Available online at www.sciencedirect.com
SCIENCE DIRECT®
Computers in Industry 56 (2005) 986–996
COMPUTERS IN INDUSTRY
www.elsevier.com/locate/compind

# A vision and robot based on-line inspection monitoring system for electronic manufacturing

Immanuel Edinbarough [a,*], Roberto Balderas [b], Subhash Bose [b]

[a] Department of Engineering, The University of Texas at Brownsville, 80 Fort Brown, Brownsville, TX 78520, USA
[b] Department of Manufacturing Engineering, The University of Texas – Pan American, Edinburg, TX 78539, USA

Received 1 December 2004; received in revised form 31 March 2005; accepted 31 May 2005
Available online 20 October 2005

**Abstract**

This paper presents a neural network-based vision inspection system interfaced with a robot to detect and report IC lead defects on-line. The vision system consists of custom software that contains a neural network database for each of the ICs to be inspected on a PCB. The vision system uses gray scale images and a single layer neural network with three outputs based on defect criteria. Each IC has a different inspection area, thus, the input vector varies for each of the ICs. The IC networks were trained with Matlab's Bayesian regularization module. Performance of each of the networks investigated was found to be 100% based on the defect criteria. This system has been implemented and tested on several electronic products using ProE, C++ and OpenGL software platforms [R. Balderas, S. Bose, Automated robotic inspection system for electronic manufacturing, MSE Thesis, Manufacturing Engineering Department, UT-Pan American, 2002; A.I. Edinbarough, J. Amieva, Experimental study on the robotics vision inspection of electronic components, BS Thesis, Engineering Technology Department, UT-Brownsville, 2002].
© 2005 Elsevier B.V. All rights reserved.

*Keywords:* Machine vision; Robotic inspection; Neural Network; Solder defects; Electronic manufacturing

## 1. Introduction

Advancements in technology have led to miniaturization of components, leading to smaller and more powerful electronics products. These products are densely packed to close tolerances and mass-produced. To insure interchangeability and precision these products are to be 100% quality checked. As a consequence, the inspection process is often expensive and cumbersome [1,2]. In a typical manufacturing plant, approximately 30% of all manufacturing tasks are related to inspection, of which 60% of inspection tasks are visual. The break down of typical defects found during visual inspection is approximately 30% part defects, 50% assembly defects (20% of which are incorrect parts or missing parts, etc.) and 20% soldering defects [3]. It should be noted that, in order to maintain a

* Corresponding author. Tel.: +1 956 882 6623;
fax: +1 956 882 6642.
*E-mail address:* immanuel.edinbarough@utb.edu
(I. Edinbarough).

0166-3615/$ – see front matter © 2005 Elsevier B.V. All rights reserved.
doi:10.1016/j.compind.2005.05.022

certain level of quality in electronics manufacturing process, an increase in the number of solder joints by a factor of 10 requires that the number of defects be reduced by a factor of 10. Therefore, the effectiveness of an inspection system used to control a process will have a direct impact on the quality of products shipped to customers. This places an extraordinary pressure on human inspectors who are trained to identify defective parts by visual examination. This is further compounded if the equipment is used by the military, which requires 100% inspection of solder joint as required by MIL-STD 2000 for electronics assemblies. A study conducted by AT&T showed that a decision repeatability of only 44% for the same inspector inspecting solder joints, and only 6% agreement between their judgments with four inspectors checking the same board. This shows that the descriptions of allowable variations are very qualitative and subject to interpretation by inspection experts and leads to increases in manual inspection costs which can be as high as 50% of assembly costs in some cases [4].

To solve the problem of inspecting pin grid arrays (PGAs) solder joints, NEC corporation developed an automated system using a XYZ robot, a specially designed 0.9 mm optical fiber scope (OFS) with light guides, a standard optical system (TV monitor, CCD camera, PC, Frame Graber), and customized software. The automated system's objective was to identify defects on each joint on the PGA as good or bad (constricted or not soldered). The automated system correctly identified defects 100% of the time, but with 0.8% false alarm rate. The author states that by using thinner OFS scope, the false alarm rate can be reduced to 0.1% [5].

A manufacturer of large-scale integrated (LSI) packages has developed an integrated automation system for final visual inspection. The inspection was the only process that was not automated and lagged behind all other manufacturing processes. The decision was made because the packages were becoming larger and the lead pitches were finer. Therefore, to reduce inspection cost, a standardized automated inspection system was developed. The inspection items were separated into three general categories based on appearance of lead, mold and mark. Inspection on leads were co-planarity, bend (pitch), evenness, marks, scratches and foreign objects. Mold inspection criteria were voids, scratches, foreign object and cracks. Mark inspection was based on off-center letters, broken characters, blurred characters and overlapping characters. Based on the inspection criteria, six 2048 pixel CCD camera systems were developed for inspection of defects in parallel using six 2048 pixel DSP board and a computer with customized software. Parallel processing was used in order to be able to implement the system on-line, and be synchronized with other manufacturing processes [6].

There have been attempts to use X-ray laminography for automated inspection of joints and application of neural networks for classification of defects. It was found that neural networks achieved poor performance because the image resolution of an X-ray machine was poor and the imaging medium is still not well understood by the industry. Developing better imaging resolution would allow accurate three-dimensional measurement of solder joint structures and improved performance for neural networks [7].

The mentioned literatures and techniques have something in common; they focus on conventional image processing and pattern recognition for classification of defects. The procedures used are intuitive but difficult to develop and are limited to particular application. Furthermore, these techniques need to be set-up carefully and monitored by a skilled operator to ensure good results. Also, these techniques are computationally expensive, slowing down the inspection process, and thus the production line. Any automated inspection system has to be able to implement an inspection system on-line, and if possible give feedback to other automated processes for continuous improvement. A neural network approach offers several advantages for automated inspection.

Neural networks have been used experimentally for decades. Neural networks are:

- Adaptive: infer solutions often capturing subtle relationships.
- Able to generalize: can handle imperfect or incomplete data.
- Non-linear: can classify defects that are not linearly separable.
- Highly parallel: large networks can be realized using parallel processors to achieve real-time speeds.
- Tolerant of unusual noise distributions.

Although neural networks offer these advantages, the network performance suffers if the resolution of the defect image is poor (blurry). Therefore, high-resolution images (clear) that accurately portray the defect being classified are preferred to yield higher detection performance.

In this work, a visual inspection system using neural networks is used to identify common defects in electronics manufacturing. A single layer network with multiple neurons is used to classify the defects. A user friendly visual method is developed in conjunction with other inspection methods, such as circuit testing, functional testing and X-ray inspection methods where appropriate, for quality assurance.

## 2. Application of robots and vision system

### 2.1. Application of robots for automation

In the electronics industry robotic assembly of PCBs reduce product cost and increase product quality. Cartesian and SCARA robots are typically used for assembling large or irregularly shaped parts onto PCBs (e.g., connectors, transformers, potentiometers, radial devices, crystals, light-emitting diodes and large DIPs). Furthermore, machine vision technology is increasingly applied to assembly and inspection tasks traditionally performed by human operators. This is mainly because miniaturization of electronic components has increased circuit density and makes human assembly and inspection virtually impossible [8]. Therefore, in this work, the integration of several disciplines is exploited. An attempt has been made in this work to introduce an automated inspection system that has seamless integration of a robot and vision system.

### 2.2. A robot model

An articulated robot with six DOF is used in the system for manipulation and control. The robot motion positions can be programmed by an external teach pendant connected to the controller or via special software developed by a user or vendor. The user can develop the software using any high or low-level language (Basic, C++, FORTRAN, etc.). The programmed positions can be easily downloaded from the computer via the standard RS232C port. In order to position PCB boards accurately and consistently, the forward (direct) and inverse (indirect) kinematics' equations of the robot are used. The forward kinematics equation is used to determine the end effector position, given that the user knows the joint angles. More realistically, the user knows the position of an object and the trajectory an object is to follow. Thus, the user can define intermediate points along the trajectory to accomplish the task and use inverse kinematics equations of the robot to find the joint angles to control the robot. The forward and inverse kinematics equations are developed for the robotic manipulation. The robot is used for loading and unloading the PCB board for automated visual inspection. The visual system then determines if there are any defects on the board. If there are defects, it will instruct the robot to place the PCB board on a rework conveyor; otherwise, it will be sent to the next process.

### 2.3. Description of the vision system

The vision system consists of a standard CCD camera, a PC computer with data acquisition board, a customized six DOF robot, a Scorbot ER-V robot, and customized neural network software. The vision system uses multi-disciplines and standard manufacturing language to make the newly developed software easy to use in a manufacturing environment. A CCD camera is attached to the end effector of the robot. The programmer uses the PCB board's 2D CAD file to define the IC types and the centroid ($x$, $y$, $z$) of the IC. It is assumed that the IC type has been previously defined and stored in a software database. If the IC type exists in the database, then the directions and positions of inspection are predefined, so that the software can compute and store the optimal focal distances needed to obtain the largest clear image of the IC under inspection. Once the optimal focal distances are computed for all required orientations of the IC, the position of the camera can easily be found from the centroid (defined by the programmer), the shape and directions predefined for the IC, and the maximum magnification and resolution of the CCD camera. With this information, the software uses the inverse kinematics equations of a six DOF robot to obtain the appropriate joint distances and angles. The software compensates for any delays and develops a sequence to achieve efficient movement and a clear image of the IC.

## 3. Neural networks algorithm for defects identification and classification

Today, neural networks have become a flexible and a successful tool. They have been applied to aerospace, automotive, banking, defense, electronics, entertainment, financial, insurance, manufacturing, medical, oil and gas, robotics, speech, securities, telecommunication and transportation.

### 3.1. Neural network topology

The automated inspection system uses a six DOF robot to position and orient the robot camera at a constant focal distance and angle to acquire IC's pin image. The positions of the ICs centroids and direction of acquisition have been previously defined by the programmer for a particular PCB. The IC's used for defect identification are gull wing pin type. A schematic of this process is shown in Fig. 1. The figure shows different regions of the image. The region of interest for defect identification is between the PCB board and the bottom bend of the gull wing pin. The defects that can be identified are the quality of solder joint, excessive pin bend, and a lifted or missing pin. For solder quality, a resolution of at least 0.025 mm/pixel is needed, but the camera used has a resolution of only 0.1 mm/pixel. Therefore, the only defects that can be identified from the image are excessive bend and lifted pin defects.

The IC packages used for training, all had the same gull wing pin and same pin heights, but the pitch varied. In real world applications the height and pitch for each different IC varies. Consequently, for the neural net topology, each IC that has different configuration is trained individually and has its own identification network. The graphical device interface (GDI) software and the automated visual inspection system select which network to be used depending on the IC being inspected.

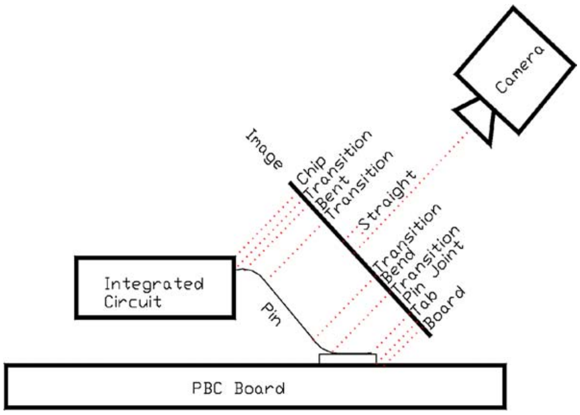



Fig. 1. Schematic of image acquisition of a six DOF robot.

ChipNumber1.txt
ChipNumber2.txt
ChipNumber3.txt

Fig. 2. Information contained within PCB1data.txt file.

### 3.2. Graphical device interface software

The GDI software has several features. One of the features is to keep a database of all the ICs network weights and biases that are trained. The training is performed offline with an acquired image data using Matlab. The GDI software has the capability of adding new ICs network weights and biases. The GDI software loads a text program of a particular PCB that contains the number of IC to inspect and a list of text files containing the ICs inspection information (Fig. 2). The files are read sequentially to extract information on an IC. The information extracted contains the network type, inspection direction, number of images to acquire and the robot execution procedure (Fig. 3). The six DOF robot uses this information to move the camera and acquire four images and saves the data in files Chip111.txt, Chip112.txt, Chip113.txt and Chip114.txt.

Chip Number 1
Chip Type 1
Number of Pins 40
Pin Height in Pixels 32
Pin Width in Pixels 3
Pin Gap in Pixels 3
Number of Images 4
Inspection Direction Sequence L B T R
Chip111.txt Chip112.txt Chip113.txt Chip114.txt
Chip Centroid (X,Y,Z) in mm 1 1 1
Mold Width in mm 120
Mold Height in mm 120
Mold Pin Separation in mm 1
Inspection Angle in (degrees) 10
Focal Length in mm 1000
Resolution in mm/pixel 0.1

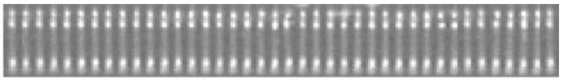
Fig. 3. File for the ChipNumber1.txt of IC contained within PCB1data.txt file.

The GDI software will then use the Chip type to select the appropriate network and Chip data text files to identify defects. If defects are found, the GDI software reports the defect type and the pin number with additional information on the screen and to a report text file for the PCB being inspected. This file can be used by an automated rework station or a rework operator to correct the defect. The above procedure is repeated until all IC text files are read. At this point, the PCB is fully inspected and another PCB file can be loaded to continue inspection. This file could be different and corresponds to a different PCB board. Thus, the GDI and inspection system are flexible and allow inspection of different PCB boards sequentially. In addition to this flexibility, the GDI software shows a visual simulation of what is being inspected as the six DOF robot performs the task on-line. This would allow monitoring of an inspection process at a remote location with feedback provided to the operator.

### 3.3. Acquisition of data

Pin defects were created on ICs of PCB boards. Since there were insufficient boards available to create defects and obtain sufficient data, several images of ICs with no defects were obtained and new images were created that simulate defects. Table 1 shows the defect criteria used by the Matlab defect generation program. The Matlab defect generation program reads in a text file (containing one integer value for each pin defect wanted) and a real image containing no pin defects (Fig. 4(a)). The defect generation program outputs an image file containing the defects (Fig. 4(b)) and a text file containing columns of the neural network responses corresponding to each pin defect shown as positive and negative ones in a row (Table 1).

(a)



(b)

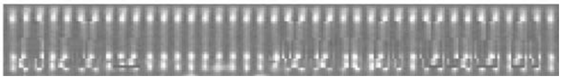


Fig. 4. (a) Real image with no defect and (b) simulated defect image.

Note the first two columns of the network output response shown in Table 1, correspond to a binary number. For example, −1 −1 corresponds to a zero decimal value, which is used to classify a straight pin, not bent. A binary number −1 1 corresponds to a decimal value of one, bent pin 1 pixel to the right or left and so on. Also the last column of the network output response corresponds to a lifted or not lifted pin state, −1 for not lifted and 1 for lifted, respectively. The image of Fig. 4(a) was acquired from a quad package containing 40 gull wing pins on each side for a total of 160 pins. The pitch of this package is 0.3 mm. The image height is 32 pixels. The width of the pin is 3 pixels, and the gap between pins is 3 pixels. Thus, the pitch is 3 pixels (0.3 mm), which corresponds to 0.1 mm/pixel CCD camera resolution. Matlab's imaging toolbox and functions were used to

Table 1
Criteria used for defect generation for data generation

| Integer | Defect type | Neural net response |
|---|---|---|
| 0 | Straight pin and not lifted | −1 −1 −1 |
| 1 | Bent pin 1 pixel to the left and not lifted | −1 1 −1 |
| 2 | Bent pin 1 pixel to the right and not lifted | −1 1 −1 |
| 3 | Bent pin 2 pixel to the left and not lifted | 1 −1 −1 |
| 4 | Bent pin 2 pixel to the right and not lifted | 1 −1 −1 |
| 5 | Bent pin 3 pixel to the left and not lifted | 1 1 −1 |
| 6 | Bent pin 3 pixel to the right and not lifted | 1 1 −1 |
| 7 | Straight pin and lifted | −1 −1 1 |
| 8 | Bent pin 1 pixel to the left and lifted | −1 1 1 |
| 9 | Bent pin 1 pixel to the right and lifted | −1 1 1 |
| 10 | Bent pin 2 pixel to the left and lifted | 1 −1 1 |
| 11 | Bent pin 2 pixel to the right and lifted | 1 −1 1 |
| 12 | Bent pin 3 pixel to the left and lifted | 1 1 1 |
| 13 | Bent pin 3 pixel to the right and lifted | 1 1 1 |

read the image file, convert the gray scale file into numerical values from 0 to 255 into an array of numbers which has a 1 to 1 correspondence with row and column of the pixel value in the image. Once this was accomplished and the locations of pin data (rows and columns) are known from the image, it is easy to generate bent pins to the left or to the right of its central location, and also lifted or broken pins from the input text file. Soldering defects (insufficient or excess solder) were not generated because the resolution of the CCD camera was 0.1 mm/pixel, which is not sufficient to detect soldering defects. A resolution of at least 0.025 mm/pixel is needed for solder defect identification. The newly generated numerical array of pins with defects is then transformed to an image with Matlab image function and saved as an image file along with its neural network response text file. The defect image and the output response text files were then used to train several networks and to find the optimum network size and the network architecture.

### 3.4. Network architecture

As stated earlier Bayesian regularization was used for training the neural networks with Matlab's trainer function. It was also discussed that each IC has its own neural network used for defect identification. The main reason is that the pitches of IC packages vary. Also, pin height, pin configuration and package technologies are different. It would be extremely difficult to find one neural network that can identify defects for all ICs with different package technologies, pin configuration, etc. Also, due to limited resolution of the CCD camera, only the defects shown in Table 1 were considered to be possible in a manufacturing environment. Obviously, the network used in this work has three outputs corresponding to the defects shown in Table 1. Generally, a network is started with sufficient neurons and layers. It is common to start with a three layer neural network and then reduce or increase the size as necessary to find an optimum network that has good performance and is able to generalize. No hidden layers were used. The following networks were tried for one IC type of the following sizes 288:6:3:3, 288:6:3, 288:3:3, 288:2:3 and 288:3. The value of 288 is the size of the input vector corresponding to one pin inspection area. The inspection area consists of a width of 9 pixels (3 pixel gap + 3 pixel pin + 3 pixel gap) multiplied by 32 pixel height for a pin totaling 288 pixel values or inputs. The input pixel values are each divided by 255 (max gray scale value) to keep the input normalized between 0 and 1. The numbers after each colon represent the number of neurons in its corresponding layer, respectively. For example, a 256:6:3:3 neural network is composed of an input vector of 256 normalized pixel values with 6 neurons in the first layer 3 neurons in the second layer and 3 neurons in the output layer. The Matlab's satlins (symmetric saturating linear) transfer function was used for all three neurons for its ease in use with simulation.

All the networks above were successful in classifying the defects presented to the network, except 288:2:3 network which consistently failed to identify the defects in Table 1, of 3 pixels bent left or right and lifted. Therefore, the minimum network size was reduced to 288:3 neural network, 1 layer network with 3 neurons and 288 inputs and 3 outputs. The same 1 layer network with 3 neurons and different input vector size corresponding to a different IC pitch and pin height was used to train similar ICs with the same defects criteria given in Table 1. Fig. 4(b) image and output response file generated were used to train the 288:3 neural network. Fig. 5(a–c) show that each graph contains the corresponding outputs of each output neuron, the correct output response for that neuron and the error corresponding to the defect generated. Each graph has three lines, correct output response, network output response and the difference between the actual value and the network output for that neuron (error). It can be seen from the graphs, the error response is flat and very close to zero. This indicates perfect identification by the neural network. One other network was trained using the same procedure discussed previously on IC-two and the results were satisfactory. The training time for the networks was from 8 to 20 min on a 400 MHz personal computer. The weights and biases for the ICs were saved as text files for use by an on-line monitoring system discussed in the following section.

## 4. On-line robotic inspection and monitoring system

The on-line monitoring system was developed using Pro-E, C++ and Scorbot ER-V language. Open

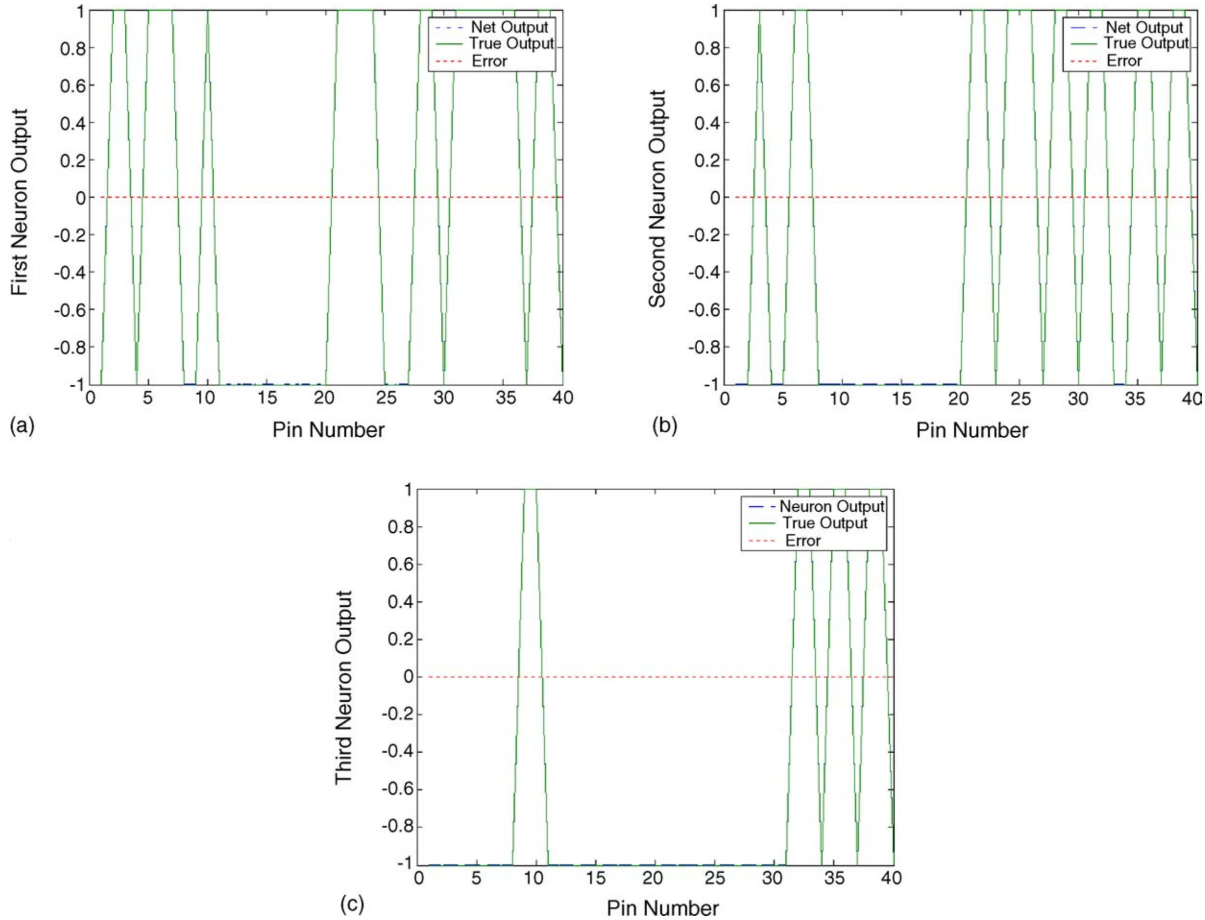



Fig. 5. (a) Output performance of first neuron, (b) output performance of second neuron and (c) output performance of third neuron.

Graphics Library (OpenGL) was used for high speed rendering of the Scorbot ER-V robot and for the animation of the inspection system. Microsoft Visual C++ (Ver. 6.0) compiler was used to compile and debug the software. Microsoft WIN32 platform was used for developing high speed rendering using OpenGL. Pro-Engineer (Pro-E) was used to create all solid models, for example, components of the robots, ICs and PCB. The Pro-E models were exported as binary stereo lithography (STL) files and were used by the on-line monitoring system to create three-dimensional animations. Eye Image Calculator developers guide and software from IO industries was used to acquire gray scale images from the CCD camera and saved as TIF graphics format. The TIF files were used for training using Matlab and for defect identification by the on-line monitoring system. Matlab neural network modules were used to generate data for training the neural network, generate appropriate weights for the neural network, and export of the weights, biases and pixel data as ASCII files to the on-line monitoring system software. The robot's direct and inverse kinematics equations were used by the on-line monitoring system to synchronize the robot motions with the animation (monitoring system). The authors had the choice of selecting a robot from the brands of Mitsubishi, Sony or Scorbot ER-V for the system. The following sections discuss the graphical features of the on-line monitoring system.

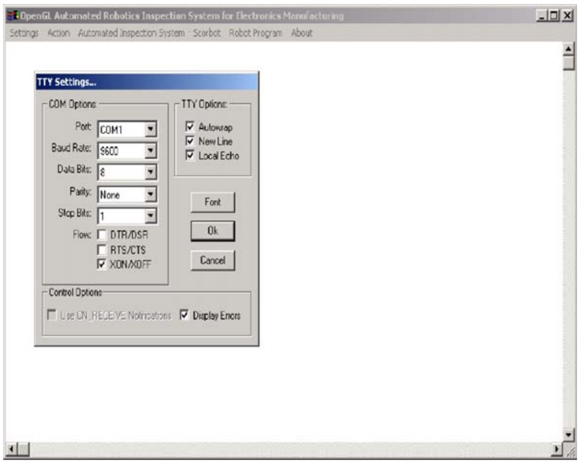



Fig. 6. Snapshot of the on-line monitoring system communications options.

### 4.1. Graphical features

The on-line monitoring system is a typical windows program with its set of menus, windows, radio buttons, check boxes, edit boxes, etc. (Fig. 6). The program utilizes both Microsoft graphics libraries (slow) and OpenGL graphics (fast) and EYE IMAGE software libraries for image acquisition. The on-line monitoring system is capable of direct communication with the robot's controller using the RS232C communications port. The user selects the correct communications port, settings, text display format, font size, echo, etc. After this, the user simply selects connect. At this time the user can type any legal controller command, and the robot will execute the command. The user can also select the shift + ~key, while connected, to shift from manual joint mode (press j key) to Cartesian mode (press c key). In joint mode the user can press the 1/Q combination to move the robot's base to the left or the right, respectively, 2/W combination to move the robot's upper arm up or down, 3/E combination to move the robot's forearm up or down, 4/R combination to move the robot's wrist pitch up or down, 5/T combination to move the robot's wrist roll clockwise or counter clockwise and 6/Y to open and close the jaw. Similarly, in Cartesian mode, the same keys are used to move the robot's hand in the ±X, ±Y, ±Z, ±pitch and ±roll. The user can press the shift + ~key to exit manual mode and enter direct mode. In direct mode the user can define and save positions in the controller for future use by the on-line inspection system.

The on-line monitoring system keeps an internal database of all weights and biases used by the neural network for defect identification. The user can add or delete IC's weights and biases as needed. The on-line system has an option to simulate the inspection process or to start monitoring an on-line process. If on-line monitoring system option is selected, the computer directly controls the inspection process of image acquisition of the PCB with the robot, the visual inspection system movements, image acquisition by the CCD camera, defect reporting, and good/rework bin placement using the robot if any defects are found on the PCB. While the robot is being commanded on-line, a synchronized animation of the process and the status report of defects classification are displayed on the computer screen for monitoring. The on-line monitoring system keeps track of the PCB and its defects for the rework process.

### 4.2. On-line inspection

To use the on-line option, the user has to select the communication settings correctly under the settings menu shown in Fig. 6 and then select the connect option under the action menu. The settings shown in Fig. 6 are the default settings. At this point the user can type any legal controller command, for example, home. If the robot homes successfully, communication is set correctly. If everything has been set correctly and the user can communicate with the robot controller, on-line monitoring of inspection of PCB could begin.

The on-line option under the automated inspection system menu uses real data (previously acquired using the EYE image software) and monitors the inspection system on-line while an animation of the process is visualized on the screen (Fig. 7(a–h)). A report of the pin data is saved in a report text file for the PCB under inspection. Fig. 8 shows a sample of the report file. If the on-line system inspects five PCB boards, there will be five PCB board defect reports. Currently, these reports are saved as text files, but in the future these files will be saved as binary files and possibly compressed to minimize disk space.

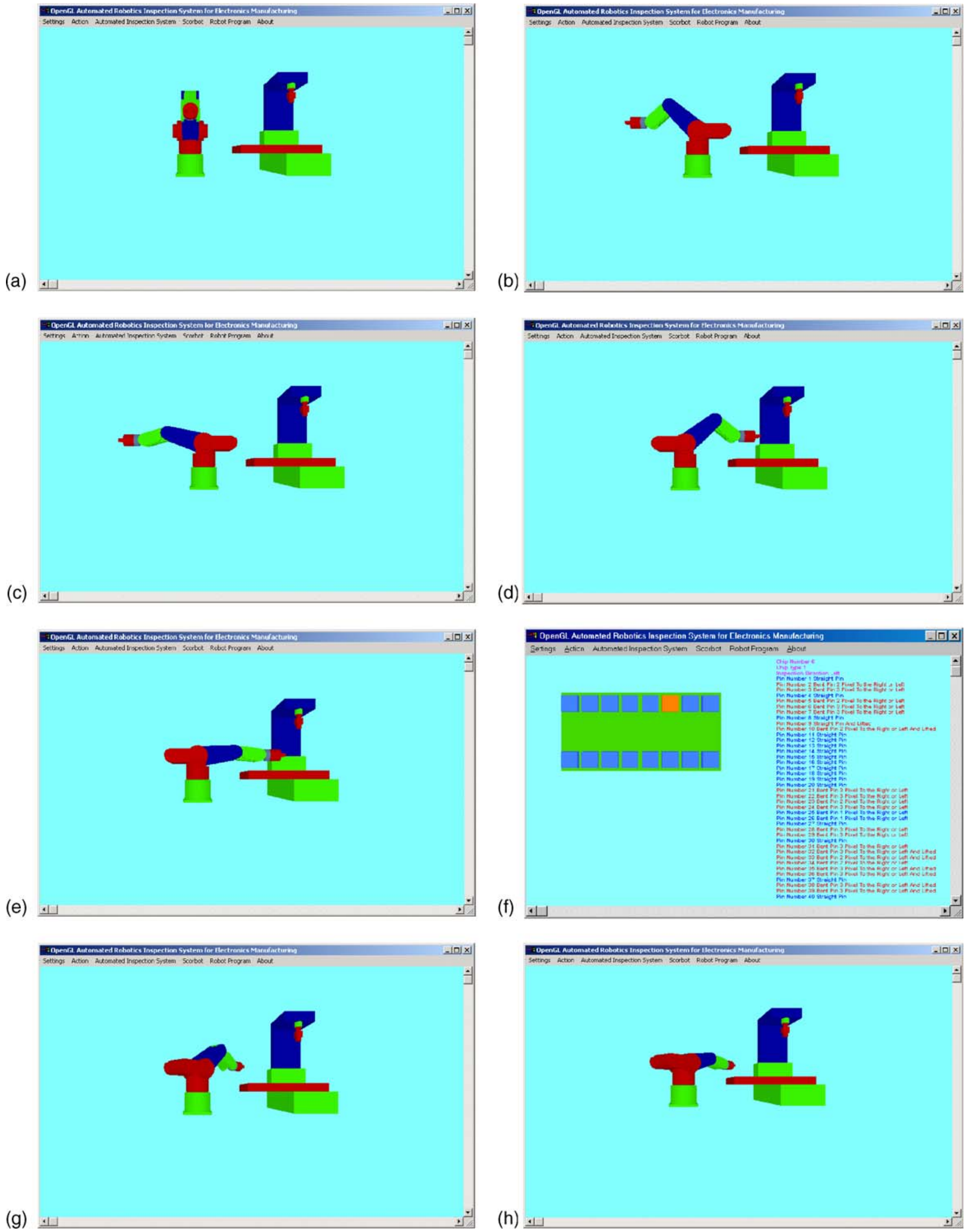



Fig. 7. Snapshots of the Scorbot ER-V inspection cycle. (a) Position 2, retracted, and facing the shipping station. (b) Position 3, retracted, and facing PCBs station. (c) Position 4, extended arm, and on PCB station. (d) Position 5, retracted, and facing inspection station. (e) Position 6, extended arm, and on inspection station. (f) PCB inspection station defect monitoring. (g) Position 7, retracted arm, facing rework station. (h) Position 8, extended arm, and on rework station.

Chip Number 1
Chip Type 1
Inspection Direction Left

Pin Number 1 Straight Pin
Pin Number 2 Bent Pin 2 Pixel To the Right or Left
…
Pin Number 39 Bent Pin 3 Pixel To the Right or Left And Lifted
Pin Number 40 Straight Pin

Chip Number 1
Chip Type 1
Inspection Direction Bottom

Pin Number 1 Straight Pin
Pin Number 2 Bent Pin 2 Pixel To the Right or Left
Pin Number 3 Bent Pin 3 Pixel To the Right or Left

Chip Number 1
Chip Type 1
Inspection Direction Right

Pin Number 1 Straight Pin
Pin Number 2 Bent Pin 2 Pixel To the Right or Left

Chip Number 2
Chip Type 2
Inspection Direction Right

Pin Number 1 Straight Pin

Fig. 8. PCB board defect report.

## 5. Discussion and recommendation

A neural networks algorithm based PCB automated visual inspection method is demonstrated in the paper. A method to monitor the inspection process in real time, including the robot manipulation is discussed in detail. Based on the defect criteria given in Table 1, it was determined that the optimum neural network size for the ICs being inspected was a single layer network with three outputs. Several of the simulated and real defect data were used to test the robustness of the neural network. The performance of the network was found to be 100% accurate. The defect criteria were kept constant for all ICs. The only thing that varied was the inspection area used for each IC due to the fact that the ICs have varying lead pitch, lead height and number of leads. Each of the ICs was trained separately with each IC having a varying input vector, and thus a different network with its own weights and biases. The weights and biases for each of the networks were then saved as text files and used by an on-line monitoring system, which stores the information in an internal database.

The on-line monitoring system was developed and implemented using Pro-E, C++ and Scorbot ER-V language. The on-line system simulation is by no means complete or perfect, but it lays the foundation for future work that will have potential for integration into the larger information technology and communication systems of the enterprise. Improvements in weight and bias database could be made. For example, instead of using text files, binary files could be used and if necessary using data compression techniques. This will reduce disk space necessary to maintain a large database of weights and biases. Data management techniques could be explored to add and delete information from the database. The defect report database could be converted to binary files, eliminating unwanted information to reduce disk space. A CCD camera of at least 0.025 mm/pixel is required in order to check the quality of solder joint leads. Overall, the proposed automated PCB inspection method along with the robot integration and on-line monitoring is found to be a straightforward implementation and effective for electronic manufacturing industries.

## References

[1] R.T. Chin, C.A. Harlow, Automated visual inspection: a survey, IEEE Transaction on Pattern Analysis and Machine Intelligence 4 (6) (1982) 557–573.
[2] R.T. Chin, Automated visual inspection: 1981 to 1987, Computer Vision Graphics Image Processing 41 (1988) 346–381.
[3] S. Hata, Vision Systems for PCB Manufacturing in Japan, IEEE Transaction, Production Engineering Research Laboratory Hitachi Ltd., 1990, pp. 792–797.
[4] V. Sankaran, B. Chartrand, D.L. Millard, M.J. Embrechts, R.P. Kraft, Automated inspection of solder joints—a neural network approach, in: IEEE Int'l Electronics Manufacturing Technology Symposium, 1995, 232–237.
[5] A. Kashitani, N. Takanashi, N. Tagawa, A solder joint inspection system for surface mounted pin grid arrays, in: IEEE Transaction, Mechatronics Research Laboratory, Functional Devices Research Laboratories, NEC Corporation, 1993), pp. 1865–1870.
[6] T. Okabe, M. Akaiwa, T. Shirakawa, T. Yokouch, T. Sugimoto, Final visual inspection system for LSI packages, IEEE Transaction 3 (1993) 1877–1881.
[7] V. Sankaran, A.R. Kalaukin, R.P. Kraft, Improvements to X-ray laminography for automated inspection of solder joints, IEEE Transactions on Components, Packaging, and Manufacturing Technology, Part C 21 (April (2)) (1998) 148–154.
[8] T.L. Landers, W.D. Brown, E.W. Fant, E.M. Malstrom, N.M. Schmitt, Electronics Manufacturing Processes, Vol. 3, Prentice Hall, Upper Saddle River, NJ, 1994.

**Immanuel Edinbarough** is a professor in the Department of Engineering at the University of Texas at Brownsville, Brownsville, Texas, USA. He received his PhD in Mechanical Engineering from the Bharathiar University, Coimbatore, India in 1996. Dr. Edinbarough has several years of industry experience. He has taught at the Rochester Institute of Technology, Rochester, NY before joining the University of Texas at Brownsville. His teaching and research interests are in the areas of automation, robotics, machine vision, and CAD/CAM/CIM. He has published several papers, in these areas, in various conferences and journals.

**Subhash Bose** is a beecherl professor and chair of Manufacturing Engineering Department at the University of Texas-Pan American, Edinburg, Texas. He received his MS (1984) and PhD (1987) from the University of Wisconsin-Madison, Wisconsin. Dr. Bose had taught at Wright State University, Dayton, Ohio and Utah State University, Logan, Utah before joining UT-Pan American. His areas of interests are robotics, control and CAM.

**Roberto A. Balderas** obtained a Bachelor of Science degree in mechanical engineering from the University of Texas at Austin in May 1993. He received his second Bachelors of Science degree in manufacturing engineering from the University of Texas Pan-American in May 1995 and also a Master of Science in Engineering in May 2002. He is currently seeking an MBA from Texas Christian University. He currently works for Lockheed Martin Aeronautics as a Sr. electronics engineer.