Real object recognition using moment invariants

MUHARREM MERCIMEK, KAYHAN GULEZ and
TARIK VELI MUMCU

Yildiz Technical University, Electrical-Electronics Faculty, Electrical Engineering
Department, 34349 Besiktas-Istanbul, Turkey
e-mail: mercimek@yildiz.edu.tr, gulez@yildiz.edu.tr, tmumcu@yildiz.edu.tr

Abstract. Moments and functions of moments have been extensively employed as invariant global features of images in pattern recognition. In this study, a flexible recognition system that can compute the good features for high classification of 3-D real objects is investigated. For object recognition, regardless of orientation, size and position, feature vectors are computed with the help of nonlinear moment invariant functions. Representations of objects using two-dimensional images that are taken from different angles of view are the main features leading us to our objective. After efficient feature extraction, the main focus of this study, the recognition performance of classifiers in conjunction with moment–based feature sets, is introduced.

Keywords. Regular moment functions; 3-D object recognition; image processing; neural networks; fuzzy K-NN.

1. Introduction

Pattern recognition is an essential part of any high-level image analysis systems. Most of these systems share a general structure of four building blocks: Image acquisition, preprocessing of the images, feature extraction, and classification (Khotanzad & Lu 1990). To improve the classification results in various situations, many approaches have been studied including decision theory, feature selection, optimization, learning and so on (Kim & Han 1995).

An effective shape descriptor is a key component of multimedia content description, since shape is a fundamental property of an object. There are two types of shape descriptors: contour-based shape descriptors and region-based shape descriptors (Kim & Sung 2000). Regular moment invariants are one of the most popular and widely used contour-based shape descriptors is a set of derived by Hu (1962). These geometrical moment invariants have been then extended to larger sets by Wong & Siu (1999) and to other forms (Dudani *et al* 1977; Liao & Pawlak 1998). An image-processing algorithm developed by Zion *et al* (1999) has been used for discrimination between images of three fish species for use on freshwater fish farms. Zernike velocity moments were developed by Shutler & Nixon (2001), to describe an object using not only its shape, but also its motion throughout an image.

Unfortunately, regular moment invariants are not orthogonal. Hu's moment invariants and extended Zernike moments were used as feature extractors and NN-based classifiers were compared with conventional classifiers on a handwritten English character recognition problem by Khotanzad & Lu (1990). The recovery of the images from these orthogonal Zernike Moments can be achieved. Complex-log conformal mapping combined with a distributed associative memory was used to create a system, recognizing objects regardless of changes in rotation or scale by Wechsler & Zimmerman (1998) 3-D object simulations were created using computers. These were represented with geometrical moment feature vectors and classified with neural networks by Ustun (1999).

In this study, a computer vision system recognizing objects in captured images is established. The considering sensor can be located at different distances and angles according to objects; a vision system insensitive to these changes is investigated. The organization of the paper is as follows: section 2 discusses moment invariant features. In § 3, classifier basis and utilized structures are described. Section 4 reports experimental results, while § 5 gives conclusions and discussions of our study.

2. Moment invariants

An essential issue in the field of pattern analysis is the recognition of objects and characters regardless of their position, size and orientation as illustrated in figure 1. The idea of using moments in shape recognition gained prominence when Hu (1962), derived a set of invariants using algebraic invariants.

Two-dimensional moments of a digitally sampled $M \times M$ image that has gray function $f(x, y)$, $(x, y = 0, \ldots M - 1)$ is given as,

$$m_{pq} = \sum_{x=0}^{x=M-1} \sum_{y=0}^{y=M-1} (x)^p \cdot (y)^q f(x, y) \quad p, q = 0, 1, 2, 3 \cdots .$$
(1)

The moments $f(x, y)$ translated by an amount (a, b), are defined as,

$$\mu_{pq} = \sum_x \sum_y (x + a)^p \cdot (y + b)^q f(x, y).$$
(2)

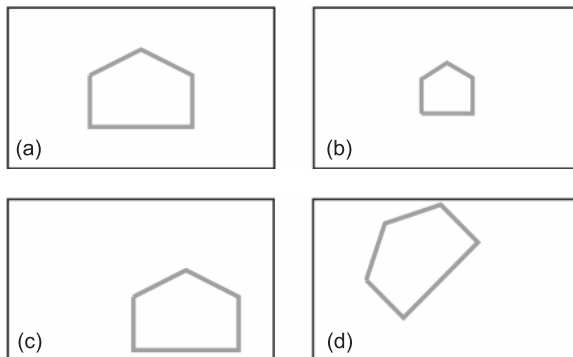**Figure 1.** (a) 2-D object, (b) change of size, (c) change of position, (d) change of orientation.

Thus the central moments m'_{pq} or μ_{pq} can be computed from (2) on substituting $a = -\overline{x}$ and $b = -\overline{y}$ as,

$$\overline{x} = \frac{m_{10}}{m_{00}} \text{ and } \overline{y} = \frac{m_{01}}{m_{00}}, \mu_{pq} = \sum_x \sum_y (x - \overline{x})^p \cdot (y - \overline{y})^q f(x, y). \tag{3}$$

When a scaling normalization is applied the central moments change as,

$$\eta_{pq} = \mu_{pq}/\mu_{00}^{\gamma}, \quad \gamma = [(p+q)/2] + 1 \tag{4}$$

In particular, Hu (1962), defines seven values, computed by normalizing central moments through order three, that are invariant to object scale, position, and orientation. In terms of the central moments, the seven moments are given as,

$$
\begin{aligned}
M_1 &= (\eta_{20} + \eta_{02}), \\
M_2 &= (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2, \\
M_3 &= (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2, \\
M_4 &= (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2, \\
M_5 &= (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] \\
&\quad + (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2], \\
M_6 &= (\eta_{20} - \eta_{02})[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] \\
&\quad + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03}), \\
M_7 &= (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] \\
&\quad - (\eta_{30} + 3\eta_{12})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2].
\end{aligned} \tag{5}
$$

3. Classification methods

Classification is the final stage of any image-processing system where each unknown pattern is assigned to a category. The degree of difficulty of the classification problem depends on the variability in feature values for objects in the same category, relative to the difference between feature values for objects in different categories Duda *et al* (2000). In this study we use MLP (multilayer perceptron), GAL (Gaining Algorithm Learning), K-NN (K-Nearest Neighbour), and fuzzy K-NN (fuzzy K-Nearest Neighbour), classifiers as pattern classifiers.

3.1 *Multilayer perceptron*

MLP has been termed a universal approximator, and can provide an optimal solution to an arbitrary classification problem. It implement linear discriminates, but in a space where the inputs have been mapped nonlinearly. The key power provided by such networks is that they admit fairly simple algorithms where the form of the nonlinearity can be learned from the training data. The models are thus extremely powerful, have nice theoretical properties, and apply well to a vast array of real-world applications. Figure 2 shows a simple three-layer MLP neural network, consisting of an input layer, hidden layer and an output layer, interconnected by modifiable weights represented by links between layers.
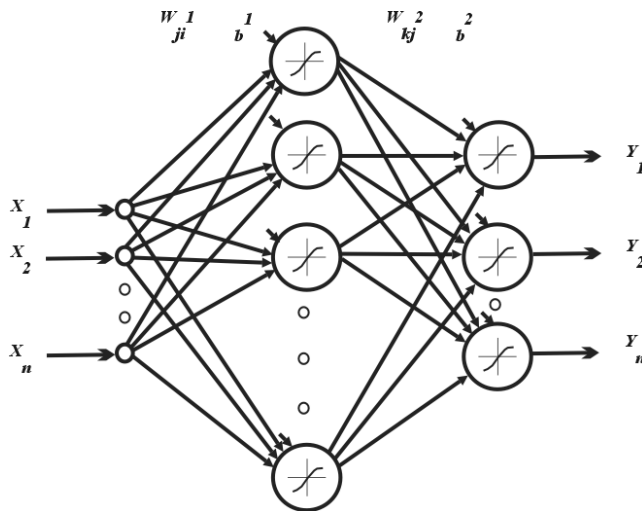
Figure 2. Three-layer MLP structure.

One of the most popular methods for training such multilayer networks is based on gradient descent in error-the backpropagation algorithm a natural extension of the LMS (least mean square) algorithm. Guided by an analysis of networks and their function we can make informed choices of the scaling of input values and initial weights, desired output values, and more. Network architecture or topology plays an important role for neural net classification, and the optimal topology will depend on the problem at hand (Duda *et al* 2000).

3.2 *Grow and learn neural networks (GAL)*

Grow and learn neural networks (GAL) determine class boundaries in the feature vector space by means of minimum distance criterion. Distances from input vector to all nodes are computed (figure 3). The class of the input vector is labeled as the class of the node that is nearest to the vector. The number of nodes is determined automatically during the training of the network (Olmez & Dokur 2003).

The first layer is used for computing the distances between the input vector and the nodes. The second layer is used to label the nodes as one of the classes. The second layer is used as a logical OR function.
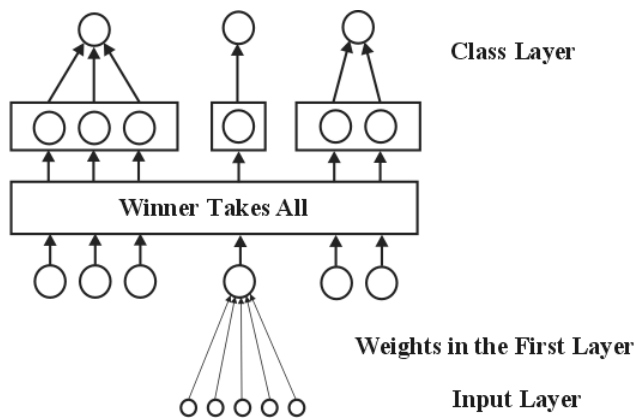

Figure 3. GAL structure.

The most significant property of the GAL is that, of determining the number of the nodes automatically. The sequence of the input vectors effects the structure of the GAL. In the networks there exist some nodes, which are initially meaningful, but then become unused. These nodes are dismissed by the forgetting algorithm. The aim of this algorithm is to find the nodes that are not influential on changing the performance of the network and to eliminate them.

After taking class representative nodes for each class, the training of the network is started. During the desired iteration time, nodes are added to the network or dismissed from it.

3.3 *K-NN classifier*

K-Nearest neighbour (K-NN) classification is a simple, yet, powerful classification method. The key idea behind K-NN classification is that similar observations belong to similar classes. The crisp 1-NN classification rule assigns an input sample pattern x to the class of its near neighbour. This idea can be extended from 1 to K, in K-NN with the input pattern x being assigned to the class that is represented by a majority among the K-nearest neighbours (Kim & Han 1995).

It can be shown that the performance of a K-NN classifier is always at least half that of the best possible classifier for a given problem. One of the major drawbacks of K-NN classifiers is that the classifier needs all available data. This may lead to considerable overhead if the training data set is large.

When using more than one neighbour for classification and giving each neighbour the same contribution for a decision, there can be a tie among classes with a maximum number of neighbours in the group of K-nearest neighbour.

The number of the nearest neighbours, k, should be odd in order to avoid ties, and it should be kept small, since a large k tends to create misclassifications unless the individual classes are well separated. Another simple method to handle this tie problem is to find the class, which has the minimum summation of distance from the sample to each neighbour.

3.4 *Fuzzy K-NN classifier*

The Fuzzy K-NN is proposed to solve the tie problem by using fuzzy set theory. This method assigns class membership to an input pattern x rather than a particular class as in K-NN. In fuzzy K-NN the result is membership information of x to each class. The advantage of using fuzzy set theory is that no arbitrary assignments are made. In addition, the membership values of input pattern would provide a level of assurance to the results of classification (Kim & Han 1995).

This method keeps to the main idea of K-NN in which class decision is made by the nearest neighbours' membership class information. By using fuzzy set theory the tie problem can be solved naturally and we can accumulate more information, which describes the whole class distribution.

4. 3-D real object recognition problem and experimental study

The major building blocks of the pattern recognition system used in this paper were briefly given in the moment invariants section (§ 2). In this section these blocks are extracted using the basis of our 3-D object recognition problem. 3-D objects used in this study are shown in figure 4. An image display/recording system is a conceptually a scanning system. A camera that can transfer captured digital images via USB (universal serial bus) is used in this paper. Information about the original colours of the objects is not used.
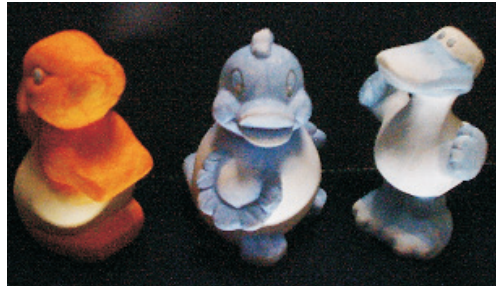
Figure 4. 3-D objects used.

4.1 *Representation of 3-D objects using 2-D images*

Synthesis of 2-D images is used practically in such applications, while transferring real life 3-D objects to computer is done by creating spatial object simulations.

Normally, objects can be rotated in three different directions. In this study objects are rotated along the y-axis only and the images thus projected on the yz plane are taken (figure 5). The camera is placed along the x-axis. With $5°$ rotations from $0°$ to $360°$ around the y-axis, 72 images for every object or a total of 216 images are obtained.

4.2 *Preprocessing of images*

A brief description of the image preprocessing steps and changes in the images that take place after these steps are as follows (figure 6),

(a) The images are minimized to 300×340 pixel dimensions in order to ease the computational burden.
(b) The minimized images in the RGB (red-green-blue) norm are converted to the gray-level coded images.
(c) Thresholding process is applied and the gray level of the pixels between 0–255 changed to either 0 or 255. Threshold value is determined as the gray level of 90. The gray levels under 90 are changed to gray level of 0, and the gray levels over 90 are changed to 255. This process is necessary for the next step, edge detection.
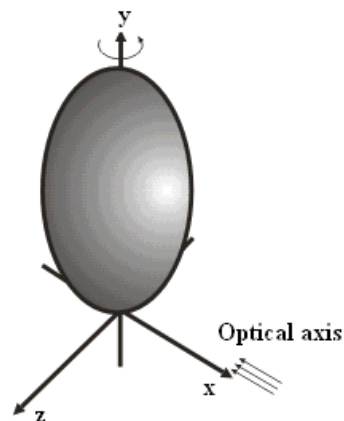
Figure 5. Imaging system.

(a)   (b)   (c)   (d)   (e)
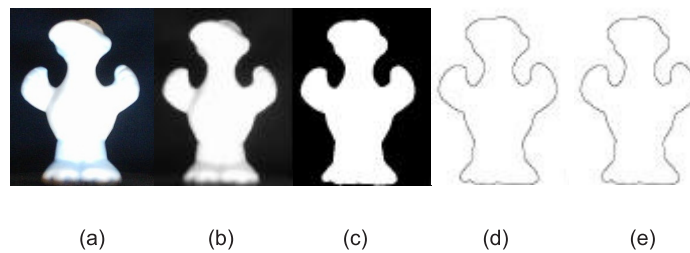
Figure 6. The image preprocessing steps.

(d) The edge of an object is one of the most meaningful parts used for image recognition. In order to obtain compact feature vectors for every image, the pixels of the edge images of the target objects are used.

(e) The last step is the converting the gray level coded images to binary images, in order to use pixels with values of 1 and 0.

All these processes are applied to totally 216 images of three objects. A number of the edge images of three objects are presented in figure 7.
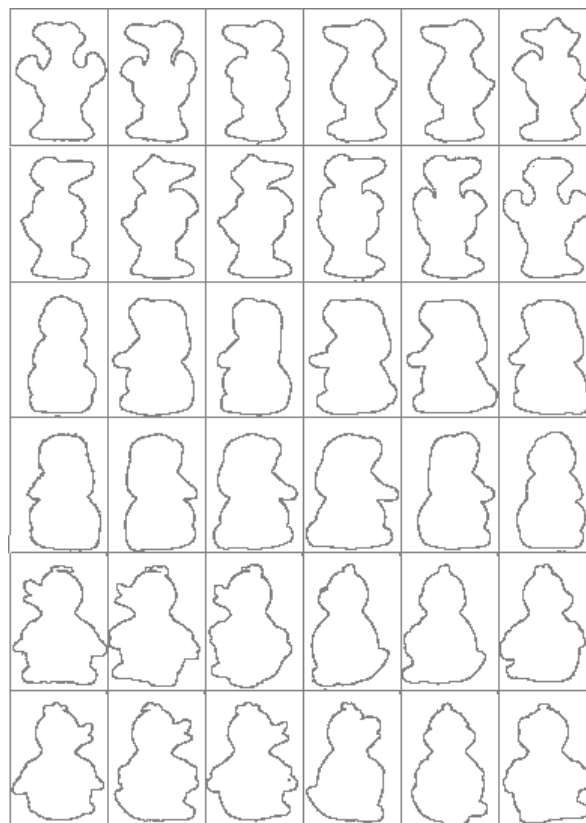
Figure 7. A number of edge images of the objects.

4.3 *Feature vector extraction*

In this paper, the idea of representing images with a set of moment invariant functions is applied. In order to normalize the different powers of the coefficients in (6), one more enhancement is done using the distance, d, between the objects and the camera. This can be achieved using the term moment function oscillation radius, r. Finally, the coefficients of moment invariant functions according to Dudani *et al* (1977) and Ustun (1999) are,

$$M'_1 = (\mu_{20} + \mu_{02})^{1/2}d = r.d,$$
$$M'_2 = M_2/r^4,$$
$$M'_3 = M_3/r^6,$$
$$M'_4 = M_4/r^6, \tag{6}$$
$$M'_5 = M_5/r^{12},$$
$$M'_6 = M_6/r^8,$$
$$M'_7 = M_7/r^{12}.$$

Totally 216 images and feature vectors representing these images with 7 coefficients of moment invariant functions were obtained previously. 108 feature vectors obtained from the images of every object, taken from the angles of $(10n)°(n = 0, 1, 2, \ldots, 35)$, are included in the training data set. 108 feature vectors obtained from the angles of $(10n + 5)°$ are included in the test data set.

4.4 *Classification*

These processed data are presented to neural classifiers for classification. A flexible recognition system that can attain the best classification is investigated in this study. MLP, GAL, K-NN, fuzzy K-NN, classifiers are created. The multidimensional input vectors $x \in \Re^7$ are assigned to one of the classes $\{C1, C2, \text{and } C3\}$. $y_1(i)$, $y_2(i)$, $y_3(i)$ outputs obtained from the classifier are presented to a classification criterion algorithm,

$$C1 = 0; C2 = 0; C3 = 0.$$

If $(0\cdot9 <= y_1(i) < 1)$ and $(0 < (y_2(i), y_3(i)) < 0\cdot1)$,

$$C1 = C1 + 1;$$

If $(0\cdot9 <= y_2(i) < 1)$ and $(0 < (y_1(i), y_3(i)) < 0\cdot1)$,

$$C2 = C2 + 1;$$

If $(0\cdot9 <= y_3(i) < 1)$ and $(0 < (y_1(i), y_2(i)) < 0\cdot1)$,

$$C3 = C3 + 1;$$

Considering the separation of the pattern in the input space, using simple classifiers (perceptron etc.) that can implement linear decision boundaries will be inadequate. MLP, which is termed a universal function approximator, is selected as one of the classifiers. A structure of MLP with an input layer of 7 neuron units, a hidden layer of 12 neuron units, and an output layer of 3 neuron units is used. The activation functions of the output layer neurons are

selected logarithmic sigmoid function. The activation functions of the hidden layer neurons are selected tangent sigmoid or logarithmic sigmoid functions. For different types of learning algorithms based on backpropagation, the network iteratively changes its parameters to bring the actual outputs closer to desired target values.

GAL is used as the second classifier. GAL determines the class boundaries in the feature vector space by mean of minimum distance criterion. In the training phase, the distance from input vector to initial nodes is computed. If the class of the input vector is the same as the class of the node that is nearest to the vector, nothing is done. Otherwise, this vector is added as one of the nodes that represent the network. Since there is only one set of training data, there exists just only one network configuration.

Training data can be renamed prototype data for K-NN and fuzzy K-NN classifiers. Using Euclidean distances from test data to prototype data, the proximity-based similarity is calculated. K-NN classification rule assigns an input sample pattern x to the class of its nearest neighbour without regarding the levels of similarity between x and other prototypes. On the contrary, in fuzzy K-NN classification, no arbitrary assignment is made. It can make more decisions based on the neighbours' membership grade, which contains more global information about the distribution of the feature vectors. For different k values, the former two types of classifiers are tested. For all configurations classification accuracies are presented in appendix A.

5. Discussions and conclusions

We presented an efficient 3-D object recognition application. Neural networks provide a new suite of nonlinear algorithms for pattern recognition. Their nonlinear function approximation ability, making no assumptions about the statistical distributions, and adapting to different input data, makes neural networks a challenging field of study. Relevance of the application is also evaluated with K-NN classifiers. To overcome the limits of conventional K-NN, not only distance information but also initial membership grade of neighbours is used with fuzzy K-NN classifiers.

In this paper, we mainly focused on feature extraction and classification. An appropriate subspace of dimensionality in the original space is investigated with moment-invariant functions set. In all the experiments, the MLP neural networks performed better than the three neural networks (table 1).

The initial assumptions about the direction of rotation of the objects make the recognition problem easier. It is possible to rotate objects around three dimensions. Besides, the recognition system works offline. There is room for improvements, if we direct our study along multi-dimensional changes and recognition.

Table 1. Overall performances of the classifiers.

Classifier	Performance (%)
MLP	100
GAL	90·74
K-NN	98·14
Fuzzy K-NN	97·22

Appendix A. Classification performances using test data.

MLP trainoss		MLP trainlm		MLP trainrp		MLP trainbfg		K-nn Classifier		Fuzzy K-nn Classifier		Gal	
Learn. rate	*Classification accuracy	Learn. rate	Classification accuracy	Learn. rate	Classification accuracy	Learn. rate	Classification accuracy	k	Classification accuracy	k	Classification accuracy	Number of nodes	Classification accuracy
0.1	105	0.1	94	0.1	108	0.1	105	1	106	1	105	22	98
0.2	104	0.2	92	0.2	102	0.2	103	2	106	2	99		
0.3	105	0.3	88	0.3	106	0.3	96	3	105	3	100		
0.4	103	0.4	83	0.4	104	0.4	107	4	105	4	102		
0.5	106	0.5	87	0.5	105	0.5	93	5	104	5	102		
0.6	105	0.6	89	0.6	106	0.6	105	6	105	6	104		
0.7	107	0.7	82	0.7	104	0.7	106	7	106	7	102		
0.8	104	0.8	87	0.8	107	0.8	102	8	106	8	102		
0.9	104	0.9	88	0.9	107	0.9	104	9	105	9	100		
1	105	1	90	1	107	1	95	10	106	10	98		
								11	104	11	91		
								12	105	12	88		
								13	104	13	86		
								14	106	14	85		
								15	101	15	81		
								16	104	16	82		
								17	96	17	82		
								18	101	18	80		

*Number of correctly classified patterns among 108 patterns of test data.
trainoss: one-step secant backpropagation
trainrp: resilient backpropagation
trainlm: levenberg-Marquardt backpropagation
trainbfg: BFGS quasi-Newton backpropagation

References

Duda R O, Hart P E, Stork D G 2000 *Pattern classification* 2nd edn (New York: John Wiley & Sons)

Dudani S A, Breeding K J, Mcghee R B 1977 Aircraft identification by moment invariants. *IEEE Trans. Comput.* C-26: 39–46

Hu M 1962 Visual pattern recognition by moment invariants. *IRE Trans. Inf. Theor.* IT-8: 179–187

Khotanzad A, Lu J-H 1990 Classification of invariant image representations using a neural network. *IEEE Trans. Acoust., Speech Signal Process.* 38: 1028–1038

Kim W-K, Sung Y 2000 A region-based shape descriptor using Zernike moments. *Signal Process. Image Commun.* 16: 95–102

Kim Y K, Han J H 1995 Fuzzy K-NN algorithm using modified K-selection. *Proc. Int. Joint Conf. 4th IEEE International Conf. on Fuzzy Systems* (FUZZ-IEEE/IFES'95) 3: 1673–1680

Liao S X, Pawlak M 1998 On the accuracy of Zernike moments for image analysis. *IEEE Trans. Pattern Anal. Mach. Intell.* 20: 1358–1364

Olmez T, Dokur Z 2003 Classification of heart sounds using an artificial neural network. *Pattern Recogn. Lett.* 24: 617–629

Shutler J D, Nixon M S 2001 Zernike velocity moments for description and recognition of moving shapes. *Proc. BMVC*, pp 705–714

Ustun A 1999 *Application of artificial neural networks to object recognition*. MSc thesis, ITU Science Institute, Istanbul, Turkey (in Turkish)

Wechsler H, Zimmerman G L 1998 2-D invariant object recognition using distributed associative memory. *IEEE Trans. Pattern Anal. Mach. Intel.* 10: 811–821

Wong W H, Siu W C 1999 Improved digital filter structure for fast moment computation. *IEE Proc. Vision, Image Signal Process.* 46: 73–79

Zion B *et al* 1999 Sorting fish by computer vision. *Comput. Electron. Agric.* 23: 175–187