Optics EXPRESS

Rapid and robust two-dimensional phase unwrapping via deep learning

TENG ZHANG,[1,5] SHAOWEI JIANG,[2,5] ZIXIN ZHAO,[3,5] KRISHNA DIXIT,[2] XIAOFEI ZHOU,[1] JIA HOU,[1] YONGBING ZHANG,[4] AND CHENGGANG YAN[1,*]

[1] *School of Automation, Hangzhou Dianzi University, Hangzhou 310018, China*
[2] *Department of Biomedical Engineering, University of Connecticut, Storrs, CT 06269, USA*
[3] *School of Mechanical Engineering, Xi'an Jiaotong University, Xi'an 710049, China*
[4] *Graduate School at Shenzhen, Tsinghua University, Shenzhen 518055, China*
[5] *These authors contributed equally to this work*
[*] *cgyan@hdu.edu.cn*

Abstract: Two-dimensional phase unwrapping algorithms are widely used in optical metrology and measurements. The high noise from interference measurements, however, often leads to the failure of conventional phase unwrapping algorithms. In this paper, we propose a deep convolutional neural network (DCNN) based method to perform rapid and robust two-dimensional phase unwrapping. In our approach, we employ a DCNN architecture, DeepLabV3+, with noise suppression and strong feature representation capabilities. The employed DCNN is first used to perform semantic segmentation to obtain the segmentation result of the wrapped phase map. We then combine the wrapped phase map with the segmentation result to generate the unwrapped phase. We benchmarked our results by comparing them with well-established methods. The reported approach out-performed the conventional path-dependent and path-independent algorithms. We also tested the robustness of the reported approach using interference measurements from optical metrology setups. Our results, again, clearly out-performed the conventional phase unwrap algorithms. The reported approach may find applications in optical metrology and microscopy imaging.

1. Introduction

Phase is a critical parameter of electromagnetic waves. In fringe pattern analysis [1, 2], phase is often obtained using the arctangent function, which results in a phase map with a sawtooth shape in the range of [-π, π]. Hence the result is given modulo 2π and discontinuities in the phase distribution. Phase unwrapping is the process of removing these discontinuities to obtain the desired continuous phase profile [3, 4]. Phase unwrapping has been applied to many imaging modalities such as synthetic aperture radar imaging (SAR), magnetic resonance imaging (MRI), surface shape measurement, interferometry, and digital holography [5], to name a few. In practice, there are many factors that could affect phase unwrapping, such as under-sampling, noise, local shadows, and so on. Therefore, phase unwrapping continues to be a challenging problem. Many spatial methods have been proposed to resolve this issue, with the three main categories being path-dependent [3, 4, 6–8], path-independent [9–13] and methods based on neural networks [14–16].

For path-dependent methods, phase unwrapping is dependent upon the integration path. The most commonly used methods are the branch-cutting method [4], quality guided method [3, 6], mask cut algorithm [7], and minimum discontinuity [8]. These four methods are only suitable for dealing with the ideal wrapped phase. For the path independent methods, the least square based methods [9] minimize error i.e. the local derivatives of the unwrapped phase match the measured wrapped phase gradients as closely as possible. Most path-independent methods are

#365340 https://doi.org/10.1364/OE.27.023173
Journal © 2019 Received 22 Apr 2019; revised 1 Jun 2019; accepted 12 Jul 2019; published 30 Jul 2019

also compromised when dealing with excessively noisy phase maps. Nearly two decades ago, Schwartzkopf et al. [14] proposed a supervised feedforward multilayer perceptron neural network with five hidden layers to detect phase discontinuities for phase unwrapping. It is a one-pass pixel-parallel low-complexity method and mainly used for optical Doppler tomography (ODT) images. Last year, Dardikman et al. [15] utilized a DCNN to predict the unwrapped images with regression network in simulative cell phase maps. It can successfully unwrap samples consisting of steep spatial gradients. Recently, PhaseNet [16] which is based on SegNet [17] used an isotropic Laplacian filter to obtain the disjoined clusters. Then, the disjoined clusters were binarized and the whole cluster was assigned to a unique wrap-count (integer jump of 2π) by obtaining the mode of the wrap-count from the output of DCNN in that region. Last, a median filter was adopted to eliminate undesirable classification along the contours of clusters in the unwrapped phase map.

Recently, with the booming development of deep learning techniques, convolutional neural networks (CNN) has seen applications in many areas such as detection [18, 19], classification [20, 21], segmentation [17, 22–25] and so on. Among these applications, semantic segmentation, which classifies the visual input into semantically interpretable classes, has grown in popularity. In 2014, Long et al. [22] proposed a fully convolutional network (FCN) based method, in which the network can be trained in an end-to-end way and the output has the same resolution as the input image. It is the first work to deploy deep convolutional neural networks to image semantic segmentation. Successively, several FCN based works such as SegNet [17] and DeepLabV3+ [24] were also proposed to push forward the semantic segmentation. Phase unwrapping operates by splitting the wrapped phase map into different phase regions, and each region is an integer jump of 2π without distinguishing which kind of 2π jumps, which is very similar to semantic segmentation rather than instance segmentation [25]. Because semantic segmentation doesn't distinguish between different individuals of the same category, instance segmentation distinguishes different individuals of the same class. Furthermore, deep convolutional neural networks have strong capabilities in feature representation and also have an anti-noise property to a certain degree. Therefore, it is a natural extension of existing techniques to extend the use of semantic segmentation networks to phase unwrapping.

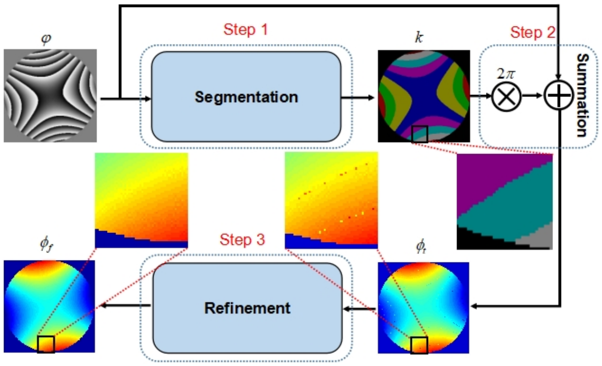

Fig. 1. The flowchart of the proposed phase unwrapping method.

In this paper, we propose a novel phase unwrapping method by using deep learning based semantic segmentation algorithm, as shown in Fig. 1. In our method, we employ the extended DeepLabV3+ network as the backbone, which consists of two parts including the encoder and the decoder. The input of the encoder, which provides rich semantic information, is a wrapped phase map. The decoder is used to recover the boundary information of the object, yielding a segmented phase area map. Then, we extend this backbone by adding a bypass between the encoder input and the decoder output. As a result, an intermediate result, i.e. a coarse unwrapped phase map, can be obtained by adding together the input of encoder and the output of decoder. Finally, to further improve the quality of the temporary result, a refinement step is employed to generate the final result.

Overall, our main contribution lies in the following two aspects:

1) We propose a novel two-dimensional phase unwrapping method using deep convolutional neural networks, which not only provides a coarse segmentation result but also shows a certain degree of anti-noise properties.

2) We tested our method with several well-established phase unwrapping methods on simulation data and real data with different shapes and resolutions, and the results demonstrate the effectiveness and superiority of our method.

2. Proposed method

Phase unwrapping could be accomplished by adding an integer multiple of 2π to the wrapped phase map. Generally, the phase unwrapping problem can be expressed as follows,

$$\phi(x, y) = \varphi(x, y) + 2\pi k(x, y), \tag{1}$$

where $\phi(x, y)$ is the unwrapped phase, $\varphi(x, y)$ is the wrapped phase, and $k(x, y)$ is the integer that needs to be solved for, which can be regarded as a pixel-wise classification problem. The overflow of our method is shown in Fig. 1, which consists of three steps including segmentation, summation and refinement. Here, we first employ the semantic segmentation network DeepLabV3+ to solve $k(x, y)$, in which we take the wrapped phase $\varphi(x, y)$ as input and give the $k(x, y)$ as output. Then, according to Eq. (1), we can obtain the temporary result ϕ_t. Lastly, a refinement step is deployed on the intermediate result, yielding the final result ϕ_f. We provide a detailed description for the proposed method below.

A. Segmentation

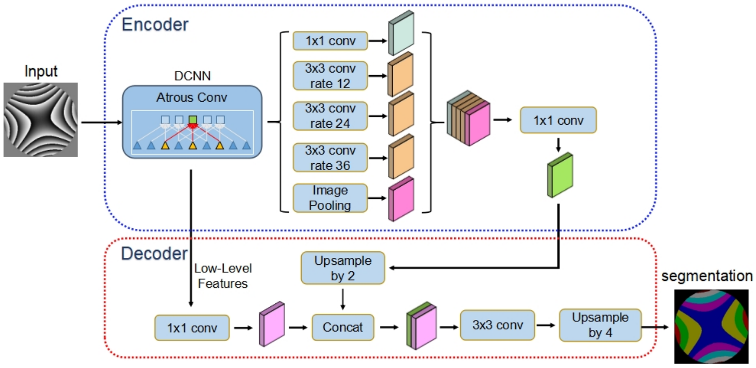

Fig. 2. The structure of DeepLabV3+.

First, we adopt the DeepLabV3+ network to obtain the unknown integer $k(x, y)$. As shown in Fig. 2, the DeepLabV3+ network, which consists of an encoder and a decoder, is widely used for semantic segmentation. In the encoder module, the DCNN is the modified aligned Xception [21, 26] network, as shown in Fig. 3, which can extract dense feature maps with higher semantic information. The modified aligned Xception consists of three parts: entry flow, middle flow, and exit flow. In our method, the wrapped phase maps first go through the entry flow, then through the middle flow which is repeated 16 times, and finally through the exit flow. The modified aligned Xception has batch normalization [27] and ReLU after all convolutional layers (including separable convolution and depthwise separable convolution). After the DCNN, there is an atrous spatial pyramid pooling (ASPP) by employing atrous convolution with different rates to get the multiple scales rich information. The rates are related to the size of input images and the ratio of input image spatial resolution to the DCNN output features resolution. Because ASPP extracts more features, and in order to balance the information of each scale and accelerate training, there is a 1 × 1 convolution to reduce the channel of features.

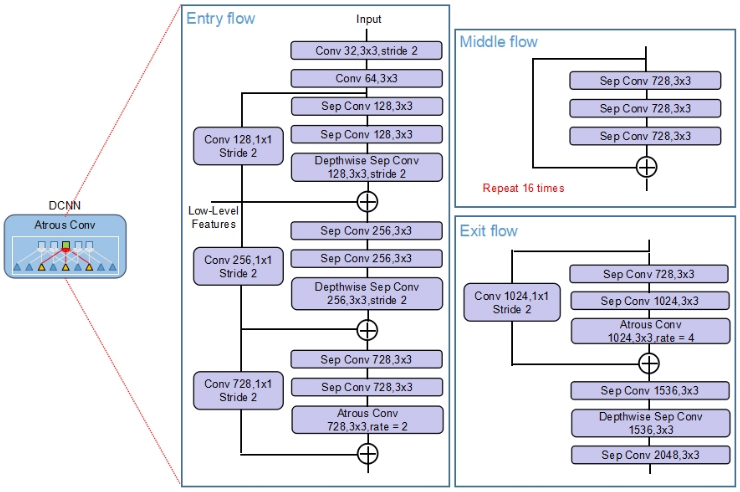

Fig. 3. The structure of modified aligned Xception.

The encoder features are first bilinearly upsampled by a factor of 2 and then concatenated with the low-level features from modified aligned Xception. Both the low-level features and the encoder features contain 256 channels, in order to make better use of higher semantic features extracted by the encoder, another 1 × 1 convolution is employed to reduce the number of channels. After the concatenation, we use a 3 × 3 kernel to refine the features followed by another simple bilinear upsampling by a factor of 4. We have tried to reduce the upsampling factor to reduce boundary errors. As a result, the error rate did not drop significantly but the training speed was much slower. Overall, the decoder module gradually recovers the spatial information and obtains sharper segmentation. We have selected the optimal parameters for phase unwrapping problem of DeepLabV3+ model by testing the simulation data set.

Therefore, step 1 of our method, i.e. segmentation, can be defined as follows:

$$k(x, y) = F_{\text{seg}}[\varphi(x, y)], \tag{2}$$

where F_{seg} denotes the DeepLabV3+ network.

B. Summation

Following the segmentation step, we can obtain the integer $k(x, y)$. Then, according to Eq. (1), we first multiply $k(x, y)$ by 2π, and then make a summation,

$$\phi_t(x, y) = \varphi(x, y) + 2\pi k(x, y), \tag{3}$$

where ϕ_t is the temporary result, as shown in Fig. 1.

C. Refinement

According to the enlarged local region of the temporary result ϕ_t shown in Fig. 1, we can see that a small number of pixels are mistakenly segmented in phase boundaries. The segmentation error at the boundaries refers to pixels, originally belonging to the left side of the boundary, that are mistakenly assigned to the right side of the boundary, which will lead to a jump of 2π or -2π on the final unwrapped phase map. Fortunately, the unwrapped phase map has continuity prior to the addition of noise. Therefore, this 2π jump can be detected and removed with a refinement operation, which can be expressed as follows:

$$\phi_f(x, y) = F_R\left[\phi_t(x, y)\right], \tag{4}$$

where ϕ_f is the final unwrapped phase map and F_R denotes the refinement operation (the step 3), as shown in Fig. 1. Notably, it can be seen from the enlarged view of ϕ_f and ϕ_t in Fig. 1, the refinement step further promotes the accuracy of unwrapped phase map. Meanwhile, to illustrate refinement operation, we provide an example shown in Fig. 4(a). We first calculate the difference between the pixel and the surrounding eight pixels, which is equivalent to a Laplace filter:

$$\Delta = \frac{a+b+c+d+f+g+h+i}{8} - e, \tag{5}$$

and then modify it as follows:

$$e = e + 2\pi \times \mathrm{Round}\left(\frac{\Delta}{2\pi}\right), \tag{6}$$

where Round is the function that takes the nearest integer.

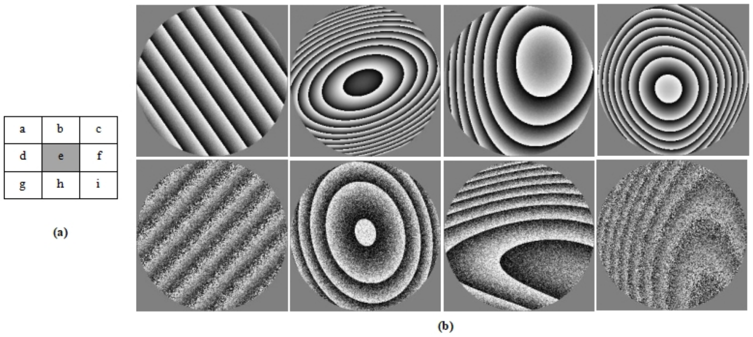

Fig. 4. (a) Illustration of the refinement operation. (b) Some wrapped phase maps in the training set.

3. Experimental results

We used the first ten Zernike polynomials to produce the training data and the test data (namely the simulation data) of wrapped phase maps with a pixel size of 256 × 256. The training set contains 25,000 images (some of them are shown in Fig. 4(b)) with different levels of Gaussian white noise (standard deviation is set from 0 to 1.5) and speckle noise. Firstly, we generate the unwrapped phase data using the first ten Zernike polynomials as shown in Eq. (7).

$$\phi(x, y) = \sum_{i=1}^{10} c_i Z_i(x, y), \tag{7}$$

where Z_i, c_i represent the i-th Zernike polynomial and its coefficient. Then we generate the corresponding wrapped phase data according to Eq. (8).

$$\varphi(x,y) = \text{ angle } (\exp(1\mathrm{i} \times \phi(x,y))). \tag{8}$$

After we generate the unwrapped and wrapped phase maps, we can compute the $k(x,y)$ as follows:

$$k(x,y) = \frac{\phi(x,y) - \varphi(x,y)}{2\pi}. \tag{9}$$

Rather than use the $k(x,y)$ as the ground truth of our network directly, we subtract the minimum value of $k(x,y)$ as the ground truth. This enables the network to learn the trend of relative phase changes. To ensure that the training data is very close to the experimental data as much as possible, different kinds of noise (additive and multiplicative noise) were added to the training data. The test set contains 1200 phase maps. To test the performance of our algorithm under the condition of different noises, we add Gaussian white noise with standard deviation from 0 to 2 to the test set. On the basis of these, we compared our method with four well-established methods including Herráez et al.'s method (SRFNP) [3,6], TIE method (TIE) and TIE iterative method (ITIE) [10], Zhao et al.'s method (RTIE) [11]. Among these methods, SRFNP is path dependent method, and TIE, ITIE, RTIE are path independent methods.

Furthermore, during training, we used the stochastic gradient descent (SGD) with momentum 0.9 for 200000 iterations, base learning rate 0.005, mini-batch size 8 and learning power 0.9. When testing, we employed our method on the simulation data and real data, respectively.

A. Experiments on simulation data

We first performed two experiments on simulation data with noise free and heavy noise (standard deviation equals 2.0) condition. Then the performance of our proposed method under different noise levels was analyzed. We have compared the results obtained by the proposed method with other well-established methods. We have also shown the computation cost of all methods.

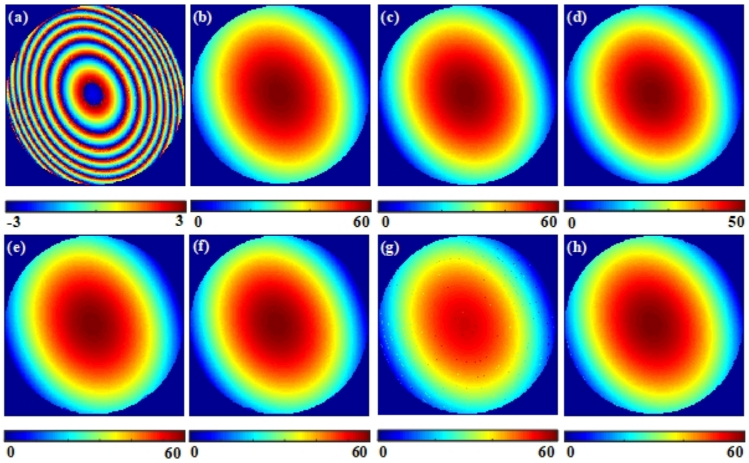

Fig. 5. Phase unwrapping on simulation data without noise. (a) Simulated wrapped phase map. (b) Ground truth. Unwrapped phase maps are generated by using SRFNP (c) with RMSE $3.5e^{\wedge-13}$, TIE (d) with RMSE 0.9067, ITIE (e) with RMSE 0.2096, RTIE (f) with RMSE 0.1627, Ours without refinement (g) with RMSE 0.5618, Ours with refinement (h) with RMSE 0.1410.

In Fig. 5, we give a comprehensive comparison for our method and the aforementioned four methods on simulation data without noise in terms of the root mean square error (RMSE). It can be seen that our method with the refinement step performs better than all methods, except the SRFNP. Meanwhile, it clearly demonstrates the effectiveness of the refinement step in our method.

Second, to demonstrate the superiority of our method over path-based and least-square based methods, we added Gaussian white noise (standard deviation is set to 2.0) on wrapped phase map, and perform a comparison among all the methods, as shown in Fig. 6. We can see that our method performs the best, which demonstrates the robustness of our method against high level noise.

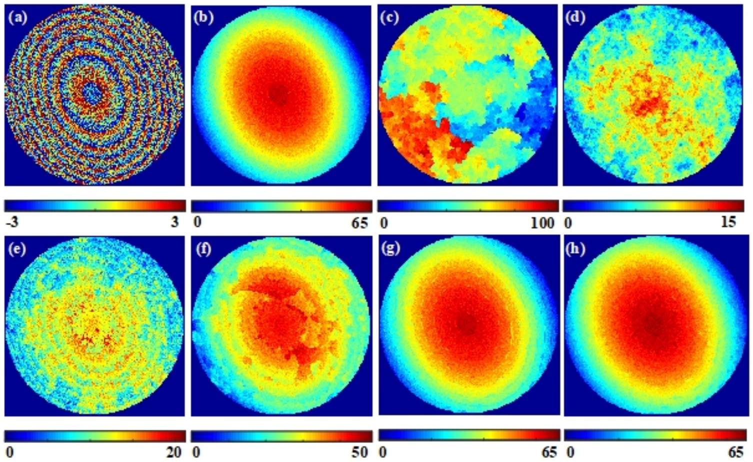

Fig. 6. Phase unwrapping on simulation data with Gaussian white noise (standard deviation is set to 2.0). (a) Simulated wrapped phase map. (b) Ground truth. Unwrapped phase maps are generated by using SRFNP (c) with RMSE 19.7591, TIE (d) with RMSE 10.9604, ITIE (e) with RMSE 10.4928, RTIE (f) with RMSE 6.9038, Ours without refinement (g) with RMSE 1.2325, Ours with refinement (h) with RMSE 1.5608.

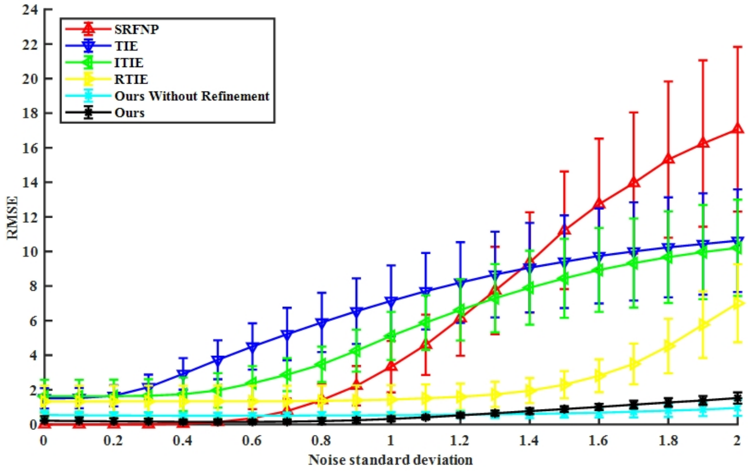

Fig. 7. The effects of noise with different standard deviations.

Lastly, Gaussian white noise with different standard deviations (from 0 to 2, 0 being no noise) is added to the test set. Figure 7 shows the mean RMSE values under the condition of different standard deviations for all methods. When the standard deviation of the noise level is larger than 0.5, our method with refinement and our method without refinement perform better than the other four methods. When the standard deviation of noise standard deviation is less than 0.5, SRFNP can perfectly unwrap the phase maps, which is consistent with the results of Fig. 5. However, it should be noted that our method also achieves the best performance except for SRFNP. From Fig. 7, we can see that when dealing with phase maps with low-level noise, our model with refinement is perfect. Otherwise, our method without refinement is ideal. The reason why the method without refinement performs slightly better than the method with refinement during heavy noise is that noise has a limited effect on our post-processing. We also find that the fluctuation of the RMSE value of our method is the smallest, which indicates the robustness of our method.

In regards to the computation cost, all of the methods were implemented on a PC with CPU E5-1620 v4 and GPU NVIDIA TITAN Xp, and the average processing time of the five methods is approximately 0.4608(SRFNP), 0.0344(TIE), 0.0712(ITIE), 2.5172(RTIE), and 0.0393(Ours) seconds, respectively. This shows that our method is faster than most traditional methods.

B. Experiments on real data

To evaluate the robustness of the proposed method, two experiments were carried out to produce the real data of wrapped phase maps. The first experiment was conducted to test an optical sphere using a Twyman-Green interferometer as shown in Fig. 8(a). By introducing a small defocus, a series of fringe patterns, as shown in Fig. 8(c), can be obtained with the help of a phase shifter (PZT) attached on the reference mirror. It can be seen that the fringe contains very little noise and high contrast. Another experiment was conducted to test the out-of-plane deformation of a rough planar surface using a speckle interferometer as shown in Fig. 8(b). The speckle fringe patterns, as shown in Fig. 8(d), can be obtained by subtracting the speckle patterns captured before and after loading to the rough surface. It can be seen that the speckle fringe patterns are corrupted by strong speckle/coherent noise, which heavily degrades the contrast of the fringe patterns.

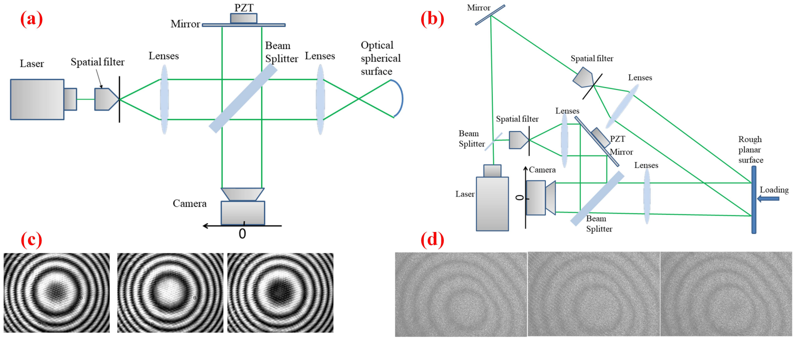

Fig. 8. Experimental setup. (a) Twyman-Green interferometer for testing an optical sphere. (b) Speckle interferometer for measuring deformation of a rough planar surface. (c) Phase shifted interferograms obtained by (a). (d) Phase shifted speckle fringe patterns obtained by (b).

To validate our method using real data, three phase maps, i.e., rectangular shape with low-level noise, circular shape with severe noise and rectangular rectangle shape with severe noise, as shown in Figs. 9(a), 10(a) and 11(a), were adopted to perform comparisons among all five methods. All the wrapped phase maps were obtained using the phase shifted fringe patterns with the help of the generalized principle component analysis-based method [28]. All results in this section are obtained using the proposed method with the refinement step.

The first one, as shown in Fig. 9(a), was obtained from the Twyman-Green interferometer with a pixel size of 300 × 400. Figs. 9(b)-9(f) show the results obtained using the five methods. Since the wrapped phase map was obtained with low-level noise, all methods can successfully unwrap the wrapped phase map. This experiment also shows that our network can handle phase maps whose pixel size is different from that of the training set (256 × 256).

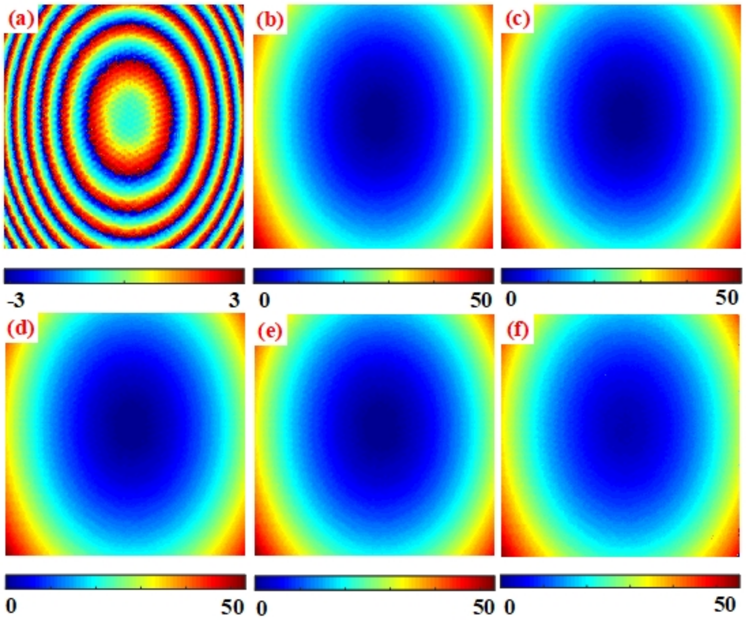

Fig. 9. Phase unwrapping on real data with low-level noise. (a) Wrapped phase map. Unwrapped phase maps are generated by using SRFNP (b), TIE (c), ITIE (d), RTIE (e), Ours (f).

The second form of real data was obtained by measuring the deformation of a rough planar surface with digital speckle interferometry. As can be seen from Fig. 4(b), the training images are circular, and the pixel size is 256 × 256. As a result, we firstly resize the original map to the pixel size 256 × 256 with a circular mask as shown in Fig. 10(a). Then five methods were used to unwrap the wrapped phase map and the corresponding results are shown in Figs. 10(b)-10(f) respectively. It can be seen that, due to severe noise, the first three methods fail to unwrap the phase map. The result of RTIE, as shown in Fig. 10(e), has a certain regional dislocation. In stark contrast, our method, as shown in Fig. 10(f), achieves the best performance.

Figure 11(a) shows the original wrapped phase data in rectangular profile with a pixel size of 1600 × 2400, which is obtained using the phase shifted speckle fringe patterns shown in Fig. 8(d). Although our network can deal with images of arbitrary scale, the unwrapping performance becomes worse as the scale of the wrapped phase map becomes much larger than 256. To achieve better performance for maps with large scale, we down sample the phase map to 256 × 256 before unwrapping for all methods. The corresponding unwrapped results are shown in Figs. 11(b)-11(f), respectively. The TIE and ITIE methods fail to unwrap. The result of SRFNP algorithm is not continuous enough, and the result of RTIE has some regional solving errors. In contrast, our method performs best.

Generally speaking, we can conclude that our method is suitable for real wrapped phase maps in circular and rectangular shape profiles with severe noise, and the corresponding results also demonstrate the effectiveness of our method.

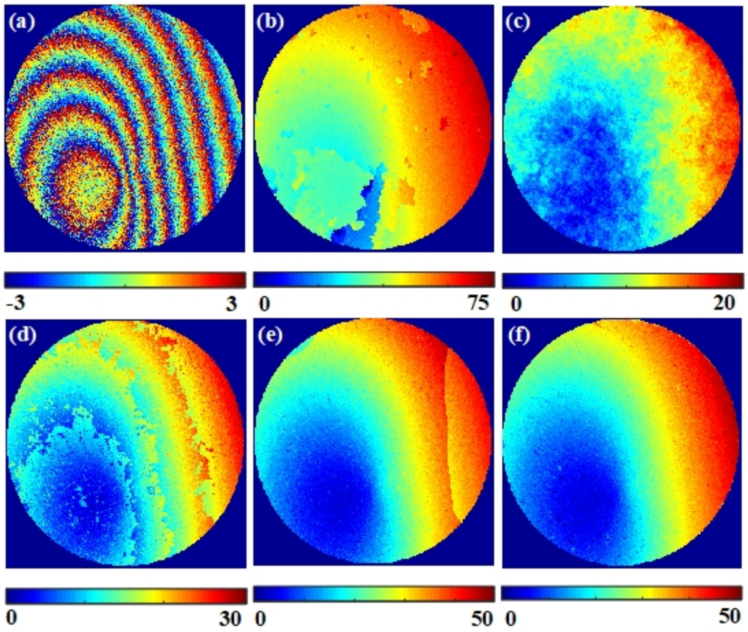

Fig. 10. Phase unwrapping on real data in circular shape with severe noise. (a) Wrapped phase map. Unwrapped phase maps are generated by using SRFNP (b), TIE (c), ITIE (d), RTIE (e), Ours (f).

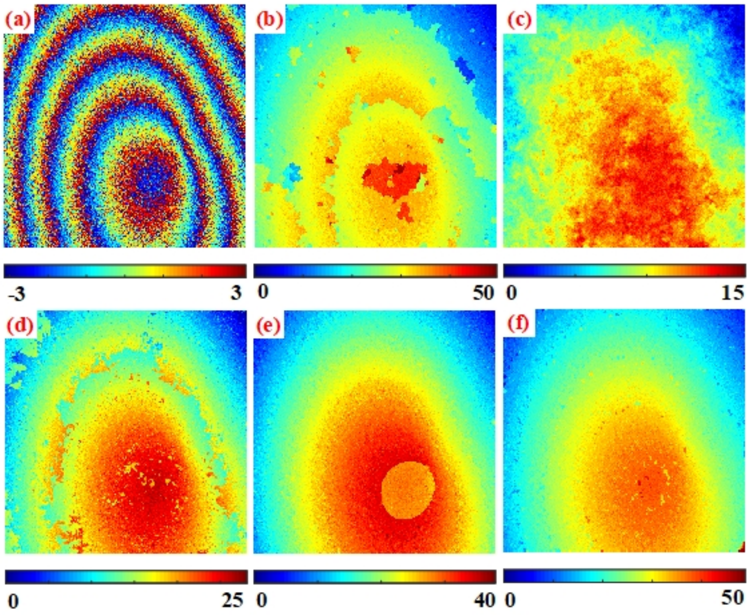

Fig. 11. Phase unwrapping on real data in rectangular shape with severe noise. (a) Wrapped phase map. Unwrapped phase maps are generated by using SRFNP (b), TIE (c), ITIE (d), RTIE (e), Ours (f).

4. Discussions and conclusion

It should be noted that down sampling to 256 × 256 is highly undesirable when the fringe density is relatively high because the information will be lost after down sampling. To make our proposed method applicable to the real phase maps with large pixel count and high fringe density, a stitching strategy or retraining with larger image size can be used. Considering the GPU memory and training speed of larger image size, we think image stitching to unwrap larger image size is a better solution instead of training images with large image size. As shown in Fig. 12, we use a simulation data for demonstration. We use a wrapped phase map with size 500 × 500 to unwrap via image stitching. Firstly, we split the whole wrapped phase map into four wrapped phase maps with a resolution of 256 × 256. Secondly, each sub-phase map is used as an input to the network separately and the corresponding segmentation results can be obtained. Then, all the sub-segmentation results can be stitched together with a suitable stitching algorithm. Finally, we can perform summation and refinement operations on the stitched image to get the final unwrapped phase map. The result in Fig. 12 shows that the stitching strategy is promising. However, further research is needed. How much overlap between the adjacent images is sufficient? What is the optimal overlap? All of these problems need further investigation. Recently, Feng et al. [2] proposed a method of using a convolutional neural network to perform fringe analysis. Hence phase retrieval and phase unwrapping with a network will also be a further research direction.

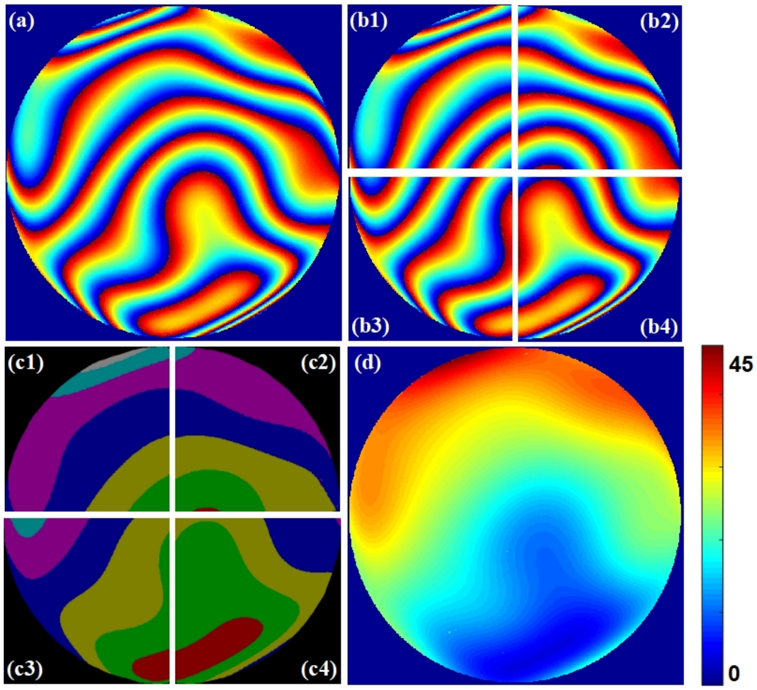

Fig. 12. Phase unwrapping via image stitching. (a) Original wrapped phase map with a resolution of 500 × 500. (b1) - (b4) Four sub-images with an overlap of 12 pixels. (c1) -(c4) Segmentation results of (b1) - (b4). (d) Final stitched unwrapping result.

In this paper, we present a DCNN based two-dimensional phase unwrapping method, which consists of three steps including segmentation, summation, and refinement. The key advantage of our method lies in the introduction of deep neural networks, which provide both the coarse segmentation result of the wrapped phase map and a degree of anti-noise capability. Extensive experiments were performed on simulated and real data, and the results demonstrate the effectiveness of our method. Notably, our method can obtain a satisfactory unwrapping result even under a severe noise condition.

Funding

National Key Research and Development Program of China (2017YFC0820604); National Natural Science Foundation of China (NSFC) (61671196,61525206,61701149,51705404); Zhejiang Province Nature Science Foundation of China (LR17F030006); China Scholarship Council Foundation (CSC) (201806285004); National Natural Science Major Foundation of Research Instrumentation of China (61427808); Key Foundation of China (61333009); National Key Basic Research Program of China (2012CB821204).

Acknowledgments

We thank the Smart Imaging Laboratory and Intelligent Information Processing Laboratory members for their useful discussion.

References

1. D. J. Bone, "Fourier fringe analysis: the two-dimensional phase unwrapping problem," Appl. Opt. **30**, 3627–3632 (1991).
2. S. Feng, C. Qian, G. Gu, T. Tao, Z. Liang, H. Yan, Y. Wei, and Z. Chao, "Fringe pattern analysis using deep learning," Adv. Photonics **1**(2), 025001 (2019).
3. M. A. Herráez, D. R. Burton, M. J. Lalor, and M. A. Gdeisat, "Fast two-dimensional phase-unwrapping algorithm based on sorting by reliability following a noncontinuous path," Appl. Opt. **41**, 7437–7444 (2002).
4. R. M. Goldstein, H. A. Zebker, and C. L. Werner, "Satellite radar interferometry: Two-dimensional phase unwrapping," Radio Sci. **23**, 713–720 (1988).
5. H. Xia, S. Montresor, R. Guo, J. Li, F. Yan, H. Cheng, and P. Picart, "Phase calibration unwrapping algorithm for phase data corrupted by strong decorrelation speckle noise," Opt. Express **24**, 28713–28730 (2016).
6. M. F. Kasim, "Fast 2D phase unwrapping implementation in MATLAB," https://github.com/mfkasim91/unwrap_phase/.
7. C. Prati, M. Giani, and N. Leuratti, "SAR interferometry: A 2-D phase unwrapping technique based on phase and absolute values informations," in *Proceedings of IEEE Conference on International Geoscience and Remote Sensing Symposium,* (IEEE, 1990), pp. 2043–2046.
8. T. J. Flynn, "Two-dimensional phase unwrapping with minimum weighted discontinuity," JOSA A **14**, 2692–2701 (1997).
9. H. Takajo and T. Takahashi, "Least-squares phase estimation from the phase difference," JOSA A **5**, 416–425 (1988).
10. J. Martinez-Carranza, K. Falaggis, and T. Kozacki, "Fast and accurate phase-unwrapping algorithm based on the transport of intensity equation," Appl. Opt. **56**, 7079–7088 (2017).
11. Z. Zhao, H. Zhang, Z. Xiao, H. Du, Y. Zhuang, C. Fan, and H. Zhao, "Robust 2D phase unwrapping algorithm based on the transport of intensity equation," Meas. Sci. Technol. **30**, 015201 (2018).
12. C. Zuo, "Connections between transport of intensity equation and two-dimensional phase unwrapping," arXiv preprint arXiv:1704.03950 (2017).
13. J. Arines, "Least-squares modal estimation of wrapped phases: application to phase unwrapping," Appl. Opt. **42**, 3373–3378 (2003).
14. W. Schwartzkopf, T. E. Milner, J. Ghosh, B. L. Evans, and A. C. Bovik, "Two-dimensional phase unwrapping using neural networks," in *Proceedings of IEEE Conference on Image Analysis and Interpretation,* (IEEE, 2000), pp. 274–277.
15. G. Dardikman and N. T. Shaked, "Phase unwrapping using residual neural networks," in *Computational Optical Sensing and Imaging,* (Optical Society of America, 2018), pp. CW3B–5.
16. G. E. Spoorthi, S. Gorthi, and R. K. S. S. Gorthi, "Phasenet: A deep convolutional neural network for two-dimensional phase unwrapping," IEEE Signal Process. Lett. **26**, 54–58 (2019).
17. V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," IEEE Transactions on Pattern Analysis Mach. Intell. **39**, 2481–2495 (2017).
18. S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in Neural Information Processing Systems,* (IEEE, 2015), pp. 91–99.
19. W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *Proceedings of the European Conference on Computer Vision,* (Springer, 2016), pp. 21–37.
20. K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," arXiv preprint arXiv:1409.1556 (2014).

21. F. Chollet, “Xception: Deep learning with depthwise separable convolutions,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition,* (IEEE, 2017), pp. 1251–1258.
22. J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional networks for semantic segmentation,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition,* (IEEE, 2015), pp. 3431–3440.
23. O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” in *Proceedings of the International Conference on Medical Image Computing and Computer-assisted Intervention,* (Springer, 2015), pp. 234–241.
24. L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, “Encoder-decoder with atrous separable convolution for semantic image segmentation,” in *Proceedings of the European Conference on Computer Vision,* (Springer, 2018), pp. 801–818.
25. K. He, G. Gkioxari, P. Dollár, and R. Girshick, “Mask r-cnn,” in *Proceedings of the IEEE International Conference on Computer Vision,* (IEEE, 2017), pp. 2961–2969.
26. J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei, “Deformable convolutional networks,” in *Proceedings of the IEEE International Conference on Computer Vision,* (IEEE, 2017), pp. 764–773.
27. S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” arXiv preprint arXiv:1502.03167 (2015).
28. J. Vargas, C. Sorzano, J. Estrada, and J. Carazo, “Generalization of the principal component analysis algorithm for interferometry,” Opt. Commun. **286**, 130–134 (2013).