Applied Mathematics & Information Sciences
An International Journal

http://dx.doi.org/10.12785/amis/081L35

Computer-Integrated Micro-Assembling with Image-Servo System using ANN

Chih-Jer Lin[1,*], *Chii-Ruey Lin*[2] *and Shen-Kai Yu*[1]

[1] Institute of Automation Technology, National Taipei University of Technology, Taipei, Taiwan, R.O.C.
[2] Department of Mechanical Engineering, National Taipei University of Technology, Taipei, Taiwan, R.O.C.

Received: 11 May. 2013, Revised: 5 Sep. 2013, Accepted: 6 Sep. 2013
Published online: 1 Apr. 2014

Abstract: A micro-assembling task using an image-servo positioning system is studied in this work. To improve the micro-assembling efficiency using manual operations, an image-servo automatic assembling system is established using a XYQ stepping positioning stage with a self-developed image recognition system. The circle outlines and the cross marks of the micro parts are used for this proposed system to control the XYQ-stepping stage to achieve assembling tasks. For the proposed image-servo system, the coarse positioning task and the fine positioning task are designed. First, the Sobel operator is used to find circle outline of positioning mark for the coarse positioning. Second, calculating the centre and radius of the positioning circle using a least-mean-square error is to guide the XYQ-stepping stage to perform the positioning. After performing the coarse positioning task, Artificial Neural Network (ANN) systems are studied to improve the positioning precision via compensating positioning errors due to the image distortion. The main contributions of this paper are using BP and RBF neural networks to perform the nonlinear geometry transformation from image coordinates of the pixels to the actual positions in the global coordinate system.

Keywords: Micro-positioning Stage, Image Recognition, Vision guidance, Line Detection, Micro-assembly, Neural network.

1 Introduction

For the tasks of assembling micro components made by micro electro- mechanical system (MEMS) technology, manual operation is too slow and makes the mass-producing for micro systems difficult in the industry. There are two major limiting factors of the micro assembling process for the manual operation. One is the quantity problem, because the speed of the automatic assembling system is larger than the manual system. The other is the quality problem, because the manual operation cannot be guaranteed the assembling precision. As assembling minute parts, the manual operation is unstable due to the worker's eye strain. Therefore, the automatic micro assembling with high precision and high speed is very necessary for the industry. There are two types of micro-systems and they are monolithic and hybrid micro-systems [1]. In this study, an automatic optical inspection (AOI) system is proposed for assembling micro systems via a designed image-servo system. In literature [2], the micro-assembling task of the micro system was discussed and the image recognition of the positioning mark was used to achieve the automatic positioning process.The positioning mark is created by etching processes, so there are many blur in the outline of the image. To solve this problem, a sub-pixel method of outline processing is proposed to improve geometric pattern. In the visual-servo system designs, the positioning task is separated into two steps, which are the coarse positioning task and the fine positioning task. For the coarse positioning task, the Sobel operator is used to find outline of circular positioning mark and then to calculate circle center and radius by using the way of least-mean-square error algorithm [3].To increase the precision of the coarse positioning task by the image recognition, the neural network systems are studied to correct the twisting image errors of the CCD camera. For the fine positioning task, the Hough transform is applied to find the straight line of the cross positioning mark [4, 5]. Finally, a computer integrated micro-assembling system is implemented for achieve micro-assembling of a PZT inkjet head with image-servo system.

[*] Corresponding author e-mail: cjlin@ntut.edu.tw

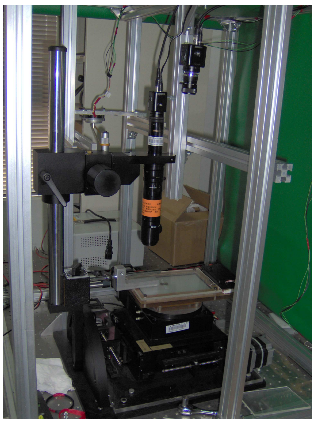


Fig. 1: Automated micro assembling system diagram [2].

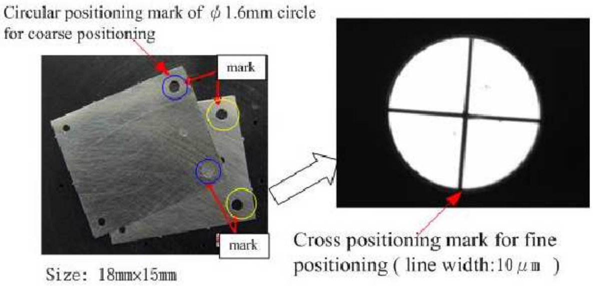

Circular positioning mark of ϕ 1.6mm circle for coarse positioning
mark
mark
Size: 18mm×15mm
Cross positioning mark for fine positioning (line width:10 μm)

Fig. 2: Micro components for assembling and its positioning-aided mark [2].

2 Experimental Setup

In this paper, the same automated micro assembling system studied in literature [2] is used to investigate the image servo task based on the artificial neural networks. The system (shown in Fig. 1) consists of the assembling mechanisms and the visual system. The assembling mechanisms are a upper clamping apparatus and a lower XYQ-stepping stage. The XYQ-stepping stage consists of XY positioning stage (NAPLES COOMBE 29000 and 5000, 100mm×100mm travel, 1mm pitch, 0.72 stepping angle) and the rotary stage (NAPLES COOMBE 5000, with 90:1 gear system, 0.008 per pulse, 500 pulse/round). The XYQ-stepping stage's motors are actuated by the micro-stepping drivers (Panasonic KR-535M). The motion command is produced using a four axes motion card (GALIL DMC-1842) and the control code is developed in Borland C++ builder to integrate the motion control with the vision system. The vision system consists of a global visual CCD camera equipped with 2× wide lens and a micro vision via a CCD camera equipped with 600× micro lens. The visual images are captured via the MATROX Meteor-II capturing card. The global vision is used to amend the position and orientation of micro parts. The image-servo procedures are as follows. First, the positioning circle mark is captured by the global CCD and the positioning marks are recognized from the work space by the designed code. Second, the micro vision is used to achieve precision positioning via the cross mark in the positioning circle mark. If the cross mark has the smaller line width (as shown in Fig. 2), the proposed computer-integrated system can guarantee the more precision for the micro-assembling tasks.

3 Vision-based Positioning

As discussed in the literature [2], the positioning task consists the coarse positioning and fine positioning tasks. The coarse positioning uses the image captured by the global visual CCD camera equipped with 2× wide lens. However, the distorted image from the wide lens (as shown in Fig. 4) may degrade the precision of the proposed system. The relation between the CCD-matric and the field of view can be described in Fig. 3, where its optical axis is orthogonal to the ground plane. After obtaining the image coordinates of the positioning mark center in the image domain, there exist a mapping relation between the image coordinates of the pixel (u ,v) and the global coordinate (X_R,Y_R) as follows.

$$x = \frac{(u-w'/2)\cdot W}{w'} \quad , \quad y = \frac{(v-h'/2)\cdot H}{h'} \tag{1}$$

$$\begin{aligned} X_R &= x_C + x\cos\theta_C - y\sin\theta_C \\ Y_R &= y_C + x\sin\theta_C + y\cos\theta_C \end{aligned} \tag{2}$$

, where w' and h' are the dimensions of the CCD matrix, and W and H are the dimensions of the field of view; (u,v) are the image coordinates of the positioning mark's center; $X_C = [x_C, y_C, \theta_C]^T$ is the vector of the camera position and orientation in the global frame; (X_R,Y_R) are the actual coordinate in the global frame. Once the positions in image coordinates are known, they can be computed by using the known dependence between the dimensions in the image and in the reality. However, the distorted image from the wide angle (or fish-eye) camera is shown in Fig.4 [6].The mapping relation is usually nonlinear and can be

written in the following general form:

$$\begin{bmatrix} X_R \\ Y_R \end{bmatrix} = F\left(\begin{bmatrix} u \\ v \end{bmatrix}, X_C\right) \tag{3}$$

, where $F(\bullet)$ is a nonlinear mapping function between the image and the global coordinates. Therefore, the image-servo system using the vision-based sensor is needed to be calibrated with respect to the global coordinate system. The main contributions of this paper are using BP and RBP neural network methods to perform the nonlinear geometry transformation from

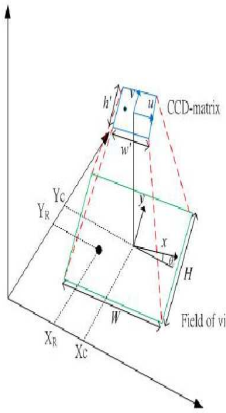
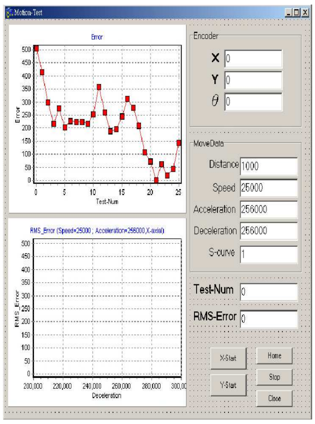


Fig. 3: Overhead vision system geometry.



Encoder

X 0

Y 0

θ 0

MoveData

Distance 1000

Speed 25000

Acceleration 256000

Deceleration 256000

S-curve 1

Test-Num 0

RMS-Error 0

Fig. 5: The self-developed UI for the controller setup and the experimental display.

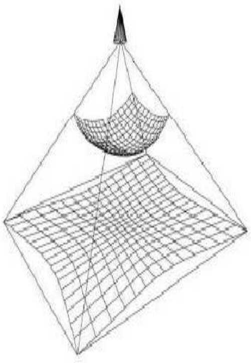


Fig. 4: The distorted image geometry for a wide angle lens [6].

image coordinates of the pixels to the actual positions in the global coordinate system.

Tuning the controller of XY positioning stage. To guarantee the accuracy and repeatability of the vision-based positioning system, we need to calibrate the above vision system via the other more precise positioning sensors. On one hand, to obtain the precise positioning data to train the following neural network systems, the optical encoders and the linear scale with 0.1μm resolution (Mercury 2000, made by MicroE systems in USA) are equipped on the XY positioning stage as the positioning sensors as shown in Fig. 5. On the other hand, the motors of the XY positioning stage are actuated by the micro-stepping drivers to achieve precise positioning and the XY positioning stage is driven by the stepping motors via the open-loop control. Therefore, to improve the performance of positioning for the XY stage, we need to tune the control parameters, such as speed, acceleration and deceleration), via experimental tests. With using the self-developed C++ code as shown in Fig. 5, the proposed vision system is tested by the specified positioning tasks and the actual regulation errors are stored automatically in the computer. We can obtain the optimal control parameters, which are described in Table 1, via implementing the specified experiments for the controller. The results of the root-mean-square (RMS) errors in Table 1 are obtained by the average of twenty experiments using the optimal parameters. For long positioning distance, using S-curve motion profile can help to smooth the motion and reduce the possibility of exciting a vibration or oscillation, for example we study the influence of the S-curve parameter for the 1000 μm step command as shown in Fig. 6.

Table 1. Nonlinear geometry transformation via ANN.

Optimal speed 2700μm/s, optimal acceleration 27000μm/s^2, and optimal deceleration 26000μm/s^2	
Distance (μm)	RMS-Error (μm)
10	0.308
500	0.483
1000 (s-curve coefficient: 0.6)	0.424

After self-tuning the controller of the XY stage, the lower component of the micro system is carried by the AMAS to perform a positioning task as shown in Fig. 7. The global coordinates (X_R, Y_R) of the positioning mark, which is measured using the linear encoders (Mercury 2000), are compared from the vision-based coordinates (X_R', Y_R'), which are obtained using Eq. (1)-(2); the vision-based positioning error can be defined as follows.

$$error = \sqrt{(X_R - X_R')^2 + (Y_R - Y_R')^2} \tag{4}$$

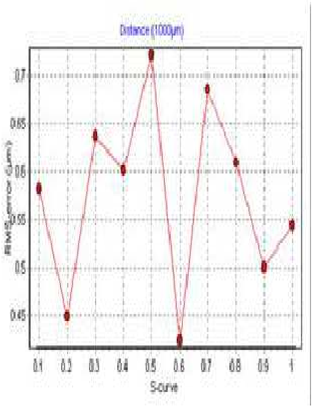


Fig. 6: The S-curve for the motion and the RMS error for the different S-curve's parameter.

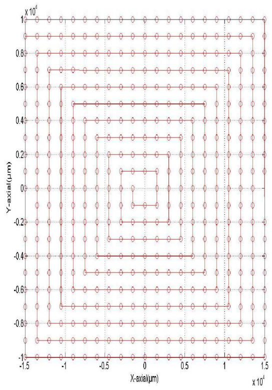


Fig. 7: The designed point-to-point positioning task to calibrate the vision-based positioning system.

Due to the distorted image from the wide angle lens, Fig. 8 shows that the greater errors would appear if the picture was taken from a much greater distance with respect to the center of the FOV. The maximal error before the calibration is almost 150 μm. Because the mapping between the vision and the global coordinates is nonlinear and the geometry correction transformation is difficult to obtain, we use artificial neural network (ANN) methods to perform this nonlinear geometry transformation from image coordinates of the pixels to the actual positions in the global coordinate system. The training process of the ANN is described as follows. Control the XY positioning stage using the optimal parameters to follow the point-to-point positioning task as shown in Fig. 7. Capture the image and obtain the image coordinate (u ,v) for the center of positioning mark for the micro component at each positioning point. Obtain the global coordinates (X_R, Y_R) using linear encoder at each point simultaneously. Store the image coordinates (u,v) and the global coordinates (X_R, Y_R) for each point. Use the input data of the image coordinates (u,v) and the output data the global coordinates (X_R, Y_R) to train the ANN. Fig. 9 shows the block diagram for this nonlinear geometry transformation via ANN.

4 Main Results

Generally, the ANN models can be divided into three types: supervised learning algorithm, unsupervised learning algorithm, and associative memory learning algorithm. In this study, the image-servo problem is suitable to use the supervised learning ANN. Therefore, we investigate the back-propagation network (BPN) and radial basis function network (RBFN) to achieve the nonlinear geometry transformation between the image coordinate and the global coordinate. Because the relation between the input data (u,v) and output data (X_R, Y_R) is

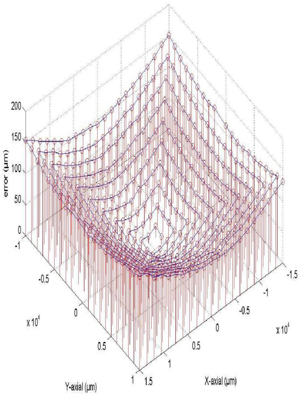


Fig. 8: The calibration errors for the linear transformation between the global coordinates and image positioning coordinate.

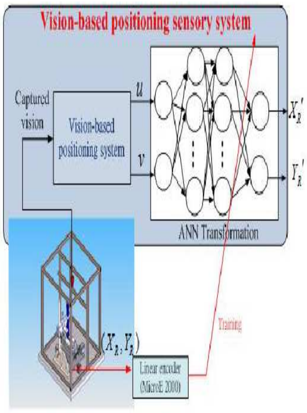

Vision-based positioning sensory system
Captured vision
Vision-based positioning system
u
v
X_R'
Y_R'
ANN Transformation
(X_R, Y_R)
Linear encoder (MicroE 2000)
Training

Fig. 9: The proposed vision-base positioning sensory system using ANN

very nonlinear, three case studies are carried out to discuss the performance of the proposed method. In Case 1, we do not change the training data to train the BPN for this mapping relation. In Case 2, we used the RBFN to learn this mapping relation. In Case 3, we transfer the training data in the form of the polar coordinate and use BPN to learn this nonlinear mapping.

Case study 1. The multilayer BP artificial network consists of input layer, hidden layer and output layer (as shown in Fig. 9). Each layer has many neurons which are connected to the other neurons in the other layer using the synaptic weights. There is only a single input and output layer in the ANN, but there are one or more hidden layers. The neurons in the ANN model are called processing elements. Figure 10 shows the mathematical model of a neuron, where Y_j is the output element, $X_1, X_2, \cdots, X_n$are the input elements, $W_1, W_2, \cdots, W_n$are the synaptic weights, and $\varphi(\bullet)$ is the activation function. The value of NET_j refers to the weighted sum of all the inputs to the *jth* neuron Y_j for n inputs as follows.

$$NET_j = \sum_{i=1}^{n} W_{ji} X_i \tag{5}$$

The activation function $\varphi(\bullet)$ is used to reflect the weighted sum to the results of the output element and it is usually designed as the Sigmoid function as shown in Fig. 11. Therefore, the relation between the input layer and the *jth* neuron Y_j can be described as follows.

$$Y_j = \varphi(NET_j) = \frac{1}{1 + e^{-NET_j}} \tag{6}$$

The operation procedure of BPN can be divided into two stages, which are the learning stage and the recalling stage. For the learning stage, there are initialization of the network and network training work. In this case, we generate random values between -1 and 1 for the weight and bias initialization in each layer. The network training work will adjust each weight and bias to make the output of network approach the desired output. After the learning stage is finished, the relationship between the input data (u,v) and output data (X_R, Y_R) is established.

The learning algorithm of the BPN consists of two stages: feed-forward and back-propagation. In the feed-forward stage, each neuron calculates the output values using Eq. (5) with the fixed weights. Finally, the BPN calculates the actual output and the network can obtain the mean square error via comparing the desired output from the actual output. The mean square error values E can be defined as:

$$E = \frac{1}{2} \sum_j (T_j - Y_j)^2 \tag{7}$$

, where T_j is the desired output value, Y_j is the actual output value and E is the mean square error. To make the mean square error will approach to zero; the weight values of BPN should be modified until the output conforms to the desire output. Therefore, in the back-propagation stage, the weight values of neuron are updated using the
following equation.

$$W_{ij}(n+1) = W_{ij}(n) + \Delta W_{ij}(n) \tag{8}$$

where W_{ij} is the weight value, index n represents the iteration number in the learning procedure, and ΔW_{ij} is the network correction value. To make the ANN have good convergence, the correction valued are suggested as the following equation.

$$\Delta W_{ij}(n) = -\eta \cdot \frac{\partial E(n)}{\partial W_{ij}(n)} = -\eta \cdot \frac{\partial E(n)}{\partial Y_j(n)} \cdot \frac{\partial Y_j(n)}{\partial W_{ij}(n)} = -\eta \cdot \delta(n) \cdot (-X_i(n)) \tag{9}$$

, where η is the learning rate for the ANN and $\delta(n)$is the local gradient function. Then, the update law of the weight values can be shown in Fig. 13 and the update law is described as follows.

$$W_{ij}(n+1) = W_{ij}(n) + \eta \cdot \delta(n) \cdot X_i(n) \tag{10}$$

The flowchart of the BPN learning algorithm is shown in Fig. 13. In this study, we design the BPN's setup for this case as follows: one input layer has 2 neurons; two hidden layers have 10 neurons at each layer; one output layer has 2 neurons; the learning rate is chosen as 0.9; the input and output data are normalized in the range from 0 to 1 and the output value are inverted to the original range in the recalling stage. In this study, the BPN is coded in LabView language and Fig. 14 shows the block-diagram in Labview. The procedures for the BPN are as follows?G
The architecture of BPN is set for 4 layers with an input layer with 2 nodes, the first hidden layer with 10 nodes, the second hidden layer with 10, and the output layer
with 2 nodes.

1.The BPN of training numbers is set as 200000.
2.The learning rate is set 0.9.
3.Initial the weight and theta by random between -1 and 1.
4.Normalize the input value from 0 to 1.
5.Normalize the output value from 0 to 1.
6.Start the learning task.
7.Calculate the hidden and output layer value by weight and theta.

$$\begin{aligned} net_k &= \textstyle\sum_i W_{ik} x_i - \theta_k \\ h_k &= \tfrac{1}{1+\exp(-net_k)} \\ net_j &= \textstyle\sum_k W_{kj} h_k - \theta_j \\ y_j &= \tfrac{1}{1+\exp(-net_j)} \end{aligned}$$

8.Calculate the output and hidden error value.

$$\begin{aligned} \delta_j &= (t_j - y_j) \cdot y_j \cdot (1 - y_j) \\ \delta_k &= (\textstyle\sum_j \delta_j \cdot W_{kj}) \cdot h_k \cdot (1 - h_k) \end{aligned}$$

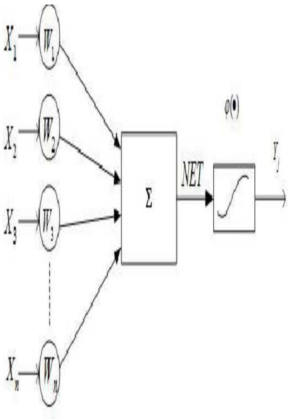


Fig. 10: The mathematical model of the neuron for the BPN.

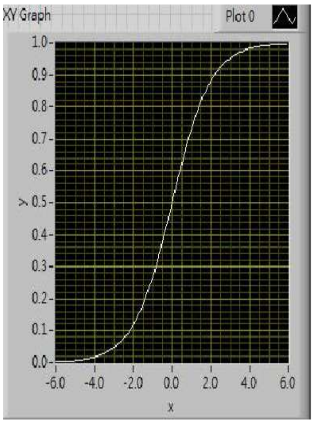


Fig. 11: The activation function of the Sigmoid function.

9. Calculate the weight and theta Correction of hidden and output layers.

$$
\begin{aligned}
\Delta W_{kj}(n) &= \eta\delta_j h_k + \alpha\cdot\Delta W_{kj}(n-1)\\
\Delta\theta_j(n) &= -\eta\delta_j + \alpha\cdot\Delta\theta_j(n-1)\\
\Delta W_{ik}(n) &= \eta\delta_k x_i + \alpha\cdot\Delta W_{ik}(n-1)\\
\Delta\theta_k(n) &= -\eta\delta_k + \alpha\cdot\Delta\theta_k(n-1)
\end{aligned}
$$

10. Update the weight and theta of hidden and output layers.

$$
\begin{aligned}
W_{kj} &= W_{kj} + \Delta W_{kj}\\
\theta_j &= \theta_j + \Delta\theta_j\\
W_{ik} &= W_{ik} + \Delta W_{ik}\\
\theta_k &= \theta_k + \Delta\theta_k
\end{aligned}
$$

11. When the learning task is finished, the recalling task is enabled.
12. Calculate the output hidden and output layer value.
13. Unnormalize the output value to map the global coordinates.
14. Compare the output value and current value.

Fig. 15 shows the errors between the actual global coordinate (X_R, Y_R) and the output (X_R', Y_R') of the BPN using the input data from the vision-based positioning. The resulting data shows that the errors are small at most area except the edge of FOV. The maximal error in this calibration is almost 75.448 μm.

Case study 2. For the radial basis function network, the bell shaped curves in the hidden nodes indicate that each hidden layer node represents a bell shaped *radial basis function* that is centered on a vector in the feature space. Fig. 16 shows a radial basis function neural network. There are no weights on the lines from the input nodes to the hidden nodes. The input vector is fed to each *n-th* hidden node where it is put through the nodes using radial basis function as follows.

$$
\varphi_j(\mathrm{x}) = \exp\left(-\frac{\left\|\mathrm{x}-m_j\right\|^2}{2\sigma_j^2}\right) \tag{11}
$$

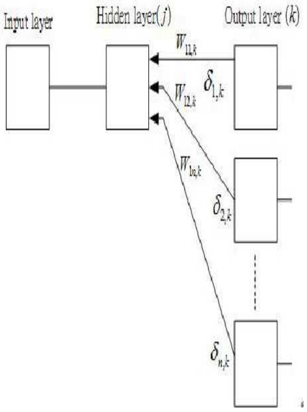


Fig. 12: The update law for the back-propagation method.

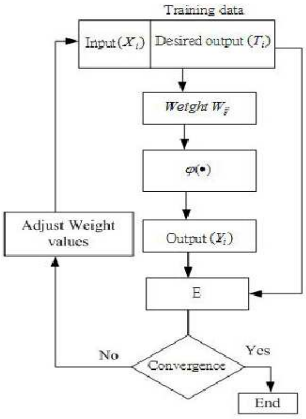


Fig. 13: The flowchart of the training process for the nonlinear mapping transformation.

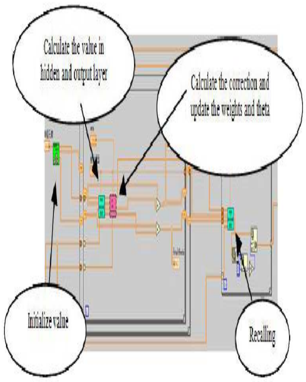

Calculate the value in hidden and output layer
Calculate the correction and update the weights and theta
Initialize value
Recalling

Fig. 14: The flowchart of the BPN for the block-diagram in Labview

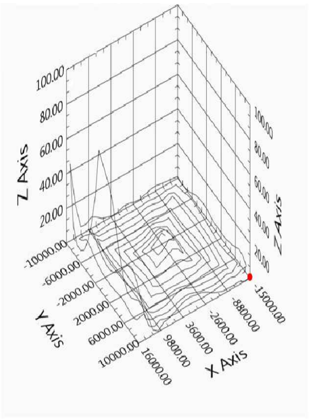


Fig. 15: The calibration errors for the BPN transformation in Case 1.

, where $\left\|\mathrm{x}-m_j\right\|^2$ is the square of the distance between the input feature vector **x** and the center vector m_j for the radial basis function. The output of RBFN is the linear combination of the hidden nodes as follows.

$$y=\sum_{j=0}^{J} w_j\varphi_j(\mathrm{x}) \tag{12}$$

Fig. 17 shows the architecture for the RBFN. Therefore, RBFN has the following characteristics: there is only one hidden layer; the output units are linear; distance based aggregation function for the hidden units; activation functions with radial symmetry for hidden units; the convergence rate is faster than the BPN. In this case study, there are 35 units in the hidden layer and the learning rate is chosen as 0.8.

The computing procedures for the RBFN are as follows?G

1. The architecture of RBFN is designed as 3 layers with 2-35-2 nodes in the input, hidden, and output layer, respectively.
2. The training numbers is set as 35000.
3. The learning rate is set as 0.8.
4. Initialize the weight, theta, Qk, Vik and Cik by radom.
5. Normalize the input value from 0 to 1.
6. Normalize the output value from 0 to 1.
7. Start the learning task.
8. Calculate the hidden and output layer value by weight and theta.

$$\begin{aligned}&net_k=Q_k^2\sum_i V_{ik}^2(x_i-C_{ik})^2\\&h_k=\exp(-net^2)\\&net_j=\sum_k W_{kj}h_k-\theta_j\\&y_j=\tfrac{1}{1+\exp(-net_j)}\end{aligned}$$

9. Calculate the output and hidden error value.

$$\begin{aligned}&\delta_j=(t_j-y_j)\cdot y_j\cdot(1-y_j)\\&\delta_k=-(\sum_j\delta_j\cdot W_{kj})\cdot h_k\end{aligned}$$

10. Calculate the weight and theta Correction of hidden and output layers.

$$\begin{aligned}&\Delta W_{kj}(n)=\eta\delta_j h_k+\alpha\cdot\Delta W_{kj}(n-1)\\&\Delta\theta_j(n)=-\eta\delta_j+\alpha\cdot\Delta\theta_j(n-1)\\&\Delta C_{ik}=2\eta\delta_k Q_k^2 V_{ik}^2(x_i-C_{ik})\\&\Delta Q_k=2\eta\delta_k Q_k\sum_i V_{ik}^2(x_i-C_{ik})^2\\&\Delta V_{ik}=2\eta\delta_k Q_k^2 V_{ik}(x_i-C_{ik})^2\end{aligned}$$

11. Update the weight and theta of hidden and output layers.

$$\begin{aligned}&W_{kj}=W_{kj}+\Delta W_{kj}\\&\theta_j=\theta_j+\Delta\theta_j\\&C_{ik}=C_{ik}+\Delta C_{ik}\\&Q_k=Q_k+\Delta Q_k\\&V_{ik}=V_{ik}+\Delta V_{ik}\end{aligned}$$

12. When the learning task is finished, the recalling task is executed.
13. Calculate the output hidden and output layer value.
14. Unnormalize the output value to map the global coordinates.
15. Compare the output value and current value.

Fig. 18 shows the block-diagram of the RBFN using in LabVIEW and Fig. 19-20 compare the actual global coordinates (X_R,Y_R) from the output (X_R',Y_R') of the RBFN using the proposed vision-based positioning. In Fig. 20, the resulting data also shows that the errors are small at most area except the edge of FOV as similar as BPN. The RMS error of the all points in this calibration is almost 139.029 μm. The positioning error for the RBFN is larger than the one of the BPN, but the training time of RBFN (1832 seconds) is much less than BPN (14412 seconds).

Case study 3. For the results in Case studies 1 and 2, the errors for the vision-based positioning are small at most of the FOV; however, the system may fail at the edge of the FOV. To improve the precision and repeatability of the proposed vision-based positioning sensory system

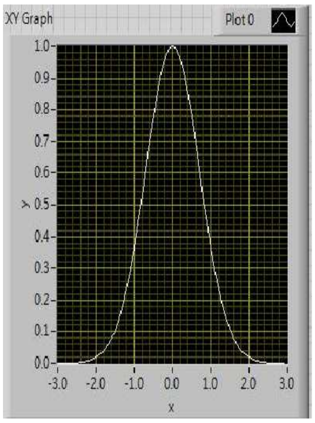


Fig. 16: A bell shaped radial basis function for the RBFN.

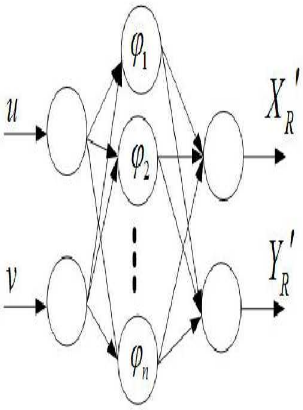


Fig. 17: The architecture of the RBFN.

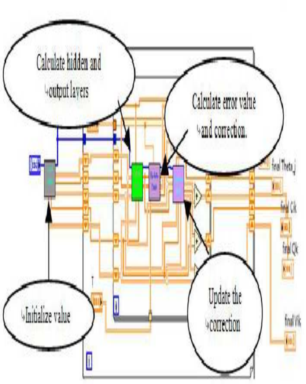

Calculate hidden and output layers
Calculate error value and correction.
Initialize value
Update the correction

Fig. 18: The block-diagram of RBFN using in Labview.

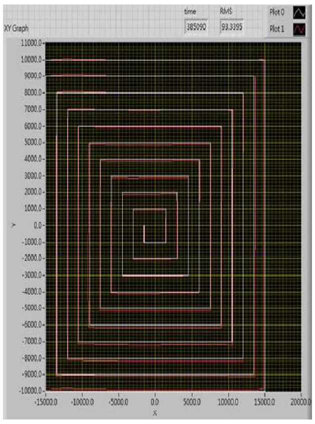


Fig. 19: Compare the actual global coordinate from the output of the RBFN positioning transformation.

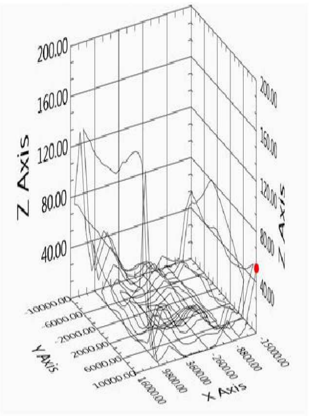


Fig. 20: The calibration errors for the RBFN transformation in Case 2

(VBPSS) as shown in Fig. 9, we make some algorithm's modification for the artificial neural network. Observing the trendy of the vision-based positioning error in Fig. 8, we can find the error is related to the distance from the CCD matrix's center. Therefore, the training data of BPN is transferred from the Cartesian coordinate to the polar coordinate at first; after the training of BPN is finished and the output value is taken the inverse transformation from the polar coordinate to the Cartesian coordinate. Fig. 21 shows that the errors of VBPSS at all FOV are much smaller than the above and the maximal error is 2.3075 μm. Therefore, the proposed VBPSS can achieve the precision requirement for the system.

5 Conclusion

In this paper, we have investigated the vision-based positioning using BP and RBF neural networks to perform the nonlinear geometry transformation from

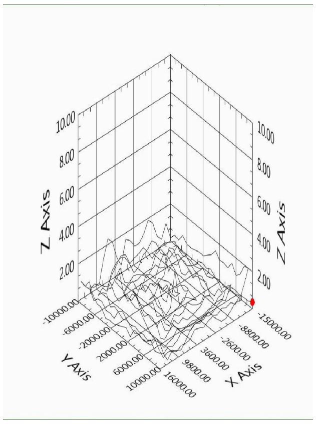


Fig. 21: The calibration errors for the BPN transformation using the polar coordinates in Case 3.

image coordinates of the pixels to the actual positions in the global coordinate system. There are two major conclusions can be made in this study. (a) The transformation from image coordinates to the global coordinates is very nonlinear and BPN and RBFN cannot establish the mapping relation at the edge of FOV. (b) Using the polar coordinate transformation with the BPN can improve the precision of the vision-based positioning sensory system and the accuracy is under 3μm. Finally, we have presented the integration of micro-stepping positioning stage with the designed vision-based positioning system to achieve the precise assembling task automatically.

Acknowledgements

This research was supported by grant NSC 100-2221-E-027-031 and NSC 100-2221-E-027-015 from the National Science Council. The authors thank the reviewers for their comments.

References

[1] S. Hutchinson, G. A. Hager and P. I. Corke. A Tutorial on Visual Servo Control. IEEE Transactions on Robotics and Automation, **5**, 651-670 (1996).

[2] C. J. Lin, G. Z. Chen, Y. X. Huang and J. K. Chang, Computer-integrated micro-assembling with image-servo system for a microdroplet ejector. Journal of Material Processing Technology, **201**, 689-694 (2008).

[3] K. Hanahara, T. Maruyama and T. Uchiyama, A Real-Time Processor for the Hough Transform. IEEE. Transactions on Pattern Analysis and Machine Intelligence, **10**, 121-125 (1988).

[4] N. Guil, J. Villalba, E. L. Zapata, A Fast Hough Transform for Segment Detection. IEEE. Transactions on Image Processing, **4**, 1541-1548 (1986).

[5] D. J. Kerbyson, T. J. Atherton, Recent Advances in Clinical Neurophysiology. IEEE Conference on Image Processing and its Applications, Edinburgh, UK, (1995).

[6] P. Skrzypczynski, Uncertainty models of vision sensors in mobile robot positioning. Int. J. Appl. Math. Comput. Sci., **15**, 73-88 (2005).

Lin, Chih-Jer received the MS and PhD degree in Mechanical Engineering from National Cheng Kung University, Taiwan. He is currently a Professor in Graduate Institute of Automation Technology, National Taipei University of Technology, Taiwan (R.O.C.). His research interests are in the areas of mechatronics, precision positioning control, robotic motion planning and control, computer-integrated manufacturing, and computational intelligence.

Lin, Chii-Ruey received the PhD degree in Mechanical Engineering from National Chiao-Tung University, Taiwan. He is currently a Professor in Mechanical Engineering, National Taipei University of Technology, Taiwan (R.O.C.). Professor Lin efforts focus on mechanical engineering, coating technology, mechatronic, nanotechnology(esp. in diamond coating technology), just like diamond films, DLC, CNTs, nano-diamond films, ultra-nano-diamond films, untra-nano-diamond wire(rods) etc..

Yu, Shen-Kai received the Bachelor degree in Mechanical Engineering from National Taipei University of Technology, Taiwan. He is currently a PhD student in Graduate Institute of Mechanical and Electrical Engineering, National Taipei University of Technology, Taiwan (R.O.C).