Available online at www.sciencedirect.com
ScienceDirect
Expert Systems with Applications 35 (2008) 1966–1977
Expert Systems with Applications
www.elsevier.com/locate/eswa

Invariant 2D object recognition using eigenvalues of covariance matrices, re-sampling and autocorrelation

Te-Hsiu Sun [a], Chi-Shuan Liu [b], Fang-Chih Tien [b,*]

[a] Department of Industrial Engineering and Management, Chaoyang University of Technology, Taiwan
[b] Department of Industrial Engineering and Management, National Taipei University of Technology, Taiwan

Abstract

This study presents a novel invariant object recognition method for two-dimensional object. The proposed method employs the eigenvalues of covariance matrix, re-sampling, and autocorrelation transformation to extract unique features from boundary information, and then use minimum Euclidean distance method (MD) and backpropagation neural networks (BPN) for classification. The boundary of the binary digital part is first extracted and represented as the sequence of the smaller eigenvalues of covariance matrix over a given region of support. Then the sequence is re-sampled into a pre-determined number, and transformed using autocorrelation function. The experimental results reveal that the proposed method successfully derives translation, rotation, and scaling invariant features which can be classified easily with MD and BPN.

Keywords: 2D invariant object recognition; Eigenvalues of covariance matrix; Autocorrelation coefficient; Feature extraction

1. Introduction

Computer vision has extensively adopted in industry for the last two decades. It enhances productivity and quality management, and is flexible, efficient, inexpensive, and reliable. The applications of computer vision mainly relate to part inspection and measurement (e.g., food, wood, electronic components, glass, textile, machine parts, printing products and integrated circuits) and defect classification (Malamas, Petrakis, Zervakis, & Petit, 2003; Noble, 1995). In this area, the process of using computer vision usually performs in the following sequence: data acquisition, part recognition, feature extraction and measurement and decision-making. Ullman (1996) noted that object recognition is the most difficult of these tasks owing to the varying positions, orientations and scales of objects. Hence, the recognizing objects, even when they are translated, rotated, or scaled, is of priority concern issue. In particular, using computer vision for object recognition plays a critical role that first determines which process plan should be adopted for the manufactured work piece in a flexible manufacturing system (FMS), assembly line or robot system (Haralick & Shapiro, 1992).

Many methods have been developed for invariant object recognition. Sarkodie-Gyan, Lam, Hong, and Campbell (1996) presented an object recognition scheme for inspecting in manufacturing engine components using a fuzzy membership function. Kreutz, Vöpel, and Janben (1996) proposed a framework of a translation and scale-invariant image recognition system with high-order autocorrelation features. Lee, Moon, and Lee (1997) described an automated object recognition method using the signature and autocorrelation function for invariant feature extraction, and backpropagation neural networks for object recognition. Tsang (1997) presented an affine-invariant recognition method for single, near-planar object shapes from broken boundaries by genetic algorithms. Bruckstein and Shaked (1997) proposed a skew-symmetry detection approach based on invariant signatures for planar object. Khalil and Bayoumi (2000) developed a 2D invariant object

* Corresponding author. Tel.: +886 2 2771 2171; fax: +886 2 2731 7168.
E-mail address: fctien@ntut.edu.tw (F.-C. Tien).

0957-4174/$ - see front matter © 2007 Published by Elsevier Ltd.
doi:10.1016/j.eswa.2007.08.106

recognition method using continuous wavelet transform and neural networks. Jones and Bhanu (2001) developed an invariant SAR (Synthetic aperture radar) recognition method using SAR scattering center locations and magnitudes as features. Wöhler and Anlauf (2001) presented an object detection and recognition system in the traffic environment based on the adaptable time delay neural network. Oswald and Levi (2001) developed an approach to recognize objects based on the usage of cooperation at object level. Moreover, Khalil and Bayoumi (2002) defined three invariant functions with dyadic wavelet transformation of the object boundary to object recognition. Zhang, Zhang, Krim, and Walter (2003) proposed an invariant 2D object recognition approach by measuring the geodesic distance between the observed object and a model in the shape space. Cao, Hao, and Wang (2004) employed the direction basis function (DBF) neural networks for successful object recognition. Kyrki, Kamarainen, and Kälviänen (2004) utilized a Gabor filter to extract invariant features for object recognition. Li and Lee (2004) presented a Hopfield neural network model for invariant object recognition using projective transformations and the projective invariance was embedded into the compatibility constraint for finding point correspondences such that the problem was formulated by minimizing the predefined energy function through a Hopfield network. Huang, Wang, and Zhang (2005) proposed a scheme based on independent component analysis (ICA) for object recognition with affine transformation and for affine motion estimation between video frames. Recently, Sookhanaphibarn and Lursinsap (2006) proposed a method for extracting the invariant features of a color image based on the concept of principal component analysis and a competitive learning algorithm. Yu and Bennamoun (2007) developed two complete sets of similarity invariant descriptors using Fourier–Mellin transform and the analytical Fourier–Mellin transform frameworks, and then adopted 2D-PCA to simplify the invariant descriptor for face recognition.

Artificial Neural Networks (ANNs) are computational modeling tools that have emerged in 1990 and found extensive applications in many disciplines for modeling complex real-world problems. A variety of areas have adopted ANN as a tool to solve complex problems (Smith & Gupta (2000)). ANNs may be defined as structures comprised of densely interconnected adaptive processing elements (neurons or nodes) that are capable of performing parallel computations for data processing and knowledge representation (Hecht-Nielsen, 1990; Schalkoff, 1997). The attractiveness of ANNs can be accredited to their remarkable information processing characteristics of the biological system such as nonlinearity, high parallelism, robustness, fault and failure tolerance, learning, ability to handle imprecise and fuzzy information, and their capability to generalize. There are at least five reasons of using ANNs with such characteristics: (i) nonlinearity allows better fit to data, (ii) noise-insensitivity provides accurate prediction when the data with uncertainty and measurement errors, (iii) high parallelism implies fast processing and hardware failure-tolerance, (iv) learning and adaptivity allow ANNs to modify their internal structures in response to changing environment, and (v) generalization enables applications of the model to unlearned data (Basheer & Hajmeer, 2000). In general, the objective of ANN-based computing is to develop mathematical algorithms that will enable ANNs to learn by mimicking information processing and knowledge acquisition in the human brain. Among various types of neural networks, backpropagation neural networks (BPN) are the most popularly used networks for pattern recognition, prediction, and other applications. Therefore, this study adopts BPN as a pattern classifier.

This study proposes a new transition, rotation and scale-free 2D object recognition method based on the boundary information of the object. Feature extraction is performed using the eigenvalues of covariance matrices, re-sampling and autocorrelation transformation, and objects are finally classified using both backpropagation neural networks and minimum Euclidean distance method. The remainder of this study is organized as follows. Section 2 describes the new feature extraction technique and the classification using backpropagation neural networks and minimum Euclidean distance method. Next, Section 3 summarizes the experimental results. Conclusions are finally drawn in Section 4, along with recommendations for further research.

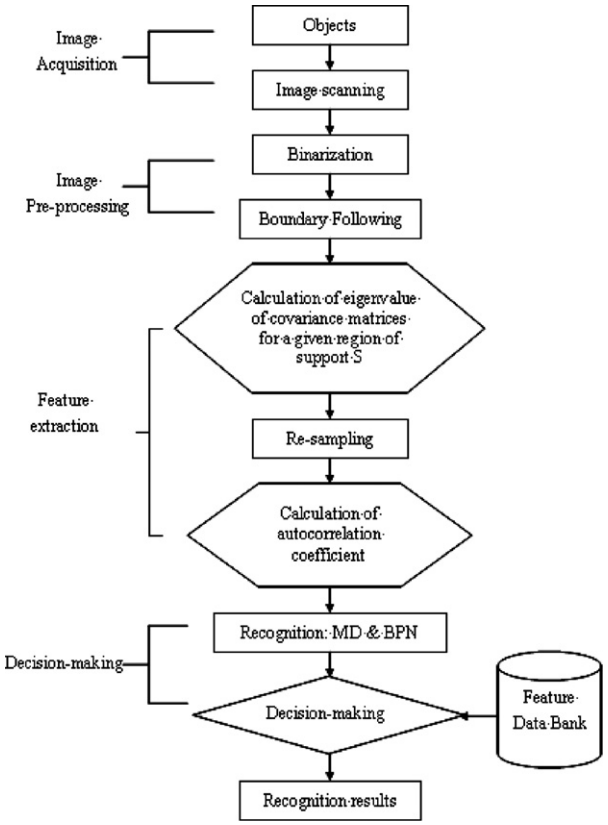


Fig. 1. Flow of proposed method.

2. The proposed method

The proposed invariant object recognition method is broadly divided into four stages as depicted in Fig. 1. Each stage is described in the following sections.

2.1. Image acquisition and segmentation

Image acquisition refers to acquiring the digital information of objects through a digitizer. Once images have been acquired, image segmentation is performed to separate the digitized object from the background. Image segmentation attempts to derive effective information from a digital image. A conventional segmentation method is thresholding, in which each pixel is converted into a binary value with a derived or pre-specified value (T). Pixels with grey values above the threshold are given the binary value 255 (white), while pixels with grey values below the threshold are given the binary value 0 (black). Thresholding is generally performed using the following rule (Gonzalez & Woods, 2004).

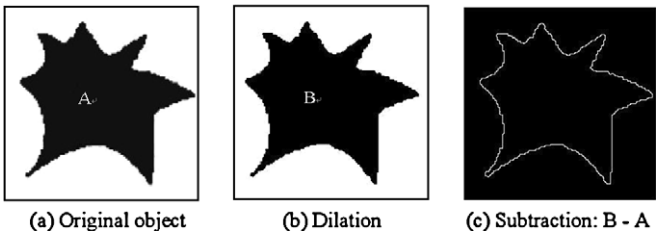


Fig. 2. Boundary extraction using morphological operation.

$$f_a(i,j) = \begin{cases} 255, & \text{if } f(i,j) \geqslant T \\ 0, & \text{if } f(i,j) < T \end{cases}, \quad (1)$$

where $f(i,j)$ denotes the grey value at coordinates (i,j); T represents the specified threshold, and $fa(i,j)$ is the new grey value after segmentation.

2.2. Boundary extraction

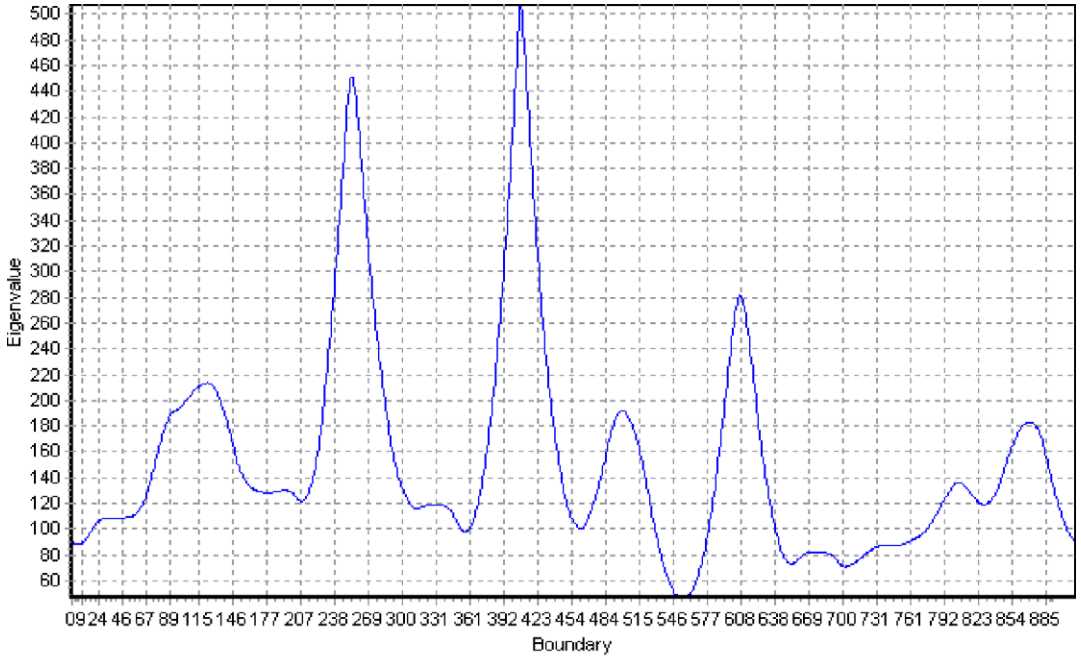
Morphological operations are conventionally adopted in image processing and feature extraction. The basic operations of mathematical morphology include: dilation, erosion, opening and closing. The dilation is a quick and simple operation to extract the boundary information and is utilized in this study. As illustrated in Fig. 2, the boundary of an object is derived as below:

$$\text{Boundary of object} = \mathbf{B} - (\mathbf{A} \cap \mathbf{B}), \quad (2)$$

where the original binary object **A** is extended one pixel using a dilation operation to derive new binary object **B** as shown in Fig. 2b. Then, the intersection of objects **A** and **B** is subtracted from the object **B** to obtain the boundary, as shown in Fig. 2c. Then, a commonly used method, the boundary-following algorithm of Tien, Yeh, and Hsieh (2004), is then employed to extract the boundary information. Finally, the boundary points, totally 912 points in this example, are stored in a sequential array.

2.3. Feature extraction

The proposed feature extraction process derives the transition, rotation and scaling-invariant features based



Note: there are 912 boundary points (s=91) in total.

Fig. 3. Boundary representation using the smaller eigenvalue of covariance matrices.

on the obtained sequential boundary. This study represents the boundary by the smaller eigenvalue of covariance matrices to obtain the transition invariant property; uses a re-sampling process to obtain scaling-invariant property; and employs the transformation of autocorrelation to ensure the orientation-free property.

2.3.1. Boundary representation using eigenvalue of covariance matrices

As a curvature-based boundary descriptor developed by Tsai, Hou, and Su (1999), the eigenvalue of the covariance matrix method adopts the smaller eigenvalue of covariance matrices to estimate the curvature of a two-dimensional boundary point over a small region of support. Let the sequential n digital pixels describe a boundary $\mathbf{P}$,

$$\mathbf{P} = \{p_i = (X_i, Y_i), i = 1, 2, 3, \ldots, n\}, \tag{3}$$

where p_{i+1} is adjacent to p_i on $\mathbf{P}$. Additionally, let $\mathbf{N}_s(p_i)$ denote a small boundary segment, called a region of support, centered on point p_i over the region between points p_{i-s} and p_{i+s} for some integer s, such that

$$\mathbf{N}_s(p_i) = \{p_j | i - s < j < i + s\}. \tag{4}$$

Hence, the covariance matrix $\mathbf{M}$ of a boundary segment $\mathbf{N}_s(p_i)$ is expressed as

$$\mathbf{M} = \begin{bmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{bmatrix}, \tag{5}$$

where

$$m_{11} = \left[\frac{1}{2s+1}\sum_{j=i-s}^{i+2} x_j^2\right] - \bar{x}_i^2,$$

$$m_{22} = \left[\frac{1}{2s+1}\sum_{j=i-s}^{i+2} y_j^2\right] - \bar{y}_i^2,$$

$$m_{12} = m_{21} = \left[\frac{1}{2s+1}\sum_{j=i-s}^{i+2} x_j y_j\right] - \bar{x}_j \bar{y}_j,$$

$$\bar{x}_i = \frac{1}{2s+1}\sum_{j=i-s}^{i+2} x_j,$$

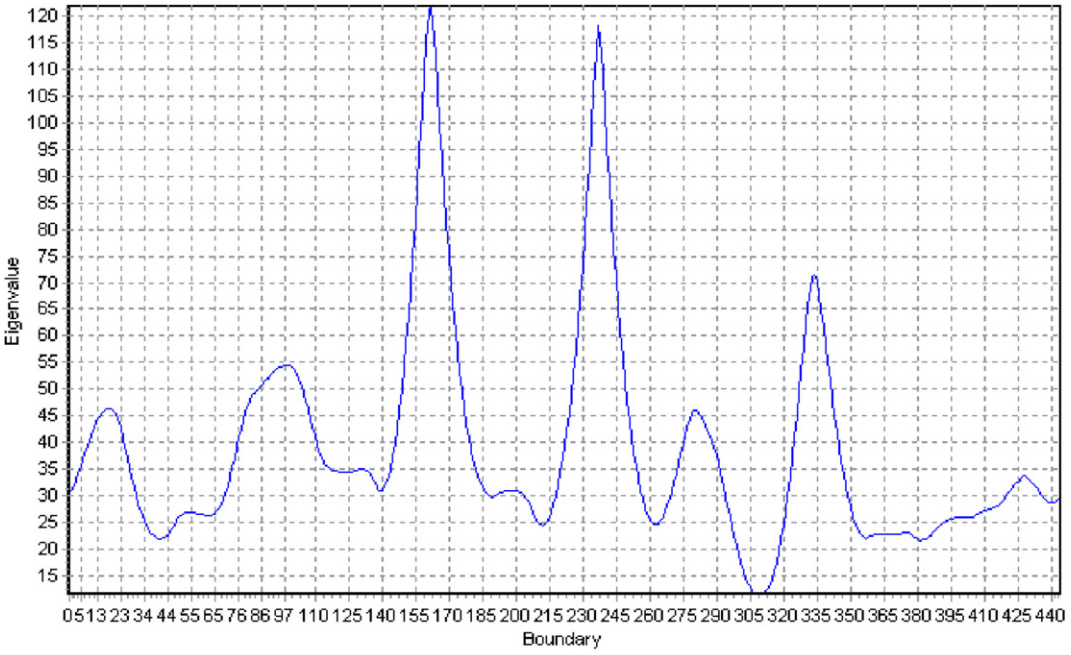
$$\bar{y}_i = \frac{1}{2s+1}\sum_{j=i-s}^{i+2} y_j,$$

where $\bar{x}_i$ and $\bar{y}_i$ are the geometrical center of $\mathbf{N}_s(p_i)$. The covariance matrix M is a 2×2, symmetric and positive semi-definite matrix. The eigenvalues λ_L and λ_S ($\lambda_L > \lambda_S$) of matrix $\mathbf{M}$ are obtained as follows:

$$\lambda_L = (m_{11} + m_{22} + \sqrt{(m_{11} + m_{22})^2 + 4(m_{11}^2 - m_{12}^2)}), \tag{6}$$

$$\lambda_S = (m_{11} + m_{22} - \sqrt{(m_{11} + m_{22})^2 + 4(m_{11}^2 - m_{12}^2)}). \tag{7}$$

For an elliptical object, the square roots of λ_L and λ_S are the semi-major and semi-minor axial lengths of the object, respectively. The λ_S approximates to zero when point p_i is on a straight line or on a flat curve. The smaller the radius of a circle, the larger the λ_S obtained,and vice versa. In particular, the λ_S of a corner point on the boundary segment is generally a local maximum, which is a peak when representing the boundary as a 1D sequence, while the λ_S of



Note: There are 455 boundary points (s=44).

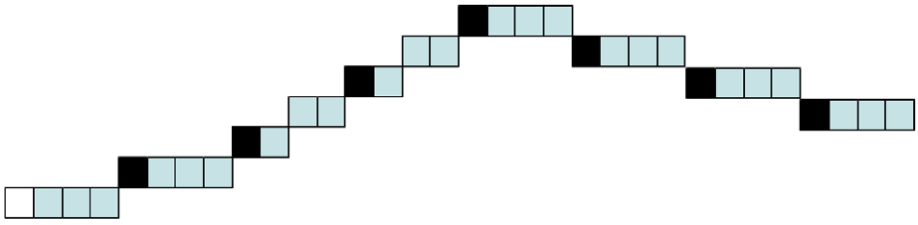
Fig. 4. Boundary representation with 1/4 scale.

the boundary points gradually decreases away from the corner. Using the region of support $s = 30$, the boundary representation of the digital object shown in Fig. 2 is illustrated in Fig. 3.

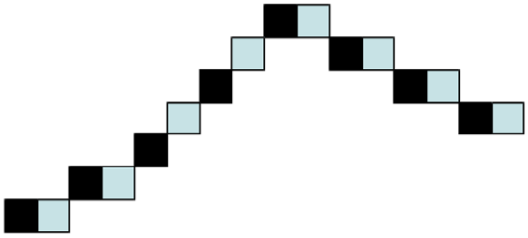
2.3.2. Re-sampling

The boundary representation derived previously is scale-dependent when using the same starting-point. A digital object with different scales is converted into different numbers of eigenvalue, λ_S. Fig. 4 illustrates the boundary representation of the original object in Fig. 2 when the size is reduced to 1/4. The total number of boundary points is decreased from 912 (with $s = 91$) to 445 (with $s = 44$). Very minor difference can be differentiated from the curves shown in Figs. 4 and 5. This is achieved by selecting a proper relationship between the number of boundary points (n) and the region of support (s), which will be derived empirically.

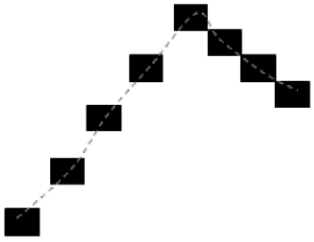
The purpose of re-sampling is to obtain the same number of λ_S for a given object with different scales such that the scale-invariant property is derived. This process can be achieved by simply sampling the eigenvalues



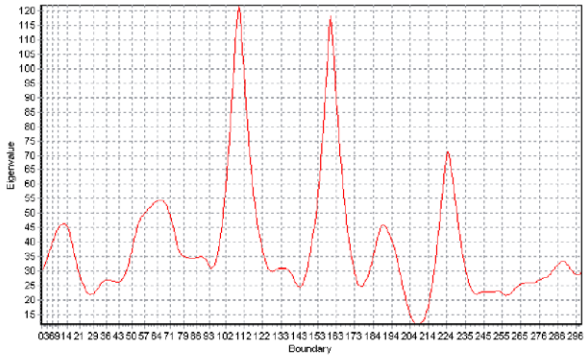
(a) the numbers of λ_S=32



(b) the numbers of λ_S=16



(c) the numbers of λ_S=8 (after re-sampling)



(d) Boundary representation after re-sampling (n=300)

Fig. 5. The process of re-sampling.

proportionally to the numbers of λ_S extracted. For instance, given a digital object of two different scales in Fig. 5a and b, the numbers of two λ_S sequences are 32 and 16. To obtain the same number of features such as 8, the re-sampling process sequentially samples one of every four features along the first sequence, and one of every two features along the second sequence. Theoretically, this process does not alter the profile of boundary representation as shown in Fig. 5c. After re-sampling the boundaries of the objects with different scales, the re-sampled boundary then has the same number of pixels. An example is also shown in Fig. 5d to demonstrate the boundary representation of the object, originally presented in Fig. 4, after re-sampling to $n = 300$.

2.3.3. Autocorrelation

Autocorrelation coefficients have been extensively applied as features for 1D and 2D signal classification in various applications such as character recognition, texture classification, face detection and recognition, signal classification and so forth (Popovici & Thiran, 2004). Autocorrelation coefficient provides important information about matching pattern in time-series data (Makridakis & Wheelwright, 1978). The formula of autocorrelation coefficient of time lag K is as follows:

$$\rho_K = \frac{\sum_{t+1}^{n-K}(r_t - \bar{r})(r_{t+K} - \bar{r})}{\sum_{t=1}^{n}(r_t - \bar{r})^2}, \tag{8}$$

where ρ_K denotes the autocorrelation coefficient; K is the length of the time lag $(K = 1, 2, \ldots, n)$; n represents the number of observations; r_t denotes the value of the variable at time t; and $\bar{r}$ denotes the mean of r_t. The benefit of adopting autocorrelation coefficients is in their shift-invariant property. After applying Eq. (8), the re-sampled boundary sequence becomes invariant to starting points. Fig. 6 is an illustration of boundary sequence with different starting points after applying the autocorrelation transformation with $K = 250$. After the boundary representation with the eigenvalue of covariance matrix, re-sampling and autocorrelation transformation, the derived feature patterns are invariant to translation, rotation and scaling.

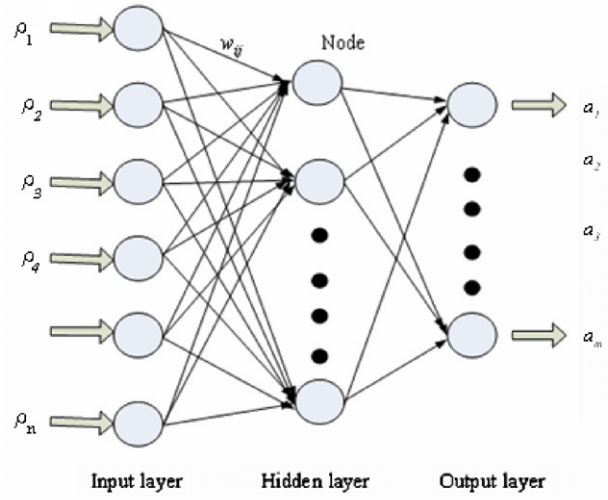


Fig. 7. Architecture of the network for object recognition.

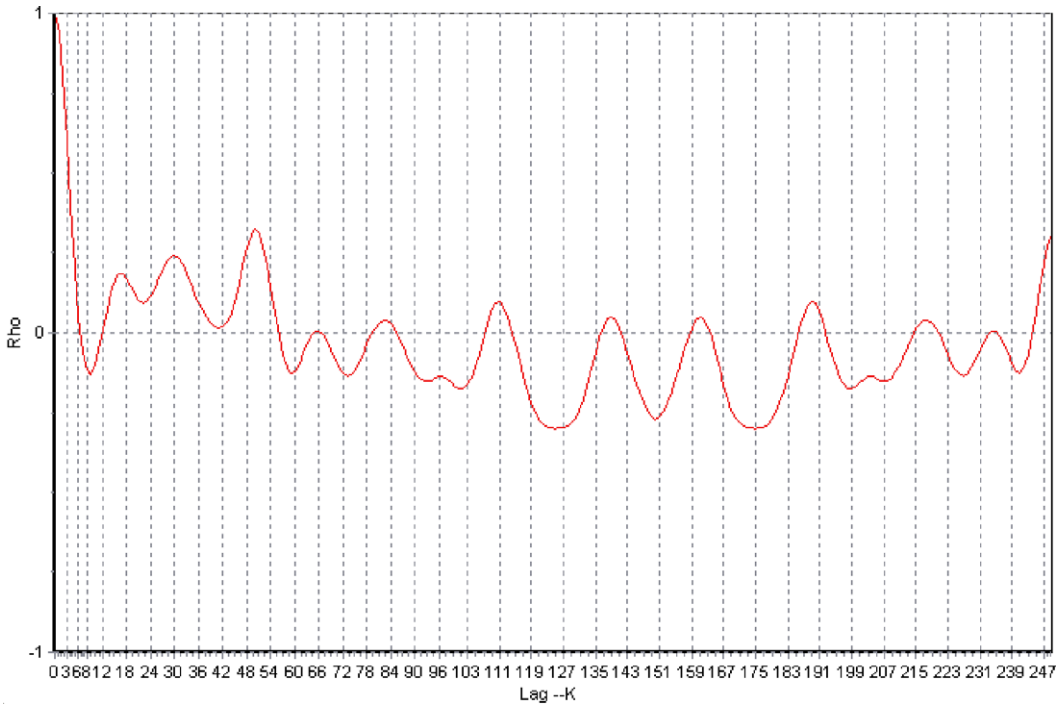


Fig. 6. Boundary sequence after autocorrelation transformation.

2.4. Decision making

2.4.1. Minimum euclidean distance

Calculating the minimum Euclidean distance between patterns is a simple but useful pattern recognition method. Assuming patterns **A** (standard pattern), and **B** (test patterns) with n feature elements denoted as

$$\mathbf{A} = [a_1, a_2, \ldots, a_n],$$
$$\mathbf{B} = [b_1, b_2, \ldots, b_n].$$

Euclidean Distance between **A** and **B** is calculated as follows:

$$\text{Distance } (\mathbf{A}, \mathbf{B}) = \sqrt{\sum_{i=1}^{n} (a_i - b_i)^2}. \tag{9}$$

Hence, the index of the targeting pattern with the minimum distance to the testing pattern is determined as the class of the testing pattern belongs.

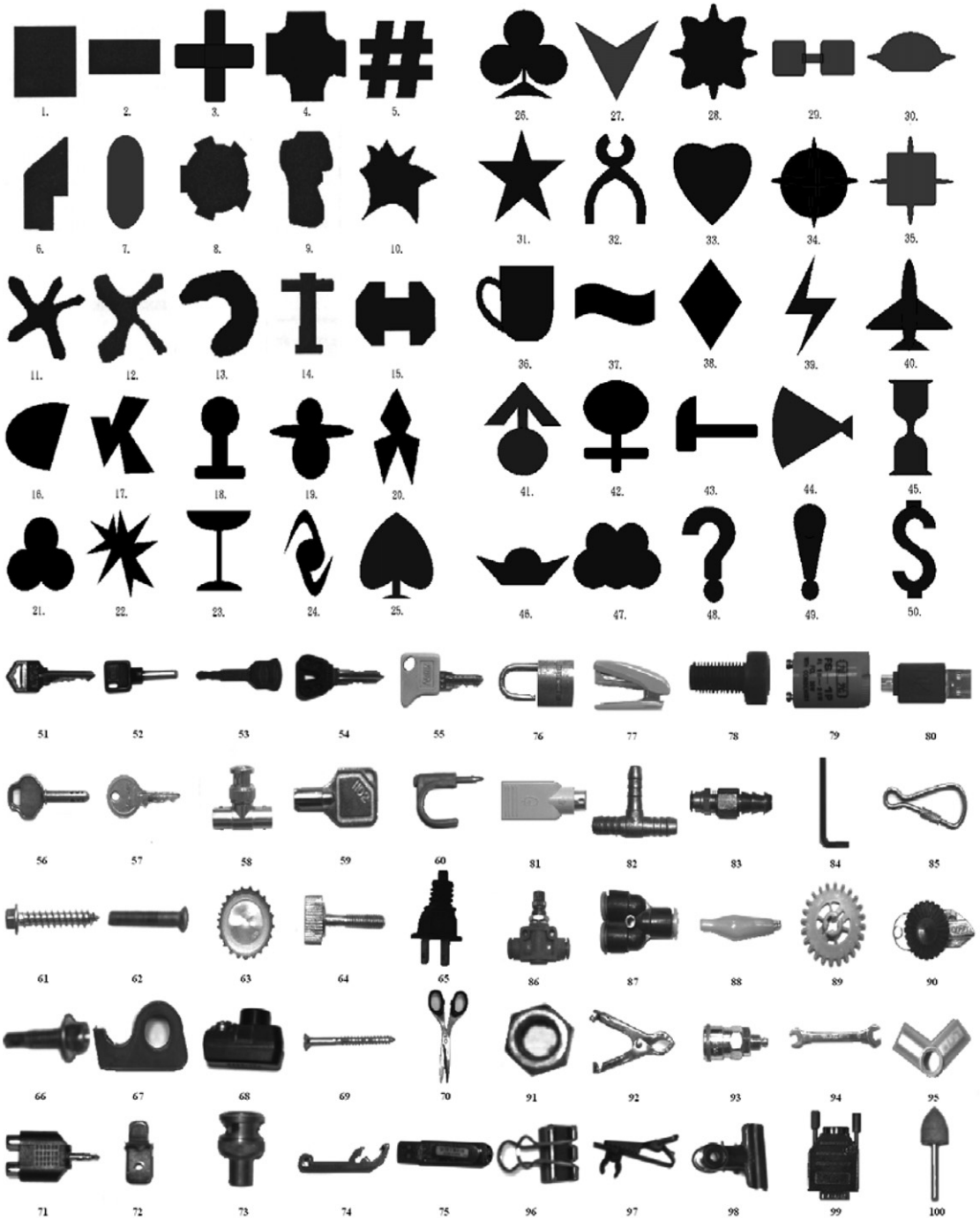


Fig. 8. Standard patterns.

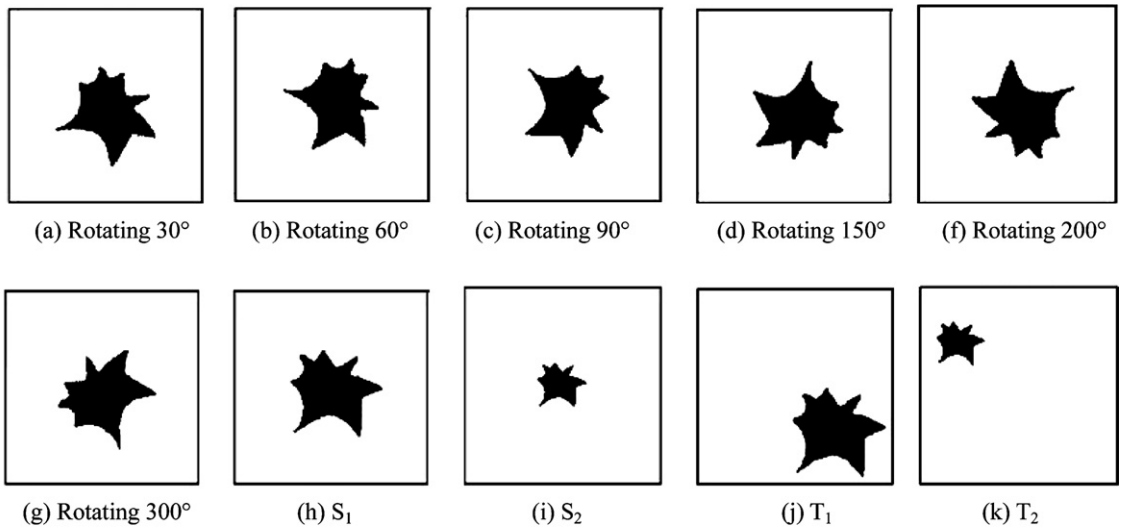


Fig. 9. Ten test patterns with various translation, rotation and scaling factors.

2.4.2. Backpropagation neural network

Backpropagation neural networks (BPN) are the most widely used ANN networks and are considered as the work horse of artificial neural networks (Rumelhart, Hinton, & Williams, 1986, chap. 8). As shown in Fig. 7, a BPN is a multilayer feed-forward network consisting of (1) an input layer with nodes representing input variables to the problem, (2) an output layer with nodes representing the dependent variables, and (3) one or more hidden layers containing nodes to help capture the nonlinearity in the data. The neurons between layers can be fully or partially interconnected between layers with weight (w_{ij}). The data are fed forward from the input layer, through hidden layer, to output layer without feedback. Then, based on the feed-forward error-backpropagation learning algorithm, BPN will search the error surface using gradient descent for point(s) with minimum error. This network is versatile and has been broadly applied to the areas of data modeling, classification, forecasting, control, data and image compression, and pattern recognition (Hassoun, 1995).

The purpose of using BP neural network in this study is to adopt the characteristics of memorizing and referencing properties that recognize the testing 2D boundary pattern based on the learned standard patterns. The input of the network is the 2D-invariant boundary feature (ρ_K), and the target is the designated index of the object. When training the BPN, the input pattern ρ_K is fed into the network, through the hidden layers and to the output layer. Using the gradient descent method, the network calculates the deviation between output, a_i, and the target pattern, T_i, and backpropagates errors into the networks. These extracted features are continuously fed into a BPN and the network will self-adjust until a set of weights (w_{ij}) with pre-specified error (SSE or MSE) is reached. Then these weights are stored and used for recognition later on. Fig. 7 also represents the architecture of the network for object recognition, in which ρ_i is the element of feature vectors with predetermined dimension n, and $a_i = 0$ or 1 is the output with the dimension of number of recognizing objects m.

When using BPN for object recognition, the processes, learning, testing, and validation, are three required stages. First, the collected standard 2D object boundary features are divided into 60%, 20%, and 20% for the use of learning, testing and validation (Basheer & Hajmeer (2000)). Then, the learning data set is fed into different structures of BPN, and is tested with the 20% testing data, in order to prevent from overtraining or under training. After learning and testing, the validation data (20%) are used to determine a proper BPN structure with the best recognition performance.

Table 1
Recognition rates with different s

Region of support (s)	t	Correct matches	Recognition rate (%)
$d = 30$	1	226	0.904
	2	225	0.900
	3	227	0.908
	4	228	0.912
	5	229	0.916*
$d = 20$	1	239	0.956
	2	240	0.960*
	3	240	0.960*
	4	238	0.952
	5	233	0.932
$d = 10$	1	250	1.000*
	2	243	0.972
	3	243	0.972
	4	242	0.968
	5	233	0.932

Note: 1. This table only shows the results with $d = 10$, 20, 30, and $t = 1$, 2, 3, 4, 5.
2. This experiment was undertaken using the Patterns 1–50 (with five extra testing images each).
* The best recognition rate(s).

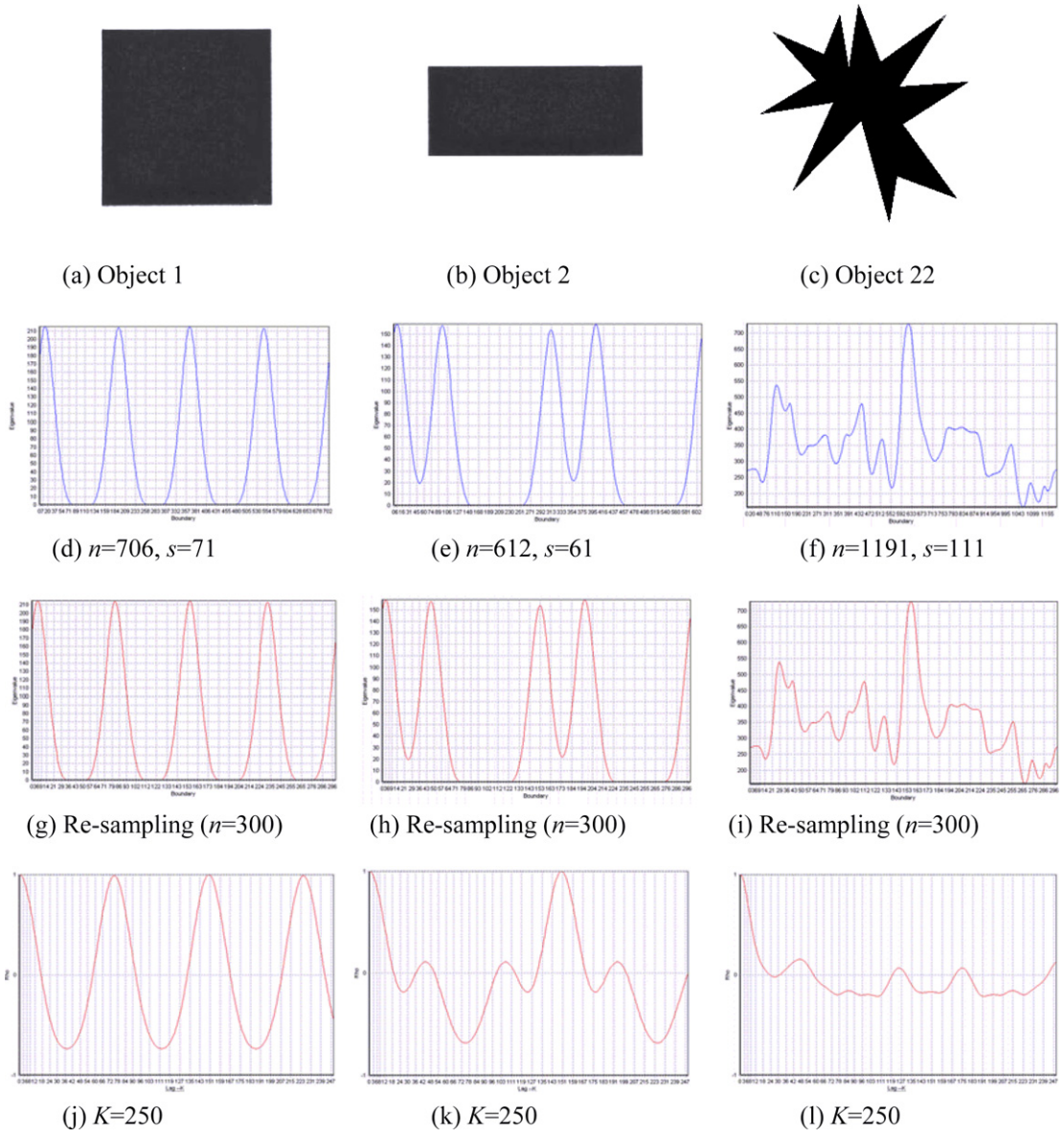
For the training process, the parameters including learning rate, momentum, types of transfer function, and structure of networks (number of hidden layer, and number of nodes in hidden layers) must be determined in advance. In this study, the nonlinear Sigmoid function, the most commonly used transfer function, and is used in both hidden and output layers as below.

$$f(x) = \frac{1}{1+e^{-x}}. \tag{10}$$

The number of input nodes will be determined by the dimension of feature vectors and the number of output nodes is dependent to the size of object database. The number of hidden layers was arbitrarily used one layer for simplicity. The number of nodes in hidden layer is first set according to the most commonly used formula as below.

$$\text{No. of hidden nodes} = \frac{\text{Input notes} + \text{output nodes}}{2}. \tag{11}$$

Neurons are added or deleted based on the performance of learning. However, the learning rate, momentum and final architecture of networks will be determined by an experimental design. During the experiment, the network is stopped at a pre-determined number of epochs or when SSE (or MSE) reaches a small number. The general setup of BPN, including data partition, normalization, input/output representation, weight initialization, learning rate, momentum, convergence criteria, training methods, hidden layer size, and parameter optimization, is implemented



(a) Object 1 (b) Object 2 (c) Object 22

(d) n=706, s=71 (e) n=612, s=61 (f) n=1191, s=111

(g) Re-sampling (n=300) (h) Re-sampling (n=300) (i) Re-sampling (n=300)

(j) K=250 (k) K=250 (l) K=250

Fig. 10. Recognition process of Objects 1, 2, and 22.

based on the fundamentals discussed in Basheer and Hajmeer (2000).

3. Implementation

3.1. System configuration

The proposed method was implemented using Borland C++ Builder on a personal computer with a Pentium IV-2.8G CPU, RAM DDR 512M. The images were acquired with a computer vision system comprising a Teli-8205i monochrome CCD camera, a Piccolo-Pro 2 (Euresys Ltd.) frame grabber with 256×256 pixels and 8-bit grey levels.

3.2. Experiments

One hundred standard patterns, illustrated in Fig. 8, were considered as training patterns (target) for system validation. Among them, Patterns 1–50 were synthetic patterns, while Patterns 51–100 were acquired using real objects. Ten additional test patterns with various positions, rotations (30°, 60°, 90°, 150°, 200°, 300°) and scales (1/4 and 1/8) were created for each object. Fig. 9 displays the 10 test patterns generated using standard Pattern 10. Hence, 1000 test patterns were created in total. All standard and test patterns were then segmented by a pre-determined threshold and stored as binary images.

Following the proposed framework in Fig. 1, the boundary was first extracted using the morphological operations and boundary following algorithm. For the boundary representation with the eigenvalue of covariance matrix, the region of support, s, was correlated with the number of pixels in the boundary (n). The relationship between s and n can be presented as

$$s = (n/d) + t. \tag{12}$$

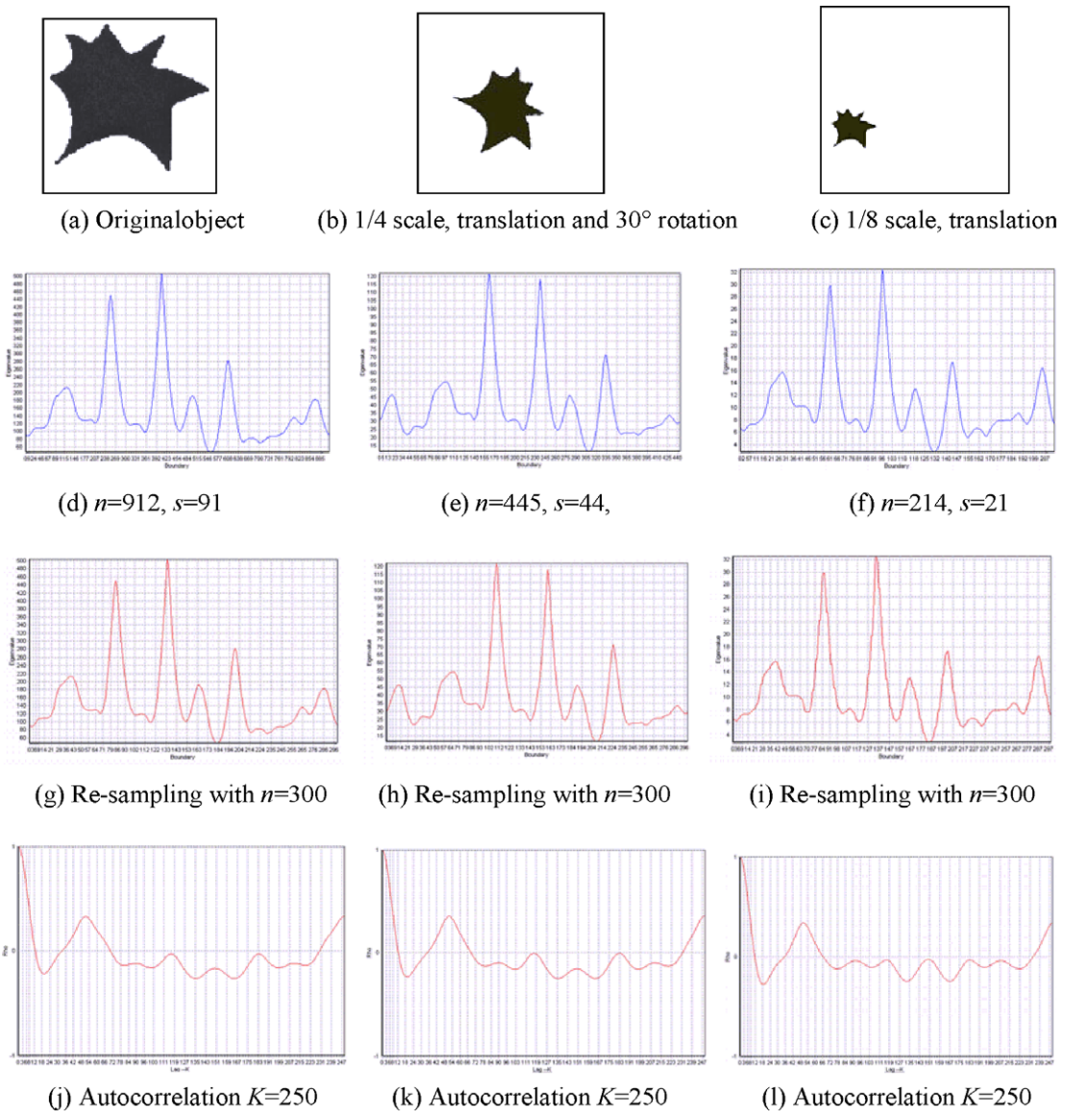


Fig. 11. Recognition using different positions, orientations, and scales.

Table 2
Recognition results of BPN and MD

Parameters	Methods	
	BPN	MD
Region of support (s)	20	20
Re-sample	200	200
Lag (K)	100	100
Recognition rate(%)	99.5%	97.2%

An experiment was undertaken to determine the parameters d and t and experimental results show that $s = (n/10) + 1$ gives the best recognition rate, as presented in Table 1.

Eventually, we used Objects 1, 2, and 22, for instance, to demonstrate the robustness of the proposed method. Fig. 10 illustrates the extracted invariant features after applying the proposed method. The boundary representation using the smaller eigenvalue of covariance matrices for Objects 1, 2, and 22, are shown in Fig. 10d–f, which have 706, 612, and 1191 boundary points, respectively. After that, the boundary sequences were re-sampled into 300 features as shown in Fig. 10g–i. Then, Fig. 10j–l depict the patterns after autocorrelation transformation with $K = 250$. Obviously, the square pattern (Object 1) and rectangle pattern (Object 2) have distinct sequences which can be easily differentiated. In addition, the extracted features of Object 22 have conspicuous difference to Objects 1 and 2's.

Furthermore, Fig. 11 shows the effect of applying the proposed method on Object 10 with different positions, orientations and scales. Fig. 11a–c illustrate the original digitized images of Object 10. Their boundaries were extracted as 912, 445, and 214 points, and represented by the eigenvalue of covariance matrices. Fig. 11d–f depict the boundary representation with different numbers of point (n) and region of support (s), respectively. The boundary sequences were then re-sampled into 300 points. After that, the features were further transformed using autocorrelation function, and are shown in Fig. 11j–l. Accordingly, the extracted features are obviously similar series, and can be utilized for object recognition.

After obtaining the invariant feature patterns, BPN and minimum Euclidean distance methods were implemented for classification. After an experimental design, we selected a two-layer BPN with learning rate = 0.5, momentum = 0.8, and 300–190–100 architecture based on the ability of minimizing the mean square error without overtraining. The experimental results, with the selected best parameter combination, are listed in Table 2, which shows that both methods obtained high recognition rates and BPN slightly outperforms MD.

4. Conclusions

Computer vision systems are extensively adopted in industrial product inspection and quality control tasks. To fulfill the requirement, automated object recognition with invariant properties becomes an important issue. This study presents a novel boundary-based automated object recognition method that incorporates the eigenvalue of covariance matrices, re-sampling, and autocorrelation transformation to extract unique and invariant features. Experiments were undertaken to verify the proposed method. Accordingly, we conclude that: (1) the smaller eigenvalue of covariance matrices is an effective boundary descriptor that owns the translation invariant property; (2) re-sampling process successfully produces the scaling-free property; (3) autocorrelation transformation achieves the ideal rotation invariant property with a pre-determined lag. After applying backpropagation neural networks for object recognition with minimum Euclidean distance method as a benchmark, we conclude that BPN slightly outperformed MD method and obtained almost perfect recognition. This implies that the extracted invariant features can be easily applied to object recognition using simple classifiers. The proposed method has been practically applied to various applications in industrial manufacturing such as object sorting, robot positioning and automated inspection systems. However, the appropriate values of threshold (T) and region of support (s) must be determined by trial-and-errors at this stage, for which is left for future study.

Acknowledgements

The authors would like to thank the National Science Council of the Republic of China, Taiwan for financially supporting this research under Contract No. NSC 95-2221-E-027-058.

References

Basheer, I. A., & Hajmeer, M. (2000). Artificial neural networks: Fundamentals, computing, design, and application. *Journal of Microbiological Methods, 43*, 3–31.

Bruckstein, A., & Shaked, D. (1997). Skew-symmetry via invariant signatures. *Pattern Recognition, 31*(2), 181–192.

Cao, W., Hao, F., & Wang, S. (2004). The application of DBF neural networks for object recognition. *Information Sciences, 160*, 153–160.

Gonzalez, R. C., & Woods, R. E. (2004). *Digital image processing*. Englewood Cliffs NJ: Prentice Hall.

Haralick, R. M., & Shapiro, L. C. (1992). *Computer and robot vision*. Reading, MA: Addison-Wesley.

Hassoun, M. H. (1995). *Fundamentals of artificial neural networks*. Cambridge, MA: MIT Press.

Hecht-Nielsen, R. (1990). *Neurocomputing*. Reading, MA: Addison-Wesley.

Huang, X., Wang, B., & Zhang, L. (2005). A new scheme for extraction of affine invariant descriptor and affine motion estimation based on independent component analysis. *Pattern Recognition Letters, 26*, 1244–1255.

Jones, G., III, & Bhanu, B. (2001). Recognizing articulated objects in SAR images. *Pattern Recognition, 34*, 469–485.

Khalil, M. I., & Bayoumi, M. M. (2000). Invariant 2D object recognition using the wavelet modulus maxima. *Pattern Recognition Letters, 21*, 863–872.

Khalil, M. I., & Bayoumi, M. M. (2002). Affine invariants for object recognition using the wavelet transform. *Pattern Recognition Letters, 23*, 57–72.

Kreutz, M., Vöpel, B., & Janben, H. (1996). Scale-invariant image recognition based on higher-order autocorrelation features. *Pattern Recognition, 29*, 19–26.

Kyrki, V., Kamarainen, J.-K., & Kälviänen, H. (2004). Simple Gabor feature space for invariant object recognition. *Pattern Recognition Letters, 25*, 311–318.

Lee, Y. H., Moon, S., & Lee, H. (1997). A new approach for automated parts recognition using time series analysis and neural networks. *Journal of Intelligent Manufacturing, 8*, 167–175.

Li, W., & Lee, T. (2004). Projective invariant object recognition by a Hopfield network. *Neurocomputing, 30*, 1–18.

Makridakis, S., & Wheelwright, S. C. (1978). *Forecasting methods and application*. New York: John Wiley & Sons.

Malamas, E. N., Petrakis, E., Zervakis, M., & Petit, Laurent (2003). A survey on industrial vision systems, applications and tools. *Image and Vision Computing, 21*, 171–188.

Noble, J. A. (1995). From inspection to process understanding and monitoring: A view on computer vision in manufacturing. *Image and Vision Computing, 13*(3), 197–214.

Oswald, N., & Levi, P. (2001). Cooperative object recognition. *Pattern Recognition Letter, 22*, 1273–1282.

Popovici, V., & Thiran, J.-P. (2004). Pattern recognition using higher-order local autocorrelation coefficients. *Pattern Recognition Letters, 25*, 1107–1113.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning internal representation by error propagation. In D. E. Rumelhart & J. L. McClleland (Eds.), *Parallel distributed processing: Exploration in the Microstructure of Cognition* (Vol. 1). Cambridge, MA: MIT Press.

Sarkodie-Gyan, T., Lam, C. W., Hong, D., & Campbell, A. W. (1996). An efficient object recognition scheme for a prototype component inspection. *Mechatronics, 7*(2), 185–197.

Schalkoff, R. J. (1997). *Artificial Neural Networks*. New York: McGraw-Hill.

Smith, K. A., & Gupta, J. N. D. (2000). Neural networks in business: Techniques and applications for the operations researcher. *Computer and Operations Research, 27*, 1023–1044.

Sookhanaphibarn, K., & Lursinsap, C. (2006). A new feature extractor invariant to intensity, rotation, and scaling of color images. *Information Sciences, 176*, 2097–2119.

Tien, F. C., Yeh, C. H., & Hsieh, K. H. (2004). Automated visual inspection of microdrill for PCB production. *International Journal of Production Research, 15*(42), 2477–2495.

Tsai, D. M., Hou, H.-T., & Su, H.-J. (1999). Boundary-based corner detection using eigenvalues of covariance matrices. *Pattern Recognition Letters, 20*, 31–40.

Tsang, P. W. (1997). A genetic algorithm for affine invariant recognition of object shapes from broken boundaries. *Pattern Recognition Letters, 18*, 631–639.

Ullman, S. (1996). *High-level vision*. Cambridge, MA: MIT Press.

Wöhler, C., & Anlauf, J. K. (2001). Real-time object recognition on image sequences with the adaptable time delay neural network algorithm - applications for autonomous vehicles. *Image and Vision Computing, 19*, 593–618.

Yu, H., & Bennamoun, M. (2007). Complete invariants for robust face recognition. *Pattern Recognition, 40*, 1579–1591.

Zhang, J., Zhang, X., Krim, H., & Walter, G. G. (2003). Object representation and recognition in shape spaces. *Pattern Recognition, 36*, 1143–1154.