ELSEVIER

Computers in Industry 27 (1995) 225–236

COMPUTERS IN INDUSTRY

Neural networks for precise measurement in computer vision systems

Chao-Ton Su [a], C. Alec Chang [b,*], Fang-Chih Tien [b]

[a] *Department of Industrial Engineering and Management, National Chiao Tung University, Hsinchu, Taiwan 30050, ROC*
[b] *Department of Industrial Engineering, 113 EBW, University of Missouri–Columbia, Columbia, MO 65211, USA*

Received 21 April 1994; revised 24 January 1995

Abstract

Although computer vision systems have been successfully applied to some inspection tasks, they were generally not considered as precise measurement tools due to dimensional distortion and errors. This paper presents procedures to correct these errors for precise measurement. The first step is to formulate calibration models for image coordinate systems using neural networks. Then neural networks to model dimensional errors from the initial measurement are structured in a learning stage using standard parts. Finally these models are used to correct measurement errors in measurement tasks. These proposed procedures are implemented as an example.

Keywords: Precise measurement; Dimensional inspection; Coordinate calibration; Error correction; Measurement correction; Neural networks; Back propagation; Computer vision

1. Introduction

Since the computer vision system was first developed, it has been successfully applied to automated inspection tasks in many industrial processes such as printed circuit board, chip alignment and bonding and profile matching for machined parts [1]. These inspection tasks mainly utilize image subtraction, feature matching, diffraction, and spatial filtering to find differences between a part and its standard datum. In contrast, the application of computer vision systems for measurement tasks has been less utilized. Due to image distortion and other system deformities, computer vision systems are seldom considered as precise measurement devices.

When an industrial part is brought into the computer vision system, this part is scanned by a camera. The image of this part will be digitized both spatially and in amplitude. Digitization of the spatial coordinates is called image sampling and amplitude digitization is referred to as gray-level quantization. After an image has been digitized, the thresholding technique and boundary extraction method can be applied to detect the edge points representing the profile of the part. Then a segmentation procedure can be used to decompose the part profile into linear edges or circular curves at certain joints. The K-curvature thresholding method is an effective method for this boundary segmentation task [2]. By calculating the change of the K-curvature for each edge

* Corresponding author.

SSDI 0166-3615(95)00024-0

point of the profile, break points of the profile can be detected. Finally, dimensional measurements can be made for these segmented edge lines.

Generally the measurement errors of a computer vision system come from two major sources. They are system errors from hardware and measurement errors from software. System errors include digitization error, geometrical dissimilarity due to image distortion, curvature of fields of view and others. These system errors will seriously deform the extracted boundary of a scanned part. Measurement errors are usually generated from the adopted algorithm that fits represented lines or curves for a segmented set of edge points. These errors could cause a large deviance from true dimensions. Wagner [3] suggests an uncertainty of at least ± 1 pixel resolution value for each edge transition in a measurement. Ho [4] finds that the digitizing error of various geometric features must be expressed as a perimeter of the object. Chang, Chen and Lin [2] develop a two-step method to reduce the discarded edge points between two boundary lines for better precision in measurement. They also explore the representation errors for the measurements of a straight line edge, a circular arc and angles in their research [5]. However, none of these papers present a complete operational procedure to correct dimensional measurement errors in a computer vision system.

Without an effective error correction procedure, computer vision systems can never be used as precise measurement and inspection tools. The development of a systematic procedure to correct these errors in computer vision inspection systems is an urgent task for automated manufacturing systems. Therefore, the purpose of this paper is to present an effective error correction procedure that will reduce the errors of dimensional measurement in computer vision systems.

2. Error correction for dimensional measurements

Laboratory experiments consistently show that measurement errors estimated from the digitization approximation of a part profile into discrete image pixels are mostly underestimated. There are other errors from minute shadow of part edges, lighting variation and other unknown causes that cannot be easily approached analytically. In order to correct most errors in the use of vision systems, the empirical approach is proposed to formulate measurement correction procedures. Geometric factors such as part size, geometric shapes, part location and orientation in the field of view can be associated with sources of measurement errors. Since a generic error correction model for all parts on all systems cannot be built readily, a system-dependent and part-dependent procedure is proposed. In this case, the part position and part orientation will be the only influencing factors in error correction procedures. Errors associated with part sizes and geometric shapes are implicit in mod-

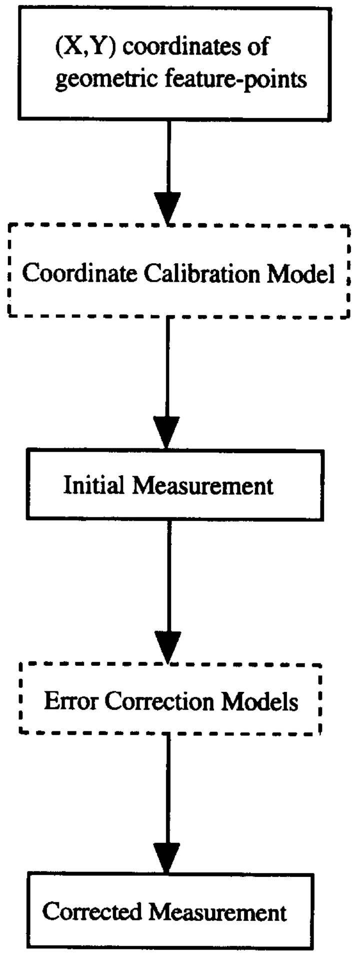


Fig. 1. Error correction procedure for precise measurement.

els for each part. The position errors due to the surface distortion of the field of view can be corrected by a proper calibration of the coordinate system of a vision equipment set-up. Therefore, part orientations will be the independent factor for measurement correction models. Fig. 1 shows this proposed framework of error correction procedures. The stage of formulating error correction models is the "learning process", while the stage in implementing the developed models in a computer vision system is the "operational process".

For measurement correction, the following relationship can be established in a learning process:

$$\varphi_\pi = f(\theta_\pi), \tag{1}$$

where φ_π is the ratio of the measured dimension to its true dimension of a geometric feature, and θ_π is the orientation of the part scanned. Then, the corrected estimate of true dimension, $\hat{\pi}_t$, can be obtained in the operational process by

$$\hat{\pi}_t = \pi_i / \varphi_\pi, \tag{2}$$

where π_i is the initial dimension of a geometric feature measured.

Neural networks can be a general mapping procedure for the input–out patterns as shown in (see equation 1). The back propagation neural network does not require any prior information of functional relationships to map the input patterns to the output patterns [6–8]. When this method is applied to develop error correction models, it will generate the required correction ratio, φ_π, in Eq. (1) from a part orientation.

3. Error correction using neural network models

Neural network is a powerful technology that has been successfully applied to many tasks of manufacturing systems. For example, Sasaki, Casasent and Natarajan [9] apply a neural network fed with optically generated features to IC inspection. Javed and Sanders [10] use a multi-layer neural network to structure a quality control monitor for zinc coated steel. Using a back propagation neural network, Neubauer [11] develops an optical inspection system to detect and classify the defects on treated metal surfaces. Kroh, Durrani and Chapman [12] develop a new neural network architecture for feature recognition in binary images. Masory [13] proposes a neural network model to find the relationship between multi-sensor readings and actual tool wears. Ipakchi et al. [14] develop a neural network-based analytical technique for instrument calibration. Their technique can predict the reading of a target instrument using data from other dissimilar instruments. Hou, Lin and Scott [15] propose an automated inspection system using a Hough Transform and a back propagation network for surface mount devices. Ker, Lynch and Kalale [16] develop a neural network approach to check radii of circular parts and differentiate between good and defective products. Hwarng and Hubele [17] present a pattern recognition method for quality control charts based on the back propagation algorithm. Their algorithm can identify six types of unnatural patterns on $\bar{X}$ control charts, namely, trends, cycles, stratification, systematic, mixtures and sudden shift.

When a set of input/output patterns is given to a feed forward neural network, the weights and the bias of structural nodes keep adjusting to decrease the difference between the network's output and the target patterns. The back propagation learning algorithm involves a forward pass and a backward pass. Both passes are done for each pattern presentation during training. After the network reaches a satisfactory level of performance, the relationships between input and output patterns are defined and they can be used to estimate the output of new incoming patterns. In other words, the information is fed from the input nodes through the output nodes in the forward pass, and the calculations and updating are done layer by layer. Mathematically the net input to node i for pattern p is expressed as

$$\text{net}_{pi} = \sum_{j \in \text{previous layer}} w_{ij} a_{pj} + b_i, \tag{3}$$

where a_{pj} is the activation value of unit j for the pattern p, w_{ij} is the weight from unit j (sending unit) to unit i (receiving unit), and b_i is a bias associated with unit i.

After the incoming sum is computed, a sigmoid function f is usually used to compute the output of unit i for the pattern p:

$$a_{pi} = f(\text{net}_{pi}) = \frac{1}{1 + \exp(-\text{net}_{pi})}. \tag{4}$$

The difference between the network's output and the target patterns is calculated and errors for hidden units are computed. An error function is then formed based on the total sum of square differences of the target value and actual output value of the ith output unit for the pth pattern from training data. The back propagation algorithm uses gradient descent on the error function with respect to weights. A recursive procedure is applied to calculate incremental values that are used to compute the weight changes in Eq. (3) for the network. In this recursive procedure, new incremental weights are obtained from the results of each run. When the total sum of squared error (tss) of all the output units is less than a selected criterion, the recursive procedure can be terminated. The structured neural network derives an error correction ratio (φ) that will be multiplied to the initial measurement to produce a better measurement.

4. Calibrating the image coordinate system

When coordinates of edge points of an object are used to calculate its dimension, the pixel sizes of an image should be known precisely. However, in most systems, not only is the unit length of the pixel in the X direction not the same as that in the Y direction, but also the pixel size changes at different locations. For example, the ratio between the unit length in the X direction and that in the Y direction is 1.73 for a pixel around the center of an ITEX 100 Image Processing System. It changes to a different ratio when a pixel is away from the center. To improve measurement accuracy, a proper calibration of the image coordinates system is necessary prior to employing the computer vision inspection system. Veeder [18] suggests that by measuring a certified master part and comparing the measured results with the true values according to the standard part, enough information can be obtained to correct the error from the video measuring system. Rodriguez, Mandeville and Wu [19] adopt an array of squares to calibrate the optical distortion of the camera. Improving these methods, we can develop a modeling approach to calibrate the different variations for different coordinate locations.

The ratio between the length in the X and Y directions for a single pixel at different positions is associated with different amounts of errors. Different coordinate calibration ratios must apply to image points at different positions. Fig. 2 shows an example of calibration points that can be used to build the relationships between the true object coordinates and their image coordinates at different positions. After scanning these circles, data sets of elliptical images are obtained because of the different unit lengths in the X direction and in the Y direction. The geometric centers of the elliptical images are used for mapping the coordinates of the geometric center of the physical circles. Once the relationship between the image position and its true physical location is defined, the image coordinates of scanned image points can be calibrated.

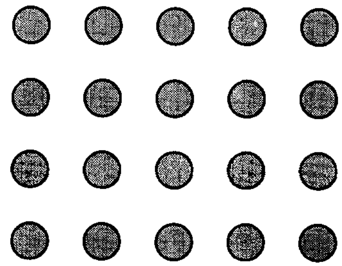


Fig. 2. Data points to structure neural networks for coordinate calibration.

Assuming that N data sets are used on the entire image plane, the following procedure can be used to estimate the coordinates of the geometric center for each image I_i. Let (p_{ij}, q_{ij}) be the coordinates of the jth edge point for the ith image and n_i is the number of edge points for the ith image. The geometric center for the ith image can be expressed as

$$(x_i, y_i) = \left(\frac{\sum p_{ij}}{n_i}, \frac{\sum q_{ij}}{n_i} \right), \tag{5}$$

where $i = 1,2,\ldots,N$ and $j = 1,2,\ldots,n_i$. N is the number of circle images to be captured. Because the distortion is usually symmetrical around the center of the image plane and its distortion function is usually quadratic, at least three calibration points along one side of the image plane are required. At least nine data sets are required for the calibration of the whole image plane.

Based on the discussion above, the proposed procedure for the calibration of the image coordinate system can be summarized as follows:

Procedure 1. Calibration for the image coordinates
Position a precise X–Y table within the field of view;
Attach a thin gauge block on the X–Y table and define the initial position of the ring contour as the origin of the field of view;
Scan the ring gauge block to obtain an image, I_1;
Decide the number of calibration points required, N ($N \geq 9$);
FOR $i = 2,3,\ldots,N$ **DO**
  Translate the X–Y table a predetermined step distance (as Fig. 2);
  Record the true readings (x_i, y_i) from the X–Y table;
  Scan the ring gauge block to obtain an image, I_i;
END FOR;
FOR each image I_i **DO**
  Get the coordinates of the edge points;
  Calculate the coordinates of the geometric center (x_i, y_i) by using Eq. (5);
END FOR;
Model the relationships between (x_i^*, y_i^*) and (x_i, y_i) by using neural networks.

5. Correcting dimensional measurement

In order to apply the back propagation procedure for a feed forward neural network, a set of input/output patterns should be obtained. The initial dimensional measurement and orientations of the part should be obtained for the learning process. To collect an effective set of input/output data, one can position the rotary table within the field of view and place the part to be measured close to the center of the rotary table. By rotating the rotary table θ degrees in a counterclockwise (or clockwise) direction, where θ is a predetermined increment for the part orientation, one can measure geometric features of the part at different orientations. Thus, a dimension ratio can be obtained by

$$\varphi_{\pi_i} = \pi_\mathrm{i}/\pi_\mathrm{t}, \tag{6}$$

where π_i is the initial measurement of a geometric feature after conducting coordinate calibration, π_t is the true dimension and φ_{π_i} is the dimension ratio of the ith observation with orientation θ_i. This procedure is presented to obtain an effective input–output set for network training:

Procedure 2. Data collection for network training
Position a precise rotary table within the field of view;
Place the part to be measured close to the center of the rotary table;
Scan the profile of the part at present location to obtain an image, I_1;
Let $\theta_1 = 0$;
FOR $i = 2,3,\ldots,k$, $k = \mathrm{int}[360/\theta]$, **DO**
  Rotate the rotary table θ degrees in a counterclockwise (or clockwise) direction, where θ is a predetermined increment for the part orientation;
  $\theta_i = \theta_{i-1} + \theta$;
  Scan the profile of the part to obtain an image, I_i;
END FOR;
FOR each image I_i **DO**
  Get the coordinates of the edge points;
  Calibrate these coordinates;
  Compute the radius size r_i, length L_i and angle ϑ_i by employing proper measurement methods;
  Set $\varphi_{r_i} = r_i/\boldsymbol{r}$, $\varphi_{L_i} = L_i/\boldsymbol{L}$ and $\varphi_{\vartheta_i} = \vartheta_i/\boldsymbol{\vartheta}$, where $\boldsymbol{r}$, $\boldsymbol{L}$ and $\boldsymbol{\vartheta}$ are the true values for the radius size, length and angle of the part to be measured;
END FOR.

When the architecture of a neural network is selected to build error correction models, the orientations of the geometric features to be measured are input to the input layer. To reduce the number of input variables, an orientation angle can be used for adjacent features as long as the same practice is used in its operational stage. If the value of the input pattern in the network is greater than 3, the value of the sigmoid function will be close to 1; and if the value of the input pattern is less than -3, the value of the sigmoid function will be close to 0. In case there are too many values of the input pattern which are greater than 3 or less than -3, those input patterns will block the back propagation procedure [20,21]. The output layer will have nodes corresponding to the number of correction ratios for geometric features. Since the observations of the target patterns must be within [0, 1] and the observations of the input patterns should be between -3 and 3, the data sets can be scaled and shifted. For example, the following scaling and translation is used on the data set in this paper:

Input pattern:

$$(\theta_r', \theta_L', \theta_\vartheta')^t = \left(\frac{\theta_r}{100}, \frac{\theta_L}{100}, \frac{\theta_\vartheta}{100} \right)^t, \tag{7}$$

Target pattern:

$$(\varphi_r', \varphi_L', \varphi_\vartheta')^t = \left(\varphi_r - \tfrac{1}{2}, \varphi_L - \tfrac{1}{2}, \varphi_\vartheta - \tfrac{1}{2} \right)^t, \tag{8}$$

where θ_r, θ_L and θ_ϑ are the orientations of the circular arc, straight line edge and angle, respectively, and φ_r, φ_L and φ_ϑ are the correction ratios for the measurement of radius, length and angles, respectively.

Following the proposed data collection procedure, a set of training patterns $(\boldsymbol{x}_1, \boldsymbol{t}_1)$, $(\boldsymbol{x}_2, \boldsymbol{t}_2)$, ..., $(\boldsymbol{x}_s, \boldsymbol{t}_s)$ can be collected, where

$$\boldsymbol{x}_i = \left(\theta_{r_i}', \theta_{L_i}', \theta_{\vartheta_i}' \right)^t, \tag{9}$$

$$\boldsymbol{t}_i = \left(\varphi_{r_i}', \varphi_{L_i}', \varphi_{\vartheta_i}' \right)^t. \tag{10}$$

When the learning rate and momentum coefficient are selected properly, the back propagation network can be used effectively in estimating the mapping function between φ' and θ' by choosing a reasonable number of layers and nodes in the hidden layers. The value of the total sum of squared errors (tss) can be used as an index for the performance of the trained network. If tss reaches a stable condition (or is less than some criterion), the training process is terminated.

This proposed error correction procedure using neural networks can be summarized as follows:

Procedure 3. Learning stage for network building

1. Obtain a set of observed data, $(\varphi_{r_i}, \theta_{r_i})$, $(\varphi_{L_i}, \theta_{L_i})$ and $(\varphi_{\vartheta_i}, \theta_{\vartheta_i})$, $i = 1,2,\ldots,k$, by using the proposed data collection procedure.
2. Transform the observed data sets into a set of training patterns, $(\boldsymbol{x}_i, \boldsymbol{t}_i)$, $i = 1,2,\ldots,k$.
3. Choose a set of network's architecture. Determine reasonable learning rate and the momentum coefficient.
4. Train each network until the difference of the tss of two successive iterations is less than a predetermined tolerance.
5. Choose a trained network with the smallest tss.

Procedure 4. Operational stage for measurement correction

1. Measure the radius r, length L and angle ϑ of the part profile.
2. Estimate the orientations of the circular arc, straight line edge and angle, i.e., decide $(\theta_r, \theta_L, \theta_\vartheta)^t$.
3. Set $(\theta_r', \theta_L', \theta_\vartheta')^t = (\theta_r/100, \theta_L/100, \theta_\vartheta/100)^t$.
4. Present $(\theta_r', \theta_L', \theta_\vartheta')^t$ to the trained network from the above learning procedures and compute the output $(\varphi_r', \varphi_L', \varphi_\vartheta')^t$.
5. Correct radius size by r/φ_r, length by L/φ_L and angle by $\vartheta/\varphi_\vartheta$, where $\varphi_p = \varphi_p' + \rho$, $p \in \{r, L, \vartheta\}$ and ρ is a translation constant as in the learning stage.

6. Implementation

This proposed measurement error correction procedure is implemented on an ITEX 100 Image Processing System using a CCD camera with 512 by 512 resolution. The measurements are carried out in the laboratory at 20°C room temperature. A precise mechanical part with known dimensions is used to verify the developed procedure. The four geometric features, the radius, two angles, and length of the segment are illustrated in Fig. 3.

A coordinate calibration step is conducted first. The camera distance is set at 41.1 cm for a proper

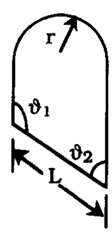


Fig. 3. A test part (aluminum alloy, thickness: 0.0260" ±0.0005").

field of view of the experiment. By following the proposed Procedure 1, a set of data points are derived by moving a circular gauge block according to a predetermined format in the field of view. In the present example, the number of calibration points is arbitrarily set as shown in Fig. 2. The boundary of the circular gauge block is scanned and its geometrical center is calculated as the input (X_i, Y_i) of the network. The corresponding target values (X_i', Y_i') are the readings from the X–Y table. After the coordinates of the geometric centers are determined, the relationships between the image coordinates and their true coordinates can be established by using the neural network method following Procedure 1. This coordinate correction procedure is applied to all scanned boundary points of the part profile.

In order to locate the pattern values into the feasible range to train neural networks, the input–output patterns, (X_i, Y_i), are transformed as $(X_i/100, Y_i/100)$. For network training, the Parallel Distributed Processing Software (PDP) is used [22,23]. After several pilot runs, the learning rate (η) and momentum (κ) are decided as 0.03 and 0.9. The networks with one and two hidden layers, 2–∗–2 (such as 2–4–2) and 2–∗–∗–2 (such as 2–4–4–2), are trained. The 2–6–2 network has the best performance in mapping. Its weights and biases are listed in Table 1.

When the coordinates are calibrated, error correction models can be structured following the proposed procedures. Many sharewares from Internet such as

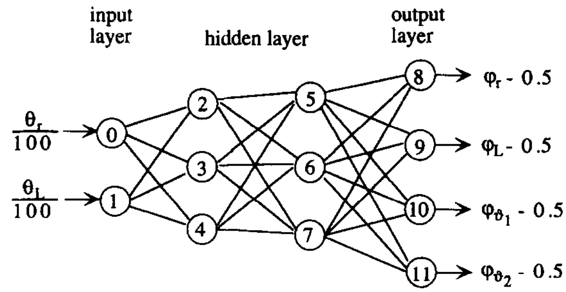


Fig. 4. Architecture of the 2–3–3–4 network.

the Laplacian operator or other gradient based methods can be used to find edge lines. The Sobel's edge detection method is used in this example for its speed. By applying the proposed Procedure 2, forty observed data sets are collected from two repetitions with 18 degrees of angle increment. They are transformed into a set of training patterns $(\boldsymbol{x}_i, \boldsymbol{t}_i)$ for the proposed Procedure 3, where

$$\boldsymbol{x}_i = \left(\frac{\theta_{r_i}}{100}, \frac{\theta_{L_i}}{100}\right)^t,$$

$$\boldsymbol{t}_i = \left(\varphi_{r_i} - \tfrac{1}{2}, \varphi_{L_i} - \tfrac{1}{2}, \varphi_{\vartheta 1_i} - \tfrac{1}{2}, \varphi_{\vartheta 2_i} - \tfrac{1}{2}\right)^t,$$

$$i = 1,2,\ldots,40.$$

There is no good algorithm indicating the starting values of leaning rate, moment, tss and an initial set of weights. The designation of these values of a feed-forward neural network is mainly a trial and error process. However, experiences of many re-

Table 1
Weights and biases of the 2–6–2 network for coordinate calibration

j–i	w_{ij}	j–i	w_{ij}	node i	b_i
0–2	1.019298	2–8	−4.38538	2	−3.854295
1–2	0.009901	3–8	−1.913018	3	0.498228
0–3	0.088061	4–8	2.118208	4	1.365475
1–3	−0.292487	5–8	0.942664	5	2.115479
0–4	−0.387209	6–8	2.535533	6	0.698492
1–4	0.03951	7–8	−0.716094	7	0.132813
0–5	−0.015648	2–9	0.388515	8	−0.011987
1–5	−0.520544	3–9	−3.604298	9	1.075127
0–6	−1.921038	4–9	0.276105		
1–6	−0.006703	5–9	−3.739314		
0–7	0.017369	6–9	−0.152341		
1–7	0.505321	7–9	4.500251		

searchers provide us with some guidelines that can be summarized as follows [23,24]:

(1) A heuristic rule recommended by Eberhart and Dobbins states that a reasonable number of neurons in each layer is around the square root of he number of input plus output neurons. As a result, the starting number of neurons in the hidden layer in this example is 2 and 2–2.

(2) A random number generator supplies the initial weight set. The range of randomization should be within ± 0.3. A range too wide may lead to network oscillation.

(3) McClelland and Rumelhart frequently use values of 0.5 and 0.9 for learning rate and momentum, respectively. In this project, these values are used as a place to start.

(4) The tss is set to 0.01 initially. A maximum iteration is set at 100 000. If the network stops before reaching the maximum number of iterations allowed, reduce tss further. Otherwise, plot a tss graph. If the graph shows the trend that tss will further decrease, a larger learning rate can be used and the network trained again until it shows little sign of improvement.

Several different networks have been tried. A 2–3–3–4 network, as shown in Fig. 4, demonstrates the best performance. Accordingly, the model of the 2–3–3–4 network is chosen to estimate the required correction ratios when a new input pattern $(\theta_r/100, \theta_L/100)^t$ of an incoming part is given.

The correction models obtained from Procedure 3 for the ratios of radius, length, angle 1 and angle 2 of the test part are as follows:

$$\varphi_r = \frac{1}{2} + \frac{1}{1+\exp(-\text{net}_{p8})},$$

$$\varphi_L = \frac{1}{2} + \frac{1}{1+\exp(-\text{net}_{p9})},$$

$$\varphi_{\vartheta 1} = \frac{1}{2} + \frac{1}{1+\exp(-\text{net}_{p10})},$$

$$\varphi_{\vartheta 2} = \frac{1}{2} + \frac{1}{1+\exp(-\text{net}_{p10})}, \tag{11}$$

where $\text{net}_{pi} = \Sigma_j w_{ij} a_{pj} + b_i$, j belongs to previous layer based on network 2–3–3–4, $a_{p0} = \theta_r/100$ and $a_{p1} = \theta_L/100$ for the input layer. θ_r is the orientation of the circular arc and θ_L is the orientation of the straight line edge between angle 1 and angle 2. The weights and biases of this network model, w_{ij} and b_i, are listed in Table 2.

This proposed error correction method is further conducted using 20 testing data sets for the structured networks from Procedure 3. A summary of corrected measurements using Procedure 4 is shown in Table 3. The absolute errors after a measurement correction of the radius, length, angle 1 and angle 2 are much less than the errors of raw measurement. For example, the average absolute error of radius is

Table 2
Weights and biases of the 2–3–3–4 network for measurement calibration

j–i	w_{ij}	j–i	w_{ij}	node i	b_i
0–2	0.684261	4–7	1.764118	2	−4.966747
1–2	1.637231	5–8	2.329209	3	−8.085383
0–3	−5.062571	6–8	3.590328	4	−1.009515
1–3	4.898856	7–8	1.861937	5	−3.031289
0–4	2.812193	5–9	1.079458	6	1.789951
1–4	−2.510104	6–9	2.39857	7	−3.141014
2–5	6.998819	7–9	−4.743825	8	−3.110659
3–5	−2.892356	5–10	−0.685486	9	−0.750603
4–5	1.020161	6–10	−0.819291	10	0.709865
2–6	−3.22185	7–10	0.43645	11	−0.498287
3–6	−1.868098	5–11	0.101489		
4–6	−1.104647	6–11	0.948275		
2–7	−0.11002	7–11	5.410086		
3–7	6.60991				

Table 3
A sumary of measurement coorection

Feature		Raw measurement from scanned images		Corrected measurement without coordinate calibration		Initial measurement after coordinate calibration		Corrected measurement using coordinate calibration and error correction models	
		Measured dimension	Absolute error	Measured dimension	Absolute error	Measured dimension	Absolute error	Measured dimension	Absolute error
Radius	Mean	0.402″	0.027	0.384	0.009	0.368	0.007	0.376	0.001
	Std	0.298		0.226		0.010		0.002	
Length	Mean	0.9088″	0.0078	0.8945	0.0065	0.8880	0.0130	0.9032	0.0022
	Std	0.3211		0.1694		0.0200		0.004	
Angle 1	Mean	122.52°	1.07	123.03	0.56	123.04	0.55	123.59	0.00
	Std	3.53		1.27		2.07		0.24	
Angle 2	Mean	57.35°	1.13	56.50	0.28	56.56	0.34	56.25	0.03
	Std	3.55		1.62		1.44		0.16	

Note: 1. Mean and Std are the sample mean and the standard deviation of 20 observations.
2. Absolute error = |measured dimension − true dimension|.

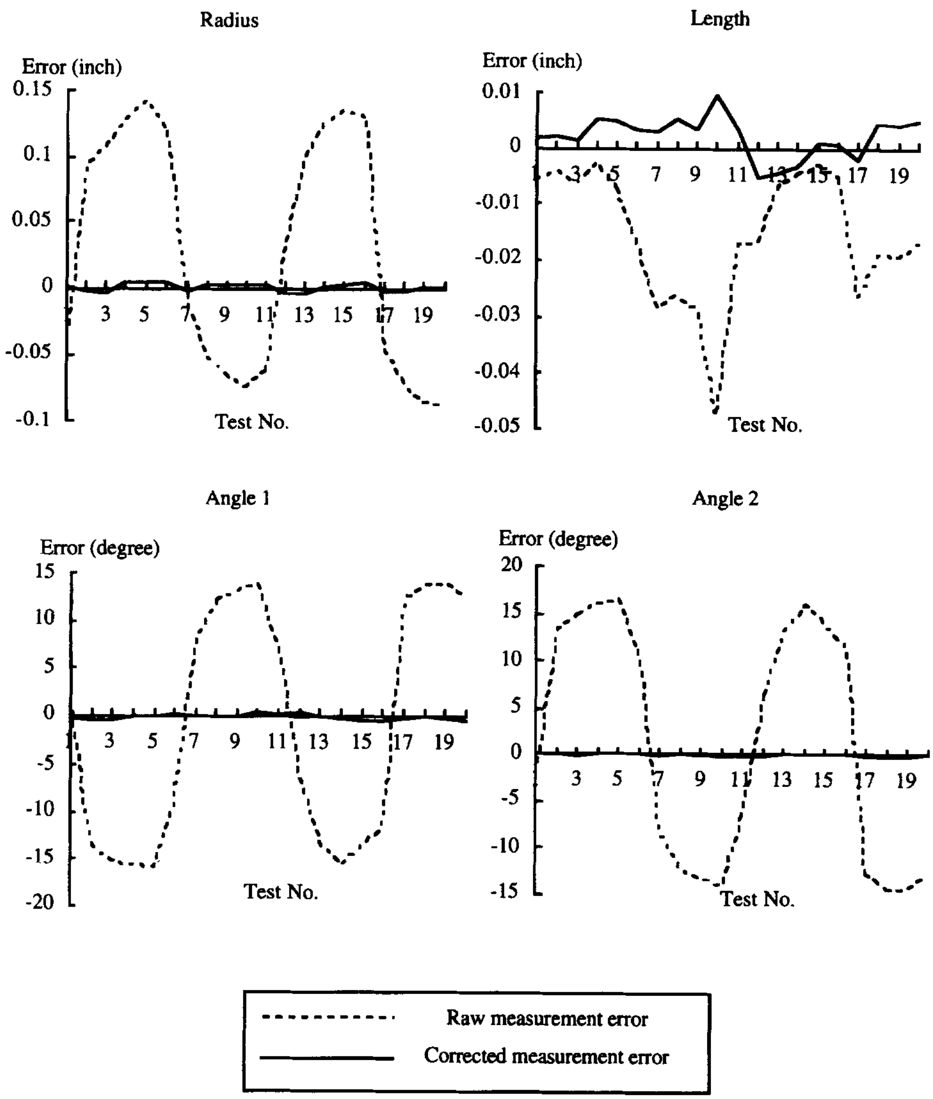


Fig. 5. A comparison of raw measurement and corrected measurement.

decreased from 0.027 to 0.001. Its standard deviations of corrected measurements are reduced from 0.298 to 0.002. A comparison of the errors for each test data before and after error correction is shown in Fig. 5.

7. Discussion and Recommendation

Due to the ability to map any set of input–output patterns, a practitioner may tend to structure one network to model the error correction ratio from raw data without a coordinate calibration step. In this approach, the inputs of the net work are four initial measurements, the position of the arc center, and the two orientations for the arc and the bottom edge obtained from the scanned image. For the training of such a network, the position (X_i, Y_i), and the orientations (θ_1, θ_2) can be transformed into $(X_i/100, Y_i/100)$ and $(\theta_1/100, \theta_2/100)$ in order to locate the input values within $[-3, 3]$. The target patterns of correction ratios $(\varphi_r, \varphi_L, \varphi_{\vartheta 1}, \varphi_{\vartheta 2})$ can be adjusted into the interval [0, 1] by subtracting 0.5. The 8–12–4 network is the most efficient network, since it has a faster convergence and the lower tss among different networks. However, even though the measurement errors of Angle 1 and Angle2 are reduced to 0.56 and 0.28 respectively, these errors and their standard deviations are still too large to satisfy common precision requirement in industry. Therefore, the one-step network structure for measurement correction is not recommended unless we can use a very large amount of learning data. If resources for learning procedures are limited, a calibration of the coordinate system should be applied before training neural networks for measurement correction. Moreover, this coordinate calibration model can be used for different parts using the same field of view. Finally, the setup condition of a system should be recorded and maintained properly for each use of measurement correction models.

Although the use of more data sets for network training may generate better results, it will also take longer time in learning. It takes about three hours in a personal computer with a 486DX2 processor computer to train the coordinate calibration model and additional four hours for the measurement correction model. Fortunately, this training is required only once during a system setup. The use of learned model in implementation stage takes only a fractional second.

There are also statistical methods to derive mapping equations to calibrate coordinate systems. For example, the regression method is also an effective procedure to formulate calibration models. In our experiments, results of corrected coordinates using regression models show compatible precision as using neural network models. If the network training of coordinate correction models is tedious or cannot converge properly, this is an alternative to be considered. However, the search of good function format for the regression method can be a very difficult task.

With current technology advancement, computer vision systems can become a very versatile non-contact inspection system for precise measurement. However, a correction of raw measurements is essential in using computer vision systems for measurement purposes. In this paper, we propose a neural network-based approach to correct the system errors in hardware and the measurement errors in measurement algorithms. The task of precise measurement can be accomplished using fine cameras and this proposed measurement correction procedure. With further research, effective neural network models to calibrate 3D vision systems can be expected in the near future.

References

[1] R.T. Chin and C.A. Harlow, "Automated visual inspection: A survey", *IEEE Trans. Pattern Anal. Mach. Intell.* PAMI-4(6)1 (1982) 557–573.

[2] C.A. Chang, L.-H. Chen and H.-D. Lin, "Profile segmentation and merging for automated measurement by computer vision, *Proc. First Industrial Engineering Research Conf.*, 1992, pp. 383–387.

[3] G.G. Wagner, "Vision methodology as it applies to industrial inspection applications", in: *Gauging: Practical Design and Application*, Society of Manufacturing Engineers, 1983, pp. 410–418.

[4] C.S. Ho, "Precision of digital vision systems", *IEEE Trans. Pattern Anal. Mach. Intell.* PAMI-5(6) (1983) 593–601.

[5] C.A. Chang, L.H. Chen, L.G. David and C.T. Su, "Error correction models for measurement and inspection in computer vision system", *Proc. Second Industrial Engineering Research Conf.*, 1993, pp. 629–633.

[6] B. Joseph, F.H. Wang and S.S. Shieh, "Exploratory data analysis: A comparison of statistical methods with artificial neural networks", *Comput. Chem. Eng.* 16(4) (1992) 413–423.

[7] T.R. Holcomb and M. Morari, "PLS/neural networks", *Comput. Chem. Eng.* 16(4) (1992) 393–411.

[8] L. Marquez, T. Hill, R. Worthley and W. Remus, "Neural network models as an alternative to regression", *Proc. 24th Annu. Hawaii Int. Conf. on System Sciences*, Vol. 4, 1991, pp. 129–135.

[9] K. Sasaki, D. Casasent and S. Natarajan, "Neural net selection of features for defect inspection", *Proc. SPIE* 1384 (1991) 228–233.

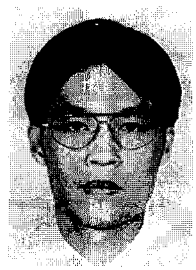
[10] M.A. Javed and S.A.C. Sanders, "Neural networks based learning and adaptive control for manufacturing systems", *Proc. IEEE/RSJ International Workshop on Intelligent Robots and Systems – IROS '91*, 1992, pp. 242–246.

[11] C. Neubauer, "Fast detection and classification of defects on treated metal surfaces using a back propagation neural network", *1991 IEEE Int. Joint Conf. on Neural Networks*, 1991, pp. 1148–1153.

[12] J.R. Kroh, T.S. Durrani and R. Chapman, "Neural network architecture foe circular features extraction in binary patterns", *Electron. Lett.* 27(20) (1991) 1879–1880.

[13] O. Masory, "Monitoring machining processes using multisensor readings fused by artificial neural network, *J. Mater. Process. Technol.* 28(1–2) (1991) 231–240.

[14] A. Ipakchi, M. Khadem, C.M. Peng and R.W. Colley, "Neural network applications to measurement calibration verification in power plants, in: *Instrumentation in the Power Industry*, Proceedings 34, Instrument Society of America, Research Triangle Park, NC, 1991, pp. 501–511.
[15] T.H. Hou, L. Lin and P.D. Scott, "A neural network based automated inspection system with an application to surface mount devices", *Int. J. Prod. Res.* 31(5) (1993) 1171–1187.
[16] J.I. Ker, M. Lynch and A. Kalale, "A neural network approach to machine vision based radius inspection", *Intelligent Engineering Systems Through Artificial Neural Networks* 2 (1992) 541–546.
[17] H.B. Hwarng and N.F. Hubele, "Back-propagation pattern recognition for $\bar{X}$ control charts: Methodology and performance, *Comput. Ind. Eng.* 24(2) (1993) 219–235.
[18] V. South III, "Automatic precision video measurement for quality inspection", in: *Gauging with Vision System*, Society of Manufacturing Engineers, 1987, pp. 70–78.
[19] A.A. Rodriguez, J.R. Mandeville and F.Y. Wu, "System calibration and part alignment for inspection of 2D electronic circuit patterns, *SPIE* 132 (Optical Testing and Metrology III: Recent Advances in Industrial Optical Inspection) (1990) 25–35 .
[20] R.P. Lippmann, "An introduction to computing with neural nets, *IEEE Acoustics Speech ignal Process. Mag.* (April 1987) 4–22.
[21] K. Funahashi, "On the approximate realization of continuous mappings by neural networks", *Neural Networks* 2 (1989) 183–192.
[22] D.E. Rumelhart and J.L. McClelland, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, Vol. I, MIT Press, Cambridge, MA, 1989.
[23] J.L. McClelland and D.E. Rumelhart, *Explorations in Parallel Distributed Processing: A Handbook of Models, Programs, and Exercises*, MIT Press, Cambridge, MA, 1991.
[24] R.C. Eberhat and R.W. Dobbins. *Neural Network PC Tools*, Academic Press, San Diego, CA, 1990.



Chao-Ton Su is an associate Professor and Director of Graduate Studies in the Department of Industrial Engineering and Management at National Chiao Tung University, Taiwan, ROC. He received his B.S. and M.S. from Chung-yuan Christian University, Taiwan, and Ph.D. from University of Missouri–Columbia, USA, all in industrial engineering. His current research activities include automated inspection and neural networks in a variety of industrial applications.

C. Alec Chang is an associate professor in the Department of Industrial Engineering at University of Missouri–Columbia. He has been an adjunct professor at Polytechnic University of Valencia, Spain and at National Cheng Kung University, Taiwan. He also worked for the US Army Human Engineering Laboratory to conduct projects related to application of computer vision in indirect viewing systems. His current research activities include automated inspection, neural networks and information combination.

Fang-Chih Tien is a doctoral candidate in the Department of Industrial Engineering at University of Missouri–Columbia. He received his B.S. from Chung-yuan Christian University and M.S. from University of Missouri–Columbia, both in industrial engineering. His research activities include stereo vision and neural networks for automated inspection.