Int J Adv Manuf Technol (1999) 15:843–850

The International Journal of Advanced Manufacturing Technology

A Computer Vision-Based Shape-Classification System Using Image Projection and a Neural Network

Tung-Hsu Hou and Ming-Der Pern

Institute of Industrial Engineering and Management, National Yunlin University of Science and Technology, Yunlin, Taiwan

In this research, a computer vision-based shape-recognition system which combines an image projection and the back propagation neural network is proposed. The system uses the image projection and features of the projection histogram to represent the shape, and uses the back propagation neural network to classify the shape. In this research, a fast normalisation scheme is developed which is especially designed for the image projection. It can find the same projection axis no matter how the object is oriented. Therefore, the proposed system can be invariant to scale, translation, and orientation variations. This research finds that the direct representation scheme of the back propagation neural net has better accuracy than the binary coding representation scheme. This research also finds that classification accuracy of the proposed system is greater than 98.8% and its processing speed is also very fast. Therefore, the proposed system is feasible for use in shape classification.

Keywords: Image projection; Neural networks; Shape classification

1. Introduction

Computer vision-based shape recognition techniques have been widely applied in automatic assembly tasks, industrial inspection, and product identification. In an automated assembly task, a part has to be recognised and classified and its orientation has to be calculated automatically before a robot can be driven to grasp the part. In an automatic inspection task, the test part has to be recognised first in order to retrieve the standard specification for the part. Since different parts have different shapes, a computer vision-based shape-recognition system or classifier is usually used to recognise or classify parts in industry. Shape-recognition techniques have also been applied in optical character recognition, hand-written character recognition, and medical image recognition, etc.

Correspondence and offprint requests to: Dr T.-H. Hou, Department of Industrial Engineering, National Yunlin Institute of Technology, Touliu, Yunlin 640, Taiwan.

In a part-recognition system, the part may be rotated at any angle or translated to any position because of the vibrations of the transport devices. The part's image scale may be also be changed when the image acquisition distance changes. Therefore, a part-recognition system should be invariant to scale, translation, and orientation, in order to be used in practice.

A shape-recognition system generally contains two procedures: a shape-representation procedure and a shape-matching procedure.

Many shape-representation schemes have been proposed. Horaud et al. [1] used geometric features such as area, perimeter length, centre of gravity, maximum radius, minimum radius, etc. to represent the shape of a part. Although the geometric features are useful to represent a shape and are easy to calculate, they are sensitive to scale and orientation variations.

Gupta and Srinath [2] created a signature model which was an ordered sequence representing the Euclidean distance between the centroid and all the boundary points, and extracted the moments of the signature as features to represent the shape. Gupta and Srinath [2] concluded that moments of the signature were invariant to the variations of scale, translation and orientation. However, their system's recognition accuracy was only about 95%. In addition, it was a time-consuming process to form the signature model.

Amin and Wilson [3] combined a morphological skeleton transform and a neural network to classify characters. Cordella et al. [4] also used a skeleton transform and structural descriptions for optical character recognition. However, the morphological skeleton transformation is time consuming, and the transformation is also sensitive to scale variations.

Kim and Nam [5] applied a Fourier transform to the boundary coordinates of a shape and used Fourier coefficients to represent the shape. They then applied a back propagation neural network to classify an object. However, the Fourier transformation is also a time-consuming procedure.

Image projection of a shape has been used to represent a shape. Breuer and Vajta [6], and Wang [7] applied image projection on characters, extracted the moments of the projection histogram as features to represent the shape of the characters, and then used a statistical decision classifier to classify a character. Although image projection is not a new technique, it is not much used. The reason may lie in the projection histogram which is sensitive to scale and orientation variations. Objects with different scales will have different scale projection histograms, although the shapes of the projection histograms are similar. In addition, the same object with a different orientation will result in a different projection histogram if it is projected at a fixed direction. The scale variation problem can be solved when the feature extraction procedure is well developed, while the orientation variation usually cannot be solved until an orientation normalisation procedure is applied. Traditionally, a normalisation procedure has to orient the whole image, which is a time-consuming process. We have developed a fast normalisation scheme which can find the same projection axis however the object is oriented, and we have developed a unique feature extraction procedure to extract features from the image projection histogram. Therefore, the classifier developed in this research is able to take a projection histogram and extract the required features very quickly.

The other component of a shape-recognition system is the shape-matching procedure. Many shape-matching methods have been developed, and these methods can be classified into four categories [8]:

1. Template matching.
2. Syntactic and structural methods.
3. Decision theoretic analysis.
4. Neural networks.

The template-matching method classifies the test shape by matching it with some predefined templates. Therefore, this method requires much memory space. Syntactic and structural methods decompose a shape into some basic primitives which are represented by a sentence in a language specified by a grammar. Then, the shape is recognised by using a grammar parser. However, the parsing procedure and the grammar are complex and computationally extensive. Decision theoretic analysis uses the features extracted from the shape to form a feature vector, and uses statistical methods to classify the shape. The well-known decision theoretic methods are the minimum-distance classifier, the nearest-neighbour classifier, and the minimum-mean-distance classifier.

Artificial neural networks, which simulate human information processing, are inherent in parallel processing and have the ability to learn. When a neural net is trained with a series of examples, it is able to learn the governing relationships between the examples. The trained neural net is then used to classify a test pattern based on the learning result. Many neural networks have been proposed and the back propagation neural net developed by Rumelhart et al. [9] is one of the most useful and popular. It has been successfully applied in hand-written character recognition [3,10], object recognition [5], OCR recognition [4], shape classification [2], printed character recognition [11], solder joint inspection [12], and integrated circuit chip inspection [13].

In this research, we combine image projection and the back propagation neural net to classify a shape. This classifier uses the image projection and the features extracted from the projection histogram to represent the shape, and uses the back propagation neural net to classify the shape. A fast normalisation scheme which can find the same projection axis however the object is oriented is also developed so that the classifier can achieve orientation invariance.

2. The Shape-Classifier Framework

The system framework of the proposed shape classifier is shown in Fig. 1. The classifier consists of an image binarisation, an image projection, an orientation normalisation scheme, a feature extraction procedure, and the back propagation neural network.

2.1 The Image Binarisation Procedure

When a charge-couple device (CCD) camera is used to capture the image of a part, it produces a grey-level image. Therefore, an image binarisation procedure is required to segment the part from the background. The image binarisation procedure is used to convert the grey-level image into a binary image in which there are only two grey levels: black (representing the background) and white (representing the object). Many thresholding techniques have been proposed. Some of the most popular global thresholding techniques which threshold an entire image with a single value are the P-tile method, the mode method, the Otsu method, the histogram concavity analysis method, the entropy method, the moment-preserving method, and the minimum error method [14]. Sahoo et al. evaluated these techniques with many kinds of images and found that there was no technique which would always yield the best performance. Instead, the performance of thresholding techniques depended on the shape of a histogram which might be unimodal, bimodal, or multi-modal. Sahoo et al. also found that the Otsu method had the best performance in terms of uniformity and shape factor for a bimodal image. Since the

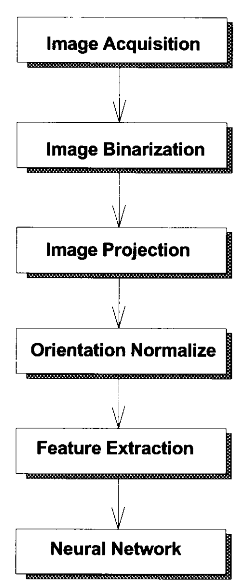

Fig. 1. The system framework.

image in shape recognition usually has homogeneous regions and has a favourable contrast between the object and the background, it has a bimodal histogram. Thus, the Otsu method is used in the proposed shape classifier. Detailed algorithms of the Otsu method can be found in [12].

2.2 Image Projection and Feature Extraction

Image projection is used to acquire shape information for an object, and the features extracted from the projection histogram are then fed to the back propagation neural network to classify the object. In the image-projection procedure, the projection histograms are created along the X- and Y- axes of the binary image. The so-called X-axis projection is to count the number of the object points (white pixels) along the Y-axis for all X-values. The so-called Y-axis projection is to count the number of object points along the X-axis for all Y-values. If we let $P(x,y) = 1$, which represents an object point, and $P(x,y) = 0$, which represents a background point, then the X-axis projection results in the following projection histogram:

$$f(x_i) = \sum_{Y=1}^{512} P(x_i,Y) \quad (i = 1,2,\ldots,512) \tag{1}$$

Similarly, the Y-axis projection results in the following projection histogram:

$$f(y_i) = \sum_{X=1}^{512} P(X,y_i) \quad (i = 1,2,\ldots,512) \tag{2}$$

The X-axis and Y-axis projection histograms of a part are shown in Fig. 2.

A major feature of the projection histogram is that the shape of the histogram remains the same no matter how the object is translated or scaled. Another feature of the projection histogram is that different objects will result in different projection histograms. Therefore, the projection histograms are useful for shape representation.

Wang [7] extracted the moments of the projection histogram as features to classify a shape. In this research, we propose another unique feature extraction procedure. First of all, we segment the X-axis projection histogram into K_X equal-length blocks, and then calculate the area under each block for all the K_X blocks, and finally divide each block area by the total area to acquire K_X the block area ratios. Similarly, we can acquire the K_Y block area ratios for the Y-axis projection histogram. The main feature of the block area ratios is that their values remain the same even if the object is translated or scaled. This feature-extraction algorithm contains the following steps:

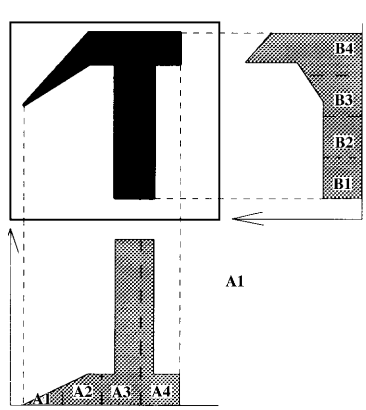

Fig. 2. Image projection histograms.

1. Let the grey level of an object pixel located at (x,y) be $P(x,y) = 1$, and the grey level of the background at (x,y) be $P(x,y) = 0$.
2. Calculate the area of the object, denoted as A, and

$$A = \sum_{X=1}^{512} \sum_{Y=1}^{512} P(X,Y) \tag{3}$$

3. Find the starting point and ending point of the X and Y projection histograms, denoted as (S_X,S_Y) and (E_X,E_Y), respectively, and segment the X and Y projection histograms into K_X and K_Y blocks.
   The length of each block of the X-axis projection is:

$$l_X = \frac{E_X - S_X}{K_X} \tag{4}$$

   The length of each block of the Y-axis projection is:

$$l_Y = \frac{E_Y - S_Y}{K_Y} \tag{5}$$

4. Calculate each block area of the X-axis projection histogram as follows:

$$A_i = \sum_{X=S_X+(i-1)\times l_X}^{S_X+i\times l_X} \sum_{Y=S_Y}^{E_Y} P(X,Y) \quad (i = 1,2,\ldots,K_X) \tag{6}$$

   and calculate each block area of the Y-axis projection histogram as follows:

$$B_i = \sum_{Y=S_Y+(i-1)\times l_Y}^{S_Y+i\times l_Y} \sum_{X=S_X}^{E_X} P(X,Y) \quad (i = 1,2,\ldots,K_Y) \tag{7}$$

5. Calculate the block area ratios of the X-axis projection as follows:

$$E_i = \frac{A_i}{A} \quad (i = 1,2,\ldots,K_X) \tag{8}$$

   and calculate the block area ratio of the Y-axis projection as follows:

$$E_{(i+K_X)} = \frac{B_i}{A} \quad (i = 1,2,\ldots,K_Y) \tag{9}$$

2.3 The Orientation Normalisation Procedure

Since the projection histogram is different when the object is oriented at any other angle, therefore, an orientation normalisation procedure is required. In this research, a unique normalisation procedure especially designed for the image projection is developed, and the procedure is shown in Fig. 3. The normalisation procedure contains the following steps:

1. Finding the centroid of the object.

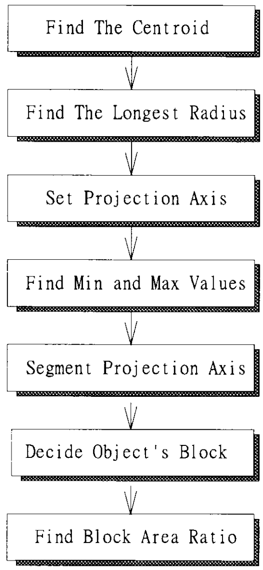

Fig. 3. The orientation normalisation procedure.

2. Finding the longest radius.
3. Setting the projection axis.
4. Finding the maximum and minimum values of the projection axis.
5. Segmenting the projection axis into some equal-length blocks.
6. Deciding the block for all the object points.
7. Calculating the block area ratio.

The algorithms for each step are described in the following:

1. *Finding the centroid of the object.* If an object has N pixels and (X_i,Y_i) is the coordinate of the ith pixel, then the centroid of the object (X_c,Y_c) is calculated as follows:

$$X_c = \frac{1}{N}\sum_{i=1}^{N} X_i, \quad Y_c = \frac{1}{N}\sum_{i=1}^{N} Y_i \tag{10}$$

2. *Finding the longest radius.* Since the longest radius should occur at the boundary points and should be located at the extreme left and right points of each row of the image, we scan the image from the left most to the right most pixel, row by row, until we find the extreme left boundary point whose coordinates and distance to the centroid are recorded. Next we scan from the right most to the left most pixel, row by row, to find the extreme right boundary point and record its coordinates and its distance to the centroid. The scanning process is repeated for all the rows. Finally, the boundary pixel which has the longest radius will be found by a simple comparison procedure. The locations where the longest radius may or may not occur are illustrated in Fig. 4.

By using the scanning algorithm, we can find the longest radius and save much processing time because we do not need to calculate all the distances between the boundary points and the centroid for all the boundary points. Instead, we calculate the distance between the centroid and the extreme left and right boundary points.

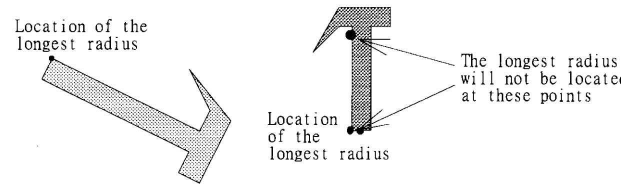

Fig. 4. Locations of the longest radius.

3. *Setting the projection axis.* For an object without any orientation variation, we take the projection histogram along the X- and Y-axis. However, if the object is oriented, the projection histogram taken along the X- and Y-axis will be different from that of the one without orientation variation. To resolve this problem, we set the longest radius as the projection Y-axis and the axis orthogonal to the longest radius as the projection X-axis, and then we take the projection histograms along these two axes. In this manner we can create the same projection histogram no matter how the object is oriented.

If the coordinate of the centroid is $O(X_c,Y_c)$ and the coordinate of the longest radius boundary point is (A_1,B_1), then the slope of the projection Y-axis, denoted as m_1, is calculated as follows:

$$m_1 = \frac{B_1 - Y_c}{A_1 - X_c} \tag{11}$$

and the slope of the projection X-axis, denoted as m_2, is calculated as $m_2 = -1/m_1$.

4. *Finding the maximum and minimum values of the projection axis.* When we put all the boundary points, which are scanned in the way described in the second step, into the equation $y - m_1x = b$, we can find all the corresponding b values and the maximum and minimum b values of the X-axis, denoted as $b_{\max}$ and $b_{\min}$, respectively. The algorithm is illustrated in Fig. 5. Similarly, when we put all the boundary points into the equation $y - m_2x = c$, we can find the maximum and minimum values of the Y-axis, denoted as $c_{\max}$ and $c_{\min}$, respectively.

5. *Segmenting the projection axis into some equal-length blocks* After we find the minimum and maximum values of the X-axis, we segment the interval of $[b_{\min},b_{\max}]$ into K_X equal-length blocks, and obtain $b_1,b_2,\ldots,b_{(K_X-1)}$ values. Similarly, we segment the interval of $[c_{\min},c_{\max}]$ into K_Y equal-length blocks, and get $c_1,c_2,\ldots,c_{(K_Y-1)}$ values. The segmenting algorithm is also illustrated in Fig. 5.

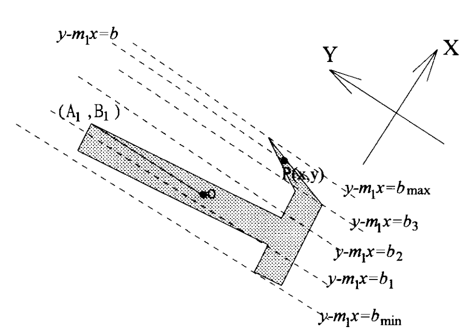

Fig. 5. Projection axis segmentation.

6. *Deciding the block for all the object points.* When we put the coordinate of an object point (x,y) into equations $y - m_1x = b$ and $y - m_2x = c$, we obtain the b and c values. From these values and the, $b_{\min}, b_1, b_2, \ldots, b_{(K_X-1)}, b_{\max}$ and the $c_{\min}, c_1, c_2, \ldots, c_{(K_Y-1)}, c_{\max}$, we know the interval to which the object point belongs.

When we sum up the number of the object points which belong to a specified block, we can acquire the area of the specified block. By this procedure, we can acquire all the areas for all the projection blocks.

7. *Calculating the block area ratio.* After the area for each equal-length block is obtained, we can calculate the block area ratio for each block by the equation described in Section 2.2.

2.4 The Back Propagation Neural Network

The neural network contains an input layer, a hidden layer, and an output layer, as shown in Fig. 6. Features of the projection histogram are fed into the nodes of the input layer. When there are eight features, there are eight nodes in the input layer. The output layer is used to represent the category of the classified shape. Two kinds of representation scheme can be used on the output layer. One is direct representation which uses eight nodes at the output layer if there are eight categories of shapes. In this representation, only one node gets the 1 value and the node number that has the 1 value is the shape category. For example, if the test shape belongs to the fifth category, then the output of the output layer will be 0010000. The other representation scheme is called binary coding representation which uses three nodes at the output layer for eight categories of shape. For example, if a shape belongs to the fifth category, then the output of the layer will be 101.

The hidden layer is used to represent the interaction between the input layer and the output layer. Usually the node number of the hidden layer is set to be half of the sum of the input nodes and output nodes.

The neural net uses a set of examples in a training database as input, the generalised delta rule algorithm to adjust the weights and the sigmoidal activation function to derive an output. After several iterations, the neural net will converge to a reasonable set of weights which represent the relationships. Then the neural network can be used to classify a new incoming shape.

The training database contains some examples that consist of attribute values and known categories. These attributes are fed into the nodes in the input layer (O_i), and the known category is set as t_{pk}. These attributes are then transmitted to the nodes in the hidden layer, and the net input to node j is

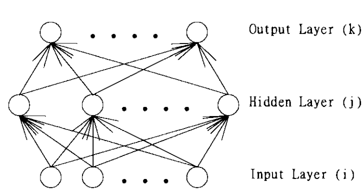

Fig. 6. Framework of the back propagation neural net.

$net_j = \Sigma w_{ji} o_i$. By applying the sigmoidal activation function, we can obtain the output of the node j for the hidden layer. The sigmoidal activation function is as follows [9]:

$$o_j = f(net_j) = \frac{1}{1 + \exp\left[\frac{(net_j + \theta_j)}{\theta_0}\right]} \tag{12}$$

where θ_j is the threshold or bias and θ_0 is the scaling factor. Similarly, the output from nodes in the hidden layer is fed into the nodes in the output layer. The process is called the feed forward stage.

After the feed forward process, output O_{pk} can be calculated from nodes in the output layer. In general, the output O_{pk} will not be the same as the desired known category t_{pk}. Therefore, the average system error is

$$E = \frac{1}{2p} \sum_p \sum_k (t_{pk} - o_{pk})^2$$

where k is the output node number and p is the example number.

If the error is greater than the preset value, then the error is back propagated from nodes in the output layer to the nodes in the hidden layer by using a gradient search method and is used to adjust the weights. The weight between node k and node j is adjusted according the following equation:

$$\Delta_p w_{kj} = -\eta \frac{\partial E}{\partial w_{kj}} = \eta \delta_k o_j \tag{13}$$

where, $\delta_k = (t_k - o_k) f'_k\,(net_k)$, and η is the learning rate. Similarly, the delta value in the hidden layer is $\delta_j = f'_j(net_j) \sum_k \delta_k w_{kj}$ and $\Delta_p w_{ji} = \eta \delta_j o_i$. The process is called the back propagation stage. After all the examples are trained, the system will collect adjusted weights according to $\Delta w_{ji} = \sum_p \Delta_p w_{ji}$. If the system error in nodes of the output layer is less than the threshold, then the current weights are desirable. The system will stop training and use the weights in the continuing recognition stage. Otherwise, the system will adjust weights according to the following equation:

$$\Delta w_{ji}(n+1) = \eta \delta_j o_i + \alpha \Delta w_{ji}(n) \tag{14}$$

where α is the momentum coefficient. Then the system will continue the next iteration. Selections of the learning rate η, momentum coefficient α, and number of nodes in a hidden layer have effects on computation efficiency. General rules for the selections have been reported by [15].

3. A Feasibility Study

3.1 A Pre-Test

A pre-test is conducted in this research to test the performance of the proposed system and to compare the output representation schemes of the neural net. Ten digits shown in Fig. 7 are tested in this pre-test. At the training stage, three images for each digit at three capturing distances are taken. The

Table 1. Classification accuracy of the binary coding representation.

Digit number	0	1	2	3	4	5	6	7	8	9	Average
Hit number	107	107	105	103	104	108	108	104	108	108	106.2
Test number	108	108	108	108	108	108	108	108	108	108	108
Accuracy (%)	99	99	97	95	96	100	100	96	100	100	98.3

1 2 3 4 5 6 7 8 9 0

Fig. 7. The pre-test digits.

capturing distances are 50 cm resulting in an image with 145×58 pixels, 65 cm resulting in an image with 112×44 pixels, and 80 cm resulting in an image with 90×36 pixels. Therefore, there are three training examples for each digit in the training database. At the classification stage, the test digit may be scaled at any of the above three capturing distances and may be translated at any location. However, the orientation remains unchanged in the test. For each digit 36 translated images for each scale are tested. Therefore, 108 images are tested for each digit.

At the image projection and feature extraction stages, the *X*-axis and *Y*-axis projection histograms are segmented into four equal-length blocks, and the eight block area ratios are calculated and fed into the back propagation neural net.

The two representation schemes of the output layer of the neural net are tested. For the direct representation scheme, ten nodes are used at the output layer. For the binary coding representation scheme, only four nodes are used at the output layer. Therefore, node numbers of the three layers of the direct representation neural net are 8, 9, and 10, respectively. For the binary coding representation scheme, the node numbers of the neural net are 8, 6, and 4, respectively. The learning rate and momentum rate are both set at 0.4 for both of the neural nets which can converge.

The pre-test results are shown on Tables 1 and 2. Table 1 shows that the classification accuracy of the neural net with the binary coding representation, is 98.3% in average. Table 2 shows the classification accuracy of the neural net with the direct representation scheme, which is 100%. Therefore, it is clear that the direct representation scheme has a better classification result than the binary coding representation scheme, and should be used in the back propagation neural network.

3.2 The Feasibility Study

In order to demonstrate the feasibility of the proposed shape-classification system, two case studies are conducted. The first case is general part classification and the second case is similar shape classification. In these two cases, a part may be scaled or translated and oriented at any location.

3.2.1 General Part Recognition Case

Twelve general parts are tested. The twelve parts and their projection histograms are shown in Fig. 8. In the neural net training database, three images taken at three orientations with a fixed scale for a part are used. Therefore, there are only three training examples for a part. However, if a part has more than two longest radii, the image projections along these radii are all taken, and these histograms are all included in the training database. At the feature extraction stage, the *X*-axis and *Y*-axis projection histograms are segmented into four equal-length blocks. Therefore, there are eight features in total.

Since there are eight features, eight nodes are used at the input layer of the neural net. Twelve nodes are used at the output layer of the direct representation neural net because there are twelve kinds of part. The hidden layer contains nine nodes which can lead to the convergence of the neural net. In addition, the learning rate and momentum rate are both set at 0.4 at which the neural net can come to convergence.

At the classification stage, three scales and thirty-two random orientations for each scale are tested. Therefore, 96 images for a part are tested. Experimental classification accuracy is shown in Table 3. This table shows that the proposed system on average has up to 98.8% classification accuracy. This classification accuracy is greater than 95%, and is thus feasible for practical use.

The proposed system takes only 6.5 s to analyse and classify a whole part using a 386 PC and the DT-2851 image frame grabber. The proposed system is also feasible with respect to the processing time.

3.2.2 Similar Shape Recognition Case

Four similar shapes and their projection histograms are shown in Fig. 9. In the training database, one shape contains only three training examples which are taken at a fixed scale and at three different orientations. Since the shapes have similar projection histograms, we segment the *Y*-axis histogram into six equal-length blocks and the *X*-axis projection histogram

Table 2. Classification accuracy of the direct representation.

Digit number	0	1	2	3	4	5	6	7	8	9	Average
Hit number	108	108	108	108	108	108	108	108	108	108	108
Test number	108	108	108	108	108	108	108	108	108	108	108
Accuracy (%)	100	100	100	100	100	100	100	100	100	100	100

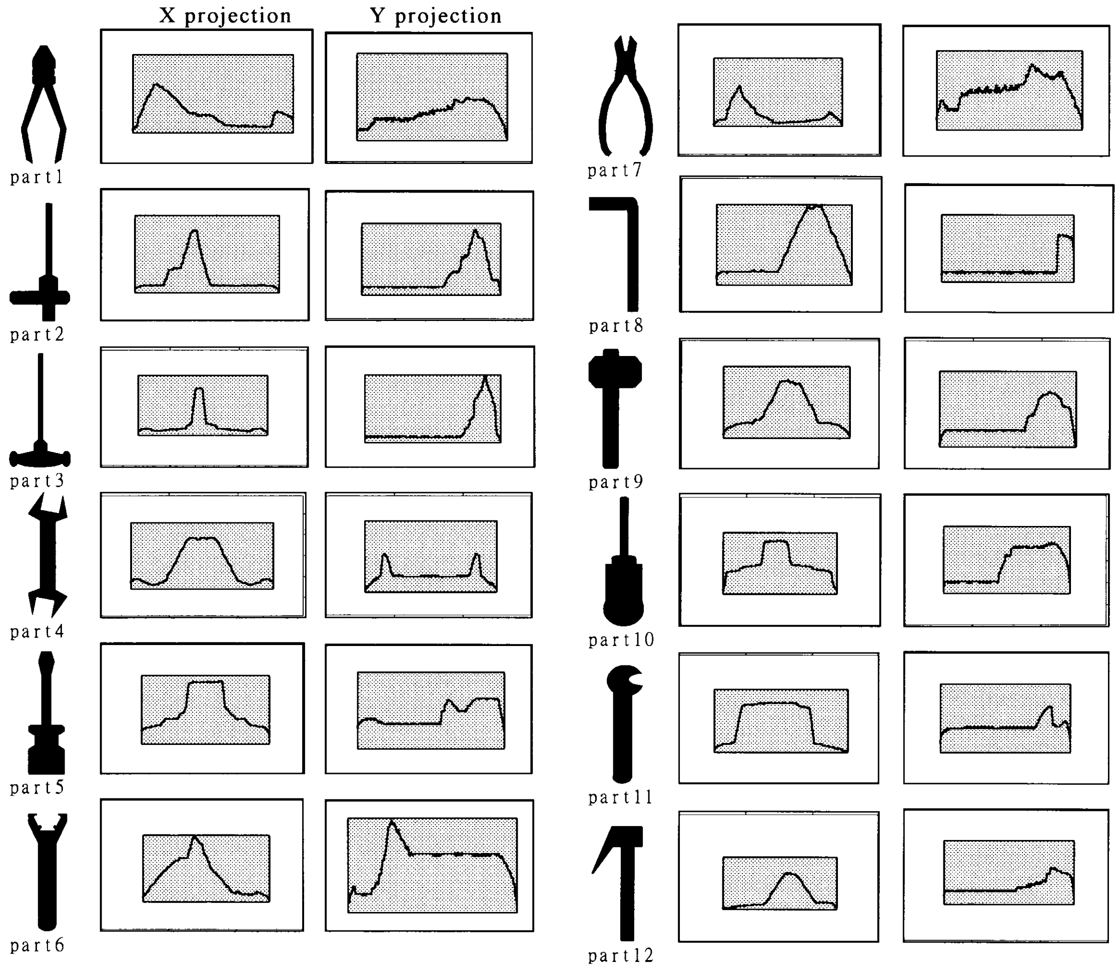

Fig. 8. Parts and their projection histograms.

Table 3. Classification accuracy of the general parts.

Part number	1	2	3	4	5	6	7	8	9	10	11	12	Average
Hit number	96	96	96	96	96	96	90	96	96	95	89	96	94.8
Test number	96	96	96	96	96	96	96	96	96	96	96	96	96
Accuracy (%)	100	100	100	100	100	100	93.8	100	100	99	92.7	100	98.8

into four blocks. Therefore, there are ten features extracted from the projection histogram, and ten nodes are used at the input layer of the neural net. Since there are four kinds of shape, four nodes are used at the output layer. Seven nodes are used in the hidden layer, and the learning rate and momentum rate are both set at 0.4 in which the neural net can come to convergence.

At the classification stage, two kinds of scale and 32 random orientations for a shape are tested. Therefore, there are 64 test images for a shape in this experiment. Classification results are shown in Table 4 which show that the proposed system has up to 100% accuracy. Therefore, we can improve the classification accuracy by increasing the number of segmented blocks. However, the block length should be large enough in order to extract the block area ratio. The system also takes only 6.5 s to analyse and classify a shape. In short, the feasibility test has demonstrated that the proposed system has very high classification accuracy and very good processing speed, and is feasible for use in practice.

4. Conclusions

In this paper a computer vision-based shape-classification system is presented. The proposed system contains an image binarisation, an image projection, an orientation normalisation, a feature extraction, and a back propagation neural network. The image binarisation procedure is used to threshold a grey-

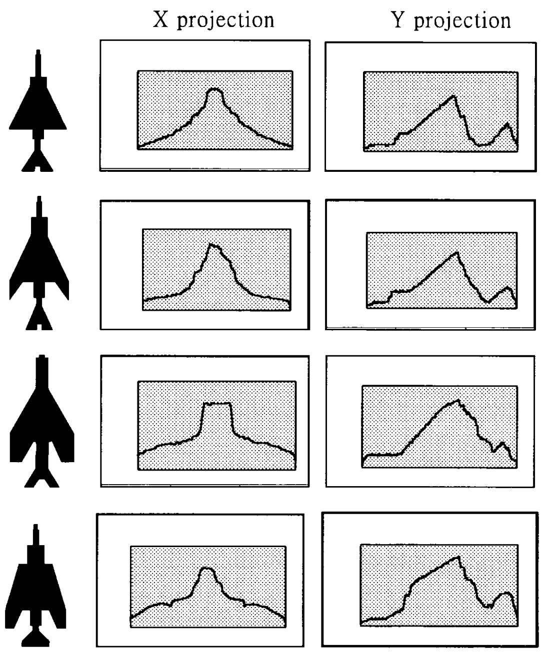

Fig. 9. Similar shapes and their projection histograms.

Table 4. Classification accuracy of the similar shapes.

Shape number	1	2	3	4	Average
Hit number	64	64	64	64	64
Test number	64	64	64	64	64
Accuracy (%)	100	100	100	100	100

level image into a binary image, and then the image projection procedure is used to acquire the *X*-axis and *Y*-axis projection histograms. An orientation normalisation procedure, especially suitable for the image projection is also developed. Therefore, the proposed system is able to classify a shape with scale, translation, and orientation variations.

A feature extraction procedure is also presented in this research. Traditionally, moments of the projection histogram were used to classify a shape. In this research, we segment the projection histogram into some equal-length blocks, and calculate the block area ratios which are the ratios of the block areas to the total area. These ratios are then fed to the back propagation neural net to classify a test shape.

A pre-test is conducted to verify the feasibility of the system and to compare the output layer representation schemes. The pre-test results show that the proposed system has very high classification accuracy, and the direct representation has better accuracy than the binary coding representation scheme.

Finally, a feasibility study is conducted to test the performance of the system for objects with variations in scale, translation, and orientation. Two cases are tested, one with general parts, and the other with similar shapes. These two cases demonstrate that the proposed classifier has accuracy up to 98.8% for the general parts and 100% for the similar shapes. In addition, the system is very fast. Therefore, the system is feasible for use in practice, considering both accuracy and speed. Although no part with an internal hole is tested in the proposed system, the authors argue that the image projection procedure can detect this internal hole feature, and therefore, the system is still able to classify the part. In addition, the centroids of parts 1 and 7 used in the first case are not inside the object, and the proposed system is able to classify them with 100% and 93.8% accuracy, respectively. Therefore, the proposed system can be used to classify objects with centroids outside the object. In short, this research has demonstrated that the new shape classifier proposed in this paper is feasible both in speed and accuracy.

References

1. R. Horaud, S. Olympieff and J. P. Charras, "Shape and position recognition of mechanical parts from their outlines" The First International Conference on Robot Vision and Sensory Controls, pp. 125–134, 1981.
2. L. Gupta and M. D. Srinath, "Contour sequence moments for the classification of closed planar shapes", Pattern Recognition, 20(3), pp. 267–272, 1987.
3. A. Amin and W. H. Wilson, "Hand-printed character recognition system using artificial neural networks", Proceedings of the Second International Conference on Document Analysis and Recognition, pp. 943–946, 1993.
4. L. P. Cordella, C. D. Stefano and M. Vento, "A neural network classifier for OCR using structural descriptions", Machine Vision and Applications, pp. 336–342, 1995.
5. H. Kim and K. Nam, "Object recognition of one-DOF tools by a back-propagation neural net", IEEE Transactions on Neural Networks, 6(2), pp. 484–487, 1995.
6. P. Breuer and M. Vajta, "Structural character recognition by forming projections", Problems of Control and Information Theory, 4, pp. 339–352, 1975.
7. Y. R. Wang, "Characterization of binary patterns and their projections", IEEE Transactions on Computers, pp. 1032–1035, 1975.
8. J. Mantas, "Methodologies in pattern recognition and image analysis – a brief survey," Pattern Recognition, 20, pp. 1–6, 1987.
9. D. E. Rumelhart, G. E. Hilton and R. T. Williams, "Learning internal representation by error propagation", in D. E. Rumelhart and J. L. McClleland (ed.), Parallel Distributed Processing, Vol. 1: Foundations, pp. 318–362, MIT Press, 1986.
10. Y. Lecun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Backpropagation applied to handwritten zip code recognition", Neural Computation, 1, pp. 541–551, 1989.
11. T. Yanagita, T. Tanaka and A. Toshima, "A neural network recognition system for machine printed characters on coils", Conference Record of the 1995 IEEE Industry Applications Conference, 2, pp. 1795–1799, 1995.
12. Y. K. Ryu and H. S. Cho, "Visual inspection scheme for use in optical solder joint inspection system", Proceedings of the 1996 IEEE International Conference on robotics and Automation, pp. 3259–3264, 1996.
13. K. Sasaki, D. Casasent and S. Natarajan, "Neural net selection of features for defect inspection", SPIE 1384, pp. 228–233, 1990.
14. P. K. Sahoo, S. Soltani and A. K. C. Wong, "A survey of thresholding techniques", Computer Vision, Graphics, and Image Processing, 41, pp. 233–260, 1988.
15. R. P. Lippmann, "An introduction to computing with neural nets", IEEE ASAP Magazine, 4, pp. 4–22, 1987.