Deep Learning Based Period Order Detection in Fringe Projection Profilometry

Budianto, Y.H. Chan and Daniel P.K. Lun
Department of Electronic and Information Engineering
The Hong Kong Polytechnic University, Hong Kong
budianto@ieee.org, enpklun@polyu.edu.hk, enyhchan@polyu.edu.hk

Abstract**—** Fringe projection profilometry (FPP) is a popular optical 3-dimensional (3D) scanning method which can obtain an object's 3D model at low cost, and achieve high resolution, fast speed and full-field measurements. Traditional FPP methods suffer from the ambiguity problem that only the wrapped phase information can be measured while the true phase information is required to obtain the 3D model of the scene. Various phase unwrapping methods were suggested to recover the wrapped phase in FPP methods. However, most of them will fail when the captured fringe images contain complex structures such as having discontinuities due to sudden jumps in object's height profile. To solve this problem, we propose in this paper to embed the fringe pattern with a set of textural patterns to encode the period order of the true phase information. During the offline phase, a convolutional neural network (CNN) is trained to learn a set of filters that will be activated when they see the code patterns. When the encoded fringe image is captured, the modified morphological component analysis is first performed to extract the code pattern. It is then decoded by the trained CNN to estimate the K-map, which contains the period order of the true phase information. Experimental results show that the proposed method can measure the 3D profile of objects with abrupt jumps in height profile, where the conventional phase unwrapping algorithms often fail to perform. It also has a much higher computational efficiency due to the effective utilization of GPU by CNN.

I. Introduction

The fringe projection profilometry (FPP) has been widely used as a non-invasive three-dimensional (3D) scanning method. It allows a fast, high resolution, and full-field measurement of the 3D model of objects. In a typical setup of FPP, a projector projects a sinusoidal fringe pattern onto the target object and a camera captures the deformed light pattern due to the object's height profile. By analyzing the displacement of the fringe pattern on the object surface, the height profile of the object can be measured.

There are two major classes of patterns used in FPP: aperiodic and periodic fringe patterns. The former requires the system to project a code pattern or a set of special code patterns unto the target object, such as the Gray code pattern and De Bruijn pattern. Although this kind of approaches can decode the object's 3D information directly by absolute codification, it is susceptible to the interference from the global illumination.

This paper focuses on the periodic FPP which offers a salient advantage of being resistant to the global illumination. By employing repeated fringe pattern (periodic sinusoid), the influence of distortion from global illumination and object's texture can be mitigated. However only wrapped phase information can be obtained from any conventional FPP method. Therefore, additional phase unwrapping procedure needs to be performed to obtain the true phase required to reconstruct the 3D model of the object. By assuming that the Itoh smoothness condition [2] is satisfied, i.e., the true phase difference between two neighboring pixels is equal or less than π, conventional phase unwrapping methods simply integrate the phase differences of the wrapped phase to obtain the true phase. Unfortunately, this assumption does not always hold good when there are occlusions or sudden jumps in the object's height profile. They lead to the situation that some fringes are missing from the camera's view; the phase difference between neighboring pixels can thus be greater than π. Hence any traditional phase unwrapping algorithm will fail in integrating the phase differences of the wrapped phase.

Most recently, many methods were suggested to extend this phase unwrapping principle by using additional period order information, i.e., the number of 2π jumps in the phase angles that is hidden in the wrapped phase data. When the period order information is known, the true phase can be easily obtained. This period order information can be detected using multiple cameras [3, 4] or various coded patterns such as Gray code patterns [5] [6], phase code patterns [7], and temporal phase-stepped patterns [8]. However, these approaches require additional fringe projections which slows down the acquisition process and thus increases the processing time to reconstruct the 3D model.

Alternatively, to avoid additional fringe pattern projection, recent approaches embed the period order information into the fringe patterns. To encode the period order into the fringe pattern, the approaches uses patterns of multiple wavelength or multiple frequencies, multiple colors, random patterns, or structural markers, etc. Theoretically, no additional fringe patterns are projected onto the scene. However, some approaches [9-11] can only be used for fringe images of simple scenes, e.g. a scene contains only a single simple object. Furthermore, the accuracy of the estimated period order in these approaches is often low since the embedded code patterns can introduce additional distortions to the fringe image.

Recently we proposed a novel approach which embeds to the fringe pattern some code patterns that indicate the true phase information [1]. More specifically, for each period of the fringe pattern, a unique codeword (period order number) is assigned

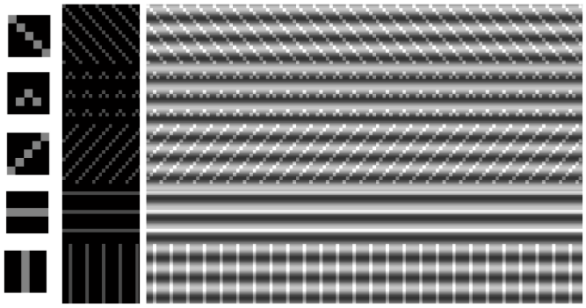


Fig. 1. 5 × 5 pixel binary textons (first column); the code pattern generated by the corresponding texton (second column); and the encoded fringe pattern with $N = 3$ (third column).

to a set of pixels and forms a code pattern. The fringe pattern is projected to the object and the captured fringe image is processed by a modified morphological component analysis (MMCA) method and the period order detection using discriminative dictionary learning method. The first is essential to separate the fringe pattern and code patterns and the latter is for the decoding of the coding patterns. Since the decoding process can be considered as classification and/or segmentation problem, the machine learning based approaches have a great potential application in FPP. In our previous work, we have employed dictionary learning approach in FPP method to assist a multi-level quality guided phase unwrapping algorithm to reconstruct the 3D model of the object [1]. However this approach is inefficient and less accurate due to the patch based operation. The recent trend in machine learning approach is deep learning [12] which has been successfully used for image classification [13, 14] and segmentation [15, 16]. To our knowledge, this approach has not been used in FPP before and more importantly it provides superior performance by integrating the feature extraction within the training process. While the performance of dictionary learning is in general satisfactory, further improvement can be achieved by a more comprehensive deep learning procedure.

In this paper, we introduce a new way of decoding the coded pattern. This new strategy is based on the convolutional neural network (CNN) which is efficient and provides more accurate pixelwise prediction due to the massive training process. The proposed CNN can be considered as a simplified version of U-Net architecture [15]. It has less dyadic scale decomposition layers and no additional skip connection is added to the network. It also works with few training patches and can decode the input coded pattern of any size as in [16]. More importantly, it yields a higher accuracy than our previous discriminative dictionary learning method.

This paper is organized as follows. We first present in Section II an overview of the FPP method. The proposed period order estimation using the CNN is explained in Section III. Finally, the experimental results and conclusions are presented in Section IV and V, respectively.

II. Proposed FPP Framework

For the proposed FPP method, we employ the three-step phase shifting profilometry (PSP) method to reconstruct the 3D model of the object from the captured fringe images. The three-step PSP method requires to project to the object a set of three fringe patterns with constant phase offset of $2\pi/3$ regardless of the starting phase. They are projected unto the object sequentially and captured by a camera. Given $n = \{0,1,2\}$, the three captured images can be represented mathematically as,

$$I_n(x,y) = a(x,y) + b(x,y)\cos\left(\phi(x,y) - \frac{n2\pi}{3}\right), \quad (1)$$

where $I_{n\in\{0,1,2\}}$ are the three phase shifted fringe images; a is the bias component; b is the amplitude of the sinusoid; and ϕ is the phase that carries the information of interest. In (1), the three fringes have constant phase shift of $0, \frac{2\pi}{3}$, and $\frac{4\pi}{3}$ respectively. In this case, the solutions for the phase information ϕ in (1) can be obtained from the fringe images as,

$$\hat{\phi}(x,y) = \tan^{-1}\left[\frac{\sqrt{3}(I_1(x,y) - I_2(x,y))}{2I_0(x,y) - I_1(x,y) - I_2(x,y)}\right]. \quad (2)$$

$\hat{\phi}$ thus obtained from (2) is bounded from $-\pi$ to π due to the tangent function. So the next step in the PSP process is to remove such 2π discontinuities. As mentioned, the basics of the unwrapping process as described by Itoh in [2] is to examine the phase differences between two neighboring pixels. In principle the phase unwrapping can be achieved by adding a multiple of 2π to the wrapped phase. It can be written as,

$$\phi(x,y) = \hat{\phi}(x,y) + k(x,y)2\pi. \quad (3)$$

where k is an integer step function indicating the multiple of 2π. In (3), k is the so-called K-Map which is unknown. In this paper, our goal is to accurately determine all k-values in the K-Map for all $\hat{\phi}$. In [1, 10, 17], the estimated $k(x,y)$ is used to assist the phase unwrapping algorithm. Additional voting algorithm is employed to determine the consistency of the estimated absolute phase. On the other hand, our proposed method estimates the K-map (all k-values) by performing pixelwise segmentation as in [18]. To get an accurate pixelwise K-map, an additional simple refinement method is performed. Details of this decoding stage will be explained in the next section.

Recall our previous work in [1, 17], the key idea of the proposed method is to encode the period order k in (3) with some unique textural patterns and embed them to the fringe pattern. Thus, the captured fringe image embedded with the textural code patterns can be formulated as,

$$X = X_1 + X_2, \quad (4)$$

where X_1 denotes the sinusoidal fringe pattern, i.e, the second term in (1), and X_2 denotes the code patterns that encodes the k-value. It is defined by,

$$X_2 = \Theta\{K(\phi)\}; \quad K(\phi) = \left\lfloor\frac{\phi + \pi}{N2\pi} - \pi\right\rfloor, \quad (5)$$

where $\lfloor x \rfloor$ is the floor function that gives the closest integer number smaller than x; Θ is an encoding function which assigns a textual pattern for each N consecutive 2π regions

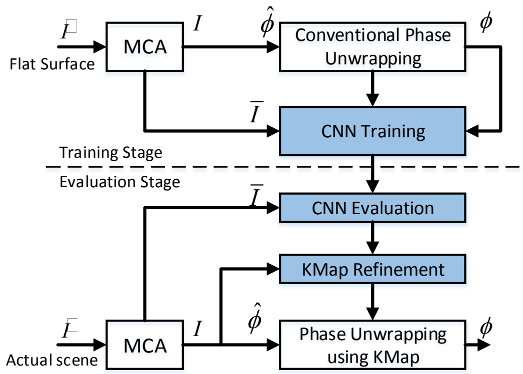


Fig. 2. Proposed FPP framework

having the same k-value. For instance, for $N = 1$, the k-value is unique for each 2π region. Hence every 2π region will have a different textual pattern from the neighboring ones. In this paper, we construct the textural pattern by concatenating textons and embedding into three 2π regions, thus $N = 3$. An example of the encoded fringe pattern image is depicted in Fig. 1.

The overall framework of the proposed FPP method is illustrated in Fig. 2. Both in the offline and online stages, the MMCA procedure is performed to separate the fringe pattern and codes patterns. We refer readers to our previous work in [1, 17] for the detailed information about the MMCA procedure. In this paper, we focus on the period order or K-map estimation. In the offline stage, the pixelwise k-values and the code patterns are known. A supervised CNN can be trained to learn a set of filters that will be activated when they see the code patterns. In the online stage, the trained CNN is used to decode K-map. The whole procedure consists of two steps: estimation of the K-Map regions using the proposed CNN and pixelwise K-Map refinement. They will be introduced in the next section.

III. CNN BASED PHASE UNWRAPPING

A. Network Architecture

In this section, we present the proposed CNN based phase unwrapping method. The network architecture of the proposed CNN is designed based on U-Net [15]. It is for segmenting the detected code patterns into different regions according to their representing k-values. It is similar to the segmentation problem in [18]. An example of the CNN output segmentation map is shown in Fig. 6 (second column). Recall that the embedded code patterns are specifically designed of having highly repetitive structure as shown in Fig. 1. They are also periodic and its local 'order' is repeated over a small region. By taking advantage of these properties, we propose a simplified version of the U-Net network as illustrated in Fig. 3. Specifically, we reduce the number of contracting (red arrows) and expansive path (green arrows) and remove the skip line connections. It is because the contracting feature map and the up-sampling map are not directly correlated as in the neuronal structure segmentation problem in [15]. In the figure, the basic block of this network is a 3x3 convolution (Conv) followed by batch normalization (BN) and a ReLU (blue box). All basic blocks have the same numbers of input and output except the first layer whose input is a single coded pattern image.

In the contracting step, the 2-dimensional (2D) max pool with size 2x2 for dyadic scale decomposition is employed similar to that in the multiresolution wavelet decomposition. Each contracting path and expansive path consists of a four-consecutive basic block which acts as a multichannel filter. To maintain the size of the resulting segmented map, the padding process is performed at each convolutional procedure. Hence, no additional extrapolation is required as in U-Net. In the final step, a 3x3 convolutional layer is added to produce the final segmentation map (orange arrow) which indicates the pixelwise k-value of the scene.

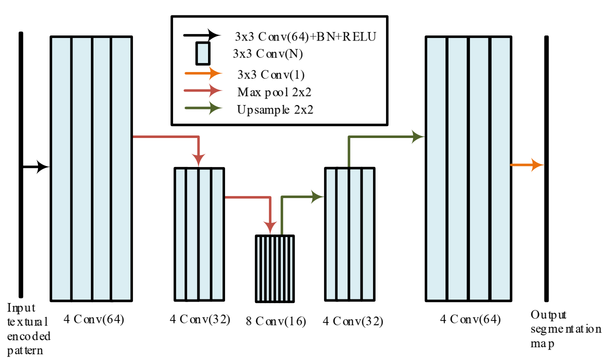


Fig. 3. Proposed CNN architecture

B. Training Stage

To train the network, we employ the Adam algorithm of the PyTorch framework [19]. Since all convolutions are padded, the output segmentation map is of the same size as the input. One of the difficulties of training CNNs is to collect sufficient number of training samples. As it is time consuming to set up an FPP system to obtain the fringe image and the ground truth of the code patterns, it will take a long time to collect sufficient training samples. To have more training data, we favor small patches. The input code pattern patches and their

Table 1. % of segmentation correctness of the proposed CNN trained with various patch sizes (16, 32, and 64).

Patch size	16 × 16	32 × 32	64 × 64
% Correctness	24.51	91.49	93.28

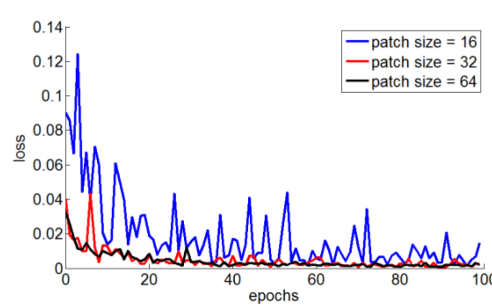


Fig. 4. Convergence rate of the proposed CNN

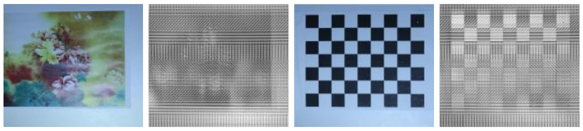
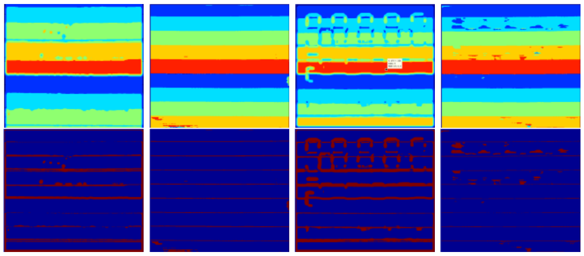


Fig. 5. Planar surfaces with vivid textures (1st and 3rd columns) used for validation and their coded pattern images (2nd and 4th columns) obtained by the modified MCA [1].



86.08% 94.91% 76.6% 95.02%

Fig. 6. The segmentation results. (1st and 3rd column) LC-KSVD segmentation in [1]; (2nd and 4th column) proposed CNN decoding. The first row is the decoding result and the second row is the error map against the ground truth.

corresponding segmentation map patches are used for the training. To generate more training data, we employ the data augmentation method by slightly modifying the image patches that let them resemble the real image patches obtained in typical FPP system. The three types of modification are: 1) performing affine transformation to accommodate deformation of fringe due to object's shape; 2) introducing two additional artifacts, namely additive Gaussian noise and Gaussian blur to accommodate artifacts due to lens' distortion or the medium of transmission; and 3) adding and multiplying augmentation to accommodate various changes in intensity due to object's texture.

It is known that the receptive field size of neural networks correlates with the effective patch size to capture more context information for segmentation. Meanwhile, we prefer smaller patch size since it is not only simple but also gives more detailed segmentation on the boundary in particular. By fixing the learning rate, we determine the effective patch size by comparing the loss of three patch sizes, i.e., 16, 32, and 64 (see Fig. 4). As illustrated in the figure, we can observe that the larger patch size can give faster and more stable convergence (black line).

Table 1 shows a comparison of the segmentation correctness of the proposed CNN when training with various patch sizes. As shown in the table, the patch size of 16x16 can give only 24.51% correct segmentation while the larger patch size, i.e., 32x32 and 64x64, can give more than 91% correct segmentation. For this reason, a patch size of 64x64 is selected when training the proposed CNN.

C. *K-map estimation and Refinement*

Although the output segmentation map given by the CNN is a good estimation of the regions of different code patterns, it still has not reached the pixelwise accuracy as required to generate the K-map. To further refine the CNN output segmentation map, we use the local information of the segmented regions to estimate the K-map as well as to improve the misclassified regions. It consists of two steps: region refinement and k-value estimation.

Table 2. Comparison in terms of computation time of the proposed CNN and LC-KSVD [1]

Method	**LC-KSVD**			**Proposed method**
Amount of overlapping of the 16x16 patches	Nil	8x8 pixels	12x12 pixels	
Computation time (seconds)	6.53	27.00	131.60	0.3283

Table 3. Comparison in terms of SNR of the phase error for planar surface

Number of periods	**PSP + Goldstein**	**PSP-Speckle**	**Proposed method**
Shiny pattern	21.73	16.58	26.55
Flower pattern	20.53	20.88	23.12
Checkered pattern	16.96	16.72	23.24

For FPP, the boundary of the region can be determined by analyzing the discontinuity of the wrapped phase information as in [10]. Note that discontinuity is caused either by the 2π discontinuities due to the fringe periodicity, which occurs only in the direction perpendicular to the fringe orientation, or by the abrupt surface change or occlusion, which can occur in any direction. By detecting these discontinuities, we can refine the region boundaries.

After refining the region boundaries, we can estimate the k-value of each region. Suppose there are N_C regions, $\{R_m\}_{m=1,\dots,N_C}$, the k-value for each region R_m can be determined by,

$$K(R_m) = \Theta^{-1}\{X_2(R_m)\}, \quad (6)$$

where Θ^{-1} is the decoding function for determining the k-value of a region. It is defined as follows:

$$\Theta^{-1}(X) = Refinement\left(V\left(CNN(X)\right)\right) \quad (7)$$

where $V(\cdot)$ is the voting function to determine the most popular label within the region R_m; $Refinement$ is a function to ensure that the estimated k-value is valid; and $CNN(X)$ is the output of the proposed CNN for a given code pattern. More specifically, the refinement function is to ensure that the order of k-values are valid. For instance, if $N = 1$, the refinement function is to ensure that $k_{R_i} < k_{R_{i+1}}$. On the other hand, if $N = 2$, $\left\lfloor \frac{k_{R_i}}{3} \right\rfloor = \left\lfloor \frac{k_{R_{i+1}}}{3} \right\rfloor < \left\lfloor \frac{k_{R_{i+2}}}{3} \right\rfloor = \left\lfloor \frac{k_{R_{i+3}}}{3} \right\rfloor$ must be hold.

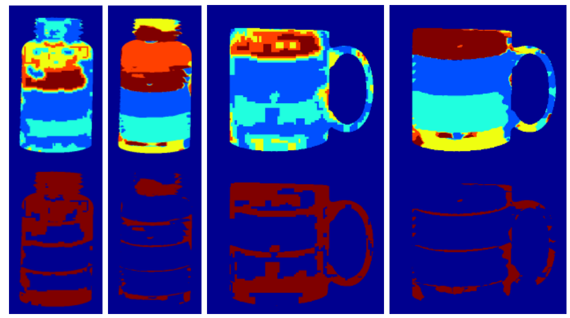
IV. EXPERIMENT

All experimental results were performed using a real FPP hardware setup, which contains a digital projector and a camera. The camera is equipped with a 22.2 x 14.8mm CMOS sensor and a 17-50mm lens. Both devices are placed at a distance 700mm-1200mm from the target object and are connected to a personal computer with a 3.4GHz CPU and 16GB RAM for image processing. All programs were developed in the MATLAB and Python environment. More specifically, the CNN was built using the PyTorch framework with GPU acceleration.

The first experiment is to verify the performance of the proposed CNN for segmenting the code patterns in the fringe image. We used a flat board with size 500mm x 400mm as the target object in the experiment. Since the *k*-values of a flat board can be easily determined, it was used to train the proposed CNN as well as to verify its accuracy. To train the network, we first projected fringe patterns unto a white and a black board. Then the proposed CNN was trained by sampling 30,000 patches of the captured fringe images with size 64x64 pixels. They were randomly selected as the training data. The learning rate and batch size are set to be 0.001 and 128 respectively. To verify the performance of the network, we used two flat boards with flower and checkered patterns as shown in the first and third columns of Fig. 5 respectively as the testing objects. The proposed algorithm is compared with a method we previously proposed based on the discriminative dictionary learning [1]. The segmentation result and error map are shown in the first and second row of Fig. 6 respectively. As shown in the figure, the proposed method can clearly segment the code patterns even when the object has vividly changing textures on its surface. Unlike the segmentation result obtained from using the discriminative dictionary learning, the proposed method can mitigate the distortion around the boundary; and is also more efficient due to the effective utilization of GPU by CNN. Table 2 shows the run time performance of the proposed method and LC-KSVD. As shown in the table, the proposed method can achieve much faster performance by more than 16 times compared with the fastest performance of LC-KSVD (the non-overlapping case). It also does not suffer from the blocking effect as in LC-KSVD.

In the second experiment, we compare the performance of the propose method against two traditional approaches including: three step phase shifting profilometry with the Goldstein phase unwrapping algorithm (PSP+Goldstein) [20, 21], and PSP with speckles (PSP-Speckle) [10]. The PSP+Goldstein method is popularly used nowadays whereas the PSP-Speckle was newly proposed to tackle the phase unwrapping problem in FPP. They are used to reconstruct a flat board with shiny and textured surfaces as in Fig. 5. Table 3 shows the comparison results in terms of SNR. They show that the proposed method can give a more accurate measurement compared to PSP+Goldstein and PSP-Speckle. It does not introduce additional distortions due to the embedded code patterns as the PSP-Speckle method.

Fig. 7. The code patterns of a bottle and a glass in Fig. 9 obtained from the modified MCA (1st and 3rd columns) and the ground truth of the segmentation (2nd and 4th columns).[1]



43.30% 81.85% 47.28% 84.84%

Fig. 8. The segmentation results of objects in Fig. 7. (1st and 3rd column) LC-KSVD segmentation in [1]; (2nd and 4th column) proposed segmentation. The first row is the segmentation result and the second row is the error map against the ground truth.

Fig. 9. The testing objects (1st column), their ground truth 3D models (2nd column), and their 3D models reconstructed using PSP+Golstein (3rd column), PSP-Speckle (4th column), and the proposed algorithm (5th column)

In the third experiment, we compare the proposed method when measuring the 3D model of real-life objects with abrupt changes in their height profile. In the experiment, a cup and a bottle were used as shown in Fig. 9. Similar to the previous experiment, we first verify the CNN performance. Fig. 7 shows the extracted code pattern obtained from the modified MCA and the ground truth of the segmented regions determined manually. We used the pre-trained CNN from the previous experiment to determine the segmented regions. The experimental results are shown in Fig. 8. As shown in the figure, the proposed method can obtain more accurate segmentation (2^{nd} and 4^{th} column) than LC-KSVD (1^{st} and 3^{rd} column), even for the relatively small parts of the object, e.g., the handle of glass (4^{th} column).

Finally, we compare the performance of the proposed algorithm with the conventional PSP+Goldstein [20, 21] and PSP-Speckle [10] for measuring the real-life objects in Fig. 9. As illustrated in the figure, the PSP+Goldstein method generates incorrect depth information because when the object has abrupt changes in the height profile (as compared to the background), the Goldstein phase unwrapping algorithm cannot generate the unwrapped phase without additional period order information. Here we do not assume any physical marker is added as in the traditional approaches to aid the phase unwrapping. Although the PSP+Speckle method can recover the depth information, the embedded speckles introduce artifacts to the reconstructed 3D model as can be seen in Fig. 9. Meanwhile, the proposed algorithm can measure the 3D model of the objects accurately and are similar to the ground truth.

V. Conclusion

This paper presented a novel CNN based phase unwrapping algorithm for FPP. CNN is employed to decode the period order information in the fringe pattern and solve the ambiguity problem when evaluating the true phase information in FPP. The proposed method first embeds the code patterns which carry the period order information in the fringe pattern. They are then extracted by the MMCA procedure and decoded by the proposed CNN. The new algorithm is highly efficient, which stems from the efficient utilization of GPU by the proposed CNN. Experimental results have demonstrated the superiority of the proposed algorithm over the conventional methods in terms of computation speed, robustness and accuracy.

Acknowledgment

This research is fully supported by the Hong Kong Research Grant Council under General Research Fund no. PolyU 5210/13E.

References

[1] Budianto and D. P. K. Lun, "Robust Fringe Projection Profilometry via Sparse Representation," *IEEE Trans. Image Process.,* vol. 25, no. 4, pp. 1726-1739, 2016.

[2] K. Itoh, "Analysis of the phase unwrapping algorithm," *Appl. Opt.,* vol. 21, no. 14, p. 2470, 1982.

[3] R. R. Garcia and A. Zakhor, "Consistent Stereo-Assisted Absolute Phase Unwrapping Methods for Structured Light Systems," *IEEE J. Sel. Topics Signal Process.,* vol. 6, no. 5, pp. 411-424, 2012.

[4] Y. Wang, K. Liu, Q. Hao, X. Wang, D. L. Lau, and L. G. Hassebrook, "Robust Active Stereo Vision Using Kullback-Leibler Divergence," *IEEE Trans. Pattern Anal. Mach. Intell.,* vol. 34, no. 3, pp. 548-563, 2012.

[5] J. Salvi, J. Pagès, and J. Batlle, "Pattern codification strategies in structured light systems," *Pattern Recognition,* vol. 37, no. 4, pp. 827-849, 2004.

[6] Z. Song, R. Chung, and X.-T. Zhang, "An Accurate and Robust Strip-Edge-Based Structured Light Means for Shiny Surface Micromeasurement in 3-D," *IEEE Trans. Ind. Electron.,* vol. 60, no. 3, pp. 1023-1032, 2013.

[7] Y. Wang and S. Zhang, "Novel phase-coding method for absolute phase retrieval," *Opt. Lett.,* vol. 37, no. 11, pp. 2067-2069, 2012.

[8] J. M. Huntley and H. Saldner, "Temporal phase-unwrapping algorithm for automated interferogram analysis," *Appl. Opt.,* vol. 32, no. 17, pp. 3047-52, 1993.

[9] Y. Zhang, Z. Xiong, Z. Yang, and F. Wu, "Real-Time Scalable Depth Sensing With Hybrid Structured Light Illumination," *IEEE Trans. Image Process.,* vol. 23, no. 1, pp. 97-109, 2014.

[10] Y. Zhang, Z. Xiong, and F. Wu, "Unambiguous 3D measurement from speckle-embedded fringe," *Appl. Opt.,* vol. 52, no. 32, pp. 7797-7805, 2013.

[11] P. Cong, Z. Xiong, Y. Zhang, S. Zhao, and F. Wu, "Accurate Dynamic 3D Sensing With Fourier-Assisted Phase Shifting," *IEEE J. Sel. Topics Signal Process.,* vol. 9, no. 3, pp. 396-408, 2015.

[12] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature,* vol. 521, p. 436, 05/27/online 2015.

[13] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," presented at the Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1, Lake Tahoe, Nevada, 2012.

[14] O. Russakovsky *et al.*, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision,* journal article vol. 115, no. 3, pp. 211-252, December 01 2015.

[15] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, Cham, 2015, pp. 234-241: Springer International Publishing.

[16] E. Shelhamer, J. Long, and T. Darrell, "Fully Convolutional Networks for Semantic Segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 39, no. 4, pp. 640-651, 2017.

[17] B. Budianto and D. P.-K. Lun, "Gabor feature based discriminative dictionary learning for period order detection in fringe projection profilometry," 2015. Available: https://doi.org/10.1109/APSIPA.2015.7415521

[18] V. Andrearczyk and P. F. Whelan, "Texture segmentation with Fully Convolutional Networks," *CoRR,* vol. abs/1703.05230, / 2017.

[19] A. a. G. Paszke, Sam and Chintala, Soumith and Chanan, Gregory and Yang, Edward and DeVito, Zachary and Lin, Zeming and Desmaison, Alban and Antiga, Luca and Lerer, Adam, "Automatic differentiation in PyTorch," 2017.

[20] R. M. Goldstein, H. A. Zebker, and C. L. Werner, "Satellite radar interferometry: Two-dimensional phase unwrapping," *Radio Science,* vol. 23, no. 4, pp. 713-720, 1988.

[21] P. S. Huang and S. Zhang, "Fast three-step phase-shifting algorithm," *Appl. Opt.,* vol. 45, no. 21, pp. 5086-5091, 2006.