Translation, Rotation, and Scale-Invariant Object Recognition

L. A. Torres-Méndez, J. C. Ruiz-Suárez, Luis E. Sucar, and G. Gómez

***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), in which 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 boundary deformations and random noise was tested. Results for character recognition in "real" images of car plates are presented as well.**

***Index Terms*—Character recognition, holographic nearest neighbor, invariant-object recognition.**

I. 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 [6], [17], boundary-based analysis via Fourier descriptors [9], [12], neural-network models [1], [5], [7], [18], [19], invariant moments [2], [11], [13], and genetic algorithms [14]. However, most of these methods are too computationally expensive or are not invariant under the three types of transformations: scaling, rotation, and translation.[1]

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 considered two dimensional (2-D) binary images and tested our algorithm for invariant-character recognition. The method could be extended easily for multilevel images, and we present the results in recognition of characters in the real images (grey scale) of car plates.

In Section II, we will describe the preprocessing stage of our model. In Section III, we introduce the HNN algorithm and discuss the way this is used in IOR. In Section Iv, 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 V, we briefly describe related work in IOR. Conclusions and future work will be the object of the last section.

Manuscript received October 23, 1996; revised June 8, 1998 and May 31, 1999. This work was supported in part by CONACT, Mexico.

L. A. Torres-Méndez is with McGill University, Montreal, PQ, Canada H3A 2A7 (E-mail: latorres@cim.mcgill.ca).

J. C. Ruiz-Suárez is with the Department of Applied Physics, CINVESTAV-IPN, Unidad Mérida, Yucatán, México (E-mail: cruiz@kin.creamer.conacyt.mx).

L. E. Sucar and G. Gómez are with the Department of Computing, Instituto Tecnológico y de Estudios Superiores de Monterrey, Campus Morelos, Cuernavaca Morelos, México (e-mail: esucar@campus.mor.itesm.mx; gegomez@campus.mor.itesm.mx).

Publisher Item Identifier S 1094-6977(00)00365-5.

[1]A brief description of other methods is presented in Section V.

II. 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. Fig. 1 illustrates the way that feature vectors are created.

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

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

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

$$\boldsymbol{I} = \sum_{i=1}^{N} d_i^2 = \sum_{i=1}^{N} \left((x_i - C_x)^2 + (y_i - C_y)^2 \right) \tag{1}$$

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 in the object.

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

The possibility that two or more different images have the same or very close $\boldsymbol{I}_N$ 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 2-D). The first part of the heuristic considers the number of intensity pixel changes on a circular boundary of some radius inside the object as it crosses it. This simple coding scheme extracts the topological characteristics of the object regardless of 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 $\boldsymbol{M}$ and $\boldsymbol{N}$, see Figs. 2–4). In the second part of the heuristic, we take into account the difference in size of the two largest arcs (for each circle) that are not part of the object. These correspond to d_1 and d_2 in Fig. 3. For achieving size normalization, we divide this difference by the circumference.

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

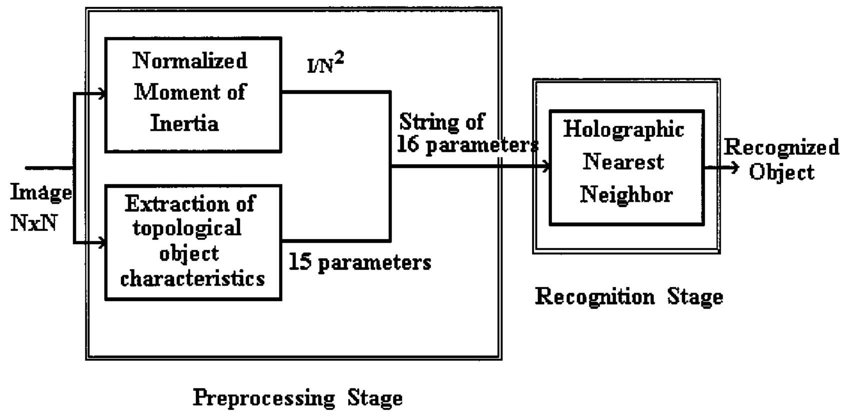


Fig. 1. General diagram showing the steps of the IRO system.

Fig. 2. Difference between the letters M and N.

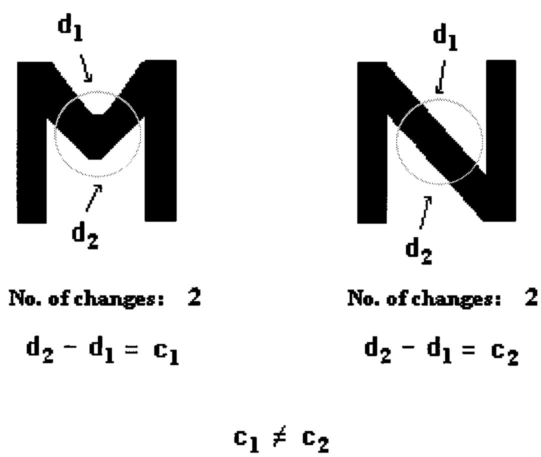
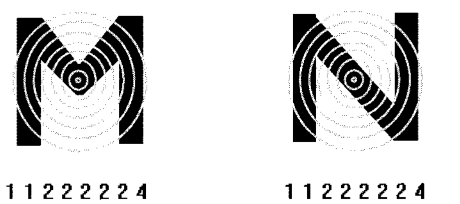


Fig. 3. Difference between the number of pixels in the changes of the letters M and N.

1) Obtain the centroid of the object.
2) Generate K equidistant concentric circles C_i around the centroid. The spacing is equal to the distance between the centroid and the furthest pixel of the object divided by K.
3) For each circular boundary, count the number of intensity changes (zero to one or one to zero) 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 arc pixels, obtain the difference, and divide by the circumference. 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 circumference.

The radial coding can be represented by the following vector:

$$R_1,\, R_2,\, \cdots,\, R_K,\, D_1,\, D_2,\, \cdots,\, D_K$$



Normalized differences:

Circle / Letter	2	3	4	5	6	7	8
N	0	0.08	0.08	0	0.03	0.02	0.01
M	0	0.18	0.41	0.26	0.18	0.06	0.01

Fig. 4. Parameters obtained for the letters M and N.

considering K circles. R_i is a positive integer, and D_i is a real value in [0, 1]. Fig. 5 shows this coding scheme for different sizes and orientations of the letter $\boldsymbol{E}$.

In summary, we define three sets of invariant features to be obtained in the preprocessing stage: 1) normalized central moment of inertia; 2) radial coding; and 3) differential radial coding. All these features are invariant to translation, rotation, and scaling, and together, they provide a robust feature set for the recognition or classification stage.

III. 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) [20]. The main motivations for using this technique are as follows.

1) Unlike other neural-network architectures, learning and recall with the HAM is very fast (see Section IV).
2) The HNN algorithm has, in general, a better performance (recognition rate) than a simple nearest-neighbor technique.

As in optical holography, the HNN algorithm bases its operation on the principle of unfolding information of different phases in a single plane (see [20]).

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

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

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.

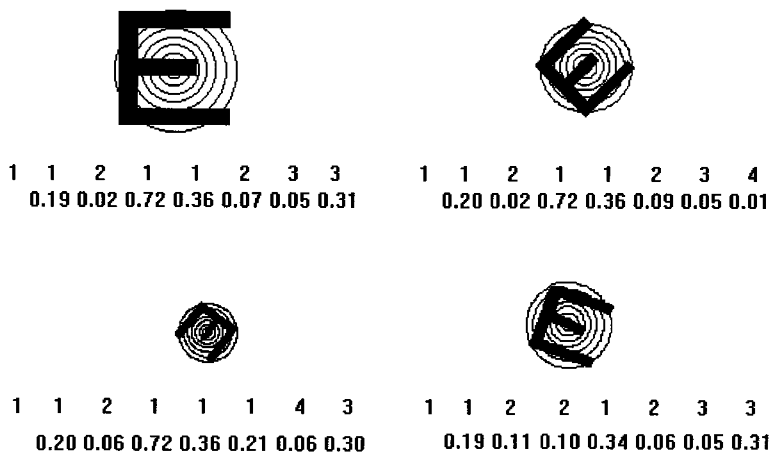


Fig. 5. Extraction of topological characteristics for different rotations and sizes of the letter E. The first line indicates the number of intensity pixel changes in each one of the eight circumferences, and the second line represents the normalized differences over the largest seven circumferences.

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

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

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 zero and 2π.

The HNN algorithm is similar to the well-known nearest-neighbor algorithm (NN) [13]. This latter algorithm is based on the idea that the minimum Euclidean distance between the input and each training vector can be used to classify the input vector. Moreover, to prevent the domination of a subgroup of features, the NN algorithm normalizes these features. The normalization consists of subtracting the mean from each feature variable and dividing the result by the standard deviation of the corresponding feature in the training set.

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 (θ_t) and training phases ($\theta_{\exp}$) 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 superior to the normal NN algorithm [22].

IV. 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.

A. Learning

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

B. 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 × 100 pixels, and the smallest has 20 × 20 pixels. Fig. 6 shows a few of the generated images of the letter $\boldsymbol{E}$.

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, different amounts of noise can be added to the letters to be recognized. The noise was applied to the boundary (Fig. 7) or randomly (Fig. 8). The boundary 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, where the percentage of altered pixels is varied from 10% to 90%.

C. Results

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

Noisy images (see Figs. 7 and 8) are recognized correctly by our model as well, indicating the robustness of it. An accuracy of 98% is obtained, with images having up to 60% of random noise.

As was mentioned before, both preprocessing and recognition are computationally efficient. The system was tested in an IBM RS6000-H50 workstation with the following results. The mean time for the preprocessing stage, including the moment of inertia and the radial coding, is 0.105 462 s, with a standard deviation of 0.0187 s.

TABLE I
TRAINING VECTORS OF LETTERS A (65), F (70), M (77), AND T (84) IN FOUR DIFFERENT ROTATIONS: 0°, 35°, 70°, AND 105°

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

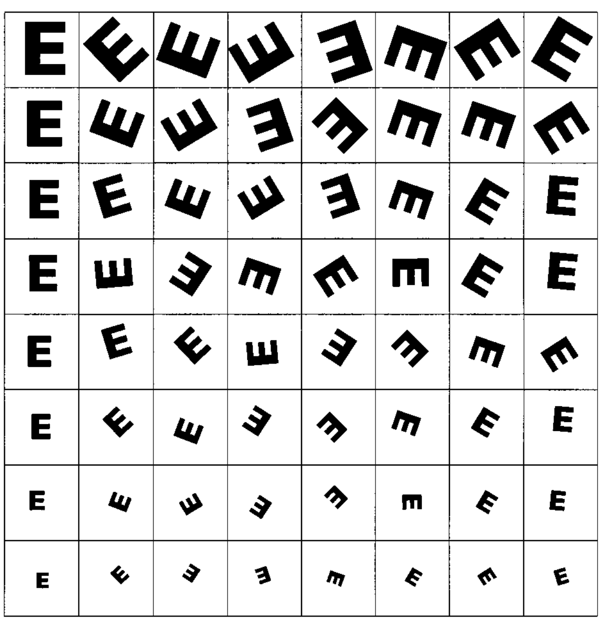


Fig. 6. Testing images of the letter E.

The mean time for recognition using the HNN algorithm is 0.108 846 s, with a standard deviation of 0.034 49 s. So the total mean time for IOR in our experiments is approximately 0.2 s and is very stable. The training time for the HNN is the same as the recognition time for each pattern used for training, which is approximately 0.1 s times the number of training patterns.

D. 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 using 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 threshold at the following intensity value [3]:

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

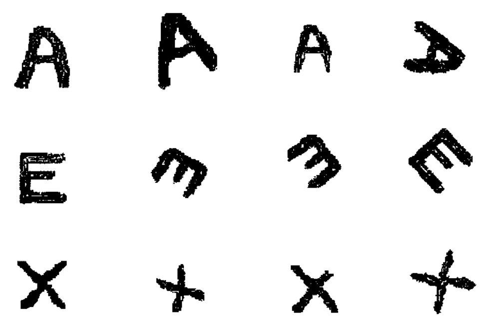


Fig. 7. Letters A, E, and X with some degree of boundary noise recognized with 100% accuracy.

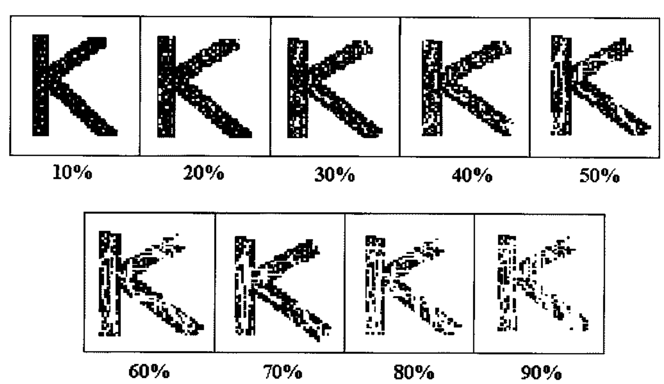


Fig. 8. Different percentages of random noise applied to the letter K.

TABLE II
AVERAGE PERCENT OF RECOGNITION FOR EACH OF THE 26 LETTERS FOR DIFFERENT SIZES

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

where

- Th threshold
- m median of the intensity values
- $\max$ maximum
- $\min$ minimum
- k constant that is obtained empirically.

The proposed methodology for IOR has been applied for recognizing characters in car plates [8]. In this case, the images have 256 gray levels and are binarized using the previous method. For this application, we segmented the plate in the image, and we automatically isolated the area in which each character is located. So, as in the previous experiments, we can assume that there is a single object in the image to be recognized, and in this case, it is a single character. The system was tested with 90 images of car plates obtained under natural-illumination conditions. Fig. 9 shows an example of an image of a car plate, and Fig. 10 shows the segmented characters with the circles used to obtain the radial coding. The recognition results (assuming the segmentation of the characters is correct) are similar to those obtained with the artificially generated characters, with nearly 90% correct recognition.

Fig. 9. Example of an image of a car plate.

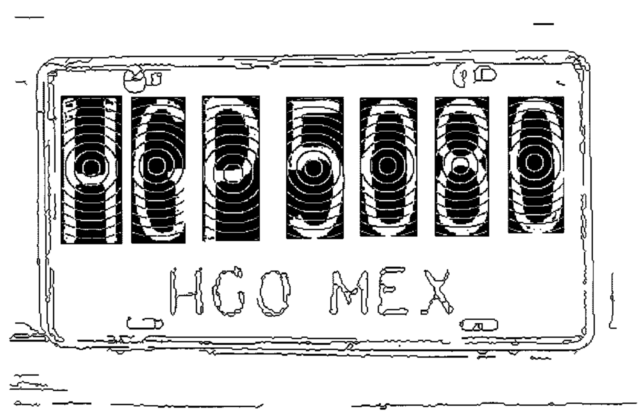


Fig. 10. Segmented characters illustrating the radial coding for the car plate image in Fig. 9.

V. RELATED WORK

A number of IOR methods have been proposed in the literature. These can be classified as: optical techniques [6], [17], boundary-based analysis via Fourier descriptors [9], [12], neural-networks models [1], [5], [7], [18], [19], invariant moments [2], [11], [13], and genetic algorithms [14].

It is important to mention recent IOR research based on optical techniques such as composite-harmonic filters [17] or scale, translation, and in-plane rotation (STIR)-invariant transformations [4]. The former filters involve the Mellin radial harmonics for scale invariance [16], the logarithmic harmonics for projection invariance [15], and the circular harmonics for rotation invariance [10]. Fang and Hausler [4] introduced a new class of transforms that achieve STIR invariance simultaneously. In their approach, an intensity function $S(x, y)$ is mapped into a one-dimensional (1-D) 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 types 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 an alternative to IOR [9], [12]. 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 done easily 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], [19], 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 when learning is completed, it obtains 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's) have been utilized recently for invariant recognition [7], [19]. In this type of model, one has to encode the properties of invariance in the values of the synaptic weights. 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 [18], in which combinations of triplets of image pixels are used as invariant relations. The triplets form triangles representing similar angles ($\alpha,\ \beta,\ \gamma$) in any transformation of the same image. The weights are restricted in such a 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 is the main disadvantage of this approach.

IOR based on moments and invariant functions of moments is another popular invariant-recognition scheme. In 1962, Hu [11], 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 possess orthogonal properties, making reconstruction of the input image computationally difficult. To overcome this problem, Teague [21] 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 [13]. 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.* [14]. 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 four 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 variations only. This coding is an efficient way to extract topological characteristics, but it is only invariant to scaling and translation, not to rotation.

VI. Conclusion

In this work, a novel method to recognize 2-D 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 that extracts invariant topological characteristics of 2-D 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 NN algorithm. In the work reported here, only 104 patterns representing 26 letters are used for training. Regardless of its position, the HNN is able to correctly recognize orientation, size, and any upper-case letter with 17 different sizes and 14 different rotations, including some noise added randomly or to the boundary. The method is extended to handle multilevel images and applied for character recognition in images of car plates.

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

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

Acknowledgment

The authors thank the anonymous referees for their comments on previous drafts.

References

[1] M. Caudill, "Neural networks primer pt. II," *AI Expert*, vol. 3, no. 2, pp. 55–61, 1988.
[2] C.-H. Teh and R. T. Chin, "On image analysis by the methods of moments," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 10, pp. 496–513, July 1988.
[3] E. B. Davies, *Machine Vision*. London, U.K.: Academic, 1990.
[4] M. Fang and G. Hausler, "Class of transforms invariant under shift, rotation, and scaling," *Appl. Opt.*, vol. 29, pp. 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. Cybern.*, vol. 36, no. 4, pp. 193–202, 1980.
[6] E. Ghahramani and L. R. B. Patterson, "Scale, translation, and rotation invariant orthonormalized optical/optoelectronic neural networks," *Appl. Opt.*, vol. 32, no. 35, pp. 7225–7232, 1993.
[7] G. L. Giles and T. Maxwell, "Learning, invariances, and generalization in high-order neural networks," *Appl. Opt.*, vol. 26, no. 23, pp. 4972–4978, 1987.
[8] G. Gómez and L. E. Sucar, "Sistema para reconocimiento de caracteres en placas de automóviles," *Soluciones Avanzadas*, vol. 6, no. 58, pp. 47–53, Aug. 1998.
[9] R. Gonzalez and P. Wintz, *Digital Image Processing*, 2nd ed. Reading, MA: Addison-Wesley, 1987, pp. 404–411.
[10] Y. N. Hsu and H. H. Arsenault, "Optical pattern recognition using the circular harmonic expansion," *Appl. Opt.*, vol. 21, no. 22, pp. 4016–4019, 1982.
[11] M. K. Hu, "Visual pattern recognition by moment invariants," *IRE Trans. Inform. Theory*, vol. IT-8, no. 2, pp. 179–187, 1962.
[12] 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.*, vol. 17, pp. 207–210, Feb. 1995.
[13] A. Khotanzad and Y. H. Hong, "Invariant image recognition by Zernike moments," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 12, pp. 489–497, May 1990.
[14] A. D. McAulay and J. C. Oh, "Improving learning of genetic rule-based classifier systems," *IEEE Trans. Syst., Man, Cybern.*, vol. 24, pp. 152–159, Feb. 1994.
[15] D. Mendlovic, N. Konforti, and E. Marom, "Shift and projection invariant pattern recognition using logarithmic harmonics," *Appl. Opt.*, vol. 29, pp. 4784–4789, 1990.
[16] D. Mendlovic, E. Marom, and N. Konforti, "Shift and scale invariant pattern recognition using Mellin radial harmonics," *Opt. Commun.*, vol. 67, pp. 172–176, 1988.
[17] D. Mendlovic, Z. Zalevsky, I. Kiryuschev, and G. Lebreton, "Composite harmonic filters for scale-, projection-, and shift-invariant pattern recognition," *Appl. Opt.*, vol. 34, pp. 310–316, 1995.
[18] 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*, vol. 3, pp. 241–251, Mar. 1992.
[19] L. Spirkovska and M. B. Reid, "Coarse-coded higher-order neural networks for PSRI object recognition," *IEEE Trans. Neural Networks*, vol. 4, pp. 276–283, July 1993.
[20] J. G. Sutherland, *The Holographic Neural Method, in Fuzzy, Holographic and Parallel Intelligence*. New York: Wiley, 1992.
[21] M. R. Teague, "Image analysis via the general theory of moments," *J. Opt. Soc. Amer.*, vol. 70, pp. 920–930, 1980.
[22] L. A. Torres-Mendez, "Invariant 2-D object recognition," M.Sc. thesis, Graduate Program in Informatics, ITESM-Campus Morelos, México, 1995.
[23] S. Watanabe, *Pattern Recognition: Human and Mechanical*, New York: Wiley, 1985.