Multiple-Layer Neural Network Applied to Phase Gradient Recovery from Fringe Pattern

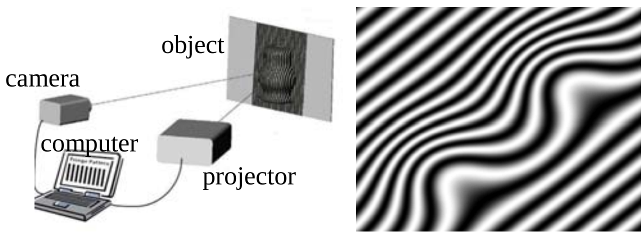
Weiguang Ding
School of Engineering Science
Simon Fraser University
wding@sfu.ca

Abstract

In kinesiology research, fringe projection profilometry is used to measure the surface shape and profile of ex-vivo beating animal heart. Deformation of projected fringe pattern will be caused by non-flat shape of surface and thus used to reconstruct the surface. In this course project, multiple-layer neural network (MLNN) is used to recover the gradient information of the surface as an intermediate step of surface reconstruction. The MLNN is trained by the fringe intensity pattern and phase gradient information extracted from synthetic data set. Various evaluation experiments are made on both parameters of MLNN and the properties of synthetic data set.

1 Introduction

Background: Surface reconstruction of ex-vivo beating animal heart is necessary in some kinesiology researches. Fringe projection profilometry provides a powerful tool to use non-contact method to measure the shape and profile of moving surface. In this system, collimated fringes (usually with sinusoidal intensity pattern) will be projected onto the target surface. Cameras would be placed from a different view angle. Deformation of fringe pattern will appear in the captured images and surface shape can be recovered from it. Fig. 1 illustrates the set-up of fringe projection profilometry system and gives an example of fringe.



(a) (b)

Figure 1: (a) Illustration of fringe projection profilometry (cited from [1]); (b) is an example of acquired fringe image

Related works: Fourier Transform Profilometry (FTP) [2] considers the problem as a modulation and demodulation process, which can be solved by analysis on frequency domain. Similar to FTP, methods including wavelet based method[3], phase-locked method[4] are also used in this application. All these methods take ‘global’ view of this problem and try to find the mapping of the whole fringe image and shape of the object.

In contrast to the above mentioned to other methods, multiple-layer neural network has been proposed to consider this problem ‘locally’ by Cuevas et al. [5]. This course project uses the idea from [5], while have differences in implementation.

Description of the problem: Fig. 1 (b) shows an example of acquired fringe image. Intensity of pixel (x, y) in this image can be expressed as Fourier series.

$$g(x, y) = r(x, y) \sum_{n=-\infty}^{\infty} A_n exp\,(jn(2\pi f_x x + 2\pi f_y y + \phi(x, y))) \tag{1}$$

Where the projected fringe pattern is represent by term $2\pi f_x x + 2\pi f_y y$, and the surface shape information is contained in term $\phi(x, y)$. To recover the surface, mapping from (x, y) to $\phi(x, y)$ is desired, and this need to be solved from the equation (1). The problem is that $\phi(x, y)$ is a 'global' property which does not only rely on the local fringe pattern information. Gradient of $\phi(x, y)$, however, can be determined without knowing information in pixels outside of the window. If the phase gradient can be acquired, the surface reconstruction can be done afterwards.

In this course project, finding the relationship between gradient of $\phi(x, y)$ and a local window at pixel (x, y) is considered to be a regression problem. Efforts on training multiple-layer neural network (MLNN) to build this mapping are made. Also, the algorithm is evaluated with various experiments.

2 Multiple-Layer Neural Network

The multiple-layer neural network (MLNN) is used to solve regression problems which are hard to find an explicit model, which is suitable for the mapping between local fringe pattern and phase gradient value. Fig. 2 illustrates the input and output of MLNN in this application.

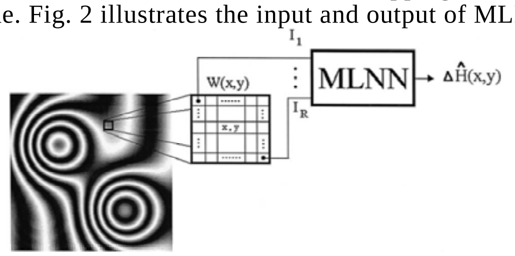


Figure 2: Input and output of MLNN (Fig. cited from [5])

The input of the MLNN is the intensity value of every pixel inside a local window. The output of the MLNN is the x and y direction phase gradient. For example, if the local windows size is chosen to be 5x5, the MLNN will have 25 inputs for each pixel on the fringe image. The 2 outputs of the MLNN is the x and y direction phase gradient at the corresponding pixel. In this application, a 2-layer MLNN is used. Parameters of the MLNN are discussed in experiment section.

3 Experiments

Synthetic data is used in these experiments because of unavailability of real data. Training data and test data are extracted from different surfaces with similar shape and are illustrated in Fig. 3. Variations in fringe direction, wavelength, noise, illumination nonuniformity (IN) and test data surface shape will be made in different specific experiments. If unspecified, the fringe image is clean (without noise and illumination nouniformity), has fixed direction and 20-pixel wavelength sinusoidal fringe on the image.

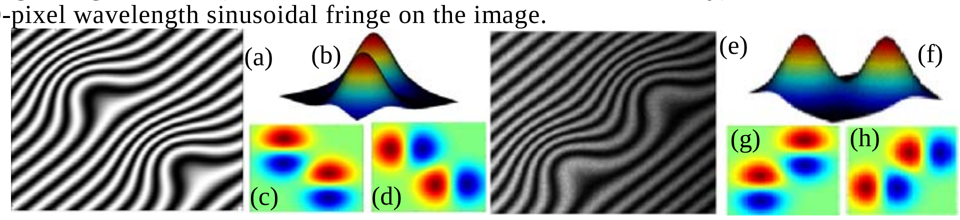


Figure 3: example of data used in MLNN training. (a) is the training fringe image; (b) is the training phase surface; (c) is the training phase gradient in vertical direction; (d) is the training phase gradient in horizontal direction; (e) is the test fringe image; (f) is the test phase surface; (g) is the test phase gradient in vertical direction; (h) is the test phase gradient in horizontal direction.

For the MLNN implementation, Netlab [6] package is used with slightly modification.

In the following experiments, both parameters of the neural network and properties of the image are considered. The MLNN method is also compared with the Fourier Transform Profilometry (FTP) method. Due to large amount of tunable parameters and properties, the following experiments are far from complete. Parameters and properties that have not been experimented will be briefly discussed.

3.1 Experiments on MLNN Parameters

3.1.1 Tunable parameters without experiments

Optimization method in following experiments is scaled conjugate gradient descent (SCG). However, determination of the optimal optimization method needs more comprehensive evaluations. The termination condition for the training process using SCG optimization is 4000 iterations over the whole training data. Activation function of hidden layer is set as the hyperbolic tangent function $\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$. Activation function of output layer is chosen as linear function, due to the regression problem [6].While logistic sigmoid function is possible for hidden layer, and sigmoid and softmax function are possible for output layer, the evaluation of their performance is left as a future work.

3.1.2 Experiments on different number of hidden neurons

5 fold cross validation is done on MLNN with different number of hidden neurons. Here, a 5x5 local window is fixed, which means that the number of input is 25. The error-iteration plots in Fig. 4 (a) and error-number of neurons plots in Fig. 4(b) shows that number of hidden neurons does NOT have a significant influence on the accuracy of the algorithm.

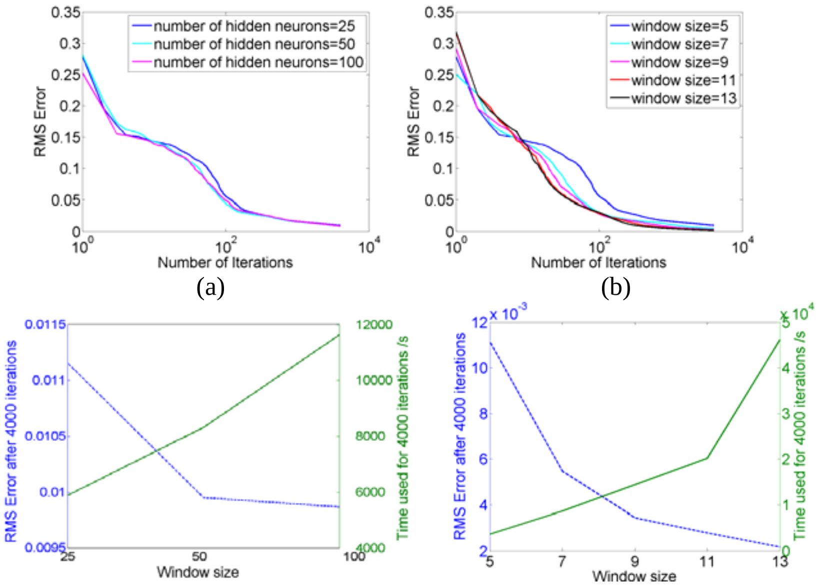


Figure 4: cross validation performance of MLNN. (a) is the error-iteration plots of trained MLNNs with different number of hidden neurons; (b) is the error-iteration plots of trained MLNNs with different number of hidden neurons; (c) and (d) the error comparisons after 4000 iterations

3.1.3 Experiments on different local window size

Choice of local window size is a tradeoff between information amount and 'locality'. 5 fold cross validation are also done for MLNN with local window size. Square local windows with size from 5 to 13 are tested and shown in Fig. 4 (b) and (d). In the 5 to 13 range, large window size will result in small errors. However, due to consideration of execution time, window size is fixed to be 5x5 in following experiments. MLNN with different window size will be tested with more experiment in the future.

3.2 Experiments on Data Properties

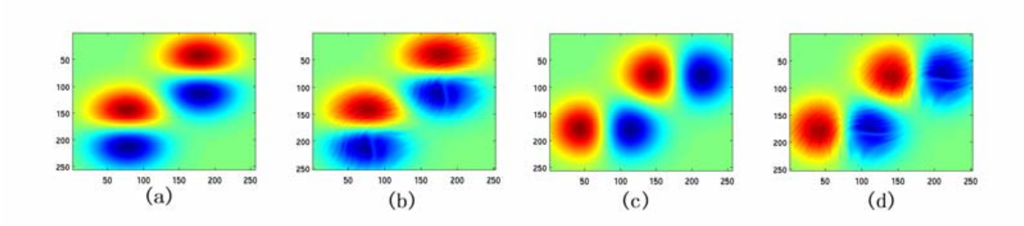


Figure 5: Learning result of clean test and training fringe image. (a) and (b) is the target vertical and horizontal phase gradient respectively; (b) and (d)is the vertical and horizontal phase gradient calculated by trained MLNN respectively

Fig. 5 shows the learning result using ‘clean’ fringe image as the training and test input. Clean means no illumination nonuniformity (IN) and no noise on the fringe image. The value of the MLNN output, the phase gradient, lies in [-0.26 0.26]. In the following experiments, root mean square (RMS) error is used. The RMS error corresponding to results shown in Fig. 5 is 0.012.

The data has many properties, which cannot be evaluated completely in this report. For example, influence of shape variance and fringe pattern other than sinusoidal fringe are not evaluated in this course project.

3.2.1 Experiments on noise and illumination

In this experiment, speckle noise and illumination nonuniformity (IN) are added to the fringe image to make it dirty. Fig. 2 (e) is an example of dirty fringe image.

First experiment use clean training data and dirty test data with various noise and IN levels. As shown in Fig. 6 (a), the algorithm is sensitive to both IN and noise, and especially very sensitive to noise. Fig. 6 (b) shows the errors of MLNN trained by ‘dirty’ training data which have same noise and IN level with test data. The algorithm is much less sensitive to IN, but still quite sensitive to noise. For comparison, the Fourier Transform Profilometry (FTP) method is also evaluated and shown in Fig. 6 (c). Different from MLNN method, FTP method is more sensitive to IN than noise.

Analysis of performance different between MLNN and FTP: In FTP method, after Fourier transform of the fringe image, if the frequency spectrum of useful information overlaps with illumination spectrum, large error would appear. MLNN is able to compensate for the constant illumination pattern, but couldn’t get accurate prediction at the presence of random noise. Therefore, if MLNN is used to process real image with considerable noise, proper denoising method need to be used.

3.2.2 Experiment on fringe direction and wavelength

Fringe direction and wavelength changes are also considered. In this experiment, input of test data, the fringe images are modified such that it has different fringe wavelength and fringe direction. The result is shown in Fig. 6 (d). It shows that large direction and wavelength difference will result in large error. However, this information is not enough to evaluate local performance. Comparison in finer scale will be done in the future.

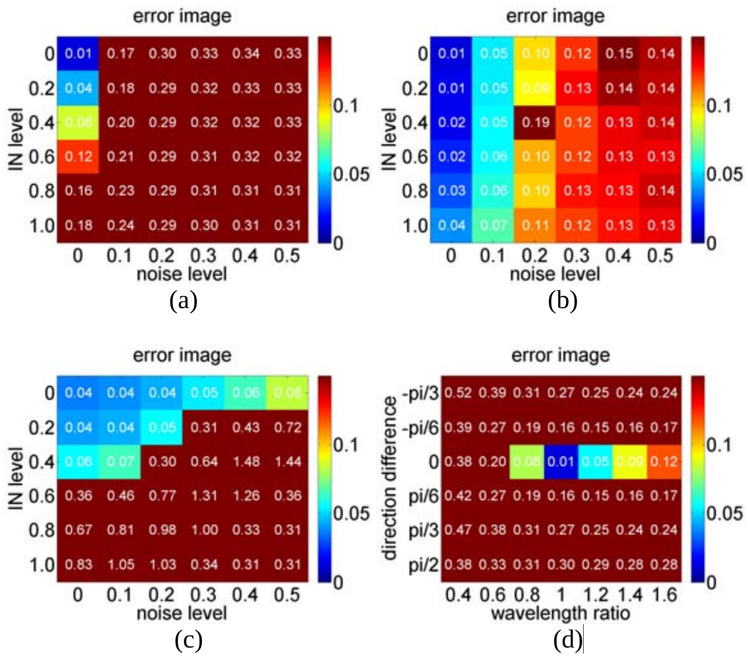


Figure 6: error images of experiments on data set properties. (a) experiment on clean training and dirty test data; (b) experiment on same level dirty training and test data; (c) experiment of FTP method on dirty test data; (d) experiment on data with different fringe direction and wavelength

4 Conclusion

In this course project, multiple-layer neural network (MLNN) is applied to recover phase gradient in fringe projection profilometry techniques. The mapping between local fringe pattern and phase gradient is found by training a MLNN. The inputs of the MLNN are the pixel values in the local window in fringe image and the outputs are the x and y phase direction gradients. Various tests on both parameters of the MLNN and properties of data are experimented. MLNN performance is not significant related to number of hidden neurons give fixed number of inputs. MLNN has higher accuracy with larger size local window for input under certain range, but due to time consideration, performance test of MLNN with large window size on dirty data set is left as a future work. MLNN is sensitive to noise and less sensitive to illumination nonuniformity (IN), which implies that proper denoising method need to be chosen in real application. If direction or wavelength difference in test data is large, the algorithm will have large error, but error with small changes directions and wavelength need to be included in future work.

References

[1] S. Gorthi and P. Rastogi, Fringe projection techniques: whither we are?. *Opt Laser Eng*, 48 (2010), pp. 133–140.

[2] M. Takeda, K. Mutoh, Fourier transform profilometry for the automatic measurement of 3-D object shapes., *Appl. Opt.* 22 (24) (1983) 3977–3982.

[3] J. Zhong, J. Weng, Spatial carrier-fringe pattern analysis by means of wavelet transform: Wavelet transform profilometry, *Appl. Opt.* 43 (26) (2004) 4993–4998.

[4] M. A. Gdeisat, D. R. Burton, M. J. Lalor, Fringe-pattern demodulation using an iterative linear digital phase locked loop algorithm, *Opt. Laser Eng.* 43 (7) (2005) 31–39.

[5] F.J. Cuevas, M. Servin and O.N. Stavrodis, et al. Multilayer neural network applied to phase and depth recovery from fringe patterns. *Opt. Commun.*, 181 (2000), pp. 239–259.

[6] Nabney I. *NETLAB: algorithms for pattern recognition.* London: Springer; 2001.