

# Inverse problem of photoelastic fringe mapping using neural networks

Gurtej S Grewal and Venketesh N Dubey

School of Design, Engineering & Computing, Bournemouth University, Talbot Campus,  
Fern Barrow, Poole, BH12 5BB, UK

E-mail: [vdubey@bournemouth.ac.uk](mailto:vdubey@bournemouth.ac.uk)

Received 29 October 2006, in final form 14 January 2007

Published 13 March 2007

Online at [stacks.iop.org/MST/18/1361](http://stacks.iop.org/MST/18/1361)

## Abstract

This paper presents an enhanced technique for inverse analysis of photoelastic fringes using neural networks to determine the applied load. The technique may be useful in whole-field analysis of photoelastic images obtained due to external loading, which may find application in a variety of specialized areas including robotics and biomedical engineering. The presented technique is easy to implement, does not require much computation and can cope well within slight experimental variations. The technique requires image acquisition, filtering and data extraction, which is then fed to the neural network to provide load as output. This technique can be efficiently implemented for determining the applied load in applications where repeated loading is one of the main considerations. The results presented in this paper demonstrate the novelty of this technique to solve the inverse problem from direct image data. It has been shown that the presented technique offers better result for the inverse photoelastic problems than previously published works.

**Keywords:** photoelasticity, fringe patterns, neural networks, inverse problem, sensing

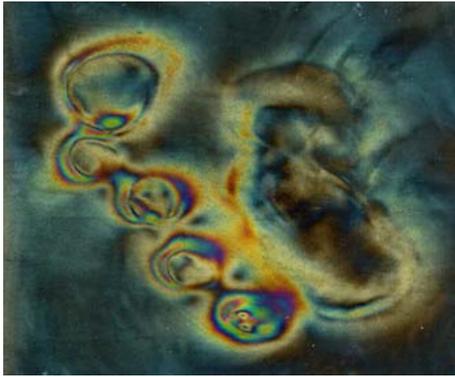
(Some figures in this article are in colour only in the electronic version)

## 1. Introduction

Photoelasticity is an optical technique for analysing stress distribution in loaded models. It is based on the phenomenon of temporary birefringence possessed by certain non-crystalline materials that are ideally isotropic but behave anisotropically when loaded [1]. The effect is temporary and persists only whilst the specimen is loaded. The technique is used to measure surface strains in structures, regardless of their shapes and sizes. The strain information is represented by fringe patterns generated due to loading which can be seen through a polariscope. If the photoelastic effect is to be used for sensing applications, the inverse problem needs to be solved, i.e. finding the applied force from the generated fringe patterns. The advantage of using this technique is to get whole-field visualization of the stress field, which may provide qualitative load information of the entire field as opposed to employing strain gauges, or load cells that only offer discrete load information. Figure 1 shows a portion

of a footprint obtained on a photoelastic medium; clearly this provides load information at every contact location of the foot. If this image can be quantified in terms of loads at all contact locations this can find use in a range of application areas. This paper discusses a neural network-based approach to analyse the whole-field image developed on a photoelastic model due to point loading or the deflection of the specimen.

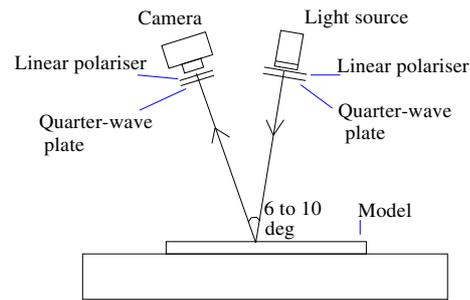
Photoelastic techniques such as RGB calibration [2–4] and phase shifting [5, 6] are useful for stress–strain applications but cannot be directly used for applications requiring load/torque estimation. The reason being the model could have complex geometry and loading conditions, and the inverse may require prior parameter determination to define precise analytical relations for the inverse problem. For such inverse problems, where it is difficult to develop analytical relations between information available and the output, use of neural networks may prove to be an effective approach, since neural networks are well suited to develop nonlinear functions between input and output [7, 8]. It is also known that



**Figure 1.** Portion of a footprint obtained on a photoelastic plate.

neural networks can tolerate slight experimental variations [9], which are likely to occur in real systems. For simpler models like beam sections under bending, disc/ring under diametral compression, which are typically used for experimental demonstrations, analytical relations can be derived but for complex shapes and loading conditions (such as shown in figure 1) it is practically impossible to derive analytical relations between the image and the applied load. Under such situations a neural network can prove to be an effective tool to solve the inverse problem. In this paper we are aiming to prove the principle of using neural networks for solving the inverse photoelastic problem. The aim is to show that direct image information from photoelastic fringes can be used to obtain the input function. The same technique can be enhanced for any generalized case involving complex fringe patterns under different input conditions for whole-field analysis. This may find application in biomedical and robotics areas such as, to assess the footprint of diabetic patients or to analyse pressure patterns of disabled subjects and thereby enacting prevention strategies by suitable footwear or bed designs. The technique may also be suitable for developing haptic interfaces in robotic applications or human–computer interface (HCI) [10].

Some attempts have been made in the past for the inverse photoelastic problems using neural networks [9, 11]. Noroozi *et al* [9] used a tensile testing machine to apply symmetrical bending load on a photoelastic beam. The fringe order was determined at known points using a polariscope and the stress difference was evaluated. The results were validated with an identical model using finite element analysis (FEA). Thus entire training data were generated from the FEA to avoid the uncertainty in the experimental data. A noise generator was used to add noise in the input data. Whilst the presented technique demonstrated the principle, it required prior calculations of fringe order for training and testing of the neural network and required a large number of training sets; data collection was another disadvantage of the system. Extracting test data required experience in photoelasticity to determine fringe order at known points in the model as inputs to the neural network. In a related work a sensor was developed using photoelastic material and a neural network to estimate torque from the fringe input [11]. Their network had an input dimensionality of 48, and 12 neurons were used in the hidden layer thus making the size of the network rather big. The network size and efficiency could have been improved if



**Figure 2.** Image acquisition rig.

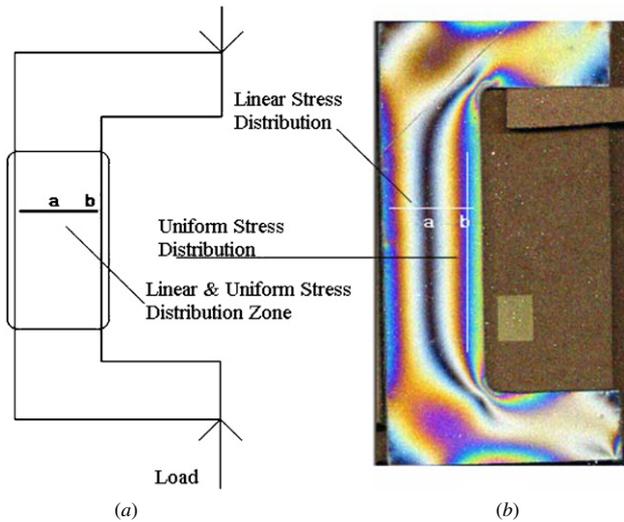
principal component analysis (PCA) or other dimensionality reduction techniques were used [12]. Photoelastic models have also been used in robotic applications for developing slip sensors [13, 14]. Cameron *et al* [13] developed mathematical models to describe a slip sensor for edge detection. A similar model was developed by Eghtedari *et al* [14] to study the effect of various design parameters such as mechanical and optical properties of photoelastic models on the resolution of the slip sensor. This paper presents an enhanced technique that uses image information from the photoelastic fringes and neural networks to determine the load information. The input given to the neural network is  $R$  intensity<sup>1</sup> from the fringe patterns and the output is the deflection of the model. The deflection is used as an output parameter since it can be easily measured during training and can be related to the applied load. The novelty of this technique is that the neural network is trained with direct image data from actual experiments and requires no prior calculations. The network size is considerably smaller than would otherwise be required and the system is generalized to analyse completely unseen data even at different camera settings. The technique does not require any knowledge of photoelasticity or FEA as used in [9].

## 2. Experimental setup

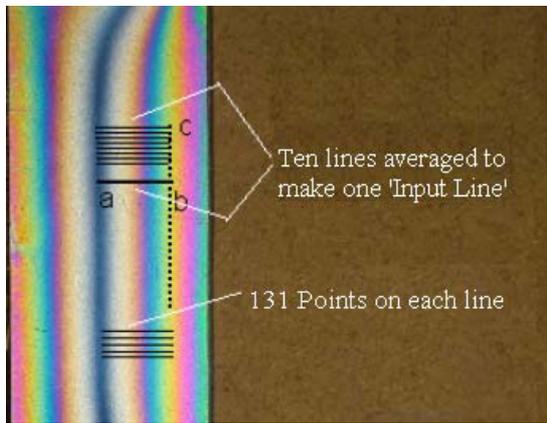
A photoelastic model of PS-1 sheet [15] with modulus of elasticity 2.5 GPa, strain optic co-efficient 0.15 and thickness 3 mm was used with reflective backing in this experiment. A 40 W incandescent bulb as the light source and a 3.2 million pixel resolution camera were used in the setup. The camera was used in conjunction with a release cable to avoid any unwanted movements during image acquisition. Figure 2 shows the basic arrangement of optical elements and the image acquisition system. The optical elements included a pair of circular polarizers kept crossed (axes are 90° apart) with each other.

The model was subjected to a bending load as shown in figure 3 and fringe patterns were obtained from the region of interest (ROI). The applied deflection was measured to one hundredth of a mm using a digital micrometer. Since within the elastic limit the model has a linear relationship between load and deflection, so training the neural network against either of the two parameters would not make any difference as far as modalities of the input functions are concerned. Thus, the deflection was set as the output target for the implemented

<sup>1</sup> Eight-bit red intensity curves across line of interest.



**Figure 3.** Model under bending load and the ROI (the rectangular area).



**Figure 4.** ROI showing multiple lines for the input data.

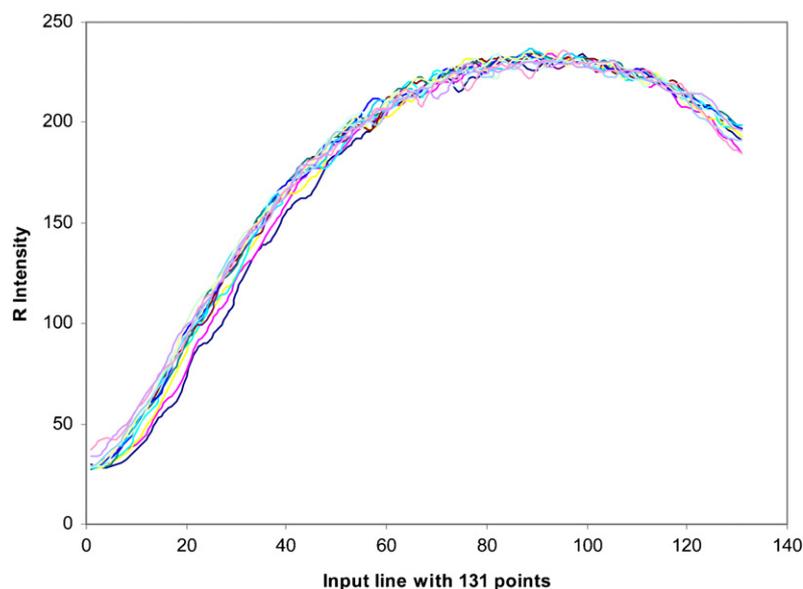
neural network. The input to the neural network was given as  $R$  intensity of line 'a-b' as shown in the figure (note that the intensity patterns of the  $G$  or  $B$  plane are similar so will give similar results). To train the network it is essential that a sufficient amount of training sets are provided.

### 3. Data acquisition

Figure 4 shows a magnified view of the ROI with input line 'a-b' showing a linear stress distribution. Considering the stress distribution along line 'c-b' in the ROI, it is evident that the stress is uniform. This facilitates extracting more than one input line from the same image thus multiple input lines are extracted along line 'c-b'. There may be some noise introduced in the data if there is uneven loading at the two ends. It has been reported, however, that noisy data can help in better 'generalization' of the neural networks [12].

For image acquisition, the camera was set in automatic mode where focus and aperture could change according to the ambient light and reflections. This introduced noise in the data in successive experiments and eliminated the need for a noise generator [9]. Images were acquired at 1 M pixel resolution and filtered for data extraction before being fed to the neural network. Image smoothing was performed to remove the irregularities present due to high sensitivity of the CCD chips. Average  $R$ -intensity of 10 input lines was extracted along 'c-b' to make one input data line (131 points each) as shown in figure 4. Similarly, 15 input data lines were extracted from each image under the load and the procedure was repeated for the successive loads. Figure 5 shows 15 input intensity curves obtained at a deflection of 0.4 mm applied to the model. The noise induced can be seen in the plot, the main consideration for selection of image data as direct input was to minimize the computational effort of [9].

Ten similar sets of experiments were performed with 25 loading and unloading (13 different loads) allowing 15 input data to be extracted from each image at a particular deflection.



**Figure 5.** Band of noise in the input data at a deflection of 0.4 mm.

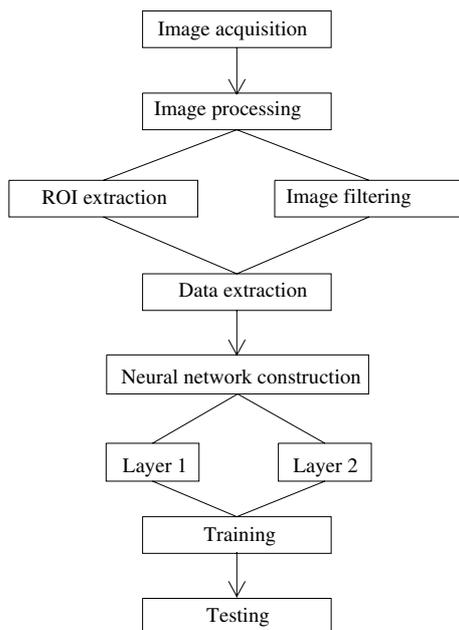


Figure 6. Flow chart of the implemented system.

Thus a total of 250 images was acquired. The deflection was incremented in steps of 0.04 mm from 0 to 1 mm using a digital micrometer. The neural network was trained with noisy data in order to make it robust against slight experimental variations that may occur during image acquisition, e.g. changes in image orientation and scaling due to loading, lighting, reflections or camera settings. Generally, one load value would provide only one input datum thus 250 input data sets from 250 images, but multi-input extraction gave 15 input sets from each image. This increased the training data from 250 to 3750 and improved the robustness of the neural network.

As there were 131 points in each input line it was important to reduce the dimensions of the input as it might lead to the dimensionality curse [8], which in turn might increase the number of training patterns required. Principal component analysis [16] was used to reduce the input dimensionality. The input data were normalized between  $-1$  and  $+1$  since neural networks are found to be more efficient if data are presented in this range [8, 17]. Another factor that can contribute to a change in input under the same load is the orientation of the optical elements. It is important that relative positioning between the two circular polarizers is always kept at  $90^\circ$  (crossed). The flow chart in figure 6 shows the step-by-step procedure for implementation of the neural network for the photoelastic fringe mapping system.

#### 4. The neural network

Figure 7 shows the structure of the neural network with two hidden layers used for this implementation. A feed forward back propagation network was constructed using MATLAB<sup>®</sup> Neural Network Toolbox [17] and training was done using the ‘Levenberg–Marquardt’ approach [8]. This offers the fastest training algorithm for networks of moderate size and has a memory reduction feature for use when the training set is large. The network trains by slowly modifying the weights

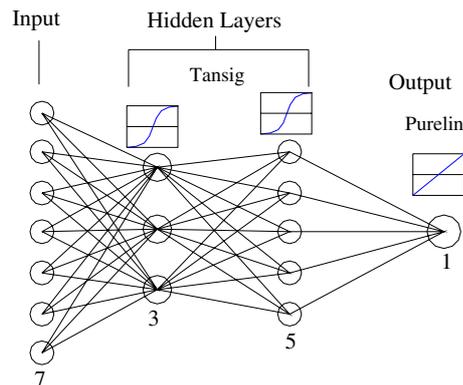


Figure 7. Structure of the neural network.

and learns from the input data sets. The input data need to be shuffled for efficient learning otherwise networks tend to memorize the last trained class of input data [7, 12]: a separate code was generated to shuffle the data.

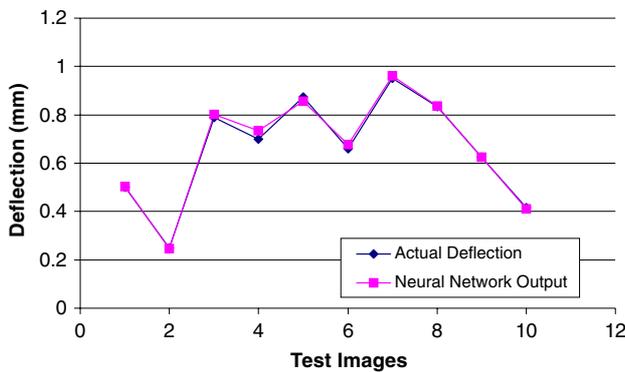
In the implementation of the network the performance goal set was based on a mean square error (MSE) function. Weights were initialized randomly and the training was performed with 50 different sets of initial weights. This is important since the error surface is unpredictable and randomly initializing the weights can help in finding the global minima of the error surface, otherwise the network may get stuck in local minima and will not generalize. The validation data (a subset extracted from total input data) were used to keep a check for generalization of the system and stop training in the case of over-fitting (memorizing). The network size was kept small, as larger networks tend to memorize instead of learning [17].

#### 5. Result and discussions

The results confirmed that ‘Levenberg–Marquardt’ was the most efficient training method for this application. The network performance was significantly affected by the input dimensionality and the size of the network. It has also been found that the network gave consistent and accurate results when two hidden layers were used as compared to one hidden layer; more than two hidden layers were found to be computationally overloading. However, in order to further determine the optimal input dimensionality and the size of the network, 49 different network configurations were tested at each input dimensionality. Table 1 shows part of the tested network configurations with different input dimensions. The error was computed for 24 different test points. By comparing the input dimensionality and the network size, the network {7 3 5} was selected as the best choice. This means the final network had input dimensionality of seven with two hidden layers, each with three and five neurons, respectively. The average percentage error was found to be 4.20 and it further dropped to 2.78 when the initial load was eliminated from the analysis: since there was very little fringe information available at such a low deflection of 0.04 mm when the network did not learn efficiently. The range of error was found to be between 4% and 9% for most of the networks as could be seen from table 1. When the error falls in the same range

**Table 1.** Part of the input dimensionality and network configuration test table.

Input dimensionality	Number of neurons–layer 1	Number of neurons–layer 2	% error 24 test points	% error 23 test points
5	4	3	5.56	4.18
6	3	3	4.97	4.68
7	3	5	<b>4.20</b>	<b>2.78</b>
9	3	6	5.98	4.15
10	3	4	6.67	2.93
11	3	4	5.54	3.26
12	3	4	9.08	3.49
13	3	3	6.08	3.90
14	5	5	5.23	5.08
15	3	5	7.51	3.92
16	3	3	6.53	5.08
17	3	8	5.97	5.02
18	4	3	5.70	5.67

**Figure 8.** Neural network output with actual deflection curve.**Table 2.** Regression analysis on the training and test data.

Data	Correlation coefficient	Slope
Training	0.993	0.985
Test	0.999	0.985

for different networks, it is better to use the network with fewer numbers of weights as they are more generalized. Once the optimal size of the network is achieved and the neural network is trained, the weights are saved and the test data can be fed without any further training. No further retraining of the network is required for this setup and it can analyse any unseen test data.

The test results for unseen data are plotted in figure 8. To check the robustness of the system, test data were collected separately in ‘manual mode’ of the camera as opposed to the ‘auto mode’ used during the training data collection. The results show that the system is capable of mapping output for the unseen data accurately. The plot between actual deflection and the deflection computed by the neural network is shown in figure 8.

Regression analysis on the training data and the test results are shown in table 2. Under the ideal conditions of perfect fit (neural network output should equal the desired target) slope and the correlation coefficient should be unity between the training and the test data.

It can be clearly seen from the plot (figure 8) and table 2 that the actual deflection and the neural network output are very close to each other. It can be further concluded that the noise

**Table 3.** Enhanced technique: a comparison.

Noroozi <i>et al</i> [9]	Our approach
Input requires calculations	Direct input from image
FEA package required	No additional package required
198 training sets	3750 training sets from 250 images
Noise generator required	Default noisy input (for robustness)
Test data: fringe order ( $N$ ) is required, thus knowledge of photoelasticity is essential.	Test data: ‘ $R$ ’ plane of digital images

due to experimental variations did not have a considerable effect on the system performance and it coped well within these variations. For efficient training at low load values, where there are insignificant changes in the fringe patterns, use of additional training sets will improve the performance.

Table 3 compares the system capabilities of our approach with the similar research conducted by Noroozi *et al* [9]. It can be clearly seen that our approach presents an enhanced technique in many aspects.

## 6. Conclusions

In this paper an enhanced technique for inverse mapping of photoelastic fringes has been proposed using neural networks. The neural network was trained from direct image data without requiring any mathematical computation. It has been shown that if sufficient and relevant inputs were provided, no prior calculations were needed on the training data and the results obtained were very close to the desired target. The output was also found to be stable against slight experimental changes such as changes in loading condition, lighting, reflections and camera settings. Large training sets were used and the PCA was implemented to reduce the dimensionality of the input whilst keeping the relevant information to make the network robust and generalized. The results have shown that in the absence of analytical solution for fringe mapping on to the applied load, use of neural networks could be a better approach for such inverse problems. The system could be further improved and generalized if training data could be combined from all three intensity planes and image acquisition be carried out with more than one light source. This technique is particularly suitable for repeated and generalized loading conditions, as may be required in various sensing

applications—this follows on from the repeated loading and image acquisition during the network training. Although in this paper only a simple deflection model has been analysed; nevertheless, the technique used the whole-field image data from the photoelastic model. The current implementation used only  $R$  intensity of the image, however, with further enhancement to include other colour planes and geometrical as well as statistical parameters of the image this technique could be potentially developed for whole-field image analysis of photoelastic fringes which may find use in various biomedical and HCI applications.

### Acknowledgment

This research was supported by a bursary awarded by the School of Design, Engineering & Computing, Bournemouth University.

### References

- [1] Ramesh K 2000 *Digital Photoelasticity: Advanced Techniques and Applications* (Berlin: Springer)
- [2] Ajovalasit A, Barone S and Petrucci G 1995 Towards RGB photoelasticity: full-field automated photoelasticity in white light *Exp. Mech.* **35** 193–200
- [3] Quiroga J, Botella A and Gomez-Pedrero J 2002 Improved method for isochromatic demodulation by RGB calibration *Appl. Opt.* **41** 3461–8
- [4] Ramesh K and Deshmukh S 1996 Three fringe photoelasticity—use of colour image processing hardware to automate ordering of isochromatics *Strain* **32** 79–86
- [5] Patterson E A 2002 Digital photoelasticity: principles, practice and potential *Strain* **38** 27–39
- [6] Patterson E A, Hobbs J and Greene R 2003 A novel instrument for transient photoelasticity *Exp. Mech.* **43** 403–9
- [7] Patterson D 1996 *Artificial Neural Networks: Theory and Applications* (Englewood Cliffs, NJ: Prentice Hall)
- [8] Bishop C 1995 *Neural Networks for Pattern Recognition* (Oxford: Oxford University Press)
- [9] Noroozi S, Amali R and Vinney J 2003 Inverse problem approach using photoelastic analysis and artificial neural networks in tandem *Strain* **40** 73–7
- [10] Brooks F, Ouh-Young M, Batter J and Kilpatrick P 1990 Project GROPE—haptic displays for scientific visualization *Comput. Graph.* **24** 177–85
- [11] Chung D 1998 Neural net based torque sensor using birefringent materials *Sensors Actuators A* **70** 243–9
- [12] Freeman J and Skapura D 1992 *Neural Networks: Algorithms, Applications and Programming Techniques* (Reading, MA: Addison-Wesley)
- [13] Cameron A, Danial R and Durrant-White H 1988 Touch and motion *Proc. IEEE Int. Conf. Robot. Autom.* 1-3 1062–7
- [14] Eghtedari F, Hopkins S and Pham D 1993 Model of a slip sensor *Proc. Int. Mech. Eng.* **207** 55–64
- [15] Vishay Measurement Groups 1992 Photoelastic materials *Bulletin* S-116-G
- [16] Bharath R and Drosen J 1994 Neural networks and statistical analysis *Neural Network Computing* (New York: McGraw-Hill)
- [17] Mathworks, MatLAB 2002