.19	0.03	0.05	0.04	0.02	77
LETTER T																
0.42	1	1	2	2	2	2	2	3	0.00	0.01	0.01	0.00	0.00	0.00	0.03	84
0.49	1	1	2	2	3	3	3	3	0.00	0.01	0.01	0.01	0.00	0.02	0.01	84
0.45	1	1	2	2	3	3	3	3	0.00	0.01	0.02	0.01	0.00	0.03	0.00	84
0.45	1	1	2	2	3	3	3	3	0.21	0.02	0.00	0.01	0.02	0.01	0.02	84

Table 1: Training vectors of letters A (65), F (70), M (77), and T(84) in 4 different rotations: 0, 35, 70, and 105 degrees.

Size	*# Letters*	A	B	C	D	E	F	G	H	I	J
100x100-45x45	168	100	100	98	100	98	100	100	100	100	100
40x40-30x30	42	100	100	98	96	90	88	88	100	100	100
25x25-20x20	28	100	98	98	94	95	89	89	100	100	99

K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
100	98	100	99	99	96	100	98	99	98	100	100	96	100	100	99
100	95	100	98	100	95	90	100	91	96	100	94	94	95	100	92
93	50	32	99	98	96	97	100	89	89	89	94	100	100	98	95

Table 2: Average percent of recognition for each one of the 26 letters for different sizes: between 100x100 to 45x45 pixels, 168 testing letters, between 40x40 to 30x30 pixels, 42 testing letters, and between 25x25 to 20x20 pixels, 28 testing letters.