Temporal phase unwrapping towards high-accuracy and high-speed fringe projection profilometry

He, Xiaoyu

2021

He, X. (2021). Temporal phase unwrapping towards high-accuracy and high-speed fringe projection profilometry. Doctoral thesis, Nanyang Technological University, Singapore.
<https://hdl.handle.net/10356/150663>

<https://hdl.handle.net/10356/150663>

<https://doi.org/10.32657/10356/150663>

This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License (CC BY-NC 4.0).

Downloaded on 26 Apr 2023 21:28:35 SGT

**NANYANG
TECHNOLOGICAL
UNIVERSITY**

SINGAPORE

**TEMPORAL PHASE UNWRAPPING TOWARDS
HIGH-ACCURACY AND HIGH-SPEED
FRINGE PROJECTION PROFILOMETRY**

HE XIAOYU

**Interdisciplinary Graduate School
Multi-plAtform Game Innovation Centre (MAGIC)**

2021

**TEMPORAL PHASE UNWRAPPING TOWARDS
HIGH-ACCURACY AND HIGH-SPEED
FRINGE PROJECTION PROFILOMETRY**

HE XIAOYU

**Interdisciplinary Graduate School
Multi-plAtform Game Innovation Centre (MAGIC)**

A thesis submitted to the Nanyang Technological University in partial
fulfillment of the requirement for the degree of
Doctor of Philosophy

2021

Statement of Originality

I hereby certify that the work embodied in this thesis is the result of original research, is free of plagiarised materials, and has not been submitted for a higher degree to any other University or Institution.

22/ Jan /2021

.....
Date

He Xiaoyu

.....
He Xiaoyu

Supervisor Declaration Statement

I have reviewed the content and presentation style of this thesis and declare it is free of plagiarism and of sufficient grammatical clarity to be examined. To the best of my knowledge, the research and writing are those of the candidate except as acknowledged in the Author Attribution Statement. I confirm that the investigations were conducted in accord with the ethics policies and integrity standards of Nanyang Technological University and that the research data are presented honestly and without prejudice.

22/ Jan /2021

.....

Date

.....

Qian Kemao

Authorship Attribution Statement

This thesis contains material from 3 paper(s) published in the following peer-reviewed journal(s) where I was the first and/or corresponding author.

Chapter 3 is published as X. He, D. Zheng, Q. Kemaο, and G. Christopoulos, “Quaternary gray-code phase unwrapping for binary fringe projection profilometry,” *Opt. Lasers Eng.*, vol. 121, no. April, pp. 358–368, 2019, doi: 10.1016/j.optlaseng.2019.04.009.

The contributions of the co-authors are as follows:

- Prof. Qian Kemaο and Prof. Zheng Dongliang provided the initial project direction.
- I prepared the manuscript drafts. Prof. Qian Kemaο and I reviewed and revised the manuscript.
- I developed the algorithms including weighted optimization for binary pattern design and a normalization-denoising-clustering process for gray code reconstruction
- I built up a fringe projection profilometry system and verified the algorithms by simulations and experiments.
- Prof. Zheng Dongliang, and Prof. Georgios Christopoulos gave comments and provided advice on the manuscript.

Chapter 4 is published as X. He and Q. Kemaο, “A comparison of n -ary simple code and n -ary gray code phase unwrapping in high-speed fringe projection profilometry,” *Opt. Lasers Eng.*, vol. 128, no. January, p. 106046, 2020, doi: 10.1016/j.optlaseng.2020.106046.

The contributions of the co-authors are as follows:

- Prof. Qian Kemaο supervised me on the overall research direction.
- I prepared the manuscript drafts. Prof. Qian Kemaο and I reviewed and revised the manuscript.
- I proposed an identification and binary classification method for the n -ary simple code phase unwrapping.

- Prof. Qian Kemaο and I designed the comparison considering different factors that influence the performance of the methods.
- I built up a fringe projection profilometry system and compares the methods by simulations and experiments.

Chapter 5 is X. He and Q. Kemaο, “A comparative study on temporal phase unwrapping methods in high-speed fringe projection profilometry,” *Opt. Lasers Eng.*, vol. 142, no. February, p. 106613, 2021, doi: 10.1016/j.optlaseng.2021.106613.

The contributions of the co-authors are as follows:

- Prof. Qian Kemaο supervised me on the overall research direction.
- I prepared the manuscript drafts. Prof. Qian Kemaο and I reviewed and revised the manuscript.
- I built up mathematic models describing the noises in the different phase unwrapping methods.
- Prof. Qian Kemaο and I designed the comparison considering different situations including the 8-bit focused system, the binary defocused system, and the geometry/continuity constrained system.
- I built up a fringe projection profilometry system and compares the methods by simulations and experiments.

22/Jan/2020

.....

Date

He Xiaoyu

.....

He Xiaoyu

Acknowledgments

First and most important, I would like to express my sincere gratitude to my supervisor Prof. Qian Kemao for his valuable guidance and constant support in my PhD research and daily life. For the part of research, he helps me revise my manuscripts again and again to let me know how to logically think and do “re”-“search”. He really likes to discuss with students and listen to their opinions, which gives me freedom in academic study and the chance to study my own interests. In addition, he emphasizes the integration of academy and industry, for which he provided me opportunities to cooperate with industrial companies and apply state-of-the-art techniques in 3D body measurement and industrial inspection. For the part of daily life, he always encourages me to do physical exercise and organizes colorful physical activities including hiking, cycling, badminton, and football in our research group, for which we have walked across almost all the natural parks in Singapore. Besides, he encourages me to read literature books and Chinese poems, and form a habit of writing, which helps me to keep thinking and understand how to write an academic paper like writing a poem. In a word, he is not only a respectable supervisor of my research work but also an amiable mentor of my life.

I would also like to thank my co-supervisor, Prof. Georgios Christopoulos, for his kind support and encouragement throughout the four years, without which I cannot finish PhD study smoothly. I still remember his interesting proposal about facial expression recognition, without which I even would not have the chance to study at NTU as a PhD student. His high requirement of integrity also has a great impact on me. Special gratitude goes out to my mentor, Prof. Seah Hock Soon, for his constructive comments and insightful questions about my research topic. He is also the main founder of Multi-plAtform Game Innovation Centre (MAGIC) which provides me the most advanced devices to do experiments and the opportunities to cooperate with colleagues. I genuinely cherish the wonderful memories in MAGIC.

Thanks also go to Prof. Zheng Dongliang for his patient guidance on the construction of the experimental platform and inspiration for my initial research direction. Thanks to my

senior Dr. Wang Tianyi for his kind help and encouragement in my academic study and daily life.

Finally, I would like to express my gratefulness to my parents and my wife. In the past 27 years, my parents always do their best to support me physically and mentally which gives me the chance to concentrate on study and research. Special thanks to my wife, Guo Wanghui, for her 7-years company from Beijing to Singapore, for her support as always without which I may never have gotten to where I am today.

Table of Contents

Summary.....	i
List of Figures.....	iii
List of Tables	vi
Chapter 1 Introduction.....	1
1.1 Optical three-dimensional measurement	1
1.2 Fringe projection profilometry	3
1.3 Wrapped phase extraction	5
1.3.1 Fourier transform (FT)	5
1.3.2 Phase-shifting (PS).....	6
1.4 Phase unwrapping.....	7
1.4.1 Spatial phase unwrapping (SPU).....	7
1.4.2 Temporal phase unwrapping (TPU).....	8
1.5 Contributions and organization.....	8
1.5.1 Motivations.....	8
1.5.2 Contributions	10
1.5.3 Organization	12
Chapter 2 Literature Review	13
2.1 TPU methods	13
2.1.1 Simple code and gray code phase unwrapping.....	13
2.1.2 Two/multiple-frequency phase unwrapping.....	15
2.1.3 Phase coding phase unwrapping.....	16
2.1.4 Stereo-vision-based methods.....	17
2.1.5 Composite-fringe-pattern-based methods	17
2.1.6 Deep-learning-based methods	18
2.2 Techniques for projection speed enhancement.....	19
2.2.1 Binary defocusing	19

2.2.2 Binary pattern generation.....	20
2.3 Techniques for TPU pattern reduction	21
2.3.1 Ternary GC phase unwrapping	21
2.3.2 Two-plus-one PS algorithm	22
2.3.3 Geometry/continuity constrained SPU	23
Chapter 3 Phase unwrapping with different numbers of codes	25
3.1 Introduction.....	25
3.2 The overall PS-GC framework for binary FPP	26
3.3 The proposed GC phase unwrapping	27
3.3.1 The conventional QGC mechanism.....	28
3.3.2. bQGC pattern design through weighted optimization	30
3.3.3. Gray code reconstruction	32
3.4 QGC with continuity/geometry constraints.....	36
3.5 Experiments	39
3.6 Discussion	42
3.7 Conclusion	43
Chapter 4 Phase unwrapping with different orders of codes	45
4.1 Introduction.....	45
4.2 Principles of nSC and nGC phase unwrapping	46
4.2.1 <i>N</i> -step PS algorithm.....	46
4.2.2 <i>n</i> SC and <i>n</i> GC phase unwrapping.....	47
4.2.3 Adaptive median filtering	49
4.2.4 Discussions	50
4.3 The proposed identification and binary classification method for nSC phase unwrapping	51
4.3.1 Analysis of errors in nSC phase unwrapping.....	51
4.3.2 J2 EP identification.....	53
4.3.3 Binary classification	54
4.4 Comparison by Simulations	56
4.4.1 Simple smooth object	57
4.4.2 Step heights and invalid regions	58

4.4.3 Influence of different error removal strategies	59
4.4.4 Influence of the defocusing level and noise level	61
4.4.5 Influence of the base n	62
4.5 Comparison by experiments	63
4.5.1 Simple smooth object	64
4.5.2 Objects with step heights.....	66
4.5.3 nSC with continuity/geometry constraints	67
4.6 Conclusion.....	70
Chapter 5 Phase unwrapping with different coding strategies.....	73
5.1 Introduction	73
5.2 Principles of temporal phase unwrapping methods	74
5.2.1 <i>N</i> -step phase-shifting algorithm	75
5.2.2 TF phase unwrapping	77
5.2.3 PC phase unwrapping	80
5.2.4 GC phase unwrapping	83
5.2.5 Summary and error rates	86
5.3 Comparison by simulations	90
5.3.1 aFPP system	90
5.3.2 bFPP system	93
5.3.3 cFPP system	97
5.3.4 Summary and denoising in practice	101
5.4 Comparison by experiments	102
5.4.1 Experiments on a flat board	102
5.4.2 Experiments on complex objects.....	106
5.5 Conclusion.....	113
Chapter 6 Conclusions and Future Works.....	115
6.1 Conclusions	115
6.2 Future works	117
6.2.1 Temporal phase unwrapping comparison towards high accuracy.....	117
6.2.2 Gray code extraction using deep learning	118

6.2.3 Temporal phase unwrapping with one additional pattern.....	118
Author's Publications.....	121
References	123

Summary

Optical three-dimensional (3D) measurement technologies, mainly including photogrammetry, interferometry, time of flight, and speckle/fringe projection profilometry, have been widely used in industrial inspection, motion capture, virtual/augmented reality, and heritage preservation. Compared with the other technologies, the fringe projection profilometry (FPP) has many inherent advantages, such as low cost, high resolution, and relatively high speed. In FPP, a continuous phase map is extracted from the captured fringe patterns and then converted to the profile of the object. To obtain the continuous phase map, spatial and temporal phase unwrapping techniques are developed. Compared with the spatial phase unwrapping, the temporal phase unwrapping (TPU) avoids the interference from the physical discontinuities on the measured objects by the pixel-by-pixel manner, thus has more applications. However, there are still some issues unclear in TPU for high-accuracy and high-speed FPP: (i) the limited measurement speed of gray code phase unwrapping by the number of codes; (ii) the impact of the different orders of codes in the gray code and simple code phase unwrapping methods; (iii) the comparison between different TPU methods with different coding strategies. Therefore, in this thesis, a systematic study of TPU methods towards high-speed and high-accuracy FPP is proposed to address the above issues.

First, to enhance the speed of gray code phase unwrapping, I propose a novel quaternary gray code method under a binary defocusing FPP system. A weighted optimization algorithm is presented for the design of special binary code patterns. The designed binary patterns are projected on the object with a defocused projector, and interestingly become quaternary patterns which are more expressive gray codes. After the distorted code patterns are captured by a camera, they will undergo the proposed normalization-denoising-clustering process to recover the desired gray codes, which are then used to recover the phase orders for phase unwrapping. Compared with the traditional binary/ternary gray code methods, the proposed method obtains the unwrapped phase with similar accuracy but fewer patterns, thus the measurement speed is enhanced.

Second, I comprehensively compare the simple code and gray code methods with the consideration of various factors, including the strategies for error removal, the numbers of codes, the step heights and invalid regions on the measured objects, defocusing level and noise level of the system. From simulations and experiments, I conclude that (i) in general, simple code with a proposed identification and binary classification method has a similar result as gray code; (ii) when accuracy is critical, binary simple code and binary gray code methods are recommended; (iii) when speed is critical, quaternary simple code method with continuity/geometry constraints is recommended.

Third, I compare three popular TPU methods using different coding strategies, two-frequency, phase coding, and gray code methods, in which three situations are considered: the traditional 8-bit focused FPP (bFPP), the high-speed binary defocused FPP (bFPP), and the geometry/continuity constrained binary defocused FPP (cFPP). In all three methods, two types of phase unwrapping errors result from the system noise: uniformly distributed errors (E^U) all over the phase map and non-uniformly distributed errors (E^N) only around particular regions. The E^U and E^N of the three methods are compared qualitatively and quantitatively, from which I find that (i) In aFPP, E^U is more noticeable in TF and PC than GC when the fringe frequency is high ($f \geq 16$), and E^N is around left and right image borders in TF, while it is around 2π phase jumps in PC and GC; GC shows the best performance; (ii) In bFPP, the overall performances of the three methods are similar to those in aFPP, but there are more E^N in PC and GC than aFPP due to defocusing; TF becomes more comparable to GC; (iii) In cFPP, compared with bFPP, E^U in TF and PC is reduced greatly and E^N in TF is removed, thus TF shows the best performance. Based on these new findings, a quick guide for TPU selection is provided.

In summary, this thesis has the following contributions for TPU in high-accuracy and high-speed FPP: (i) a framework for gray code phase unwrapping to enhance the measurement speed; (ii) the comparison of TPU methods with different orders of codes, and (iii) the comparison of TPU methods with different coding strategies. These contributions enable researchers and engineers to understand the state-of-the-art TPU methods and select the “best” TPU method for their applications.

List of Figures

Figure 1.1 Typical Methods of optical 3D measurement.....	2
Figure 1.2 Performance of various optical measurement methods.	3
Figure 1.3 A typical FPP system.....	4
Figure 1.4 Flowchart of FPP.....	5
Figure 2.1 Illustration of (a) SC phase unwrapping and (b) GC phase unwrapping.	14
Figure 2.2 Illustration of TF phase unwrapping ($f = 8$).	16
Figure 2.3 Illustration of PC phase unwrapping ($f = 8$).	16
Figure 2.4 The effect of projector defocusing.....	20
Figure 2.5 Illustration of TGC pattern generation in the binary defocused FPP.	22
Figure 2.6 Illustration of geometry/continuity constrained phase unwrapping.	23
Figure 3.1 Our PS-GC framework including a PS block (conventional), a GC block (novel), and continuity/geometry constraints block (novel but optional).	26
Figure 3.2 Flowchart of the proposed bQGC phase unwrapping.....	28
Figure 3.3 Codeword of the QGC with 16 periods.	29
Figure 3.4 The process of GC pattern optimization using the weighted optimization method.	31
Figure 3.5 Number of erroneous pixels in full-size patterns designed by different methods	32
Figure 3.6 Gray code reconstruction.....	34
Figure 3.7 Simulation of projector defocusing on the binary patterns.	35
Figure 3.8 Simulation of 16-period QGC phase unwrapping with only one pattern.	37
Figure 3.9 A comparison experiment between SPU and GcPU.	39
Figure 3.10 Experimental results of the proposed framework.	40
Figure 3.11 Experimental results of TGC phase unwrapping.....	41
Figure 3.12 Reconstructed 3D profile on multiple objects by the proposed framework with $n = 4$	42
Figure 3.13 Reconstructed 3D depth maps.	43
Figure 4.1 N-step phase-shifting algorithm.	47
Figure 4.2 Schematic diagram of the n SC (a) and n GC (b) phase unwrapping.	48

Figure 4.3 Comparison of n SC and n GC phase unwrapping.	49
Figure 4.4 Effect of AMF.	50
Figure 4.5 Schematic diagram of the n SC and n GC phase unwrapping with only one pattern.	51
Figure 4.6 Phase unwrapping errors in n SC phase unwrapping.	52
Figure 4.7 J2 EP identification based on the gradient map.	53
Figure 4.8 Code extraction with IBC.	55
Figure 4.9 n SC phase unwrapping with IBC and AMF.	56
Figure 4.10 Simulation of a simple linear phase.	58
Figure 4.11 Simulation of a complex phase with invalid regions and step heights.	59
Figure 4.12 EPs in different methods.	60
Figure 4.13 A comparison of the different phase unwrapping methods with different defocusing levels.	61
Figure 4.14 A comparison of the different phase unwrapping methods with different noise levels.	62
Figure 4.15 A comparison of the different phase unwrapping methods with different n	63
Figure 4.16 Experiment on a smooth object without step heights.	64
Figure 4.17 3D reconstruction of a smooth object without step heights.	65
Figure 4.18 A comparison of the different phase unwrapping methods on a smooth object by experiment.	65
Figure 4.19 Experiment on multiple objects with step heights.	66
Figure 4.20 3D reconstruction of multiple objects with step heights.	66
Figure 4.21 A comparison of the different phase unwrapping methods on multiple objects with step heights.	67
Figure 4.22 Flowchart of the n SC phase unwrapping with continuity/geometry constraints.	67
Figure 4.23 Geometry-constrained n SC phase unwrapping with two QSC images.	69
Figure 4.24 Geometry-constrained n SC phase unwrapping with two QSC images on complex objects.	70
Figure 5.1 Illustration of the normal and abnormal phase errors.	76
Figure 5.2 TF phase unwrapping with noises.	80
Figure 5.3 PC phase unwrapping with noises.	82
Figure 5.4 Gray code phase unwrapping with noises.	86
Figure 5.5 The probability density of the critical random variable after normalization.	87
Figure 5.6 Simulation of a smooth surface with 8-bit patterns.	91
Figure 5.7 Error distributions of the different phase unwrapping methods.	92
Figure 5.8 Error rates with different f in aFPP.	93

Figure 5.9 Differences between the 8-bit patterns and defocused binary patterns.	95
Figure 5.10 Error distributions of different phase unwrapping methods in a bFPP system.	95
Figure 5.11 Error rate of E^U and the total error rate with different f in bFPP.	97
Figure 5.12 Error distributions of different phase unwrapping methods in a cFPP system.....	99
Figure 5.13 Error rate of E^U and the total error rate with different f in cFPP.	100
Figure 5.14 Ground-truth measurement results of a flat board by 9-step PS algorithm with multi-frequency phase unwrapping.	103
Figure 5.15 Depth map reconstructed by different phase unwrapping methods with $f = 64$	104
Figure 5.16 Error rate of E^U and the total error rate with different f	105
Figure 5.17 Ground-truth measurement results of complex objects by 9-step PS algorithm with multi-frequency phase unwrapping.....	107
Figure 5.18 Depth maps reconstructed by different phase unwrapping methods in aFPP.....	108
Figure 5.19 Depth maps reconstructed by different phase unwrapping methods in bFPP.	110
Figure 5.20 Depth maps reconstructed by different phase unwrapping methods in cFPP.....	111
Figure 5.21 Multiple objects with complex geometries by different phase unwrapping methods in cFPP.....	112
 Figure 6.1 Some potential patterns can be used for single-frame phase unwrapping with deep learning.....	 119

List of Tables

Table 3.1 The number of required patterns for different numbers of codes.....	43
Table 5.1 Comparison of the phase unwrapping errors in the three methods in aFFP.....	87
Table 5.2 Error rates of the errors in the three methods in aFFP	89
Table 5.3 Comparison of the three phase unwrapping methods in aFPP.....	93
Table 5.4 Comparison of the three phase unwrapping methods in bFPP.....	97
Table 5.5 Comparison of the three phase unwrapping methods in cFPP.....	100

Chapter 1 Introduction

1.1 Optical three-dimensional measurement

Three-dimensional (3D) measurement, the process of analyzing a real-world object or environment to collect its shape information, has been widely used in industrial inspection, motion capture, virtual/augmented reality, heritage preservation, etc. Among different 3D measurement techniques, optical measurement is most popular due to its noninvasive and noncontact nature [1]. The optical measurement utilizes the properties of light and extracts the shape information (usually 3D point cloud or depth map) from the images of the measured object.

The classification of typical optical measurement methods [2]–[5] is shown in Figure. 1.1. Considering the difference of the measurement principle, we usually classify the optical measurement into two types, passive and active methods. The passive optical measurement uses the light that is naturally present in a scene, while the active optical measurement illuminates the object with active light sources containing predefined spatiotemporal structures. The passive optical measurement [6]–[9] is also called photogrammetry or stereo vision and is motivated by the human vision system. It uses two or more calibrated cameras to capture the scene from different views and reconstruct the scene based on the correspondences between the images by triangulation [10], [11]. The passive measurement systems are simple and cheap since only the cameras and ambient light are required. However, the number of feature points on the object, which is required for the determination of the correspondences, limits the density of the reconstructed point cloud. If the object is texture-less, the point cloud will be very sparse. Besides, feature detection, a key process in the passive methods, is time-consuming. To overcome these limitations, dense tracking and mapping (DTAM) is proposed [12], [13]. Nevertheless, a standard feature-based photogrammetry method is still inevitable at the initialization of DTAM and the performance of DTAM will be poor if the illumination changes. Additionally, both the method based on feature points and DTAM are inaccurate, by which the average error can be more than 1 cm [14].

Figure 1.1 Typical Methods of optical 3D measurement.

The active optical measurement “probes” structured light on the object to solve the problem of measuring texture-less objects in the passive method, thus, it is also called structured-light measurement. According to the different physical principles, the active optical measurement is classified into three major types: interferometry-based, triangulation-based, and time-of-flight-based (ToF) methods [3]. Among these methods, the interferometry-based method (interferometry or holograph) [15]–[19] is the most accurate which is generally used for the most sensitive measurements in science and industry. However, there are some limitations in the size of the measurement area and tilt angle because of optical components in the interferometry and holograph. In the triangulation-based method [4], [5], predefined patterns, such as a single dot, a line, discrete dot pattern, and fringe pattern, are projected on the object and a camera captures the distortion of the patterns resulted from the surface profile from another view. The correspondences between the projected patterns and captured images are extracted and used to recover the 3D shape information through triangulation [20], [21]. Compared with the interferometry-based method, the triangulation-based method has a larger measurement scale but lower measurement accuracy. ToF [22] measures the distance to an object by the time delay of the round trip from the emitted modulated light to the detected back-reflected light. This method has the largest measurement scale thus is widely used in control and navigation for self-driving cars [23], [24] and robots [25], [26].

The accuracy and scale of different optical measurement methods are summarized in Fig. 1.2. Additionally, the optical measurement methods also have differences in measurement speed, resolution, flexibility, and cost, thus they are applied in different fields accordingly. This thesis focuses on the triangulation-based structured light using fringe patterns, which is named fringe projection profilometry (FPP). Compared with the other methods, FPP achieves good accuracy (around 0.1 mm), high speed, high resolution, and low cost [27]. Due to these advantages, FPP has various applications in biomedicine [28]–[31], manufacturing [32]–[36], face recognition [37], [38], and heritage preservation [39]–[41].

Figure 1.2 Performance of various optical measurement methods. Image was recreated based on the image in [2].

1.2 Fringe projection profilometry

As shown in Fig. 1.3, a typical FPP system contains a projection unit (projector), an image acquisition unit (camera), a processing/analysis unit (computer), and the measured object. The projector, camera, and object form a triangulation base. The phase information included in the fringe pattern builds up correspondences between the projector and camera which are used for triangulation to reconstruct the 3D shape.

Figure 1.3 A typical FPP system.

The process of 3D measurement by FPP is shown in Fig. 1.4 which involves the following steps:

- (1) Pattern generation: for different applications, different fringe patterns, such as sinusoidal [42], trapezoidal [43], triangular [44], and squared [45] fringe patterns, with different frequencies [46], [47] are designed by the computer and sent to the projector.
- (2) Projection and acquisition: the projector projects the pre-designed fringe patterns on the object which distorts the fringe patterns from straight stripes to curved ones; then, the camera captures the distorted fringe patterns from another angle.
- (3) Wrapped phase extraction: by phase extraction techniques, such as phase-shifting (PS) [48], [49] and Fourier transform (FT) [50]–[52], a wrapped phase map in the range of $(-\pi, \pi]$ is obtained from the captured images. When multiple fringe periods are projected, multiple 2π phase jumps are resulted in the wrapped phase.
- (4) Phase unwrapping: spatial [53], [54] or temporal [55], [56] phase unwrapping techniques are used to remove the 2π phase jumps in the wrapped phase, by which a continuous phase map is obtained.
- (5) Calibration and triangulation: the camera and the projector should be calibrated to convert the unwrapped phase to 3D shape information of the object by triangulation. The calibration methods are usually classified into two types: analytical-expression-based methods [57]–[60] mapping the unwrapped phase to a depth map and stereovision-based methods [21], [61], [62] mapping the unwrapped phase to a 3D point cloud.

Compared with the other triangulation-based structured light, the unwrapped phase required in FPP is unique, which determines the quality of the reconstructed 3D shape. Therefore, some popular methods for wrapped phase extraction and phase unwrapping are reviewed in the following sections.

Figure 1.4 Flowchart of FPP.

1.3 Wrapped phase extraction

1.3.1 Fourier transform (FT)

The captured fringe image can be expressed as

$$I(x, y) = A(x, y) + B(x, y)\cos[\varphi(x, y)], \quad (1.1)$$

where (x, y) denotes the spatial coordinate on the camera image plane; A is the background intensity; B is the fringe amplitude; φ is the phase containing the depth information of the object. With Euler's formula, FT rewrites Eq. 1.1 in a complex form as

$$I(x, y) = A(x, y) + \frac{1}{2}B(x, y)e^{j\varphi(x, y)} + \frac{1}{2}B(x, y)e^{-j\varphi(x, y)}, \quad (1.2)$$

where $e^{-j\varphi(x, y)}$ is the conjugate of $e^{j\varphi(x, y)}$. The three components in Eq. 1.2 concentrate around different frequencies, thus after Fourier transform, a properly designed band-pass filter in the frequency domain can be used to remove the conjugate and the background components. The inverse Fourier transform of the filtered signal returns

$$\tilde{I}(x, y) = \frac{1}{2}B(x, y)e^{j\varphi(x, y)}. \quad (1.3)$$

The phase then can be resolved by

$$\varphi(x, y) = \tan^{-1} \left\{ \frac{\text{Im}[I(x, y)]}{\text{Re}[I(x, y)]} \right\}, \quad (1.4)$$

where $\text{Im}(\cdot)$ and $\text{Re}(\cdot)$ return the imaginary and real parts of a complex number, respectively.

It should be noted that FT requires that A and B vary slowly, the frequency of the fringe pattern is high, and the noise level is not too high, so that the three components in Eq. 1.2 are separable. Thus, FT may not work for complex objects with sharp edges, discontinuities, large texture variation, or much noise. To alleviate these problems, windowed Fourier transform (WFT) [63], [64] and wavelet transform (WT) [65]–[67] methods are proposed. All these transform-based methods retrieve phase information from only a single fringe pattern, thus are highly suitable for the measurement of dynamic surfaces. However, even with WFT and WT, the influence of the discontinuities cannot be ignored [68].

1.3.2 Phase-shifting (PS)

Unlike the FT using only one fringe pattern, PS requires three or more phase-shifted fringe patterns for phase extraction. Different algorithms, such as the three-step phase-shifting algorithm [69], least-squares algorithm [70], Carré algorithm [71], and Hariharan algorithm [72], are designed for different phase-shifted fringe patterns to calculate the phase. In FPP, N -step phase-shifted fringe patterns with equal and known phase shifts are usually used, which can be expressed as

$$I_i(x, y) = A(x, y) + B(x, y) \cos[\varphi(x, y) + 2\pi(i - 1)/N], i = 1, 2, 3, \dots, N, \quad (1.5)$$

where i is the index of the fringe pattern and $2\pi(i - 1)/N$ is the corresponding phase shift. Through least-squares fitting, the phase can be calculated as

$$\varphi(x, y) = -\tan^{-1} \left\{ \frac{\sum_{i=1}^N I_i(x, y) \sin[2\pi(i-1)/N]}{\sum_{i=1}^N I_i(x, y) \cos[2\pi(i-1)/N]} \right\}. \quad (1.6)$$

Typically, a large phase-shifting step (N) has high accuracy [73], but $N = 3$ as the minimum required number is often used for the interest of high speed. Compared with FT, PS has a lower measurement speed, but is more robust to ambient illumination, varying surface reflectivity, and discontinuities [42].

1.4 Phase unwrapping

Both the FT and PS methods use an arctangent function to calculate the phase as shown in Eq. 1.4 and Eq. 1.6, thus the resultant phase is wrapped in the range of $(-\pi, \pi]$. When multiple fringes are projected, there are 2π phase jumps around the boundary of the fringe stripes in the wrapped phase, which causes the phase ambiguity problem. To build a unique pixel correspondence between the camera and projector for triangulation, Φ should be removed. Phase unwrapping is the process to determine the phase order k for each pixel and remove the 2π phase jumps by

$$\Phi(x, y) = 2k(x, y)\pi + \varphi(x, y), \quad (1.7)$$

where Φ is the unwrapped phase map.

1.4.1 Spatial phase unwrapping (SPU)

SPU [53] calculates the phase order of a given pixel based on the phase difference with its neighboring pixels (spatial domain), as follows:

$$k(x, y) = \text{round}\left[\frac{\varphi(x', y') - \varphi(x, y)}{2\pi}\right], \quad (1.8)$$

where $\text{round}(\cdot)$ takes the closest integer of the input value, and (x', y') is an unwrapped pixel neighboring (x, y) . Eq. (1.7) and (1.8) ensure that the difference of the unwrapped phase between neighboring pixels is within $(-\pi, \pi]$. As a result, the 2π phase jumps are removed. SPU is very intuitive and works well in ideal situations. However, in real experiments, the noise, low fringe modulation, and other factors make SPU very challenging. To achieve robust phase unwrapping, quality-guided SPU methods [54], [74] are proposed. These methods use a quality map extracted from the fringe patterns to guide the phase unwrapping path, in which the high-quality phase pixels are unwrapped first before the low-quality phase pixels to avoid the error propagation in SPU.

It should be noted that, even with the quality-guided methods, SPU still can only be applied on a smooth surface without large physical discontinuities which introduce phase changes larger than π . Besides, SPU only provides a relative unwrap phase. To retrieve the absolute unwrapped phase, the fringe patterns should be encoded with certain markers [75], [76]. SPU is usually combined with FT for extreme high-speed measurement in which only one fringe pattern is required.

1.4.2 Temporal phase unwrapping (TPU)

TPU projects additional patterns (temporal domain) to encode the fringe orders and remove the 2π phase jumps by Eq. (1.7). Popular TPU methods [56] include the multi/two-frequency method, phase coding method, simple/gray code method, stereo-vision-based method, and composite-fringe-pattern-based method. Compared with SPU, TPU is pixel independent, i.e., does not require the neighboring pixel phase information, thus TPU resolves the problem of measuring objects with large physical discontinuities in SPU. To measure objects with an arbitrary geometric shape, TPU is usually combined with PS which is also pixel independent. However, both PS and TPU require multiple patterns, limiting the measurement speed and the application of dynamic object measurement. In this case, the projection time for each pattern and/or the total number of the required patterns should be reduced. The binary defocusing technique [45], [77] is an effective and widely used method to reduce the projection time for each pattern. To reduce the number of the required patterns for different TPU methods, many new methods, such as ternary gray code phase unwrapping [78], two-plus-one phase-shifting algorithm [79]–[81], geometry [81], [82] and continuity [83], [84] constrained SPU, have been proposed.

1.5 Contributions and organization

1.5.1 Motivations

As aforementioned, PS for phase retrieval combined with TPU for phase unwrapping has been widely used in recent works on FPP due to their pixel-by-pixel manner, which avoids the interference by the physical discontinuities on the measured objects. Although many potential solutions have been proposed to enhance the performance of PS-TPU-based FPP, three issues remain open for further investigation, which are the motivations of this thesis. Specifically, the motivations are listed below:

- Gray code phase unwrapping is a simple and straightforward temporal phase unwrapping method, in which a group of gray code patterns is projected to encode the fringe order of the wrapped phase. In n -ary gray code phase unwrapping, the number of codes is n . The traditional gray code method uses binary gray code,

where $n = 2$ and only 2 gray levels (black and white) are used for coding. For a wrapped phase with a frequency of f , i.e., it has f periods, $\lceil \log_n f \rceil$ binary gray code patterns are required, where $\lceil \cdot \rceil$ rounds a number towards positive infinity. Therefore, increasing the number of codes can reduce the number of required patterns and enhance the measurement speed. However, it becomes more difficult to decode the gray code correctly when more codes are used, especially under a binary defocused FPP system and when $n > 3$. Therefore, it is of great significance to build a framework to code and decode n -ary gray code effectively for binary defocused FPP with $n > 3$. Besides, it is valuable to investigate that what is the best value of n for high-accuracy and high-speed phase unwrapping.

- Simple code method and gray code method are two similar phase unwrapping methods, both of which code the phase order by the intensity of the projected pattern, i.e., gray level. In n -ary simple/gray code method, n different gray levels represent n different codes. The order of the code in the two methods are different. Due to the defocused projector and discrete nature of the camera, phase unwrapping errors are introduced around the code jumps the simple/gray code patterns. In the gray code method, all the errors can be treated as impulse noises as the width of the errors is around 2 pixels. While in the simple code method, parts of the errors cannot be treated as impulse noises as their width increases to around 10 pixels. Therefore, it is easier to remove the phase unwrapping errors in the gray code method, making it more widely used than the simple code method. However, recently, the simple code method demonstrates a higher measurement speed compared with the gray code method since fewer patterns are required when it incorporates the continuity/geometry constraints. It is thus necessary to compare simple code with gray code towards high-accuracy and high-speed phase unwrapping.
- Besides the intensity-coded phase unwrapping methods (simple/gray code), multi/two-frequency method and phase coding method are two other popular temporal phase unwrapping methods, in which the desired phase order is coded by the phase information in the projected patterns. Among them, the multi/two-frequency method has been intensively reviewed, where shows that the two-frequency method minimizes the number of required patterns, and the hierarchical multi/two-frequency method provides the best unwrapping reliability. The two-frequency, phase coding, and gray code methods are widely applied in FPP and can

be easily used to replace one another in the same FPP system, but their performances have not been compared. Therefore, it is worth evaluating and comparing them theoretically and experimentally.

1.5.2 Contributions

In order to address the above-mentioned problems, novel advanced algorithms are developed, and temporal phase unwrapping methods are reviewed and compared in this thesis. The contributions of this thesis are summarized below.

- A framework is proposed for gray code phase unwrapping in binary defocused FPP, in which a weighted optimization algorithm is developed for binary pattern design, a normalization-denoising-clustering process is used for gray code reconstruction, and geometry/continuity constrained phase unwrapping is incorporated to further reduce the gray code pattern. The proposed weighted optimization algorithm gives a larger weight for the pixels around gray code boundaries in optimization since those pixels have a greater impact on the GC recovery. Compared with the unweighted optimization and the 1D design methods, the proposed method produces fewer recovery errors. In the normalization-denoising-clustering process, the normalization removes the background and amplitude in the captured gray code patterns, median filtering reduces the noise in the normalized patterns, and k -means clustering extracts the code automatically as a general method for systems with different defocusing levels. Compared with the traditional decoding method by thresholding, the proposed normalization-denoising-clustering process is more robust to the varying background and amplitude, random noise, and projector defocusing effect. The pre-calibrated information or the continuity of the object, i.e., geometry/continuity constraint, can guide the gray code phase unwrapping which reduces the required gray code pattern to $\lceil \log_n f \rceil - 1$. Besides, $n = 4$, i.e., quaternary gray code is suggested, since it enhances the speed without the loss of accuracy. Although one more gray code pattern can be reduced when $n = 8$, it is too difficult to decode the code from 8 gray levels.
- The different orders of codes in the simple code and gray code result in that the phase unwrapping errors around the code jumps in the simple code method are more complex. Therefore, we first propose an identification and binary classification

(IBC) method to convert the errors in the simple code method into impulse noises. As a result, all the errors in the simple code and gray code unwrapped phase are impulse noises and can be removed by adaptive median filtering (AMF) easily. Besides the different orders of codes, several other factors will also influence the accuracy and speed, including the strategies on using IBC and AMF for error removal, the number of codes n , the step heights and invalid regions on the measured objects, defocusing level and noise level of the system. Therefore, comprehensive comparison considering these factors is provided by simulation and experiment, from which, we find that (i) in general, both simple code with IBC-AMF and gray code with AMF perform similarly and satisfactorily; (ii) when accuracy is critical, binary simple code with AMF and binary gray code with AMF are outstanding candidates; and (iii) when the speed is critical, quaternary simple code (simple code with $n = 4$) with IBC-AMF is suggested to be incorporated with continuity/geometry constraints.

- The three temporal phase unwrapping methods, two-frequency, phase coding, and gray code methods, are compared in three situations: the traditional 8-bit focused FPP (aFPP), the high-speed binary defocused FPP (bFPP), and the geometry/continuity constrained binary defocused FPP (cFPP). The comparison is based on their sensitivities to noise, for which noise models of the phase coding and gray code methods are built and compared with the existing noise model of the TF method. Phase unwrapping errors caused by the noise are classified into uniformly distributed errors (E^U) all over the phase map and non-uniformly distributed errors (E^N) only around particular regions. For both E^U and E^N , their distributions are analyzed, and their error rates are calculated for different TPU methods and different FPP systems by simulations and experiments. In aFPP, E^U is more noticeable in TF and PC than GC when the fringe frequency is high ($f \geq 16$), and E^N is around left and right image borders in TF, while it is around 2π phase jumps in PC and GC; GC shows the best performance. In bFPP, the overall performances of the three methods are similar to those in aFPP, but there are more E^N in PC and GC than aFPP due to defocusing; TF becomes more comparable to GC. In cFPP, compared with bFPP, E^U in TF and PC is reduced greatly and E^N in TF is removed; TF thus shows the best performance. Based on our comprehensive comparisons, a quick guide is provided for the high-frequency phase unwrapping: for static

measurement, GC in aFPP is preferred, while for dynamic measurement, TF in cFPP is suggested.

1.5.3 Organization

This thesis consists of six chapters, which is organized as follows. Chapter 1 introduces the background and the remaining issues of FPP and summarizes the contributions of the works in this thesis. Chapter 2 presents a comprehensive literature review, including the principles of different TPU methods and the related techniques of TPU developed for performance enhancement. Chapter 3 proposes a framework for gray code phase unwrapping in binary defocused FPP and discusses the influences of the number of codes in gray code phase unwrapping. Chapter 4 proposes an identification and binary classification (IBC) method to reduce the phase unwrapping errors in the simple code methods, and analyzes the influences of different orders of codes in the gray code and simple code, by simulation and experiment. Chapter 5 presents the noise models for phase coding and gray code methods to compare them with the existing noise model of the two-frequency method theoretically, and the phase unwrapped errors caused by the noise in the TPU with different encoding methods are comprehensively compared by simulation and experiment. Chapter 6 gives the conclusions of the works and provides some recommendations on the future work.

Chapter 2 Literature Review

2.1 TPU methods

TPU determines the fringe order by acquired information from the additional projected patterns. According to the different patterns and techniques used, the TPU methods are usually classified into simple/gray code method, multi/two-frequency method, phase coding method, stereo-vision-based method, and composite-fringe-pattern-based method. The principles of these TPU methods will be reviewed in this section.

2.1.1 Simple code and gray code phase unwrapping

The idea of simple code (SC) [85] / gray code (GC) [86] phase unwrapping is to code and project the phase orders k so that the true phase Φ can be easily determined by Eq. (1.7). The SC/GC with n different codes, i.e., $0, 1, \dots, n - 1$, is named n -ary SC/GC. The coding principle is much like generating n -ary numbers from digits, which can be expressed as

$$C(x, y) = \sum_{i=1}^M c_i(x, y) \times n^{M-i}, \quad (2.1)$$

where n is the base of digits and thus also the gray levels (number of codes) of the code patterns; $c_i = 0, 1, \dots, n - 1$ is the digit value and often called as code; M is the number of digits; C is the synthesized code value and often called as codeword. To uniquely code f periods in a fringe pattern, the number of SC/GC patterns is $M = \lceil \log_n f \rceil$. The mapping between the codeword $C(x, y)$ and the phase order $k(x, y)$ depends on the coding mechanism used. Figure 2.1 illustrates the coding principles of SC and GC with $n = 2$ and $M = 3$, to code $C = 0, 1, 2, \dots, 8$. In SC as shown in Fig. 2.1(a), the desired C values from 0 to 7 derived by Eq. (2.1) can be directly used as phase order k , i.e., $k(x, y) = C(x, y)$, while in GC as shown in Fig. 2.1(b), a look-up table is built to get the desired k as $k(x, y) = \text{LUT}[C(x, y)]$. In both SC and GC phase unwrapping, due to the noises and projector defocusing, the 2π phase jumps in φ and code jumps marked in red in Fig. 2.1 cannot be aligned exactly, which may result in phase unwrapping errors. Compared with GC, SC has more code jumps and will cause more phase unwrapped errors. As a result, GC is more widely used.

Figure 2.1 Illustration of (a) SC phase unwrapping and (b) GC phase unwrapping.

To remove these phase unwrapping errors, various methods including, self-correction method [87], adaptive median filtering (AMF) [88], the complementary GC method [89], and tripartite phase unwrapping [90]. These methods do not work when the errors are very dense. Compared with the other methods, AMF is simpler and thus more robust for complex object measurement, and does not require additional patterns. We will use AMF to handle the phase unwrapping errors around the code jumps in the rest of the thesis.

The AMF method includes two steps: error detection and error correction. In the error detection step, a small window (7×7) around a pixel is selected as the local area of that pixel. A pixel will be considered as an error if it is quite different from the other pixels in its local area. In the error correction step, a median filter is applied only to the detected errors. These two steps are repeated with an increasing size of the median filter from small to large until there are no errors or the number of errors no longer reduces. The AMF method prioritizes a small filter size and maximizes the chance of removing the noises without destroying the step height of a measured object.

It is highlighted that, even with the above error removal methods, part of the errors in SC cannot be removed effectively, which will be elaborated in Chapter 4. In addition, besides

the errors around the code jumps, the random noise in the system can also cause phase unwrapping errors on the whole map, the noise model of which will be presented in Chapter 5.

2.1.2 Two/multiple-frequency phase unwrapping

Two-frequency (TF) phase unwrapping projects two groups of PS patterns, one with a low frequency f_l and the other with a high frequency f_h . Then the corresponding low-frequency wrapped phases φ_l and the high-frequency wrapped phase φ_h are calculated by Eq. (1.6). The relationship between their unwrapped phases, Φ_l and Φ_h , is expressed by:

$$\Phi_h = f_h/f_l \cdot \Phi_l. \quad (2.2)$$

Different TF algorithms have been formed using different combinations of f_l and f_h [91]. Among these algorithms, the hierarchical algorithm [92], [93] provides best reliability and best noise-robustness [91], where f_h determines the quality of the wrapped phase and $f_l = 1$ determines the rough value of the absolute phase to assist phase unwrapping. We simply use f to represent f_h in the rest of the thesis.

According to the two frequencies of the hierarchical algorithm, Eq (1.7) is rewritten as:

$$\Phi_l = 2k_l\pi + \varphi_l = \varphi_l, \quad (2.3)$$

$$\Phi_h = 2k_h\pi + \varphi_h, \quad (2.4)$$

where k_l and k_h are the corresponding fringe orders and $k_l = 0$ for $f_l = 1$. Combining Eqs. (2.2)-(2.4), k_h can be determined as:

$$k_h = \text{round}\left(\frac{f \cdot \varphi_l - \varphi_h}{2\pi}\right), \quad (2.5)$$

Subsequently, φ_h is unwrapped into Φ_h by Eq. (1.7). This is the basic principle of the TF phase unwrapping method, for which, an example with ideal TF phases and $f = 8$ is illustrated in Fig. 2.2.

The TF method is sensitive to noise and the unwrapping success rate decrease with the increase of f . As a result, the fringe frequency is generally restricted to about 16 [94]. An effective strategy is projecting additional groups of fringe patterns [47], [95]. However, the multi-frequency phase unwrapping requires more patterns, which limits the measurement speed.

Figure 2.2 Illustration of TF phase unwrapping ($f = 8$).

2.1.3 Phase coding phase unwrapping

The phase coding (PC) phase unwrapping [96] also codes the phase order with phase ranging $(-\pi, \pi]$ as the TF method. However, the low-frequency PS patterns with a linear phase in TF are replaced by PS patterns with a stepped phase, which are named PC patterns. The stepped phase φ_s codes the phase order k ($0 \leq k \leq f - 1$) for phase unwrapping as follows:

$$\varphi_s = -\pi + (k + \frac{1}{2}) \cdot \frac{2\pi}{f}, \quad (2.6)$$

$$k = \lfloor \frac{x_p f}{W} \rfloor, \quad (2.7)$$

where $\lfloor \cdot \rfloor$ rounds a number towards negative infinity. The stepped phase φ_s is calculated by Eq. (1.6) from the PC patterns. The fringe order is extracted solely from φ_s as:

$$k = \text{round} \left[\frac{f \cdot (\varphi_s + \pi)}{2\pi} - \frac{1}{2} \right], \quad (2.8)$$

with which the wrapped phase φ is unwrapped by Eq. (1.7). As an example, the ideal PC phase unwrapping with $f = 8$ is illustrated in Fig. 2.3.

Figure 2.3 Illustration of PC phase unwrapping ($f = 8$).

Similar to the SC/GC phase unwrapping, there are also phase unwrapping errors around the 2π phase jumps in PC. The error compensation methods mentioned in Section 2.2.1 for the SC/GC method can be also applied in PC. Some other methods [97]–[99] has been also

proposed to enhance the accuracy of PC, however, the influence of the unavoidable noise on PC is still unclear, especially when compared with the other methods, for which the noise model of PC will be presented in Chapter 5.

2.1.4 Stereo-vision-based methods

The stereo-vision-based methods project a group of PS patterns for wrapped phase extraction and a single statistical random pattern for phase unwrapping. In this method, the projected pattern is regarded as the image acquired from the perspective of the projector. As a result, the camera image and the projector image view the object from two perspectives and establish a stereo vision system. The stereo-vision-based method [100] first builds up a coarse correspondence between the camera image and the projector image by stereo matching with the statistical random pattern; then initial fringe order is estimated for each pixel based on the coarse correspondence; at last, a refinement process is employed to adjust the misalignment between the fringe order and the 2π phase jumps in the wrapped phase, by which the unwrapped phase is obtained.

The stereo-vision-based method works well when the pixel sizes and distortions of the camera image and projector image are similar. However, in real cases, it is difficult to ensure the same pixel size of the camera and projector and the projector image is ideal without distortion. To overcome these problems, a second camera is employed [101], with which the two cameras form the stereo vision system. In such a situation, the projector is used to add features to the objects so that the correspondence between the two cameras can be derived. The framework for the dual-camera, single projector system is that in [100], in which the only difference is the input images are from the two cameras. Since one more camera is used, the cost of the system increases. Besides, such a system requires all the devices to “see” the object, thus its possible measurement area is smaller than the standard FPP system with a single camera and a single projector.

2.1.5 Composite-fringe-pattern-based methods

The aforementioned TPU methods require additional patterns besides the PS patterns for wrapped phase extraction. To speed up the measurement, composite-fringe-pattern-based

methods code the information for phase unwrapping into the PS patterns. A unit-frequency sinusoid component is combined with high-frequency PS patterns [102], from which the wrapped phase for each frequency is obtained with signal demodulation theory. As a result, the high-frequency phase can be unwrapped by the unit-frequency phase using the TF method. Another method code more complex information into the PS patterns [103], from which a slope signal can be calculated and further used for phase unwrapping. Besides, some other methods code a random pattern into the amplitude of the PS patterns [101], [104]–[106] or a stepped pattern into the value of the phase shift of the PS patterns [107].

Compared with the other TPU method, the composite-fringe-pattern-based methods do not require additional patterns, which enhance the measurement speed. However, the additional information coded in the PS patterns sacrifices the accuracy of the wrapped phase, which thus limits the measurement accuracy.

2.1.6 Deep-learning-based methods

In 2019, deep learning was first applied in FPP to recover the wrapped phase using a single fringe pattern [108], in which a deep neural network was trained to predict the intermediate numerator and denominator of the arctangent function for phase estimation. To obtain the training dataset for the neural network, fringe patterns are captured from experiments and the ground-truth wrapped phase maps are obtain by 18-step phase-shifting algorithm. Compared with the existing single-frame methods, such as Fourier transform and windowed Fourier transform methods, this deep-learning-based method provides a better performance around the edges of the object.

Besides, deep learning is also applied in phase unwrapping to improve the accuracy of TF [94]. The proposed deep-learning-based TF phase unwrapping method recovers the absolute phase from two wrapped phases with different frequencies by exploiting both spatial and temporal phase information in an integrated way. This method improves the reliability of phase unwrapping even when high-frequency fringe patterns are used. For the applications to high-speed FPP, it has also been observed that the deep learning-based method is much less affected by motion artifacts in dynamic measurement and can successfully reconstruct the surface profile of the moving and rotating objects at high speed.

However, this deep-learning-based TF phase unwrapping still requires at least 6 patterns to recover the unwrapped phase. To reduce the required patterns, the neural networks for wrapped phase extraction and phase unwrapping are combined [109], in which 3 fringe patterns with different frequencies are used. With a properly trained deep neural network, the unwrapped phase is predicted from the 3 fringe patterns and then converted into the 3D shape. To further enhance the speed, the 3 fringe patterns are merged into one color pattern [110], and as a result, with a pre-trained neural network, a unwrapped phase can be “learned” from a color-coded image without any pre/post-processing or phase error compensation. However, similar as the other color-coded projection methods, this method is unsuited for measuring color objects.

2.2 Techniques for projection speed enhancement

Dynamic object measurement requires high acquisition speed to reduce potential motion artifacts. In this case, the projection time for each pattern and/or the total number of the required patterns should be reduced in PS-TPU. The defocusing technique [111] is an effective and widely used method to reduce the projection time for each pattern.

2.2.1 Binary defocusing

The binary defocusing technique is based on the fringe pattern generation mechanism of the digital-light-processing (DLP) projector. At the core of the DLP projector, there is an extremely precise light switch, digital micro-mirror device (DMD). The DMD chip consists of an array of microscopic mirrors, each of which controls one pixel of light in a fringe pattern. For each microscopic mirror, there are only two states, on and off. The corresponding pixel is white when the mirror is switched on, and it is black when the mirror is switched off. As for the grayscale colors, the mirror controls the proportion of on and off times during one frame period to generate the colors between white and black. For example, when the gray value of one pixel is 255, the mirror is 100% on during one frame; If the gray value is reduced to 128, approximately half of the channel is filled; When the gray value is reduced to 64, a smaller portion of the signal is filled. It is shown that the output signal is irregular during one frame when the gray value is between 0 and 255, while the projected light will be solid-state if only 0 or 255 is used for each pixel [112].

For high-speed measurement, the exposure time of the camera is required to be much shorter than one frame time of the projector. However, if a gray-value image is used, the output signal during one frame is irregular, and as a result, the exposure time cannot be less than one frame. On the contrary, the output signal of binary patterns, which only contain pixels with 0 or 255, is stable. As a result, in contrast to ~ 120 Hz of the 8-bit (gray-value) pattern projection, the projection speed of binary patterns reaches tens of kHz which is more suitable for high-speed measurement [113]. To bridge the projected binary patterns and the desired structured patterns, a defocused projector is employed. The defocused projector serves as a low-pass filter and thus can blur squared binary patterns or other designed binary patterns to quasi-sinusoidal patterns. As an example, the effect of projector defocusing is simulated by Gaussian filtering as shown in Fig. 2.6.

Figure 2.4 The effect of projector defocusing. (a) Squared binary pattern, (b) defocused squared binary pattern.

2.2.2 Binary pattern generation

The squared binary pattern is the simplest and most popular pattern used in the binary defocused FPP. However, the defocused square binary pattern usually contains many high-order harmonics which result in errors in the wrapped phase [114]. To suppress the high-order harmonics, a pulse width modulation (PWM) method [115] and its variations [116]–[118] are proposed. Both the PWM-based patterns and the squared binary pattern are generated in one dimension, i.e., the pixels on the same column have the same value, limiting the accuracy of the defocused binary pattern [77]. Therefore, various two-dimensional binary pattern generation methods are presented. One effective two-dimensional method is binary dithering, in which the desired structured patterns are binarized by a dither kernel. According to the different kernels and binarization strategies

used, binary dithering is further classified as Bayer-ordered dithering [119], Floyd–Steinberg error-diffusion dithering [120], and Stucki error-diffusion dithering [121]. Besides, optimization-based two-dimensional methods are also proposed [122]–[124], which changes the value of each pixel on the binary pattern constantly to minimize the difference between the defocused binary pattern and the desired structured pattern. Some other works try to generate the best patterns for different FPP systems adaptively, for which the depth of the object [46] and the defocusing level of the system [47] are taken into consideration.

2.3 Techniques for TPU pattern reduction

To reduce the number of the required patterns for different TPU methods, many new methods have been proposed. Some typical methods related to our works are introduced in this section.

2.3.1 Ternary GC phase unwrapping

The traditional GC method using binary gray code can be easily applied in the binary defocused FPP. For a typical value of $f = 64$, $\lceil \log_2 64 \rceil = 6$ binary GC patterns are required for phase unwrapping. To enhance the measurement speed, a ternary GC (TGC) method phase unwrapping methods is proposed [78], by which the number of required GC patterns is reduced to $\lceil \log_3 64 \rceil$. In TGC, a one-dimensional binary pattern generation method is used, in which the repeated patches of 00, 10, and 11 are used to represent three different gray levels (black, gray, and white), after defocusing. The principle of the TGC patterns in the binary defocused FPP is shown in Fig. 2.7. The thresholds for gray code extraction are related to the defocusing level of the system. Simulations and experiments show that there are more phase unwrapping errors around the 2π phase jumps in the ternary method than those in the binary method. However, with a simple median filter, the ternary method performs as robustly as the traditional binary method.

Figure 2.5 Illustration of TGC pattern generation in the binary defocused FPP. (a) Designed binary patterns for TGC, (b) blurred patterns of (a).

The TGC method reduces the required GC patterns to 4 when a typical frequency $f = 64$ is used. Considering the requirement of three fringe patterns for three-step phase shifting, still seven patterns in total are needed for one measurement. It is intuitive that more gray levels (i.e., more codes) used in GC result in a higher measurement speed. However, it is difficult to apply the framework in TGC to GC with $n > 3$, effectively. A flexible and robust framework for n -ary GC phase unwrapping ($n > 3$) is required to further reduce the required patterns.

2.3.2 Two-plus-one PS algorithm

The two-plus-one PS algorithm [79]–[81] aims to reduce the required patterns in the TF or PC phase unwrapping. In the traditional TF and PC methods, at least 6 patterns including two groups of PS patterns are required. The two-plus-one PS algorithm replaces the group of PS patterns used for unwrapping, i.e., the low-frequency patterns in TF and the PC patterns in PC, with another two patterns. The two new patterns used in TF are expressed as:

$$I_1'(x, y) = A(x, y) + B(x, y)\sin[\varphi_l(x, y)], \quad (2.9)$$

$$I_2'(x, y) = A(x, y) + B(x, y)\cos[\varphi_l(x, y)]. \quad (2.10)$$

The two-plus-one PS algorithm assumes the background $A(x, y)$ to be a constant for all captured patterns. As a result, φ_l can be derived by

$$\varphi_l(x, y) = -\tan^{-1}\left\{\frac{I_1'(x, y) - A(x, y)}{I_2'(x, y) - A(x, y)}\right\} \quad (2.11)$$

where $A(x, y)$ is calculated from the PS patterns by

$$A(x, y) = \frac{\sum_{i=1}^N I_i}{N}. \quad (2.12)$$

The above equations for TF can be easily rewritten for PC by replacing φ_l with φ_s . With the two-plus-one PS algorithm, only 5 patterns in total are required for the TF or PC phase unwrapping. It should be noticed that the two-plus-one PS algorithm is inferior to the traditional N -step PS algorithm in accuracy since fewer patterns are used.

2.3.3 Geometry/continuity constrained SPU

In FPP, geometry/continuity constrained phase unwrapping is combined with the TPU methods to enhance the accuracy [81], [82] or reduce the required patterns [83], [84]. The main idea of the geometry/continuity constrained phase unwrapping is illustrated in Fig. 15 to unwrap the phase φ (blue line) which includes $f = 64$ periods. A partial phase unwrapping is performed by a TPU method to unwrap $T = 16$ periods for 4 times instead of a complete unwrapping of $f = 64$ periods immediately. As a result, a temporary phase Φ (red line) is obtained with large phase jumps ($T \times 2\pi$). Since now the phase jumps are large, they are more distinguishable from physical discontinuities of the measured objects and can be removed by adding or subtracting $T \times 2\pi$ for multiple times to satisfy the geometry/continuity constraints and obtain the final absolute phase Φ_c (yellow line). This method undergoes the partial and final phase unwrapping steps and thus is more complicated, but it brings advantages.

Figure 2.6 Illustration of geometry/continuity constrained phase unwrapping, where $f = 64$ and $T = 16$.

Chapter 3 Phase unwrapping with different numbers of codes

3.1 Introduction

PS for phase retrieval combined unwrapping with binary GC for phase unwrapping has been widely used in FPP. As the binary GC patterns have only black and white, the binary GC method can be easily applied in a binary defocused system which has a much higher projection speed than the traditional 8-bit focused system [113]. For a wrapped phase with f periods, $\lceil \log_n f \rceil$ GC patterns are required for phase unwrapping. Therefore, to reduce the number of required GC patterns for high-speed measurement, the number of codes n should be increased, for which more gray levels should be used to represent different codes. The generation and decoding of n -ary GC ($n > 2$) are trivial tasks in the traditional 8-bit focused system, but become much more difficult in the binary defocused system in which n gray levels are generated by only black and white.

The ternary GC (TGC) phase unwrapping ($n = 3$) [78] intuitively represents three different gray levels (black, gray, and white) with repeated patches of 00, 10, and 11 and extract the gray code from the captured binary defocused patterns by thresholding. However, it is difficult to extend this framework for n -ary GC with $n > 3$ robustly, which motivate us to propose a new robust and flexible framework for n -ary GC towards binary defocused FPP. In the proposed framework, a weighted optimization is developed for binary pattern design and a normalization-denoising-clustering process is presented for gray code reconstruction. Besides, continuity [53], [54] or geometry [81], [125] constraint is incorporated with the n -ary GC phase unwrapping to further reduce one pattern. In this chapter, quaternary GC with $n = 4$ (QGC) is selected as an example and the effectiveness of the proposed is proved by simulations and experiments. With the proposed framework, an unwrapped phase can be derived with only 5 patterns in total by QGC, which is fewer than 7 patterns required by TGC.

The proposed framework can be flexibly applied for GC with different numbers of codes (n), based on which the influence of the number of codes on the efficiency and accuracy of the GC phase unwrapping are analyzed. It is noticed that the QGC with $n = 4$ can provide a fast measurement without the loss of accuracy. Although the measurement speed can be improved when n is increased to 8, the success rate of decoding from octal GC patterns is decreased due to the noise and defocusing effect.

3.2 The overall PS-GC framework for binary FPP

Our overall PS-GC framework for binary-defocused FPP (bFPP) (Fig. 3.1) includes three components: the PS (the blue block) describing wrapped phase extraction, the GC (the red block with a solid boundary), and continuity/geometry constraints (the red block with a dotted boundary) describing phase unwrapping. The PS component adopts the most widely used three-step phase-shifting algorithm which requires only three fringe patterns. Although this algorithm is well-known, it is briefly described here for the completeness of the framework and easy flow for the subsequent sections. The QGC for phase unwrapping is our novel contribution and will be described in detail in Section 3.3. The continuity/geometry constraints to assist phase unwrapping is also a novel but optional component and will be described in detail in Section 3.4.

Figure 3.1 Our PS-GC framework including a PS block (conventional), a GC block (novel), and continuity/geometry constraints block (novel but optional).

In a bFPP system, the three-step phase-shifting algorithm projects three binary phase-shifted fringe patterns I_i^{bP} [115], [124], [126]. After defocusing, the patterns are modeled as:

$$I_i^{dP}(x^P, y^P) = a^P + b^P(f_x) \cos[2\pi f_x x^P - 2\pi(i-2)/3], \quad i = 1, 2, 3, \quad (3.1)$$

where (x^P, y^P) is the spatial coordinate of the projector; a^P , b^P , f_x are the background intensity, fringe amplitude, and the frequency of the fringe along x axis, respectively; the superscript P highlights that the fringe pattern is projected out from the projector. Since defocusing serves as a low-pass filter, in this model, we highlight that a^P is usually a constant ($= 0.5$), but b^P is a variable (a function of f_x). The fringe patterns are deformed by the object and then reflected (scattered) to the camera. The reflected light contains the projector light and the ambient light [30]. In general, the captured fringe images can be expressed as

$$I_i(x, y) = A(x, y) + B(x, y, f_x) \cos[\varphi(x, y) - 2\pi(i-2)/3], \quad i = 1, 2, 3, \quad (3.2)$$

where (x, y) denotes the spatial coordinate of the camera; $\varphi(x, y)$ is the phase distribution containing the 3D shape information; $A(x, y)$ and $B(x, y, f_x)$ are the background and fringe amplitude of the captured fringe patterns, respectively. Due to the environment's lighting conditions and object's surface properties, both A and B are spatially varying, i.e., non-uniform. Besides, B is related to the amplitude of the projected fringe pattern b^P , thus B is also a function of f_x .

The wrapped phase, background and fringe amplitude of these captured fringe images can be calculated as:

$$\varphi(x, y) = \tan^{-1} \left[\frac{\sqrt{3}(I_1(x, y) - I_3(x, y))}{2I_2(x, y) - I_1(x, y) - I_3(x, y)} \right], \quad (3.3)$$

$$A(x, y) = \frac{1}{3} [I_1(x, y) + I_2(x, y) + I_3(x, y)], \quad (3.4)$$

$$B(x, y) = \frac{1}{3} \sqrt{3[I_1(x, y) - I_3(x, y)]^2 + [2I_2(x, y) - I_1(x, y) - I_3(x, y)]^2}. \quad (3.5)$$

3.3 The proposed GC phase unwrapping

We select $n = 4$, i.e., QGC, to go through the details of the proposed GC phase unwrapping for bFPP.

The conventional QGC method generates and projects M quaternary patterns $q_i^P \in \{0,1,2,3\}$ ($i = 1,2, \dots, M$) for encoding, and captures and reconstructs quaternary patterns $q_i^R \in \{0,1,2,3\}$ for decoding. Both q_i^P and q_i^R contain quaternary values. Although the QGC method is simple and effective, it does not fit bFPP as bFPP is only able to project binary values $\{0, 1\}$. We thus propose a binary-defocused QGC (bQGC) phase unwrapping method, whose main idea is to generate quaternary patterns from binary projected values. The detailed process is shown in Fig. 3.2. In the encoding part, the target QGC q_i^P is normalized to q_i^{nP} which is used to design the bQGC patterns q_i^{bP} which is unknown yet. After defocusing, q_i^{bP} is blurred to q_i^{dP} and the deformed QGC image q_i^C is captured by the camera. By making q_i^{dP} as close as possible to q_i^{nP} , the unknown q_i^{bP} can be designed. In the decoding part, a new QGC q_i^R is reconstructed from q_i^C and used to unwrap the phase according to the coding mechanism.

Figure 3.2 Flowchart of the proposed bQGC phase unwrapping.

In this section, the conventional QGC mechanism is explained first; followed by the proposed bQGC pattern design method using weighted optimization and a GC reconstruction method based on a normalization-denoising-clustering process.

3.3.1 The conventional QGC mechanism

This section explains the conventional QGC mechanism. The phase obtained by Eq. (3.3) is within $(-\pi, \pi]$ due to the arctangent function. As a result, there are 2π jumps. The task of phase unwrapping is to determine the phase order k to recover the phase map $\Phi(x, y)$ without 2π jumps:

$$\Phi(x, y) = \varphi(x, y) + k(x, y) \times 2\pi. \quad (3.6)$$

For illustration, a 16-period wrapped phase with T pixels per period is shown in Fig. 3.3(a) as an example, which can be readily unwrapped with k shown in Fig. 3.3(b). The unwrapped phase by Eq. (3.6) is shown in Fig. 3.3(c). Unfortunately, in practice, when the fringe pattern is deformed, k also becomes unknown. Note that for simplicity and clarity, 16-period fringe patterns or codes are used for explanation, while in practice, 64-period fringe patterns are usually used for higher accuracy of the wrapped phase.

Figure 3.3 Codeword of the QGC with 16 periods. (a) Wrapped phase; (b) phase order for phase unwrapping; (c) unwrapped phase; (d)-(e) reconstructed gray codes from the QGC images, q_1^R and q_2^R ; (f) a unique code generated from q_1^R and q_2^R .

As discussed earlier, the gray code is currently used to determine k , where the basic idea is to code each fringe period with a phase interval of 2π by a unique gray code. The most widely used gray code is binary, which can be quickly generated and projected by a DMD projector, but the binary GC method requires $M = \lceil \log_2 f \rceil$ patterns to uniquely encode f periods [86]. Recently, an interesting TGC method has been proposed, where the projected

pattern is still binary, but it will become ternary after defocusing [78]. As a result, m is reduced to $\lceil \log_3 f \rceil$. With QGC, m can be further reduced to $\lceil \log_4 f \rceil$.

The coding mechanism of QGC is illustrated in Figs. 3.3(d)-3.3(f). For the 16-period wrapped phase in Fig. 3.3(a), 2 QGC patterns using four values $\{0, 1, 2, 3\}$ are required. In an ideal situation, the target gray codes of the two sending QGC patterns, q_1^P and q_2^P , and the recovered QGC, q_1^R and q_2^R , are the same, which are shown in Figs. 3.3(d)-3.3(e). Naturally, a unique code can be generated from the recovered QGC as

$$C(x, y) = \sum_{i=1}^M [q_i^R(x, y) \times 4^{i-1}], \quad (3.7)$$

where $M = \text{Ceil}(\log_4 f)$ is the number of QGC patterns, and thus $M = 2$ in the current example. The C distribution from Eq. (3.7) is shown in Fig. 3.3(f), which varies irregularly but is unique. Thus, it can be corresponded to the phase order k in Fig. 3.3(b) uniquely by a look-up table (LUT). This mechanism is more efficient than binary and ternary gray code since M is reduced from $\lceil \log_2 f \rceil$ and $\lceil \log_3 f \rceil$ to $\lceil \log_4 f \rceil$. Even for high-accuracy FPP, where narrow fringe patterns are used, e.g., $n = 64$, only 3 QGC patterns are required.

3.3.2. bQGC pattern design through weighted optimization

The first problem in conventional QGC is that $q_i^P \in \{0, 1, 2, 3\}$ should be projected but a bFPP system can only project binary values. To adopt QGC into bFPP, we here propose to use bQGC pattern design. To do so, the target QGC patterns $q_i^P \in \{0, 1, 2, 3\}$ are first normalized to $q_i^{nP} \in \{0, 1/3, 2/3, 1\}$. Then we aim to design bQGC patterns q_i^{bP} which, after defocusing, should turn into q_i^{nP} . Various binary pattern design methods have been reviewed in Section 2.2.2. The one-dimension binary pattern design [78] is the most intuitive by simply using the repeated patches of 000, 010, 101, and 111 to represent black, dark gray, light gray and white, respectively, after defocusing. In this chapter, the problem is modeled as minimizing the following energy function which was originally used to generate a binary fringe pattern [122]:

$$E_i = \sum_{\sigma} \sum_{x^P=1}^{S_x} \sum_{y^P=1}^{S_y} [q_i^{nP}(x^P, y^P) - G(x^P, y^P, \sigma) \otimes q_i^{bP}(x^P, y^P)]^2, \quad (3.8)$$

where G is a 2D Gaussian kernel with a mean of zero and a standard deviation of σ to reflect the defocusing, $\otimes$ denotes the convolution, and $S_x \times S_y$ represents the pattern size in pixel, $\sigma = 1, 1.5, 2, 2.5$ are used to simulate different defocusing levels.

It is realized that, in a GC pattern, the boundaries between different periods have a greater impact on the GC recovery. To pursue high fidelity around the boundaries during the optimization, a spatial weight function is added to the objective function:

$$E_i = \sum_{\sigma} \sum_{x^P=1}^{S_x} \sum_{y^P=1}^{S_y} w_i(x^P, y^P) [q_i^{nP}(x^P, y^P) - G(x^P, y^P, \sigma) \otimes q_i^{bP}(x^P, y^P)]^2. \quad (3.9)$$

The following empirical weight function works effectively:

$$w_i(x^P, y^P) = \begin{cases} 5, & D(x^P, y^P) \leq 2 \\ 1, & D(x^P, y^P) > 2 \end{cases}, \quad (3.10)$$

where D is the distance from a pixel to its nearest boundary. The objective function is minimized by successively mutating a binary pixel from 0 to 1 or from 1 to 0. Since the GC patterns are periodic, as shown in Fig. 3.3(d), only a sufficiently large patch should be optimized, and the result can be tiled into a full-size pattern. For illustration, Figs. 3.4(a)-3.4(c) show the patches (72×10 pixels) of the target gray code, the initial guess of the binary code where 0s and 1s are randomly assigned to light gray and dark gray regions, and the optimized binary code, respectively.

To test the correctness of the weighed optimization, we first defocus the optimized binary code by Gaussian filtering (Fig. 4(d)). We then use thresholds of $1/6$, $1/2$, and $5/6$ to segment the defocused pattern into four regions to recover the gray code (Fig. 4(e)) - which is seen the same as the ideal QGC pattern in Fig. 4(a).

Figure 3.4 The process of GC pattern optimization using the weighted optimization method. (a) Ideal patch; (b) random patch for initialization; (c) optimized patch; (d) defocused patch ($\sigma = 1.5$); (e) recovered gray codes of (d) by thresholding; (f) smoothed patch ($\sigma = 1.5$) with random noise.

It is noted that our weighted optimization of Eq. (3.9) is affected by the initial guess and has not involved noise. Thus, for the design of each q_i^{bP} , we do optimization 50 times with 50 different initial guesses and generate a pool of 50 pattern candidates. Each candidate is then numerically defocused by convolution, contaminated by Gaussian noise (50 trials), and segmented by thresholding to recover QGC. As an example, a smoothed patch with random noise is shown in Fig. 3.4(f). The candidate with the least average recovery error is considered as most robust to noise and used as q_i^{bP} . The same process is also used to generate the pattern from the unweighted optimization. Finally, the patterns generated by the weighted optimization are compared with the unweighted optimization and the one-dimension design. As shown in Fig. 3.5, the weighted optimization produces the least recovery errors.

Figure 3.5 Number of erroneous pixels in full-size patterns designed by different methods

3.3.3. Gray code reconstruction

The second problem in using QGC stems from the following practical issues: (i) code distortion due to spatial varying A and B ; (ii) random noise; and (iii) projector defocusing effect. Consequently, the simple thresholding used in the simulation in the last section usually does not work in practice. To address this problem, we propose a novel normalization-denoising-clustering method for gray code reconstruction.

3.3.3.1 Normalization

The QGC patterns can be seen as special fringe patterns and thus represented by Eq. (3.2) as follows,

$$q_i^c(x, y) = A(x, y) + B(x, y, f_{ci}) \cos[\varphi'_i(x, y)], \quad (3.11)$$

where f_{ci} is the frequency of the i th QGC pattern which affects the modulation, and $\varphi'_i(x, y) \in \{\cos^{-1}(-1), \cos^{-1}(-1/3), \cos^{-1}(1/3), \cos^{-1}(1)\}$ for the four code values of $\{0, 1, 2, 3\}$, respectively. The first practical issue is that the same code values are distorted to be spatially different because of the non-uniformity of A and B and this may ultimately result in wrong QGC. The code distortion is illustrated by a GC image of a flat board in Fig. 3.6(a), and its center row is shown in Fig. 3.6(b), where the distortion is obvious. We adopt normalization as a simple yet effective solution. The A and B distributions computed from captured PS fringe patterns by Eqs. (3.4) and (3.5) are used to normalize QGC patterns as follows,

$$q_i^{r1} = \frac{q_i^c - A(x, y)}{B(x, y, f_x)} = \frac{B(x, y, f_{ci})}{B(x, y, f_x)} \cos[\varphi'_i(x, y)]. \quad (3.12)$$

After normalization, the background is removed, and the code becomes spatially more uniform, as shown in Fig. 3.6(c).

(e)

Figure 3.6 Gray code reconstruction. (a) The QGC image of a plat board, q_1^c ; (b) the center row of q_1^c ; (c) the center row of q_1^{r1} ; (d) the center row of q_1^{r2} by median filtering; (e) q_1^R recovered by k -means clustering.

3.3.3.2 Random noise removal

Although the normalization step is effective, it does not help on noise suppression. Thus, the second step is denoising. In order to preserve boundaries between different periods and different objects, the normalized QGC image q_i^{r1} is filtered by a median filter (5×5 pixels), with a result of q_i^{r2} . As an example, q_1^{r2} of the flat board is shown in Fig. 3.6(d), which is deemed to be satisfactory.

3.3.3.3 k -means clustering

Observing Fig. 3.6(d) it could be noticed that the normalization is partial: q_1^{r2} ranges approximately from -1.3 to 1.3, rather than from -1 to 1. The partial normalization is due to the fact that $\frac{B(x,y,f_{Ci})}{B(x,y,f_x)}$ often deviates from the ideal value of 1, an issue which is further clarified by the simulation in Fig. 3.7. After the projector defocusing, i.e., low-pass filtering, the amplitude of fringe patterns will shrink as shown in Fig. 3.7(a), but the amplitude of the code patterns q_i^c remain about the same, as shown in Fig. 3.7(b). Consequently, $\frac{B(x,y,f_{Ci})}{B(x,y,f_x)} \geq 1$. Since $\frac{B(x,y,f_{Ci})}{B(x,y,f_x)}$ is different under different defocusing levels, q_i^{r2} also varies as shown in Fig. 3.7(c). Similarly, different q_i^{r2} will be obtained when different fringe periods are used. Therefore, if the simple thresholding is applied to recover the gray code, different thresholds should be used for systems under different defocusing levels or with different fringe periods, which is troublesome.

Figure 3.7 Simulation of projector defocusing on the binary patterns. (a) Defocused fringe patterns with different σ ; (b) defocused bQGC patterns with different σ ; (c) normalized bQGC patterns by the patterns in (a) and (b); (d) the same GC recovery of (c) by k -means clustering.

We propose to use k -means clustering to partition q_i^{r2} automatically as a general method for different systems. The k -means clustering classifies the pixels in q_i^{r2} into k sets $S = \{S_0, S_1, \dots, S_{k-1}\}$ with $c = \{c_0, c_1, \dots, c_{k-1}\}$ as their centroids, in which the sum of the squared distances in all sets is minimized:

$$\sum_{j=0}^{k-1} \sum_{(x,y) \in S_j} \| q_i^{r2}(x, y) - c_j \|^2 \rightarrow \min. \quad (3.13)$$

As the k -means clustering is only applied to the measured objects, the non-object areas are first marked out by the method in [127]. For the QGC reconstruction, $k = 4$ is set, with $\{-1, -1/3, 1/3, 1\}$ as the initial guess of $\{c_0, c_1, c_2, c_3\}$. Once $\{S_0, S_1, S_2, S_3\}$ has been obtained, the QGC code is reconstructed as $q_i^R(x, y) = j$ if $(x, y) \in S_j$ ($j = 0, 1, 2, 3$), i.e., all pixels in S_j has a code value of j .

As the simulation shown in Fig. 3.7(d), by the k -means clustering, the same gray code is recovered for systems under different defocusing levels. We also partition q_1^{r2} in Fig. 3.6(d)

by k -means clustering and the corresponding gray code q_1^R is shown in Fig. 3.6(e). When the gray codes q_i^R of all QGC images are identified, the phase order is determined by the coding mechanism of QGC and the unwrapped phase are obtained by Eq. (3.6).

3.4 QGC with continuity/geometry constraints

To further increase the acquisition speed, we attempt to remove the last pattern (q_M^{bP}) in QGC phase unwrapping. However, by simply removing q_M^{bP} , the order uniqueness is lost. As an example, in the 16-period phase unwrapping ($M = 2$), q_2^{bP} is removed and only q_1^{bP} is used. Consequently, only q_1^R can be recovered, as shown in Fig. 3.8(a). The C value is computed by Eq. (3.7) again with $q_2^R = 0$, resulting in $C = q_1^R$. The corresponding QGC-unwrapped phase Φ_A , as shown in Fig. 3.8(b), is only partially unwrapped and has many 2π and 4π jumps, which defeats the original purpose of QGC unwrapping. Indeed, such jumps are often indistinguishable from physical discontinuities of the measured objects. However, such a problematic situation can be easily avoided by re-designing the gray code of q_{m-1}^{bP} to a loop of ‘0123’, i.e., simple code [85], as shown in Fig. 3.8(c). The corresponding Φ_A is shown in Fig. 3.8(d). Although the phase has not been fully unwrapped yet (due to the removal of q_M^{bP}), all the remaining phase jumps have a jumping value of $4^{M-1} \times 2\pi$, which is 8π in this example. This jumping value is more distinguishable from physical discontinuities of the measured objects. In fact, in real applications, we often use 64-periods ($M = 3$) for projection. The phase jumping value is 32π , which can be very easily identified.

Figure 3.8 Simulation of 16-period QGC phase unwrapping with only one pattern. (a)-(b) q_1^R in conventional QGC and the corresponding Φ_A ; (c)-(d) re-designed q_1^R with simple code and the corresponding Φ_A .

Consequently, the resulting Φ_A can be easily unwrapped by any existing SPU algorithms based on the continuity constraints – if two adjacent pixels have a jump of $4^{M-1} \times 2\pi$, we add $4^{M-1} \times 2\pi$ to the phase to enforce the phase continuity. Note that the result of SPU is a relative phase map depending on where the phase unwrapping starts, i.e., the seed pixel location. To recover an absolute phase map, we require the object to have a pixel within $[0, 4^{M-1}T]$ that can be used as a seed pixel for SPU. Otherwise, we are able to recover the correct local geometry within an object but may have a wrong depth of the object overall – an issue that will be demonstrated later.

Geometry-constrained phase unwrapping (GcPU) is another solution to unwrap Φ_A . We assume that the minimum depth z_{min} of the volume of interest is known, which means that the measured objects will not be placed with a distance less than z_{min} from the FPP system. We then have the following geometry constraint for Φ :

$$\Phi \geq \Phi_{min} = h(z_{min}), \quad (3.14)$$

where h is a function obtained by system calibration which has been elaborated in [125].

This constraint provides additional information to further unwrap Φ_A as:

$$\Phi(x, y) = t(x, y) \times 4^{M-1} \times 2\pi + \Phi_A(x, y), \quad (3.15)$$

where t is determined by assuming the phase range of the measured object is within $[\Phi_{min}, \Phi_{min} + 4^{M-1} \times 2\pi)$:

$$t(x, y) = \left\lceil \frac{\Phi_{min}(x, y) - \Phi_A(x, y)}{4^{M-1} \times 2\pi} \right\rceil. \quad (3.16)$$

SPU and GcPU are demonstrated and compared in Fig. 3.9, where 64-period fringe patterns are projected and the code values in q_2^{bP} is rearranged to a simple code as mentioned earlier

so that the phase can be unwrapped by only two QGC patterns. Quality-guided phase unwrapping [74] is used as the SPU algorithm. In the first example shown in Figs. 3.7(a) and 3.7(b), a face mask is measured. As the object overlaps with the region $[0, 4^{M-1}T]$, the wrapped phase Φ_A is absolutely unwrapped by SPU, with Φ_1 as its result. GcPU also provides the correct unwrapping result Φ_2 . In the second example shown in Figs. 9(c)-9(f), two objects, a hand and a teddy-bear, are measured simultaneously. As the hand overlaps with the region $[0, 4^{M-1}T]$, it is correctly unwrapped by SPU. However, although the teddy-bear has the correct local geometry after SPU unwrapping, the entire object has a phase error of 32π , which cannot be removed by SPU. On the other hand, GcPU recovers a correct unwrapped phase for all the objects since it is a pixel-independent method. Note that the artifacts in Fig. 3.9 are phase unwrapping errors around the 2π phase jumps due to the noise and the defocused projector, which can be removed by the adaptive median filtering [88].

(a)

(b)

(c)

(d)

Figure 3.9 A comparison experiment between SPU and GcPU. (a)-(b) Unwrapped phase of a single object and its cross-section, where $\Phi_1 = \Phi_2$; (c)-(d) unwrapped phase of multiple objects by SPU and its cross-section; (e)-(f) unwrapped phase of multiple objects by GcPU and its cross-section.

In summary, the required patterns can be further reduced to $\lceil \log_n f \rceil - 1$ in our PS-GC framework by continuity/geometry constraints. The continuity constraint works perfectly under the condition that the object has a pixel within $[0, 4^{M-1}T]$ that can be used as a seed pixel for SPU. If this condition is not satisfied, SPU results have overall deviations for different objects. Note that there is a mixed use of temporal and spatial phase unwrapping, where the spatial unwrapping works robustly in this special case. On the contrary, GcPU always works as the technique remains purely temporal, but the additional cost of pre-calibrate the minimal phase as shown in Eq. (3.14) is needed.

3.5 Experiments

To verify the proposed GC phase unwrapping framework, a high-speed FPP system is set up. This system uses a TI DLP Light Crafter 4500 projector (1140×912 pixels) and a Basler acA2000-165um camera combined with a Navitar Zoom 7000 lens (1200×900 pixels). The projector and the camera are synchronized at a speed of 150 frames per second. The projector is slightly defocused and binary patterns are projected. In the following experiments, we verify the proposed framework with $n = 4$ and the geometry constraint. When the continuity constraint is used, the result is the same as the simulation in Section 3.4. Since the proposed PS-GC framework requires 3 patterns for the PS algorithm and 2 patterns for QGC phase unwrapping, total 5 patterns are required to recover one frame of 3D measurement, and the speed of measurement is 30 frames per second.

Figure 3.10 Experimental results of the proposed framework. (a) Photograph of the measured object; (b) one of the PS fringe images; (c)-(d) QGC images; (e) wrapped phase; (f) unwrapped phase with errors (red points); (g) zoom-in unwrapped phase; (h) unwrapped phase after filtering; (i)-(k) different views of the 3D profile reconstructed by (h).

A fish model is measured as shown in Fig. 3.10. A snapshot of the model is shown in Fig. 3.10(a). To obtain the wrapped phase, three binary phase-shifting patterns with a fringe period of 18 pixels are generated by the dithering technique [120] and projected onto the object. The patterns cover 50.7 (i.e. $912/18$) periods due to the projector resolution. One of the phase-shifting images is shown in Fig. 3.10(b). The computed wrapped phase is shown in Fig. 3.10(e). Two binary patterns for phase unwrapping are generated by the method proposed in Section 3.3.2 and the corresponding GC images are shown in Figs. 3.10(c) and 3.10(d). Based on the extracted gray codes and the geometry constraint, the phase is unwrapped as Fig. 3.10(f) shows. Due to the imperfection of the defocused patterns and random noise, alignment errors at 4,190 pixels (emphasized using red points) occur around the 2π phase jumps, with one zoom-in part shown in Fig. 3.10(g). The alignment errors can

be removed by the adaptive median filtering. The filtered result is shown in Fig. 3.10(h). Based on the calibration information, the unwrapped phase map with median filtering is converted to a 3D profile of the object, different views of which are shown in Figs. 10(i)-(k).

For comparison, we also unwrap the phase by TGC following the procedure described in [78]. 4 TGC patterns are projected on the object as shown in Figs. 3.11(a) – 11(d). The gray codes recognized by thresholding are used to unwrap the phase, with the result and a zoom-in part shown in Figs. 3.11(e) and 3.11(f), respectively. The TGC phase unwrapping results in even more errors (8711 erroneous pixels) due to the inferiority of simple thresholding. We use the same median filtering method to remove the errors as shown in Fig. 3.11(g). The filtered results of TGC and the QGC method are different at only 20 pixels and thus are almost identical.

Figure 3.11 Experimental results of TGC phase unwrapping. (a)-(d) TGC images; (e) unwrapped phase; (f) zoom-in unwrapped phase; (g) unwrapped phase after filtering.

The same comparison experiment is conducted on multiple objects with more complex geometries, and the reconstructed 3D profiles by QGC with the proposed framework is shown in Fig. 3.12. The results of the QGC and TGC are again almost identical. These experiments prove that the proposed QGC method results in an equivalent phase map as the TGC method but also reduces the required GC patterns from 4 to 2 for 64-period phase unwrapping.

Figure 3.12 Reconstructed 3D profile on multiple objects by the proposed framework with $n = 4$. (a) and (d) Photograph of the measured objects; (b) and (e) reconstructed 3D profiles of (a) and (d) with the proposed QGC after filtering, respectively, (c) and (f) zoom-in 3D profiles.

3.6 Discussion

The above simulations and experiments prove the effectiveness of the proposed framework when $n = 4$. This framework can be also easily applied for the GC methods with different numbers of codes n . The required patterns for the methods with different n are calculated for a typical frequency $f = 64$ as shown in Table 3.1. It is noticed that the required patterns are constantly reduced when we increase n for 2 to 4. However, the reduction speed slows down when $n > 4$, and a new reduction of the patterns happen when $n = 8$. It is highlighted that the number of required patterns is calculated under the situation without continuity/geometry constraints in Table 3.1. When the continuity/geometry constraints are used, the number of required patterns comes to $M - 1$. In such case, only 1 pattern is required for octal GC phase unwrapping ($n = 8$).

Table 3.1 The number of required patterns for different numbers of codes when $f = 64$.

n	2	3	4	5	6	7	8
$M = \lceil \log_n f \rceil$	6	4	3	3	3	3	2

We thus measure an object with the octal GC phase unwrapping and the resultant depth map is shown in Fig. 3.13(a). As a comparison, the depth map obtained by QGC is shown in Fig. 3.13(b). In both the two results, the adaptive median filtering is applied to remove the errors around the 2π phase jumps. In octal GC phase unwrapping, GC reconstruction becomes more difficult with the unavoidable noise and the defocusing effect. Compared to QGC, there are more clusters should be classified in the gray code patterns and the distance between different cluster is smaller in the octal GC. As a result, even with filtering, there are still many errors in the octal GC phase unwrapping as shown in Fig. 3.13(a). To further enhance the speed by octal GC phase unwrapping, a more robust GC reconstruction algorithm should be proposed.

Figure 3.13 Reconstructed 3D depth maps. (a) The result of octal phase unwrapping, (b) the result of QGC.

3.7 Conclusion

A novel GC phase unwrapping framework is proposed for a high-speed FPP system with a defocused projector. In the proposed framework, a binary pattern generation method using weighted optimization and a normalization-denoising-clustering process for GC reconstruction are developed. Besides, an optional continuity/geometry constraint for the GC phase unwrapping is also presented, which further increases the acquisition speed. Simulations and experiments prove the effectiveness of the framework for QGC where $n = 4$. Although the measurement speed can be improved when n is increased to 8, the success

rate of decoding from octal GC patterns is decreased due to the noise and defocusing effect. Thus, $n = 4$ is suggested for GC phase unwrapping in bFPP.

Chapter 4 Phase unwrapping with different orders of codes

4.1 Introduction

The SC [85] and GC [86] phase unwrapping are two similar TPU methods widely used in FPP, in which a group of SC/GC patterns with different gray levels are projected on the object to code the pixels in the same period with a sequence of simple/gray codes which is further converted to the fringe order used for phase unwrapping. In n -ary SC/GC, n different gray levels are projected representing n different codes. The only difference between SC and GC is the different orders of codes in the two methods. However, the influence of different orders remains unclear in high-accuracy and high-speed measurement, which motivates us to compare n -ary SC (n SC) and n -ary GC (n GC) comprehensively.

In both n SC and n GC phase unwrapping, due to the noises and projector defocusing, the 2π phase jumps in φ and code jumps cannot be aligned exactly as explained in Section 2.1.1, which may result in phase unwrapping errors. In n GC, all the errors can be treated as impulse noises as the width of the errors is around 2 pixels due to the specially designed order of codes. While in n SC, parts of the errors cannot be treated as impulse noises as their width increases to around 10 pixels. Therefore, such errors are more complex and more difficult to be removed in SC. In this chapter, an identification and binary classification (IBC) method is proposed to convert the errors into impulse noises. As a result, all the errors in n SC and n GC unwrapped phase are impulse noises and can be removed by AMF easily. Besides, the different orders of codes also result in different numbers of the required patterns. As explained in Section 3.4, n SC has the potential to incorporate the continuity/geometry constraints, thus requires fewer patterns and has a higher measurement speed than n GC.

In our comparison, different factors related to the accuracy and speed are considered, including the strategies on using IBC and AMF for error removal, the base n , the step

heights and invalid regions on the measured objects, defocusing level and noise level of the system. With comprehensive comparison by simulations and experiments, we find that (i) in general, both n SC with IBC-AMF and n GC with AMF perform similarly and satisfactorily; (ii) when accuracy is critical, binary SC (BSC) with AMF and binary GC (BSC) with AMF are outstanding candidates; and (iii) when the speed is critical, quaternary simple code (QSC, n SC with $n = 4$) with IBC-AMF is suggested to be incorporated with continuity/geometry constraints.

4.2 Principles of n SC and n GC phase unwrapping

A high-quality unwrapped phase by PS and n SC/ n GC phase unwrapping usually requires the following three steps. First, N -step phase-shifted fringe patterns are projected to calculate the wrapped phase. Second, n SC/ n GC are projected to encode the phase orders which are then used to unwrap the phase. Third, error removal methods should be used to remove the phase unwrapping errors around the code jumps. They are introduced one by one, followed by some discussions.

4.2.1 N -step PS algorithm

The N -step PS fringe patterns are sinusoidal and can be expressed as

$$I_i(x, y) = a(x, y) + b(x, y)\cos[\varphi(x, y) + 2\pi(i - 1)/N], i = 1, 2, 3, \dots, N, \quad (4.1)$$

where (x, y) denotes the spatial coordinate of the camera; a is the background intensity; b is the fringe amplitude; φ is the phase containing the depth information of the object. The wrapped phase can be calculated by a least-squares algorithm as

$$\varphi(x, y) = -\tan^{-1}\left\{\frac{\sum_{i=1}^N I_i(x, y)\sin[2\pi(i-1)/N]}{\sum_{i=1}^N I_i(x, y)\cos[2\pi(i-1)/N]}\right\}. \quad (4.2)$$

In the binary defocusing technique, the sinusoidal patterns are generated by defocusing designed binary patterns [77], such as squared binary patterns, PWM-based patterns, and binary dithering patterns. As an example, one of the captured fringe patterns of a flat board from a binary defocused system and the wrapped phase are shown in Figs. 4.1(a) and 4.1(b), respectively.

Figure 4.1 N -step phase-shifting algorithm (a) A fringe pattern of a flat board, (b) the wrapped phase of (a).

4.2.2 n SC and n GC phase unwrapping

The wrapped phase φ as shown in Fig. 4.1(b) contains 2π phase jumps which should be unwrapped. The idea of n SC/ n GC is to code and project the phase orders k so that the true phase Φ can be easily determined as

$$\Phi(x, y) = \varphi(x, y) + k(x, y) \times 2\pi. \quad (4.3)$$

The coding principle is much like generating numbers from digits, which can be expressed as

$$C(x, y) = \sum_{i=1}^M c_i(x, y) \times n^{M-i}, \quad (4.4)$$

where n is the base of digits and thus also the gray levels of the code patterns; $c_i = 0, 1, \dots, n-1$ is the digit value and often called as code; M is the number of digits; C is the synthesized code value and often called as codeword. To uniquely code f periods in a fringe pattern, the number of n SC/ n GC patterns is $M = \lceil \log_n f \rceil$. The mapping between the codeword $C(x, y)$ and the phase order $K(x, y)$ depends on the order of codes used and will be elaborated in the next paragraph.

Figure 4.2 illustrates the coding principles of n SC and n GC with $n = 4$ and $M = 2$, to code $C = 0, 1, 2, \dots, 15$. The n SC codes as shown in Fig. 4.2(a) repeat the values of 0, 1, 2, and 3 in the code patterns and are considered as the simplest and most natural order of codes. Using Eq. (4.3), the desired C values from 0 to 15 are successfully generated, which can be directly used as phase orders k , i.e., $k(x, y) = C(x, y)$. It can be noticed that n SC code patterns contain two types of code jumps: ± 1 code jumps and code jumps from $n-1$ to 0,

which are named as J1 (marked by green) and J2 (marked by red) code jumps in this chapter, respectively. In real cases, due to projector defocusing and the random noise, code jumps in c_i cannot be aligned with the 2π phase jumps in φ exactly, which results in phase unwrapping errors. To reduce the errors, nGC rearranges the order of codes to remove J2 jumps, as shown in Fig. 4.2(b). Since fewer errors are introduced by the fewer code jumps, nGC is more widely used. Nevertheless, it is also noted that the generated nGC codewords $C(x, y)$ no longer run in sequence and thus cannot be used directly as $k(x, y)$. In practice, a look-up table is built to get the desired k as $k(x, y) = \text{LUT}[C(x, y)]$.

Figure 4.2 Schematic diagram of the nSC (a) and nGC (b) phase unwrapping with two patterns where $n = 4$.

Returning to the example in Fig. 4.1, where the width of the projected patterns is 912 pixels, each fringe has 20 pixels, and the projected pattern contains $f = 912/20 = 45.6$ fringe periods. Therefore, $k = 0, 1, 2, \dots, 45$, and if quaternary patterns ($n = 4$) are projected, $m = \lceil \log_n f \rceil = 3$ code patterns are required. One of the nSC patterns and one of the nGC patterns are shown in Figs. 4.3(a) and 4.3(b), respectively, where different code jumps are emphasized by different colors. The nSC unwrapped phase, Φ_{nSC} , and the nGC unwrapped phase, Φ_{nGC} , are shown in Figs. 4.3(c) and 4.3(d), respectively, in which there are J1 erroneous pixels (EPs) around the J1 code jumps (in green) and J2 EPs around the J2 code jumps (in red). As expected, the J2 EPs are always wider and more severe than the J1 EPs, which explains the practical preference of nGC over nSC .

Figure 4.3 Comparison of nSC and nGC phase unwrapping. (a) An nSC pattern, (b) an nGC pattern, (c) Φ_{nSC} with 14,334 J1 EPs and 54,665 J2 EPs, (d) Φ_{nGC} with 21,791 J1 EPs.

4.2.3 Adaptive median filtering

Because of the EPs as shown in Figs. 4.3(c) and 4.3(d), noise suppression is necessary, for which different methods have been proposed which are reviewed in Section 2.1.1. Among these methods, the adaptive median filtering (AMF) is simpler and thus more robust for complex object measurement, and does not require additional patterns. The AMF method [88] suppresses the noise through two steps: EP detection and EP correction, both of which focus on narrow errors around 2 pixels which can be deemed as impulse noises. In the EP detection step, a small window (7×7) around a pixel is selected as the local area of that pixel. A pixel will be considered as an EP if it is quite different from the other pixels in its local area. In the EP correction step, a median filter is applied only to the detected EPs. The two steps in AMF are repeated with an increasing filter size from small to large until there are no EPs or the number of EPs no longer reduces. The AMF method prioritizes a small filter size and maximizes the chance of removing the noises without destroying the step

height of a measured object. It is noted that there are still a few isolated errors around step heights even when AMF is applied, for which we apply 3×3 median filtering on the unwrapped phase after AMF. Since the 3×3 window is very small, the isolated errors will be removed, and the details of the object will be preserved. For simplicity, we call the process of AMF with the additional 3×3 median filtering as AMF in this chapter. Even AMF performs well on impulse noise removal, AMF is not able to effectively remove the wide errors like the J2 EPs in n SC phase unwrapping which cannot be deemed as impulse noises.

As an example, AMF is applied to both Φ_{nSC} and Φ_{nGC} as shown in Fig. 4.4. For Φ_{nSC} , AMF removes all the J1 EPs but fails to remove the J2 EPs, while for Φ_{nGC} , all the EPs are removed. The J2 EPs in Φ_{nSC} are around 10 pixels in horizontal and always wider than the J1 EPs. As a result, the J2 EPs cannot be detected by the small 7×7 window used for J1 EP detection. In addition, we cannot use a large window for EP detection which may treat the step-heights on the objects as EPs, unlike in the J1 case. Therefore, a more effective method should be proposed to remove the EPs in n SC phase unwrapping.

Figure 4.4 Effect of AMF. (a) Φ_{nSC} after AMF with 42,663 J2 EPs, (b) Φ_{nGC} after AMF without EPs.

4.2.4 Discussions

Because n SC and n GC use different codewords, they have different performance in accuracy and speed:

- (i) Accuracy. Compared with nGC , nSC has additional J2 code jumps causing J2 EPs, which cannot be easily removed by the existing filters.
- (ii) Speed. Both the nSC and nGC methods require $M = \lceil \log_n f \rceil$ patterns to unwrap a phase with f periods. If the first nSC/nGC pattern is removed, the code uniqueness is lost and Φ_{nSC}/Φ_{nGC} is partially unwrapped as shown in Fig. 4.5, where the differences between Φ_{nSC} and Φ_{nGC} are emphasized by red rectangles. The phase jumps in Φ_{nSC} equals to 8π , while the phase jumps in Φ_{nGC} equals to 2π . Compared with 2π phase jumps, the 8π phase jumps are more distinguishable from physical discontinuities of the measured objects, which can be removed by continuity/geometry constraints [128]. Therefore, the nSC phase unwrapping has the potential to be incorporated with continuity/geometry constraints, which will reduce the number of the required patterns and increase the measurement speed.

Figure 4.5 Schematic diagram of the nSC and nGC phase unwrapping with only one pattern ($n = 4$). The differences between Φ_{nSC} and Φ_{nGC} are emphasized by red rectangles.

Before further comparison, we will analyze the special distribution of the J2 EPs in nSC phase unwrapping and propose an IBC+AMF method to remove J2 EPs. Then, nSC and nGC will be comprehensively compared by considering various factors, including the strategies on using IBC and AMF for error removal, the base n , the step heights and invalid regions on the measured objects, defocusing level and noise level of the system.

4.3 The proposed identification and binary classification method for nSC phase unwrapping

Since nSC is obviously inferior to nGC due to J2 errors, this section aims to improve the performance of nSC . Based on a detailed analysis of J2 errors, a two-step identification-classification method is proposed.

4.3.1 Analysis of errors in nSC phase unwrapping

The process of n SC phase unwrapping of a flat board is shown in Fig. 4.6 to explain the cause of the EPs. For clarity, we only show all the variables at the 500th row and from the 557th column to the 987th column. For both n SC/ n GC, the captured code patterns will undergo our normalization-denoising-clustering process [128], in which the normalization is used to avoid the influence of the non-uniform texture and environment light, the denoising is used to remove the random noises, and the k -means clustering is used to classify an n SC/ n GC pattern into n sets for different codes (c_i). After this process, the following EPs are observed in Fig. 6(a): (i) for a J1 code jump from c_i to $c_i + 1$, either c_i will be mistaken as $c_i + 1$, or $c_i + 1$ as c_i , but not both; (ii) for a J2 code jump from 3 to 0, due to the low-pass nature of the measurement system, it becomes a step from 3 to 2 to 1 and finally to 0. By overlaying the extracted codes with the wrapped phase φ in Fig. 4.6(b), it is then understandable that a J1 EP has a single impulse while a J2 EP has a pair of opposite impulses, as shown in Fig. 4.6(c). The J1 EPs are also narrower than the J2 EPs: in our example, the former has 1 or 2 pixels while the latter has 9 pixels around one code jump.

Figure 4.6 Phase unwrapping errors in n SC phase unwrapping. (a) Cross-sections of the normalized-denoised pattern and c_i , (b) cross-sections of φ and c_i , (c) a cross-section of Φ_{nSC} .

Thus, we have two facts regarding the J1 and J2 code jumps: (i) these jumps behave differently, as described in the previous paragraph, and (ii) the J1 EPs can be successfully removed by AMF, as described in Section 4.2.3. These two facts motivated us to identify the J2 EPs for special processing. Once the J2 EPs are identified, we will perform a binary clustering to re-classify the code either to be 3 or 0 to avoid the occurrence of 1 and 2

around the J2 code jumps. As a result, the J2 EPs will be transferred to impulse noises, which can then be removed by AMF.

4.3.2 J2 EP identification

Our J2 EP identification algorithm has three simple and natural steps: gradient-based initial identification, and J2 properties based false pixel exclusion and morphology-based missing pixel inclusion.

Gradient-based initial identification Since the J1 code jumps in nSC always equal to $+1$, while the J2 code jumps always equal to $1 - n$, they can be differentiated by the gradient of the normalized-denoised nSC pattern, $G = [G_x, G_y]^T$, which is calculated by a Sobel gradient operator [129]. The magnitude of the gradient, which is referred to as a gradient image, is calculated by

$$M(x, y) = \sqrt{G_x^2(x, y) + G_y^2(x, y)}. \quad (4.5)$$

As shown in Fig. 4.7(a), the gradient image of the flat board consists of three parts: pixels with a large gradient (in white) representing the J2 code jumps, pixels with a medium gradient (in grey) representing the J1 code jumps, and pixels with a small gradient (in black) representing the smooth region without code jumps. The cross-section of the gradient image is shown in Fig. 4.7(b). The pixels around J2 code jumps, which will be suspected to the J2 EPs, can be detected by thresholding, and in this example, the threshold, $Thre$, equals 0.6. We use a mask with the same size as the nSC patterns to describe the J2 EPs, where the J2 EPs equal to 1 while the others equal to 0.

Figure 4.7 J2 EP identification based on the gradient map. (a) M , (b) a cross-section at the 500th row of M , (c) J2 EPs.

J2-properties-based false pixel exclusion After the initial identification, there may be pixels wrongly identified due to step heights on the object and noises. An identified pixel (x, y) is directly excluded if it does not satisfy the following J2 properties: (i) value of code: since the J2 code jumps from $n - 1$ to 0, the code of a J2 EP is different from 0 and $n - 1$; (ii) gradient of value: since the J2 code jumps down if vertical fringe patterns are used, $G_x(x, y) < 0$; (iii) neighboring code: it should have neighboring pixels (x', y') that $\exists c_i(x', y') = 0 \wedge \exists c_i(x', y') = n - 1$. The size of the neighborhood region is selected as 21×21 since the width of the J2 EPs is around 10 pixels (see Section. 3.1). In addition, a median filter with a size of 3×3 pixels to the mask to further remove false pixels caused by random noise.

Morphology-based missing pixel inclusion In the gradient-based initial identification, only the pixels around the center of J2 code jumps with a large gradient can be detected. To ensure the J2 EPs around the edges of the J2 code jumps and missed pixels due to noises can be also detected, the dilation operation in morphological image processing is applied to “thicken” the J2 EPs and fill the holes in the mask. Considering the width of the J2 EPs is about 10 pixels, we select a disk-shaped structuring element with a radius of 5 in the dilation. By dilation, all the pixels around J2 code jumps are suspected, but it also takes in some non-error pixels with $c_i(x, y) = 0$ or $n - 1$ which however can be easily removed from the mask. With this three-step identification, the J2 EPs in Fig. 4.7(a) are found and shown in Fig. 4.7(c).

4.3.3 Binary classification

As the code of pixels around J2 code jumps can only be 0 and $n - 1$, we re-classify the identified J2 EPs into two classes according to the normalized-denoised n SC patterns by simple thresholding and assign new codes, 0 and $n - 1$, to these pixels. The threshold for binary classification is the average of the mean values for code 0 and code $n - 1$, which have been calculated in the previous k -means clustering step. The cross-sections of c_3 before and after binary classification are shown in Figs. 4.8(a) and 4.8(b), respectively, with the differences emphasized in red. After binary classification, the pixels with $c_i(x, y) \neq$

0 and $n - 1$ around J2 code jumps are removed, transferring the J2 EPs to J1 EPs where $c_i = 0$ are mistaken as $n - 1$, or $c_i = n - 1$ as 0.

Figure 4.8 Code extraction with IBC. (a) A cross-section of c_3 before IBC, (b) a cross-section of c_3 after IBC.

Our two-step filtering method, i.e., J2 EP identification and binary classification, is called IBC for convenience. IBC is applied to each n SC pattern except the first one where J2 code jumps do not exist. The new n SC unwrapped phase is calculated by the new c_i after IBC. As shown in Fig. 4.9(a), the remaining 18,921 EPs in Φ_{nSC} are all J1 EPs whose width is around 2 pixels. A cross-section of Φ_{nSC} after IBC is shown in Fig. 4.9(b) in which the left J2 EP is removed and the right J2 EP is converted to a J1 EP, compared with Φ_{nSC} in Fig. 4.6(c). Although those J1 EPs converted from J2 are “severe”, they are not more difficult to remove by AMF. As shown in Fig. 4.9(c), AMF is applied to the Φ_{nSC} after IBC and the J1 EPs are removed which is the same as the Φ_{nGC} after AMF in Fig. 4.4(b). The cross-sections of Φ_{nSC} after IBC and AMF are shown in Fig. 4.9(d). With IBC and AMF, the n SC method is able to perform as accurately as n GC.

Figure 4.9 nSC phase unwrapping with IBC and AMF. (a) Φ_{nSC} after IBC with 18,921 EPs, (b) a cross-section of (a), (c) Φ_{nSC} after IBC and AMF, (d) A cross-section of (c).

4.4 Comparison by Simulations

The proposed IBC improves the performance of nSC , makes nSC comparable with nGC in theory, and also makes a further comprehensive comparison towards real measurement more meaningful. We compare nSC and nGC phase unwrapping through simulation first and proceed to the experimental comparison in the next section. In real measurement, there are step heights on the discontinue surfaces and invalid regions due to the shadows or low reflectivity, resulting in difficulties in the error removal. To simulate the real situation, we create a simple smooth object in Section 4.4.1 and add step heights and invalid regions on it in Section 4.4.2. Different error removal strategies are compared in Section 4.4.3. The influences of the defocusing level and noise level are discussed in Section 4.4.4. Besides, the base n influences both the accuracy and speed of the nSC/nGC and thus is added to the comparison in Section 4.4.5.

4.4.1 Simple smooth object

A simple linear phase is simulated first. Phase-shifted patterns (1200×1200 pixels) are generated with 30 vertical fringes in each frame and 40 pixels in each fringe period. The ideal wrapped phase is calculated by sinusoidal PS patterns, one of which is shown in Fig. 4.10(a). The phase order of the fringe periods increases linearly from 0 to 29 along the horizontal direction. We inversely calculate the projected n SC and n GC patterns according to the phase order where $n = 4$ is taken. One of the n SC patterns is shown in Fig. 4.10(b). So far, the sinusoidal patterns and n SC/ n GC patterns are ideal without defocusing and noises. To simulate the real patterns in a binary defocused system, the patterns are binarized by the dithering algorithm [130], defocused by a Gaussian filter with a standard deviation of $\sigma_{defocus} = 6$, and contaminated by adding normally distributed random noises with a mean of 0 and a standard deviation of $\sigma_{noise} = 0.1$. The simulated fringe pattern and n SC pattern are shown in Figs. 4.10(c) and 4.10(d), respectively. For a better illustration, cross-sections of the ideal n SC pattern and the simulated n SC pattern at the 500th row are shown in Figs. 4.10(e) and 4.10(f), respectively. The ground truth of the unwrapped phase shown in Fig. 4.10(g) is calculated by the ideal PS patterns and the ideal n SC/ n GC patterns. Both n SC with IBC and AMF and n GC with AMF are able to reconstruct the ideal codes from the simulated real patterns, as shown in Fig. 4.10(h). This simulation further verifies the proposed method in Section 4.3 with the presence of noise and also sets up a platform for the comprehensive comparisons in the following sections.

Figure 4.10 Simulation of a simple linear phase. (a) One of the ideal fringe patterns, (b) one of the ideal nSC patterns, (c) one of the simulated fringe patterns, (d) one of the simulated nSC patterns, (e) a cross-section of the ideal nSC pattern, (f) a cross-section of the simulated nSC pattern, (g) the ideal unwrapped phase, (h) the unwrapped phase by nSC with IBC and AMF or nGC with AMF.

4.4.2 Step heights and invalid regions

Step heights and invalid regions are added into both the fringe patterns and nSC/nGC patterns as shown in Fig. 4.11(a)-4.11(c). Four circular regions with a radius of 135 pixels centering at (300, 300), (900, 300), (300, 900), and (900, 900) are modified to generate π , 3π , 5π , and 7π step heights, respectively. For invalid region simulation, NaNs (Not-a-

Number) are given to four circular regions with a radius of 135 pixels in the wrapped phase and nSC/nGC patterns, centering at (600, 300), (300, 600), (900, 600) and (600, 900). With the new fringe patterns and new nSC/nGC patterns, both the unwrapped phase by nSC and nGC methods (without any filtering) are calculated accordingly and shown in Fig. 4.11(d) and 4.11(e), respectively. Compared with Φ_{nGC} , the errors in Φ_{nSC} is severer as expected.

Figure 4.11 Simulation of a complex phase with invalid regions and step heights. (a) A simulated fringe pattern with step heights and invalid regions, (b) - (c) simulated nSC and nGC patterns with step heights and invalid regions, (d) Φ_{nSC} , (e) Φ_{dGC} .

4.4.3 Influence of different error removal strategies

Since the performance of Φ_{nSC} and Φ_{nGC} will be improved by error removal, five different methods are compared: (i) Φ_1 : nSC , (ii) Φ_2 : $nSC + AMF$, (iii) Φ_3 : $nSC + IBC + AMF$, (iv) Φ_4 : nGC , and (v) Φ_5 : $nGC + AMF$. The ground truth of the unwrapped phase (Φ_0) is calculated by the ideal fringe patterns and ideal Φ_{nSC}/Φ_{nGC} patterns, which is the same as the phase we preset. The difference between the unwrapped phase of a specific method and the ground truth phase includes the wrapped phase errors due to noise during phase retrieval

and unwrapped phase errors due to code imperfection during phase unwrapping, where the latter equals to multiple 2π and thus larger than the former. Since this chapter mainly concerns the phase unwrapping errors, only the pixels with unwrapped phase errors are regarded as EPs, which are recognized if the entire phase error is larger π . The EPs in different methods are shown in red in Fig. 4.12, from which we remark that, (i) comparison between Φ_1 and Φ_4 shows that nSC is worse than nGC , which agrees with the experiment in Section 4.2.2; (ii) comparisons between Φ_1 and Φ_2 and between Φ_4 and Φ_5 show that AMF is effective in removing J1 EPs in both nSC and nGC ; (iii) comparison between Φ_2 and Φ_3 shows the proposed IBC is effective in removing J2 EPs; (iv) the results of Φ_3 and Φ_5 show the excellent performance of $nSC + IBC + AMF$ and $nGC + AMF$, both of which are recommended for use.

Figure 4.12 EPs in different methods. (a) 124,404 EPs in Φ_1 , (b) 44,865 EPs in Φ_2 , (c) 325 EPs in Φ_3 , (d) 58,337 EPs in Φ_4 , (e) 227 EPs in Φ_5 .

4.4.4 Influence of the defocusing level and noise level

In the previous comparison, the defocusing level and noise level are constant. To compare the stability of the methods under different defocusing levels and noise levels, we repeat the simulation with different $\sigma_{defocus}$ and σ_{noise} . To simulate slight to great defocusing, different $\sigma_{defocus}$ from 1 to 6 with constant $\sigma_{noise} = 0.05$ are applied to the patterns, the results of which are compared in Fig. 4.13. For different noise levels simulation, different σ_{noise} from 0 to 0.1 with constant $\sigma_{defocus} = 5$ are applied to the patterns, the results of which are compared in Fig. 4.14. For clarity, the nSC methods and the nGC methods are shown in two charts separately in each comparison. From the two comparisons, we remark that: (i) the results of Φ_1 and Φ_4 show that nSC and nGC become worse with the increase of $\sigma_{defocus}$ and σ_{noise} ; (ii) the results of Φ_2 and Φ_3 show that $nSC + IBC + AMF$ performs consistently better than $nSC + AMF$, especially for large $\sigma_{defocus}$ and small σ_{noise} , which proves the effectiveness of the proposed IBC; (iii) it is interesting that Φ_2 increases its performance with heavier noise, which is because the heavier noise makes the J2 EPs more random and easier to be detected and removed by AMF; (iv) most importantly, the results of Φ_3 and Φ_5 show that $nSC + IBC + AMF$ and $nGC + AMF$ are insensitive to both $\sigma_{defocus}$ and σ_{noise} and have much better results than the other methods; in further comparison, we will focus on $nSC + IBC + AMF$ and $nGC + AMF$ since their stable and excellent performance.

Figure 4.13 A comparison of the different phase unwrapping methods with different defocusing levels.

(a) n SC, (b) n GC.

Figure 4.14 A comparison of the different phase unwrapping methods with different noise levels. (a) n SC, (b) n GC.

4.4.5 Influence of the base n

Since the base n influences both the accuracy and speed of phase unwrapping, n SC + IBC + AMF (Φ_3) and n SC + AMF (Φ_5) with different n are compared. In our comparison, there are 30 fringes in one fringe pattern. As a result, $\lceil \log_2 30 \rceil = 5$, $\lceil \log_3 30 \rceil = 4$ and $\lceil \log_4 30 \rceil = 3$ patterns are required for the binary, ternary, and quaternary methods, respectively. Note that there are no J2 EPs in the binary code method, thus IBC is not required and we use Φ_2 instead of Φ_3 . $\sigma_{defocus} = 1$ and $\sigma_{defocus} = 6$ are applied to the patterns to simulate slightly and greatly defocusing, respectively. Due to the randomness of noise, 10 groups of random noises with $\sigma_{noise} = 0.1$ are generated and added to the defocused patterns for each method. As a result, we get 10 groups of results for each method and we use the average number of EPs to compare the methods as shown in Fig. 4.15, from which we remark that (i) considering the size of the phase (1200×1200 pixels), all the methods get a very small number of EPs (less than 0.02%); (ii) when $n = 2$, n SC + AMF and n GC + AMF get almost the same result since there are no J2 EPs in both methods; (iii) when $n = 3, 4$, n GC + AMF is slightly better than n SC + IBC + AMF; (iv) when n increases, the number of the EPs also increases. Therefore, in general, both n SC + IBC + AMF (Φ_3) and n GC + AMF (Φ_5) can be used; when accuracy is critical, BSC/BGC + AMF should be used, which has the fewest EPs; when speed is critical, QSC + IBC + AMF should be used,

which requires fewest patterns when incorporated with continuity/geometry constraints as mentioned in Section 4.2.4.

Figure 4.15 A comparison of the different phase unwrapping methods with different n . (a) Slightly defocusing ($\sigma_{defocus} = 1$), (b) greatly defocusing ($\sigma_{defocus} = 6$).

4.5 Comparison by experiments

This section compares the n SC and n GC phase unwrapping methods by experiments on different objects and with different bases n , in which both the unwrapped phase and the reconstructed 3D profile are provided. Three phase-shifted binary patterns are projected with a defocused projector to calculate the wrapped phase by the three-step PS algorithm. Both n SC and n GC patterns with $n = 2, 3, 4$ are projected to test all the methods mentioned in the simulation to unwrap the phase. In our experiment, each fringe pattern contains 45.6 periods, therefore, $\lceil \log_2 45.6 \rceil = 6$, $\lceil \log_3 45.6 \rceil = 4$ and $\lceil \log_4 45.6 \rceil = 3$. n SC/ n GC patterns are required for the binary, ternary and quaternary methods, respectively. Since it is difficult to determine the ground truth of the unwrapped phase in real experiments, we regard Φ_5 with $n = 2$ as the ground truth since it theoretically suffers from no J2 errors and the least number of J1 errors, which is also seen in the previous simulations. We compare the unwrapped phase of each method with the ground truth phase and regard the pixels with a difference larger than π as EPs. Comparisons on a simple smooth object and objects with step heights are provided in Section 4.5.1 and 4.5.2, respectively. As mentioned in Section 4.2.4, continuity/geometry-constrained phase unwrapping can be used to reduce the number of the required n SC patterns which enhances the measurement

speed. In Section 4.5.3, we compare the results of n SC with continuity/geometry constraints using two patterns and n SC/ n GC using three patterns.

4.5.1 Simple smooth object

A smooth object without step heights is measured by the n SC and n GC phase unwrapping methods. One of the fringe images and the wrapped phase by the three-step phase-shifting algorithm is shown in Figs. 4.16(a) and 4.16(b), respectively. We regard the pixels with low modulation as invalid pixels [127] which are marked in black as shown in Fig. 4.16(c). To illustrate the J1 and J2 EPs, Φ_1 , Φ_2 , Φ_3 with $n = 4$ are selected and shown in Figs. 4.16(d) – 4.16(f), respectively, where the EPs are marked in red. As expected, AMF fails to remove the J2 EPs as shown in Φ_2 , while IBC + AMF remove all the EPs effectively as shown in Φ_3 . With a calibrated system, we reconstruct the three unwrapped phases into 3D profiles. The 3D profiles and their cross-sections at the 650th row are shown in Fig. 4.17. The outliers around the profiles are due to the phase unwrapping errors.

Figure 4.16 Experiment on a smooth object without step heights. (a) Fringe image, (b) φ , (c) φ after invalid point removal, (d) Φ_1 with 49,205 EPs, (e) Φ_2 with 17,186 EPs, (f) Φ_3 without EPs.

Figure 4.17 3D reconstruction of a smooth object without step heights. (a) 3D profile by Φ_1 ; (b) 3D profile by Φ_2 ; (c) 3D profile Φ_3 ; (d) - (f) cross-sections of (a) - (c).

The results with $n = 2, 3$ show similar phenomena. The numbers of EPs of the different methods with different bases n are compared in two charts in Fig. 4.18. From the comparison, we remark that Φ_3 and Φ_5 have a perfect performance without any EPs on a smooth object for all bases n .

Figure 4.18 A comparison of the different phase unwrapping methods on a smooth object by experiment. (a) nSC , (b) nGC .

4.5.2 Objects with step heights

To compare the performance of the n SC and n GC phase unwrapping on step heights, multiple objects with step heights are measured. One of the fringe images, the wrapped phase after invalid point removal, and Φ_3 with $n = 4$ are shown in Figs. 4.19(a) – 4.19(c), respectively. Compared with the ground truth unwrapped phase, the EPs in Φ_3 which are around the step heights are marked in red. For clarity, the reconstructed 3D profile of Φ_3 and its cross-section at the 500th row are shown in Fig. 4.20. The phase unwrapping errors around the step heights are emphasized by red circles which are difficult to be avoided because of the spatial nature of AMF.

Figure 4.19 Experiment on multiple objects with step heights. (a) Fringe image, (b) φ after invalid point removal, (c) Φ_3 with 559 EPs.

Figure 4.20 3D reconstruction of multiple objects with step heights by Φ_3 with $n = 4$. (a) – (b) the 3D profile from different views, (c) a cross-section of the 3D profile.

The numbers of EPs of different methods with different bases n are compared in two charts in Fig. 4.21 which is very similar to the comparison on a smooth object in Fig. 4.18. When

step-heights are introduced, Φ_3 and Φ_5 can still get the best performance with a very small number of EPs around step heights.

Figure 4.21 A comparison of the different phase unwrapping methods on multiple objects with step heights. (a) *nSC*, (b) *nGC*.

4.5.3 *nSC* with continuity/geometry constraints

As aforementioned in Section 4.2.4, *nSC* method can unwrap the phase using fewer patterns when incorporated with continuity/geometry constraints. In this section, we take QSC as an example and use the quaternary gray code (QGC) for comparison. In the previous experiments, three QSC patterns are projected for phase unwrapping. To increase the measurement speed, we reduce the required QSC patterns to two by removing the first QSC pattern and using continuity/geometry constraints. This idea was originally proposed in [128] with its process described in Fig. 4.22. Note that IBC has been inserted into the process for better performance.

Figure 4.22 Flowchart of the *nSC* phase unwrapping with continuity/geometry constraints.

As an experiment, the two QSC patterns are shown in Figs. 4.23(a) – 4.23(b). The QSC result after the first two steps of the above process is shown in Fig. 4.23(c), where the large

phase jumps of 32π are observed as explained in Fig. 4.5(a). These large phase jumps are then removed by the continuity/geometry constraints, as shown in Fig. 4.23(d). The unwrapping is considered successful as the result is the same as Φ_3 in Fig. 4.19(c). For comparison, two QGC patterns are used and the QGC result is shown in Fig. 4.23(e). It contains many small phase jumps of 2π as explained in Fig. 4.5(b). The subsequent application of the continuity/geometry constraints removes these small phase jumps by adding a multiple of 2π to make the phase just above the minimum phase map [125], which also removes the physical discontinuities of the measured objects larger than 2π . The unwrapped phase by QGC + geometry constraints is shown in Fig. 4.23(f) and has some obvious errors, two of which are marked in red. In fact, since the small phase jumps of 2π are removed by the continuity/geometry constraints, the gray code does not contribute to the phase unwrapping at all, defeating its original purpose.

(a)

(b)

(c)

(d)

(e)

(f)

Figure 4.23 Geometry-constrained nSC phase unwrapping with two QSC images. (a) - (b) QSC images, (c) unwrapped phase by the two QSC images, (d) unwrapped phase by QSC + geometry constraints + AMF, (e) unwrapped phase by the two QGC images; (f) unwrapped phase by QGC + geometry constraints.

We also verify the QSC + geometry constraints + AMF method on objects with more complex geometries as shown in Fig. 4.24. The two QSC patterns are shown in Figs. 4.24(a)–4.24(b). Figure. 4.24(c) is the QSC unwrapped phase with the two patterns, where large phase jumps of 32π exist. The geometry constraints are then used to remove the large phase jumps and AMF is used to remove the remaining phase unwrapping errors. The unwrapped phase using QSC + geometry constraints + AMF method is shown in Fig. 4.24(d) and the corresponding 3D point cloud is shown in Fig. 4. 24(e). It is noticed that there are still some errors around the pixels with low fringe modulation even when AMF is applied, which implies that the proposed method is sensitive to the fringe modulation.

(a)

(b)

(c)

(d)

(e)

Figure 4.24 Geometry-constrained n SC phase unwrapping with two QSC images on complex objects. (a) - (b) QSC images, (c) unwrapped phase by the two QSC images, (d) unwrapped phase by QSC + geometry constraints + AMF, (e) the reconstructed 3D point cloud from (d).

4.6 Conclusion

This chapter comprehensively compares the differences between n SC and n GC phase unwrapping methods under a binary-defocused system, which explains the influences of the different orders of codes in phase unwrapping. To prepare for the comparison, an identification and binary classification method is proposed to enhance noise suppression in n SC. In the comparison, various factors are considered, including the strategies on using AMF and IBC for error removal, the base n , the step heights on the measured objects,

defocusing level and noise level of the system. From the simulations and experiments, we observe that (i) both IBC and AMF are insensitive to the defocusing level and noise; (ii) $n\text{SC} + \text{IBC} + \text{AMF}$ and $n\text{GC} + \text{AMF}$ have excellent performance with only a few EPs around step heights; (iii) when n increases, the number of the EPs and the number of required patterns increase; (iv) $n\text{SC} + \text{IBC} + \text{AMF}$ can use fewer patterns than $n\text{GC} + \text{AMF}$ when incorporated with continuity/geometry constraints without the loss of accuracy. Therefore, in general, both $n\text{SC} + \text{IBC} + \text{AMF}$ and $n\text{GC} + \text{AMF}$ can be used; when accuracy is more concerned than the speed, BSC/BGC + AMF should be used, which provides the most accurate phase unwrapping result; when extreme high speed is required by dynamic measurement, QSC + IBC + AMF + continuity/geometry constraints should be used which requires the least number of patterns for phase unwrapping.

Chapter 5 Phase unwrapping with different coding strategies

5.1 Introduction

PS for phase retrieval combined with TPU for phase unwrapping has been widely used in recent works on FPP due to their pixel-by-pixel manner. A comparative and comprehensive review [42] has been conducted on PS algorithms. Popular TPU methods include multi/two-frequency method [16], [95], phase coding (PC) method [96]–[98], and gray code (GC) method [86], [131], in which different coding strategies are used to code the phase order with different patterns as explained in Section 2.1. Among these methods, the multi/two-frequency method has been intensively reviewed [91], where shows that the two-frequency (TF) method minimizes the number of required patterns and the hierarchical multi/two-frequency method provides the best unwrapping reliability. The TF, PC, and GC methods can be easily used to replace one another in the same FPP system, but their performances have not been compared, which motivates us to evaluate them theoretically and experimentally. The other TPU methods, such as stereo-vision-based methods [101], [104] and composite-fringe-pattern-based methods [102], [103], either require an additional camera or sacrifice the accuracy of the wrapped phase, and are not covered in this chapter.

To make our comparison focused, we evaluate the accuracy performances of TF, PC, and GC methods under the scenario of high-speed measurement. High speed has been a recent common pursuit, for which, various techniques [45], [77]–[84], [132] have been proposed by either enhancing the projection speed or reducing the number of projected patterns. For the projection speed enhancement, the binary defocusing technique [45], [77] projects designed binary patterns and defocuses the projector to generate desired structured patterns, which has a projection speed of tens of kHz, in contrast to ~ 120 Hz in the traditional 8-bit pattern projection [113]. For the projection pattern reduction, in the TF and PC methods, a two-plus-one phase-shifting algorithm [79]–[81] is applied to reduce the required TPU

patterns from 3 to 2, assuming the background to be constant during the measurement. As for the GC method, the traditional binary GC method requires $\lceil \log_2 f \rceil$ patterns. Ternary [78] and quaternary [132] GC methods reduce this number to $\lceil \log_3 f \rceil$ and $\lceil \log_4 f \rceil$, respectively. Besides, geometry/continuity constrained phase unwrapping [81]–[84], [132] is combined with the TPU methods to further reduce the number of required patterns or enhance the accuracy. As a result, all the three TPU methods can unwrap a high-frequency phase with only 2 patterns. This same requirement of the pattern numbers makes a fair and reasonable accuracy comparison possible. This chapter compares the TPU methods in three situations: the traditional 8-bit focused FPP (aFPP), the high-speed binary defocused FPP (bFPP), and the geometry/continuity constrained binary defocused FPP (cFPP).

The accuracy comparison of the three TPU methods is based on their sensitivities to noise, for which noise models of the PC and GC methods are built and compared with the existing noise model of the TF method [91]. Phase unwrapping errors caused by the noise are classified into uniformly distributed errors (E^U) all over the phase map and non-uniformly distributed errors (E^N) only around particular regions. For both E^U and E^N , their distributions are analyzed and their error rates are calculated for different TPU methods and different FPP systems by simulations and experiments. In aFPP, E^U is more noticeable in TF and PC than GC when the fringe frequency is high ($f \geq 16$), and E^N is around left and right image borders in TF, while it is around 2π phase jumps in PC and GC; GC shows the best performance. In bFPP, the overall performances of the three methods are similar to those in aFPP, but there are more E^N in PC and GC than aFPP due to defocusing; TF becomes more comparable to GC. In cFPP, compared with bFPP, E^U in TF and PC is reduced greatly and E^N in TF is removed; TF thus shows the best performance. A quick guide based on our comprehensive comparisons is provided for the high-frequency phase unwrapping: for measurement where accuracy is more concerned than the speed, GC in aFPP is preferred, while for dynamic measurement pursuing extreme high speed, TF in cFPP is suggested.

5.2 Principles of temporal phase unwrapping methods

The performance of TPU is not only determined by the TPU patterns but also the PS patterns. Therefore, for the completeness of the noise models and easy following of the

subsequent sections, a general phase-shifting algorithm is briefly introduced in Section 2.1. The principles and noise models of different TPU methods are explained in Section 2.2-2.4, in which the phase unwrapping errors caused by the system noise are classified into E^A , E^B , and E^C according to their different error sources. Besides, we also summarize the differences in the TPU methods and further classify the errors into E^U and E^N according to their different error distributions in Section 2.5. Note that this section only discusses the TPU methods in aFPP, which is basic among all FPP systems.

5.2.1 N -step phase-shifting algorithm

In FPP, various PS algorithms [42] have been proposed or extended from the field of phase-shifting interferometry. Among these algorithms, the standard N -step PS algorithm is the most widely used algorithm due to its balanced performance, in which the phase-shifted fringe patterns can be expressed as

$$I_i(x, y) = A_0(x, y) + B_0(x, y)\cos[\varphi_0(x, y) + 2\pi(i - 1)/N] + \Delta I_i(x, y), i = 1, 2, 3, \dots, N, \quad (5.1)$$

where N is the total number of PS patterns; (x, y) denotes the spatial coordinate on the camera image plane; A is the background intensity; B is the fringe amplitude; φ is the phase containing the depth information of the object; the subscript of “0” in A_0 , B_0 , and φ_0 indicates they are ideal noiseless values, which is used to differentiate from the non-ideal values calculated from real measurements; we have assumed that φ_0 is the wrapped value of the true phase, i.e., $\varphi_0 \in (-\pi, \pi]$; ΔI_i is the noise which is assumed to be Gaussian with a mean value of zero and a standard deviation of σ ; the subscript of “ i ” indicates the i th fringe pattern. Through least-squares fitting, the N -step phase-shifting algorithm calculates the wrapped phase as [70]

$$\varphi(x, y) = \varphi[I_1(x, y), I_2(x, y), \dots, I_N(x, y)] = -\tan^{-1}\left\{\frac{\sum_{i=1}^N I_i(x, y)\sin[2\pi(i-1)/N]}{\sum_{i=1}^N I_i(x, y)\cos[2\pi(i-1)/N]}\right\}. \quad (5.2)$$

Since both PS and TPU are pixel-independent, the coordinates (x, y) are omitted in the rest of this chapter. In Eq. (5.2), phase noise $\Delta\varphi$ will be caused by the intensity noise ΔI_i and can be calculated as:

$$\Delta\varphi(I_1, I_2, \dots, I_N) = \sum_{i=1}^N \frac{\partial\varphi(I_1, I_2, \dots, I_N)}{\partial I_i} \cdot \Delta I_i = -\frac{2}{B_0 N} \sum_{i=1}^N \sin\left[\varphi + (i - 1) \cdot \frac{2\pi}{N}\right] \cdot \Delta I_i,$$

(5.3)

which has a mean of zero and a standard deviation of [73]

$$\sigma_\varphi = \sqrt{\frac{2}{N} \frac{\sigma}{B_0}}. \quad (5.4)$$

It is noticed from Eq. (5.4) that the quality of the wrapped phase can be improved by increasing the signal-to-noise ratio (B_0/σ) and the PS step number N . The latter is practiced when high precision is required, but $N = 3$ as the minimum required number is often used for the interest of high speed, which is also assumed in this chapter.

Since $\varphi \in (-\pi, \pi]$, the noisy wrapped phase calculated from Eq. (5.2) can be rewritten as:

$$\varphi = \begin{cases} \varphi_0 + \Delta\varphi, & |\varphi_0 + \Delta\varphi| \leq \pi \\ \varphi_0 + \Delta\varphi \pm 2\pi, & \text{otherwise} \end{cases}. \quad (5.5)$$

As a result, the phase error due to $\Delta\varphi$ is defined as:

$$\varphi - \varphi_0 = \begin{cases} \Delta\varphi, & |\varphi_0 + \Delta\varphi| \leq \pi \\ \Delta\varphi \pm 2\pi, & \text{otherwise} \end{cases}. \quad (5.6)$$

For simplicity, we denote the two types of phase errors in Eq. (5.6) as normal and abnormal phase errors, respectively. The normal phase errors randomly occur throughout the phase map, while the abnormal phase errors usually occur around $\varphi_0 = \pm\pi$, as illustrated in Fig. 5.1. The abnormal phase errors can be automatically removed in SPU, but they cause serious errors in TPU which will be explicated in Section 5.2.2-5.2.4.

Figure 5.1 Illustration of the normal and abnormal phase errors where $\sigma = 3$ and $B_0 = 60$.

The wrapped phase φ should be unwrapped into a continuous/absolute phase Φ as

$$\Phi = 2k\pi + \varphi, \quad (5.7)$$

where k is the fringe order to be determined. In the ideal case where the phase is unwrapped correctly, the standard deviation of the unwrapped phase is the same as that of the wrapped phase, i.e.,

$$\sigma_{\Phi} = \sqrt{\frac{2}{N}} \frac{\sigma}{B_0}. \quad (5.8)$$

The unwrapped phase Φ calculated from the camera image plane is shared by the projector image plane and thus is used to determine the projector coordinate x_p (or y_p) as follows,

$$x_p = \frac{\Phi}{2\pi f} \cdot W, \quad (5.9)$$

where W is the width of the projector image plane, f is the frequency of the fringe pattern, i.e., the number of fringe periods in the fringe pattern. Therefore, the standard deviation of x_p due to the noise on the fringe patterns is

$$\sigma_{x_p} = \frac{W}{2\pi f} \sigma_{\Phi}. \quad (5.10)$$

For a certain pixel (x, y) on the camera image plane, the above determined project coordinate $(x_p$ or $y_p)$ is used for 3D reconstruction in a calibrated system. Therefore, σ_{x_p} (or σ_{y_p}) determines the quality of the reconstructed 3D profile. As W is usually a constant for the projector, a high frequency is preferred for a high-quality profilometer.

5.2.2 TF phase unwrapping

TF phase unwrapping projects two groups of PS patterns, one with a low frequency f_l and the other with a high frequency f_h . Then the corresponding low-frequency wrapped phases φ_l and the high-frequency wrapped phase φ_h are calculated by Eq. (5.2). The relationship between their unwrapped phases, Φ_l and Φ_h , is expressed by:

$$\Phi_h = f_h/f_l \cdot \Phi_l. \quad (5.11)$$

Different TF algorithms have been formed using different combinations of f_l and f_h [91]. In this chapter, the hierarchical algorithm [92], [93] is chosen for comparison due to its proven best reliability among the different TF algorithms [91], where f_h determines the quality of the wrapped phase and $f_l = 1$ determines the rough value of the absolute phase to assist phase unwrapping. We simply use f to represent f_h in the rest of this chapter.

Equation (5.7) is rewritten according to the two frequencies as:

$$\Phi_l = 2k_l\pi + \varphi_l = \varphi_l, \quad (5.12)$$

$$\Phi_h = 2k_h\pi + \varphi_h, \quad (5.13)$$

where k_l and k_h are the corresponding fringe orders and $k_l = 0$ for $f_l = 1$. Combining Eqs. (5.11)-(5.13), k_h can be determined as:

$$k_h = \text{round} \left(\frac{f \cdot \varphi_l - \varphi_h}{2\pi} \right), \quad (5.14)$$

Subsequently, φ_h is unwrapped into Φ_h by Eq. (5.13).

However, according to the TF principle shown by Eqs. (5.13) and (5.14), when noise presents, normal and abnormal phase errors occurring on both φ_l and φ_h cause errors on the weighted phase difference $f\varphi_l - \varphi_h$ and thus may cause phase unwrapping errors. There are three cases of phase unwrapping errors: (i) E^A due to the normal phase errors in φ_h and φ_l ; (ii) E^B due to the abnormal phase error in φ_h ; and (iii) E^C due to the abnormal phase error in φ_l . For the first case, the normal phase errors $\Delta\varphi_l$ and $\Delta\varphi_h$ makes the weighted phase difference deviated by

$$D_{TF}^A = f\Delta\varphi_l - \Delta\varphi_h, \quad (5.15)$$

which is a critical random variable that affects the error occurrence condition and probability of E^A . The condition can be expressed as

$$|D_{TF}^A| > \pi, \quad (5.16)$$

which gives k_h an error of ± 1 , and accordingly, gives E^A an error with an amplitude of $\pm 2\pi$. Such E^A can occur randomly over the entire phase map. How the critical random variable affects the probability of error occurrence is discussed later in Section 5.2.5. For the second case, it is very interesting that the abnormal phase error $\Delta\varphi_h \pm 2\pi$ on φ_h gives a wrong k_h by Eq. (5.14), but later is automatically compensated when Eq. (5.13) is used, and thus E^B does not happen in TF [91]. For the third case, the abnormal phase error $\Delta\varphi_l \pm 2\pi$ on φ_l introduces an error of the weighted phase difference as $f \cdot (\Delta\varphi_l \pm 2\pi) - \Delta\varphi_h$, which approximately equals to $\pm 2f\pi$ and causes E^C with an amplitude of $\pm 2f\pi$. The error condition of E^C is essentially the same as that of the abnormal phase error expressed in Eq. (5.6), and rewritten as,

$$|\varphi_{l0} + \Delta\varphi_l| > \pi, \quad (5.17)$$

where φ_{l0} is the ideal phase value of φ_l . We also define a critical random variable affecting the occurrence of E^C as

$$D_{TF}^C = \Delta\varphi_l. \quad (5.18)$$

According to Eq. (5.17), E^C occurs around $\varphi_{l0} = \pm\pi$, i.e., the left and right image borders.

As both $\Delta\varphi_l$ and $\Delta\varphi_h$ have a standard deviation of σ_φ (Eq. (5.4)), we have the following standard deviations of D_{TF}^A and D_{TF}^C :

$$\sigma_{TF}^A = \sqrt{f^2 + 1} \sigma_\varphi \approx f \sigma_\varphi, \quad (5.19)$$

$$\sigma_{TF}^C = \sigma_\varphi. \quad (5.20)$$

We emphasize that the standard deviations of critical random variables reflect the possibility of occurrence of E^A and E^C , and are irrelevant to their error amplitudes. These two equations indicate that, in the TF method, a higher frequency f causes more E^A , and a large σ_φ causes both more E^A and E^C .

Examples with different f and noise levels are simulated and shown in Fig. 5.2. Figure 5.2(a) shows that when the noise is mild, no pixel satisfies Eq. (5.16), and E^A does not appear. However, because of the abnormal phase errors around $\varphi_{t0} = \pm\pi$ (at left and right image borders) where Eq. (5.17) is satisfied, E^C appears. Figure 5.2(b) shows the situation with a higher frequency f in φ_h , which makes E^A appears randomly in the phase map; as for E^C , the number remains the same but the error amplitude increases, also due to the higher f . Figure 5.2(c) further increases the noise level. As a result, both the numbers of E^A and E^C increase. All these observations are consistent with our analysis.

(a)

(b)

(c)

Figure 5.2 TF phase unwrapping with noises. (a) $f = 8$, $\sigma = 3$, only E^C appears at left and right borders, (b) $f = 30$, $\sigma = 3$, E^A also appears due to the higher frequency, (c) $f = 30$, $\sigma = 6$, the higher noise level increases both E^A and E^C .

5.2.3 PC phase unwrapping

The PC method [96] also projects the high-frequency PS patterns as the TF method. However, the low-frequency PS patterns with a linear phase in TF are replaced by PS patterns with a stepped phase, which are named PC patterns. The stepped phase φ_s codes the phase order k ($0 \leq k \leq f - 1$) for phase unwrapping as follows:

$$\varphi_s = -\pi + \left(k + \frac{1}{2}\right) \cdot \frac{2\pi}{f}, \quad (5.21)$$

$$k = \left\lfloor \frac{x_p f}{W} \right\rfloor. \quad (5.22)$$

We highlight that, comparing the definition of φ_s in [96], Eq. (21) adds $\frac{1}{2}$ to the fringe order to avoid $\varphi_s = -\pi$ when $k = 0$ where abnormal phase errors occur. The stepped phase φ_s is calculated by Eq. (5.2) from the PC patterns. The fringe order is extracted solely from φ_s as:

$$k = \text{round} \left[\frac{f \cdot (\varphi_s + \pi)}{2\pi} - \frac{1}{2} \right], \quad (5.23)$$

with which the wrapped phase φ is unwrapped by Eq. (5.7).

As both φ and φ_s are obtained by the PS algorithm, noise on the PS and PC patterns causes normal and abnormal phase errors on φ and φ_s , which may result in phase unwrapping errors. According to the PC principle shown by Eqs. (5.7) and (5.23), similar to the TF method, there are three cases of phase unwrapping errors: (i) E^A due to normal phase errors from φ_s ; (ii) E^B due to the abnormal phase errors in φ ; and (iii) E^C due to the abnormal

phase errors in φ_s . For the first case, the normal phase error $\Delta\varphi_s$ makes $f\varphi_s$ in Eq. (5.23) deviated by

$$D_{PC}^A = f\Delta\varphi_s, \quad (5.24)$$

which is a critical random variable that affects the error occurrence condition of E^A . The condition can be expressed as

$$|D_{PC}^A| > \pi, \quad (5.25)$$

which gives k an error of ± 1 , and accordingly, causes E^A with an amplitude of $\pm 2\pi$ all over the phase map. We highlight that as the value of φ is not used in Eq. (5.23), thus the normal phase errors in φ will not cause E^A . For the second case, the abnormal phase error $\Delta\varphi \pm 2\pi$ on φ causes E^B with an amplitude of $\pm 2\pi$. The error condition is expressed as

$$|\varphi_0 + \Delta\varphi| > \pi, \quad (5.26)$$

which usually occurs around $\varphi_0 = \pm\pi$, i.e. the 2π phase jumps. We define the critical random variable affecting the occurrence of E^B as

$$D_{PC}^B = \Delta\varphi. \quad (5.27)$$

For the third case, the abnormal phase errors $\Delta\varphi_s \pm 2\pi$ on φ_s occur when $|\varphi_{s0} + \Delta\varphi_s| > \pi$ and cause E^C with an error amplitude of $\pm 2(f-1)\pi$. Consider $\varphi_{s0} + \Delta\varphi_s > 0$ first, by substituting Eq. (5.21) into this condition, followed by simple arrangements, we have

$$f\Delta\varphi_s > 2\pi(f - k - \frac{1}{2}). \quad (5.28)$$

Since the minimum of $2\pi(f - k - \frac{1}{2})$ is π achieved at $k = f - 1$, we have $f\Delta\varphi_s > \pi$. Thus, such condition is usually satisfied around the right image border. Similarly, when $\varphi_{s0} + \Delta\varphi_s < 0$, the condition becomes $f\Delta\varphi_s < -\pi$ which is usually satisfied around the left image border. In an integrated form, we have the critical random variable of E^C as

$$D_{PC}^C = f\Delta\varphi_s, \quad (5.29)$$

and the error condition of E^C

$$|D_{PC}^C| > \pi, \quad (5.30)$$

which are the same as those of E^A in Eqs. (5.24)-(5.25). Interestingly, the first and third cases share the same occurrence condition. Nevertheless, because of the different error causes, E^A occurs everywhere including the left and right image borders while E^C only occurs at the left and right image borders.

As both $\Delta\varphi$ and $\Delta\varphi_s$ follow the noise model of the PS algorithm in Eq. (5.4), we have the standard deviations for D_{PC}^A , D_{PC}^B , and D_{PC}^C :

$$\sigma_{PC}^A = \sigma_{PC}^C = f\sigma_\varphi, \quad (5.31)$$

$$\sigma_{PC}^B = \sigma_\varphi, \quad (5.32)$$

which are almost the same as Eqs. (5.19)-(5.20), suggesting that the numbers of errors in the PC method are similar to those in the TF method.

(a)

(b)

(c)

Figure 5.3 PC phase unwrapping with noises. (a) $f = 8$, $\sigma = 3$, only E^B appears at the 2π phase jumps, (b) $f = 30$, $\sigma = 3$, E^A and E^C also appear due to the higher frequency, (c) $f = 30$, $\sigma = 6$, the higher noise level increases all three types of errors.

Examples with different f and noise levels are simulated and shown in Fig. 5.3. In Fig. 5.3(a), E^B with an amplitude of $\pm 2\pi$ appears at 2π phase jumps where Eq. (5.26) is satisfied. In Fig. 5.3(b), E^A with an amplitude of $\pm 2\pi$ and E^C with an amplitude of $\pm 2(f-1)\pi$ also appear due to the higher frequency, while the number of E^B has no significant change. In Fig. 5.3(c), a higher noise level increases all three types of errors.

Compared with the results in Fig. 5.2, the overall performances of PC and TF are similar, where the main difference is the extra E^B at 2π phase jumps in PC.

5.2.4 GC phase unwrapping

The GC method also projects high-frequency PS patterns to extract the wrapped phase but uses n -ary GC patterns to code the phase order k [86]. The coding principle is much like generating a number from digits, applying to each pixel (x, y) ,

$$C(x, y) = \sum_{j=1}^M c_j(x, y) \times n^{M-i}, \quad (5.33)$$

where n is the base of digits corresponding to the number of gray levels in the GC patterns; $c_j = 0, 1, \dots, n-1$ is the j -th digital value coded by the j -th GC pattern and often called as code; M is the number of digits which equals to the number of GC patterns; C is the synthesized code with a total of n^M values from 0 to $n^M - 1$ and often called as codeword. Thus, each pixel (x, y) can have its own codes $c_j(x, y)$ and codeword $C(x, y)$. The GC method uses $C(x, y)$ to indicate the pixels' fringe order information – all the pixels belonging to the same fringe period will have the same code and codeword to be uniquely mapped to the same fringe order k i.e., $C = \text{LUT}(k)$ or $k = \text{LUT}(C)$. Once k is correctly obtained, the phase can be easily unwrapped by Eq. (5.7). To uniquely code f periods in a fringe pattern, we require $n^M \geq f$, and thus the number of required GC patterns is

$$M = \lceil \log_n f \rceil, \quad (5.34)$$

Therefore, the required patterns are reduced by increasing the base n . In this chapter, quaternary GC ($n = 4$) is selected [132], which requires fewer patterns than the binary ($n = 2$) or ternary ($n = 3$) GC.

The key to GC phase unwrapping is to use a projector to project GC patterns G_j which code c_j , and use a camera to receive the distorted GC patterns from which c_j is decoded and used for k determination. The following linear mapping is designed for coding [132]:

$$G_j = A_0 + B_0 \left(\frac{2c_{j0}}{n-1} - 1 \right) + \Delta G_j, j = 1, 2, 3, \dots, M, \quad (5.35)$$

where ΔG_j is included as the unavoidable noise; the subscript “0” is again used to emphasize the noiseless values; A_0 and B_0 are chosen the same as those for PS patterns in Eq. (5.1), thus they are estimated from the PS patterns as [70]:

$$A = \frac{\sum_{i=1}^N I_i}{N}, \quad (5.36)$$

$$B = \frac{2\sqrt{\{\sum_{i=1}^N I_i \sin[(i-1)\frac{2\pi}{N}]\}^2 + \{\sum_{i=1}^N I_i \cos[(i-1)\frac{2\pi}{N}]\}^2}}{N}. \quad (5.37)$$

Note that A and B in the above equations include errors caused by the noise on the PS patterns. Accordingly, the decoding can be done as:

$$g_j = \frac{n-1}{2} \cdot \left(\frac{G_j - A}{B} + 1 \right), \quad (5.38)$$

$$c_j = \text{round}(g_j), \quad (5.39)$$

and also there are errors in g_j and c_j .

The errors on φ and g_j cause two cases of phase unwrapping errors according to the GC principle shown by Eq. (5.7) and Eqs. (5.38)-(5.39): (i) E^A due to the errors from g_j ; and (ii) E^B due to the abnormal phase errors in φ . It is unique that g_j does not have abnormal phase error and thus does not cause E^C . For the first case, errors Δg_j on g_j result in errors of ± 1 on c_j from Eqs. (5.38)-(5.39), when

$$|\Delta g_j| > 0.5, \quad (5.40)$$

which cause E^A with an error amplitude of multiple of $\pm 2\pi$ all over the phase map. Note that the amplitude of E^A is a variable because it depends on which c_j is wrong according to the coding principle. From Eq. (40), we define the critical random variable affecting the occurrence of E^A as

$$D_{GC}^A = \Delta g_j. \quad (5.41)$$

For the second case, the abnormal phase errors $\Delta\varphi \pm 2\pi$ on φ cause E^B with an amplitude of $\pm 2\pi$ according to Eq. (5.7). The error condition of E^B is expressed as

$$|\varphi_0 + \Delta\varphi| > \pi, \quad (5.42)$$

which usually occurs around the 2π phase jumps, the same as the E^B in PC. Thus, the critical random variable E^B in GC is the same as that in PC in Eq. (5.27):

$$D_{GC}^B = \Delta\varphi. \quad (5.43)$$

D_{GC}^A introduced by image noise can be expressed as:

$$D_{GC}^A = \sum_{i=1}^N \frac{\partial g_j}{\partial I_i} \cdot \Delta I_i + \frac{\partial g_j}{\partial G_j} \cdot \Delta G_j, \quad (5.44)$$

with

$$\frac{\partial g_j}{\partial I_i} = -\frac{n-1}{2} \cdot \left\{ \frac{1}{B_0 N} + \frac{2 \cos\left[\varphi + (i-1)\frac{2\pi}{N}\right](G_{j0} - A_0)}{B_0^2 N} \right\}, \quad (5.45)$$

$$\frac{\partial g_j}{\partial G_j} = \frac{n-1}{2B_0}. \quad (5.46)$$

where $G_{j0} = A_0 + B_0(\frac{2c_{j0}}{n-1} - 1)$ is the ideal gray code pattern. As the GC patterns and the PS patterns captured under the same camera-projector system, ΔG_j and ΔI_i are modeled as the same Gaussian noise with a mean value of zero and a standard deviation of σ . Thus, we have the standard deviation of D_{GC}^A :

$$\sigma_{GC}^A = \frac{n-1}{2} \cdot \sqrt{\left[\frac{1}{B_0^2} + \frac{1}{B_0^2 N} + \frac{2(G_{j0}-A_0)^2}{B_0^4 N} \right]} \cdot \sigma = \alpha(c_{j0})\sigma_\varphi, \quad (5.47)$$

with

$$\alpha(c_{j0}) = \frac{n-1}{2} \cdot \sqrt{\frac{N+1}{2} + \left(\frac{2c_{j0}}{n-1} - 1 \right)^2}, \quad (5.48)$$

which is constant for a certain code c_{j0} . Note that Eq. (5.4) and $\frac{G_{j0}-A_0}{B_0} = \frac{2c_{j0}}{n-1} - 1$ have been used in the last step of Eq. (5.47). As an example, for $N = 3$, $n = 4$, and $c_{j0} = 1$, we have $\alpha(c_{j0}) = 2.2$ which is independent of f . The standard deviation of D_{GC}^B follows the noise model of the PS algorithm, i.e.,

$$\sigma_{GC}^B = \sigma_\varphi, \quad (5.49)$$

Examples with different f and noise levels are simulated and shown in Fig. 5.4. In Fig. 5.4(a), E^A does not appear, but E^B appears around 2π phase jumps, which is similar to the PC method. In Fig. 5.4(b), although a higher frequency is used, there is no significant impact on the number of E^B and E^A still does not appear. In Fig. 5.4(c), the number of E^B increases and E^A appears due to the higher noise level used. Besides, E^B always has a small amplitude of $\pm 2\pi$ as shown in Figs. 5.4(a)-5.4(c), while the E^A can have an amplitude of any multiple of $\pm 2\pi$, large or small, as shown in Fig. 5.4(c).

(a)

Figure 5.4 Gray code phase unwrapping with noises. (a) $f = 8$, $\sigma = 3$, only E^B appear at 2π phase jumps; (b) $f = 30$, $\sigma = 3$, the higher frequency has no impact on the errors; (c) $f = 30$, $\sigma = 6$, due to the higher noise level, E^A appears and the number of E^B increases.

5.2.5 Summary and error rates

For easy reference, Table 5.1 summarizes three types of errors, (i) E^A due to the normal errors; (ii) E^B due to the abnormal errors from the PS patterns; and (iii) E^C due to the abnormal errors from the additional patterns, in three different phase unwrapping methods, TF, PC, and GC. From Table 5.1, we can observe the following properties:

- (i) All the errors appearing in Table 5.1 are proportional to σ_φ . Thus, reducing the noise in an FPP system is always beneficial, which is well expected.
- (ii) For E^A , it is interesting to note that σ_{TF}^A and σ_{PC}^A are also proportional to f , but σ_{GC}^U is independent of f and related to the ideal code c_{j0} ; for E^B , TF is fully immune to this type of error and it exists in PC and GC around 2π phase jumps; for E^C , it exists in TF and PC only at left and right image borders, which can be simply cut off this area to avoid the problem, and it does not exist in GC.
- (iii) GC's independence of the frequency on E^A and its immunity to E^C make it most attractive in aFPP.

Table 5.1 Comparison of the phase unwrapping errors in the three methods in aFFP

Error types	Properties	TF	PC	GC
E^A due to the normal errors	Location	Whole phase map	Whole phase map	Whole phase map
	Standard Deviation	$\approx f\sigma_\varphi$	$f\sigma_\varphi$	$\alpha(c_{j0})\sigma_\varphi$
	Amplitude	$\pm 2\pi$	$\pm 2\pi$	Multiple of $\pm 2\pi$
E^B due to the abnormal errors from the PS patterns	Location	NA	2π phase jumps	2π phase jumps
	Location Standard Deviation		σ_φ	σ_φ
	Amplitude		$\pm 2\pi$	$\pm 2\pi$
E^C due to the abnormal errors from the additional patterns	Location	Left and right borders	Left and right borders	NA
	Standard Deviation	σ_φ	$f\sigma_\varphi$	
	Amplitude	$\pm 2f\pi$	$\pm 2(f-1)\pi$	

Although the standard deviations in Table 5.1 can qualitatively describe the tendency of error happenings, we further define error rates as the probabilities of the errors for easier evaluation and clearer illustration. To do so, first, the probability densities of the critical random variables of the errors are examined and shown in Fig. 5.5, where D_{TF}^A , D_{TF}^C , D_{PC}^A , D_{PC}^B , D_{PC}^C , and D_{GC}^B are found to follow the normal distribution (Fig. 5.5(a)), while D_{GC}^A only follows it approximately (Fig. 5.5(b)).

Figure 5.5 The probability density of the critical random variable after normalization. (a) D_{TF}^A , D_{TF}^C , D_{PC}^A , D_{PC}^B , D_{PC}^C , and D_{GC}^B after normalization accord with the normal distribution, (b) D_{GC}^A after normalization does not accord with the normal distribution.

Next, we consider the error rates one by one. Since D_{TF}^A is normal with a standard deviation of $\sigma_{TF}^A = f\sigma_\varphi$, its probability density function is $\text{pdf}(t) = \frac{1}{\sqrt{2\pi}\sigma_{TF}^A} e^{-\frac{t^2}{2\sigma_{TF}^A{}^2}}$, and the error rate of E^A in TF can be obtained as follows by some simple manipulation:

$$R_{TF}^A = 2 \int_{\pi}^{\infty} \text{pdf}(t) dt = 2 \left[\frac{1}{2} - \int_0^{\pi} \text{pdf}(t) dt \right] = 1 - \text{erf}\left(\frac{\pi}{\sqrt{2}f\sigma_\varphi}\right), \quad (5.50)$$

where $\text{erf}(\cdot)$ is the error function defined as:

$$\text{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt. \quad (5.51)$$

D_{TF}^C is also normal and the error rate is related to the value of φ_0 as suggested in Eq. (5.17), thus the corresponding error rate on the whole phase map can be expressed as:

$$R_{TF}^C = \frac{2 \times \int_0^{\frac{\pi}{2}} \left[1 - \text{erf}\left(\frac{\pi - \varphi_0}{\sqrt{2}\sigma_\varphi}\right) \right] d\varphi_0}{2\pi} = \frac{1}{2} - \frac{1}{2\pi} \int_0^{\pi} \text{erf}\left(\frac{\pi - \varphi_0}{\sqrt{2}\sigma_\varphi}\right) d\varphi_0. \quad (5.52)$$

For PC, E^A and E^C share the same occurrence condition which causes different errors at different locations of the unwrapped phase. The error rate is similar to E^A in TF:

$$R_{PC}^A + R_{PC}^C = 1 - \text{erf}\left(\frac{\pi}{\sqrt{2}f\sigma_\varphi}\right). \quad (5.53)$$

As has been mentioned, the error rate of E^A is related to the ideal phase order, k_0 , thus we rewrite Eq. (5.53) more specifically as:

$$R_{PC}^A = \begin{cases} \frac{1}{2} \left[1 - \text{erf}\left(\frac{\pi}{\sqrt{2}f\sigma_\varphi}\right) \right], & k_0 = 0 \text{ or } n - 1 \\ 1 - \text{erf}\left(\frac{\pi}{\sqrt{2}f\sigma_\varphi}\right), & \text{otherwise} \end{cases}. \quad (5.54)$$

E^C only occurs around the left and right borders, thus the error rate of E^C can be expressed as:

$$R_{PC}^C = \begin{cases} \frac{1}{2} \left[1 - \text{erf}\left(\frac{\pi}{\sqrt{2}f\sigma_\varphi}\right) \right], & k_0 = 0 \text{ or } n - 1 \\ 0, & \text{otherwise} \end{cases}. \quad (5.55)$$

E^B in PC has the same error condition as E^C in TF, thus $R_{PC}^B = R_{TF}^C$.

For GC, the error rate of E^A increases with M , since more noises are introduced by more GC patterns, thus R_{GC}^A is expressed as:

$$R_{GC}^A = 1 - \prod_{i=1}^M (1 - R_{c_j}), \quad (5.56)$$

where R_{c_j} is the error rate of c_j obtained from Eqs. (5.38)-(5.39). When we only keep the linear term in Eq. (5.56) and assume D_{GC}^A as the normal distribution, an approximate solution for R_{GC}^A can be expressed as:

$$R_{GC}^A \approx \sum_{j=1}^M R_{c_j} \approx 2M \times \frac{1}{2} \left[1 - \operatorname{erf} \left(\frac{0.5}{\sqrt{2}\sigma_{GC}^A} \right) \right] = M \left[1 - \operatorname{erf} \left(\frac{1}{6.2\sigma_\varphi} \right) \right], \quad (5.57)$$

where $\sigma_{GC}^A = \alpha(c_{j0} = 1)\sigma_\varphi = 2.2\sigma_\varphi$. E^B in GC has the same error condition as E^B in PC and E^C in TF, thus $R_{GC}^A = R_{PC}^B = R_{TF}^C$. All the error rates are summarized in Table 5.2 for easy reference. Note that $\operatorname{erf}(\cdot)$ can be determined by the error function table, thus all these error rates can be easily computed once the parameters σ_φ and f are given.

Table 5.2 Error rates of the errors in the three methods in aFFP

	TF	PC	GC
Error rate of E^A	R_1	$R_1 [R_1/2]$	R_2
Error rate of E^B	0	R_3	R_3
Error rate of E^C	R_3	0 $[R_1/2]$	0

Notes: $R_1 = 1 - \operatorname{erf} \left(\frac{\pi}{\sqrt{2}f\sigma_\varphi} \right)$, $R_2 = M \left[1 - \operatorname{erf} \left(\frac{1}{6.2\sigma_\varphi} \right) \right]$, $R_3 = \frac{1}{2} - \frac{1}{2\pi} \int_0^\pi \operatorname{erf} \left(\frac{\pi - \varphi_0}{\sqrt{2}\sigma_\varphi} \right) d\varphi_0$. For PC, the error rates of E^A and E^C are related to their locations as shown in Eqs. (5.54)-(5.55) with $R_{PC}^A + R_{PC}^C = R_1$, and the error rates around left and right image borders are given in brackets $[\cdot]$.

Tables 5.1 and 5.2 classify the errors according to the different TPU methods and error sources. However, to remove the errors, we concern more on the distributions of the errors, thus we re-classify the errors into uniformly distributed errors (E^U) and non-uniformly distributed errors (E^N). Hence, E^U contains E^A in all three methods, while E^N contains E^B in PC and GC and E^C in TF. As for E^C in PC, although it only occurs at image borders, it is considered as E^U because its occurrence condition is the same as E^A in PC. We have distinguished E^U and E^N by red and green in Tables 5.1 and 5.2, respectively. We note that E^U randomly distributes in the phase map, with a density heavily influenced by frequency in TF and PC, while E^N gathers in a certain area, with a density heavily influenced by defocusing which will be explained in Section 5.3.2. If the phase unwrapping errors are post-filtered, the result will be system-dependent. We thus will focus on error rates in simulation comparison to emphasize the basic error properties in different FPP systems in Section 5.3 and will consider post-filtering in real experiments in Section 5.4.

5.3 Comparison by simulations

To verify our theoretical analysis, the three TPU methods are compared in an aFPP system by simulation in Section 5.3.1. We then extend our simulation to a bFPP system in Section 5.3.2, and a cFPP system in Section 5.3.3 to investigate various TPU performances. Section 5.3.4 summarizes the simulation results.

5.3.1 aFPP system

We first simulate 8-bit patterns used in an aFPP system, which is considered well-calibrated and only suffers random noise. A smooth surface is generated by the MATLAB built-in function “peaks” with a size of 1000×1000 pixels. To obtain the wrapped phase, three-step PS patterns are simulated according to Eq. (5.1) with f varying from 4 to 64. To unwrap phase, for TF, low-frequency three-step PS patterns are generated also according to Eq. (5.1) with $f = 1$; for PC, 3 PC patterns are generated according to Eq. (5.1) and Eq. (5.21); and for GC, GC patterns are generated according to Eq. (5.35) with $n = 4$ and the number of GC patterns is $M = \lceil \log_n f \rceil$. In all the generated patterns, $A_0 = 127$ and $B_0 = 60$ are used, and zero-mean Gaussian random noise with a relatively large standard deviation $\sigma = 6$ is added to illustrate the errors more clearly. Our verification remains valid when σ is smaller. As an example, one of the PS patterns with $f = 16$ is shown in Fig. 5.6(a), which will be used for wrapped phase calculation in the three methods; Figs. 5.6(b)-5.6(d) are used for phase unwrapping, among which, Fig. 5.6(b) is one of the low-frequency PS patterns used for TF, Fig. 5.6(c) is one of the PC patterns, and Fig. 5.6(d) is one of the GC patterns. All the mentioned patterns contain random noise. The wrapped phase and unwrapped phase calculated from noise-free patterns are shown in Figs. 5.6(e) and 5.6(f), respectively, and will be used as the ground truth.

Figure 5.6 Simulation of a smooth surface with 8-bit patterns, where $f = 16$, $\sigma = 6$, $B_0 = 60$. (a) One of the high-frequency fringe patterns, (b) one of the low-frequency fringe patterns for the TF method, (c) one of the PC patterns, (d) one of the GC patterns, (e) ground truth of the wrapped phase, (f) ground truth of the unwrapped phase.

Based on the simulated patterns, we detect the phase unwrapping errors of the three methods by comparing the obtained unwrapped phase maps with the ground truth. The pixels with a difference large than π are regarded as phase unwrapping errors. As an example, shown in red in Fig.5.7 are the error distributions of the three methods with $f = 16$. As expected, E^U is randomly distributed on the whole phase map in all three methods, while E^N is concentrically distributed around the left and right image borders in TF and around 2π phase jumps in PC and GC.

Figure 5.7 Error distributions of the different phase unwrapping methods with $f = 16$, $\sigma = 6$, $B_0 = 60$. (a) TF method, (b) PC method, (c) GC method. (The size of the erroneous pixel is enlarged 4 times for better illustration).

To compare the errors in the three methods quantitatively and more intuitively, we compute the error rates R^U and R^N by simulation and compare them with their theoretical values given in Table 5.2. To compute R^U , we detect the E^U region where all the errors are E^U as follows: for TF, the region is the central region (800×800) of the patterns, and for PC and GC, the region contains all the pixels away from the 2π phase jumps, i.e., $-\pi + \varepsilon < \varphi < \pi - \varepsilon$, where $\varepsilon = \frac{\pi}{3}$ in our simulation. R^U is calculated as the number of E^U in the E^U region divided by the size of the E^U region. As for the E^N , the pixels with abnormal phase errors in φ_l in TF, i.e., E^C , and those in φ in PC and GC, i.e., E^B are regarded as E^N , and R^N is calculated as the number of E^N divided by the size of the pattern. R^U and R^N of the three methods are as shown in Fig. 5.8, along with the theoretical values of R_1 , R_2 and R_3 . The perfect agreement verifies our theoretical analysis. We also clearly observe the outstanding performance of GC when the frequency becomes larger. We have summarized the error characteristics of the three methods in Table 5.3, which is a re-arrangement of Tables 5.1 and 5.2 for easy reference and serves as a base of comparison when we extend our simulation to bFPP and cFPP.

Figure 5.8 Error rates of E^U and E^N with different f in aFPP. (a) R^U , (b) R^N .

Table 5.3 Comparison of the three phase unwrapping methods in aFPP.

		TF	PC	GC
Number of Patterns		3	3	$\lceil \log_n f \rceil$
E^U	Location	Whole phase map	Whole phase map	Whole phase map
	Source	E^A	E^A, E^C	E^A
	Rate	R_1	R_1	R_2
E^N	Location	Left and right borders	2π phase jumps	2π phase jumps
	Source	E^C	E^B	E^B
	Rate	R_3	R_3	R_3

5.3.2 bFPP system

The bFPP simulation is almost the same as aFPP except that all the patterns are binary and defocused. For binary defocused pattern generation, we first design the binary patterns by the flexible binary dithering technique [120], [130] and then apply a Gaussian low-pass filter with a standard deviation of σ_{df} , where $\sigma_{df} = 2$ and $\sigma_{df} = 5$ are selected to represent slight and great defocusing, respectively [77]. We compare the binary defocused phase unwrapping patterns with the ground-truth 8-bit patterns and plot the errors in Fig. 5.9. In all three methods, extra pattern errors from the ideal and focused situation have been observed and can cause extra phase unwrapping errors which are denoted as E^D . E^D in TF

can be ignored since the overall pattern error in the low-frequency PS pattern is small for both slight and great defocusing cases, as shown in Fig. 5.9(a). This is due to the fact that the required PS patterns are sinusoidal and smooth, which can be well approximated by defocusing. On the contrary, both the required PC and GC patterns have stepped structure, which cannot be well approximated by defocusing. Hence, E^D contributes to E^N for the obvious pattern errors around the 2π phase jumps as show in Figs. 5.9(b) and 5.9(c), respectively. Such pattern errors become more severe (wider and larger) with the increase of defocusing level which cause more E^N . We highlight that in aFPP, the inevitable projector defocusing can also cause pattern differences, but the amplitudes of such pattern differences are very small and ignorable.

It is also noticed that the amplitude of the PS patterns, i.e., B , will decrease with frequency due to the defocusing, while that of the phase unwrapping patterns almost do not change. The GC method is more sensitive to the amplitude of the PS patterns than the other methods because its decoding process involves B in Eq. (5.38), thus the number of E^U in GC will increase with the frequency. Besides, Eq. (5.39) assumes the background and amplitude are constant in the PS and GC patterns in bFPP, thus the $\text{round}(\cdot)$ operator is replaced by k -means clustering [132] for better performance.

(a)

(b)

(c)

Figure 5.9 Differences between the 8-bit patterns and defocused binary patterns. (a) TF method, (b) PC method, (c) PC method.

We add Gaussian noise to the binary defocused patterns and compute the unwrapped phase by the three methods. The phase unwrapping errors are shown in Fig. 5.10 where $f = 16$ and $\sigma_{df} = 5$. Compared with the phase unwrapping errors in aFPP shown in Fig. 5.7, the TF method has a similar result in bFPP, while the PC and GC methods produce more E^N as expected.

(a)

(b)

(c)

Figure 5.10 Error distributions of different phase unwrapping methods in a bFPP system with $f = 16$, $\sigma = 6$, $B_0 = 60$, and $\sigma_{df} = 5$. (a) TF method, (b) PC method, (c) GC method. (The size of the erroneous pixel is enlarged 4 times for better illustration)

We also compute the error rate of E^U of the three methods with different f in bFPP as shown in Fig. 5.11(a) where $B_0 = 60$, $\sigma = 6$ and $\sigma_{df} = 2$. The results are almost the same as those in aFPP shown in Fig. 5.8(a). The only minor difference is that R_{GC}^U increases with f slightly when $f \in (16, 64]$ due to the decrease of the amplitude of the PS patterns in bFPP, but GC remains to have the least errors regarding E^U . As for E^N , the error in TF is the same as that in aFPP, while extra E^N is caused by E^D in PC and GC. As a result, the error

rates, R_{PC}^N and R_{GC}^N , will be larger than R_3 . Unlike in aFPP, R_{PC}^N and R_{GC}^N cannot be calculated exactly in bFPP, since some pixels around 2π phase jumps present both E^D and the other errors, making it difficult for their differentiation. We thus calculate and illustrate the total error rate R^T , i.e., the number of all types of phase unwrapping errors divided by the size of the pattern in Fig. 5.11(b). In this simulation, as we can expect, TF is most robust to defocusing and remains a similar performance, while PC and GC deteriorate their performances due to E^N around 2π jumps. Nevertheless, GC is still shown to be most attractive in general when E^U is dominating when the frequency is high.

However, the performance of the three methods is also influenced by the signal-to-noise ratio B_0/σ and defocusing level σ_{df} . As a comparison, we repeat the experiment by reducing σ to 1.5 (less noise) and increasing σ_{df} to 4 (greater defocusing), which is in favor of TF. Indeed, as shown in Figs. 5.11(c)-5.11(d), the difference in E^U of the three methods is much smaller due to the less noise, but E^N in PC and GC becomes more serious due to the greater defocusing. As a result, TF outperforms both PC and GC in this setting. To summarize, in bFPP, TF turns to be competitive with GC, where the former is more sensitive to E^U especially under heavy noise, while the latter is more sensitive to E^N especially under great defocusing. The characteristics of the three methods in bFPP are summarized in Table 5.4 where the differences compared with aFPP are emphasized by red.

(a)

(b)

Figure 5.11 Error rate of E^U and the total error rate with different f in bFPP. (a)-(b) R^U and R^T when $B_0 = 60$, $\sigma = 6$ and $\sigma_{df} = 2$, (c)-(d). R^U and R^T when $B_0 = 60$, $\sigma = 1.5$ and $\sigma_{df} = 4$.

Table 5.4 Comparison of the three phase unwrapping methods in bFPP.

		TF	PC	GC
Number of Patterns		3	3	$\lceil \log_n f \rceil$
E^U	Location	Whole phase map	Whole phase map	Whole phase map
	Source	E^A	E^A, E^C	E^A
	Rate	R_1	R_1	R_2
E^N	Location	Left and right borders	2π phase jumps	2π phase jumps
	Source	E^C	E^B, E^D	E^B, E^D
	Rate	R_3	$> R_3$	$> R_3$

5.3.3 cFPP system

In cFPP, geometry/continuity constrained phase unwrapping is combined with the three TPU methods to enhance the accuracy [81], [82] or reduce the required patterns [83], [84], [132]. A partial phase unwrapping is performed with a TPU method, by which large phase jumps ($T * 2\pi$) are remained in Φ . Compared to the 2π phase jumps in φ , the $T * 2\pi$ phase jumps in Φ are more distinguishable from physical discontinuities of the measured objects thus can be removed by continuity/geometry constraints. Φ_c is the phase after geometry/continuity constrained phase unwrapping where all the phase jumps are removed.

In TF and PC, partial phase unwrapping of less periods ($T \ll f$) greatly increases the accuracy, according to Figs. 5.8(a) and 5.11. In GC, the partial unwrapping is actually to apply unique coding to only T periods, which allows us to use less GC patterns [132].

Following the above explanation, for $f = 64$ and $T = 16$, TF and PS use 3 phase unwrapping patterns, while GC only requires $M = \lceil \log_4 T \rceil = 2$ GC patterns. To ensure the same number of patterns used by different unwrapping methods for a fair comparison, the two-plus-one PS algorithm [79]–[81] is adopted in the TF and PC methods, which assumes that $A(x, y)$ is the same for PS patterns and unwrapping patterns. Thus, the required unwrapping patterns in TF and PC can be reduced by one since $A(x, y)$ is considered known from the PS patterns. Now, all the three TPU methods use 2 unwrapping patterns. Although the accuracy of the two-plus-one PS algorithm is inferior to the traditional three-step PS algorithm [47], the overall accuracy of TF and PC is still improved as the total number of periods to be unwrapped is reduced by the geometry/continuity constraints. Note that the geometry/continuity constrained phase unwrapping and two-plus-one PS algorithm are also applicable to aFPP, but in this chapter, they are considered only in cFPP for extreme measurement speed.

The binary patterns in cFPP are also generated by the binary dithering technique. Φ and Φ_C are computed by the generated patterns and compared with their respective ground-truth values to detect the phase unwrapping errors. The example in Fig. 5.10 with $f = 16$ is re-used but with small modification towards cFPP with $T = 4$. It should be noticed that the repeat of the phase unwrapping patterns, i.e., 4-time T -period patterns, results in extra E^N in GC, for which an identification and binary classification method is used to remove such errors [133]. The errors in Φ and Φ_C for the three methods are shown in Figs. 5.12(a)-5.12(c) and Figs. 5.12(d)-5.12(f), respectively. By comparing Figs. 5.12(a)-5.12(c) with the errors in bFPP in Fig. 10, we find that E^U in TF and PC is greatly reduced, but remains similar in GC; E^N occurs around $T \times 2\pi$ phase jumps in TF, and is still around the 2π phase jumps in PC and GC. By comparing Figs. 5.12(d)-5.12(f) with Figs. 5.12(a)-5.12(c), we notice that the geometry/continuity phase unwrapping used to remove the large phase jumps can also remove E^N in TF as shown in Fig. 5.12(d), and will not introduce new errors in PC and GC as shown in Figs. 5.12(e) and 5.12(f), respectively.

Figure 5.12 Error distributions of different phase unwrapping methods in a cFPP system with $f = 16$, $T = 4$, $\sigma = 6$, $B_0 = 60$, and $\sigma_{df} = 5$. (a) – (c) The errors in Φ by the TF, PC, and GC methods, respectively, (d) – (f) the errors in Φ_C by the TF, PC, and GC methods, respectively. (The size of the erroneous pixel is enlarged 4 times for better illustration).

We also generated patterns with f from 8 to 64 and $T = f/4$, by which the error rate of E^U of the three TPU methods are computed and shown in Fig. 5.13(a). Compared with the results in aFPP and bFPP, E^U in TF and PC in cFPP is reduced greatly since the number of phase orders to be encoded is reduced from f to $T = f/4$, and E^U in GC is reduced slightly because of the reduction of GC patterns, which is consistent with Eq. (5.57). As for E^N , in TF, E^N caused by E^C has an amplitude of $\pm 2T\pi$, which can be ideally removed by the geometry/continuity phase unwrapping, thus $R_{TF}^N = 0$, while in GC and PC, the errors are similar to those in bFPP, thus R_{PC}^N and R_{GC}^N are larger than R_3 . The total error rate of R^T is calculated and illustrated in Fig. 5.13(b), from which we notice that TF most effectively benefits from the geometry/continuity constraints and has the best performance in general in cFPP. To summarize, in cFPP, TF presents the best performance due to the two-fold benefits from geometry/continuity constraint: E^U is greatly reduced because of the fewer

fringe periods for partial unwrapping and E^N is completely removed through discontinuity removal during the final unwrapping. The characteristics of the three methods in cFPP are summarized in Table 5.5 where the differences comparing with the aFPP are emphasized by red.

Figure 5.13 Error rate of E^U and the total error rate with different f in cFPP. (a) R^U , (b) R^T .

Table 5.5 Comparison of the three phase unwrapping methods in cFPP

		TF	PC	GC
Number of Patterns		2	2	$\lceil \log_n T \rceil$
E^U	Location	Whole phase map	Whole phase map	Whole phase map
	Source	E^A	E^A, E^C	E^A
	Rate	$\ll R_1$	$\ll R_1$	$< R_2$
E^N	Location	NA	2π phase jumps	2π phase jumps
	Source		E^B, E^D	E^B, E^D
	Rate		$> R_3$	$> R_3$

5.3.4 Summary and denoising in practice

We summarize the simulation results in different FPP systems as follows:

- (i) The aFPP configuration is the same as in our theoretical analysis. As a result, the errors from the simulations perfectly match those from the theoretical prediction, where GC has the best performance.
- (ii) Compared with aFPP, bFPP dramatically increases the measurement speed by binary projection, but in the meantime introduces extra errors due to defocusing, which is much more obvious in PC and GC. The overall performances of the three methods depend on the signal-to-noise ratio and defocusing level, which determine the main error source (E^U or E^N). When E^U is the main error source, GC is best, while when E^N is the main error source, TF is most attractive.
- (iii) Compared with bFPP, cFPP further improves the accuracy and speed of the three methods by geometry/continuity constraints and the two-plus-one PS algorithm. In general, TF benefits from the geometry/continuity constraints and performs best.

In practice, the phase unwrapping errors, especially E^N in PC and GC, should be suppressed. Effective methods mainly include adaptive median filtering (AMF) [88], the complementary gray-code method [89], and tripartite phase unwrapping [90]. These methods do not work when the errors are very dense. We select AMF because it can remove E^U and E^N simultaneously, is simpler and thus more robust for complex object measurement, and does not require additional patterns. We will include it in our experimental comparisons.

The AMF method includes two steps: error detection and error correction. In the error detection step, a small window (7×7) around a pixel is selected as the local area of that pixel. A pixel will be considered as an error if it is quite different from the other pixels in its local area. In the error correction step, a median filter is applied only to the detected errors. These two steps are repeated with an increasing size of the median filter from small to large until there are no errors or the number of errors no longer reduces. The AMF method prioritizes a small filter size to maximize the chance of removing the noises without destroying the step height of a measured object.

5.4 Comparison by experiments

This section compares the three TPU methods under different situations by experiments, in which the results of both simple (Section 5.4.1) and complex objects (Subsection 5.4.2) are provided. The experiments use a TI DLP Light Crafter 4500 projector (1140×912 pixels) and a Basler acA2000-165um camera (1200×900 pixels) combined with a Navitar Zoom 7000 lens. The projector is in focus in aFPP where 8-bit patterns are used, and is out of focus in bFPP and cFPP where binary patterns are used. In aFPP, a gamma correction algorithm [134] is applied to correct the gamma nonlinearity of the 8-bit projector. In bFPP and cFPP, binary patterns are generated by the binary dithering technique [120]. In all the experiments, we obtain the ground-truth unwrapped phase by the 9-step PS algorithm for high accuracy ($N = 9$) [73] and the multi-frequency phase unwrapping method with $f = 1, 4, 16, 64$ [55]. We compare the unwrapped phase of each method with the ground truth and regard the pixels with a difference larger than π as phase unwrapping errors.

5.4.1 Experiments on a flat board

A flat board is measured by the 3-step PS algorithm with different TPU methods. The ground-truth measurement results are shown in Fig. 5.14, where Fig. 5.14(a) and 5.14 (b) are one of the fringe patterns and the corresponding wrapped phase, Fig. 5.14(c) is the unwrapped phase by the multi-frequency phase unwrapping, and Fig. 5.14(d) is the reconstructed 3D point cloud textured by the calculated background intensity A .

Figure 5.14 Ground-truth measurement results of a flat board by 9-step PS algorithm with multi-frequency phase unwrapping. (a) One of the captured fringe patterns with $f = 64$, (b) wrapped phase of (a), (c) unwrapped phase, (d) reconstructed 3D point cloud.

The unwrapped phase maps by different phase unwrapping methods with $f = 64$ are converted to depth maps as shown in Fig. 5.15, in which different situations are considered, including the aFPP system (Figs. 5.15(a)- 5.15(c)), bFPP system (Figs. 5.15(d)- 5.15(f)), and cFPP system (Figs. 5.15(g)- 5.15 (i)). In the depth maps, the outliers on the flat board are phase unwrapping errors. The error distributions basically coincide with our simulations in Fig. 5.7, Fig. 5.10, and Fig. 5.12. However, we highlight the differences between simulation and experiment include: (i) E^U is not absolutely uniformly distributed in Fig. 5.15 due to non-uniformity of B and noise in the practical experiment; (ii) the flat board is put around the central region of the projected patterns, thus there is no E^N in TF in Figs. 5.15(a) and 5.15(d); (iii) there is no E^U in Figs. 5.15(c), 5.15(f), and 5.15(g)-5.15(i) since the noise in our experiment is small. Besides, except for Fig. 5.15(g), all the other eight results suffer from visually heavy noise, but AMF [88] can successfully remove all the outliers. The filtered results look very similar to Fig. 5.15(g) and are thus not shown. However, such extreme success of the filter cannot be expected for measuring more complicated objects. In fact, pursuing precise and reliable raw data is always important in precision engineering and cannot be replaced by any subsequent data analysis, which supports our motivation for this detailed analysis work in this chapter.

Figure 5.15 Depth map reconstructed by different phase unwrapping methods with $f = 64$. (a)-(c) the TF, PC, and GC methods in aFPP, (d)-(f) the TF, PC, and GC methods in bFPP, (g)-(i) the TF, PC, and GC methods in cFPP ($T = 16$).

We also calculate the error rate of E^U and the total error rate with different f as shown in Fig. 5.16. As the signal-to-noise ratio in the experiment is high, E^U is accordingly reduced, making the overall performance of the TPU methods better than the simulation results, but the changes of E^U and E^N with respect to f agree with the simulation. We find from the results that for high-frequency phase unwrapping with $f = 64$, GC performs best for aFPP (an error rate of 1.6%), while TF performs best for bFPP (an error rate of 3.9%) and cFPP (an error rate of nearly zero). We highlight that, in bFPP, TF works better than GC because of the low noise level and the relatively high defocusing level, which agrees with our simulation in Fig. 5.11(c)-5.11(d).

Figure 5.16 Error rate of E^U and the total error rate with different f . (a)-(b) R^U and R^T in aFPP, (c)-(d) R^U and R^T in bFPP, (e)-(f) in cFPP ($T = f/4$).

Finally, although we focus on the phase unwrapping error throughout this chapter, we mention that the measurement error (RMSE) of the reconstructed depth map is $29 \mu\text{m}$ for $f = 64$ and $132 \mu\text{m}$ for $f = 16$ in aFPP, and $71 \mu\text{m}$ for $f = 64$ and $169 \mu\text{m}$ for $f = 16$ in

bFPP and cFPP. This value is only for reference as it depends on the system configuration and calibration.

5.4.2 Experiments on complex objects

We also measure a complex scene with discontinuities, step heights, and shadows, the ground-truth measurement results of which are shown in Fig. 5.17. Figure 5.17(a) is one of the high-frequency ($f = 64$) fringe patterns. We calculate the fringe amplitude B of the high-frequency fringe patterns as shown in Fig. 5.17(b) to distinguish the objects and the background [127]. The ground-truth unwrapped phase is calculated and shown in Fig. 5.17(c) with the background removed. Figure 5.17(d) is the reconstructed 3D point cloud textured by the calculated background intensity A , and Figure 5.17(e) is the corresponding depth map.

(e)

Figure 5.17 Ground-truth measurement results of complex objects by 9-step PS algorithm with multi-frequency phase unwrapping. (a) One of the captured fringe patterns with $f = 64$, (b) fringe amplitude of (a), (c) unwrapped phase with the background removed, (d) reconstructed 3D point cloud, (e) depth map.

We first compare the TPU methods under the aFPP system. The depth maps reconstructed by different methods are shown in Fig. 5.18 where the results after AMF are also included. Both results of low frequency ($f = 16$) and high frequency ($f = 64$) results are demonstrated to show the influence of frequency on phase unwrapping, while high frequency is usually preferred for a higher accuracy as explained in Section 5.2.1. Figures 5.18(a)-5.18(c) are from low-frequency phase unwrapping, in which the pixels with a small B (around the right part of the body) have a high error rate of E^U . As for E^N , there is no such error in TF as shown in Fig. 5.18(a) since the objects are around the center of the patterns, while such error appears around 2π phase jumps in PC and GC, as shown in Figs. 5.18(b) and 5.18(c), respectively. For low-frequency phase unwrapping, all the errors in the three methods can be treated as impulse noises and removed by AMF as shown in Figs. 5.18(d)-5.18(f). Figures 5.18(g)-5.18(i) are from high-frequency phase unwrapping, in which E^U in TF and PC increases dramatically, while E^U in GC and E^N in all the three methods are similar to those in the low frequency. The results of the high-frequency phase unwrapping after AMF are shown in Figs. 5.18(j)-5.18(l), where E^U is too dense to be removed in TF (Fig. 5.18(j)) and PC (Fig. 5.18(k)), while few errors remain around the step-heights and edges of the objects after AMF in GC (Fig. 5.18(l)). We highlight the white pixels in the center of the yellow region in Fig. 5.18(l) are not phase unwrapping errors but a gap on the objects. From the above experiment, we remark that, in aFPP, all

the three methods work well for low-frequency phase unwrapping, while the GC method outperforms the other two methods for high-frequency unwrapping.

Figure 5.18 Depth maps reconstructed by different phase unwrapping methods in aFPP. (a)-(c) TF, PC, and GC methods with $f = 16$, (d)-(f) the results of (a)-(c) after AMF, respectively, (g)-(i) TF, PC, and GC methods with $f = 64$, (j)-(l) the results of (g)-(i) after AMF, respectively.

We repeat the experiment in bFPP, with the depth maps shown in Fig. 5.19. Comparing Fig. 5.19 with Fig. 5.18, we notice that the error distributions in bFPP and aFPP are almost the same except that E^N in the PC and GC methods is more severe (wider) due to defocusing. As a result, when the fringes are dense ($f = 64$), it is more difficult to remove the errors in the GC method by AMF as shown in Fig. 5.19(i). In addition, we highlight that in this experiment, the overall signal-to-noise ratio is lower than that in Section 4.1, thus in the high-frequency phase unwrapping, E^U dominates and GC has the best performance. We also notice that, around the regions with a large B (around the right part of the step-shaped model), where over-exposure is likely to occur, there are more errors than the other regions in the GC method as shown in Figs. 5.19(c) and 5.19(i), which suggests that the GC method is more sensitive to over-exposure than the other methods in bFPP.

Figure 5.19 Depth maps reconstructed by different phase unwrapping methods in bFPP. (a)-(c) TF, PC, and GC methods with $f = 16$, (d)-(f) the results of (a)-(c) after AMF, respectively, (g)-(i) TF, PC, and GC methods with $f = 64$, (j)-(l) the results of (g)-(i) after AMF, respectively.

For the comparison in cFPP, we select $T = 4$ and $f = 16$ for low-frequency phase unwrapping and $T = 16$ and $f = 64$ for high-frequency phase unwrapping. We first do a T -period phase unwrapping by the three TPU methods, followed by the geometry constrained method to recover a f -period unwrapped phase. In the TF and PC methods, the two-plus-one PS algorithm is applied. The reconstructed depth maps in cFPP are shown in Fig. 5.20. Comparing Fig. 5.20 with Fig. 5.19, E^U in TF and PC is reduced greatly by the geometry constrained method. In the low-frequency phase unwrapping, with AMF, all the errors are perfectly removed as shown in Figs. 5.20 (d)-5.20(f). As for the high-frequency phase unwrapping, there are more E^U in the three methods and more E^N in PC and GC as shown in Figs. 5.20(d)-5.20(f) than those in low-frequency phase unwrapping, while AMF removes most of the errors as shown in Figs. 5.20(j)-5.20(l). In general, TF has the best result in cFPP.

Figure 5.20 Depth maps reconstructed by different phase unwrapping methods in cFPP. (a)-(c) TF, PC, and GC methods with $f = 16$ and $T = 4$, (d)-(f) the results of (a)-(c) after AMF, respectively, (g)-(i) TF, PC, and GC methods with $f = 64$ and $T = 16$, (j)-(l) the results of (g)-(i) after AMF, respectively.

We also verify the performance of the three methods on objects with more complex geometries in cFPP as shown in Fig. 5.21. The high frequency, $f = 64$, is selected which produces a high-accuracy wrapped phase. Figure 5.21(a) is one of the fringe patterns, Figure 5.21(b) is the ground-truth unwrapped phase with the background removed, and Figure 5.21(c) is the reconstructed 3D point cloud textured by the calculated background intensity A . The depth maps reconstructed by TF, PC, and GC methods are shown in Figs. 5.21(d)-5.21(f), respectively, and the corresponding results after AMF are shown in Figs. 5.21(g)-5.21(i), respectively. All these results are consistent with our previous simulations

and experiments in cFPP, where TF performs best. It should note that, in this example, PC with AMF also has a good result with 0 error which proves the ability of AMF in denoising.

Figure 5.21 Multiple objects with complex geometries by different phase unwrapping methods in cFPP. (a) One of the captured fringe patterns with $f = 64$, (b) ground-truth unwrapped phase with the background removed, (c) ground-truth 3D point cloud, (d)-(f) depth maps reconstructed by TF, PC, and GC methods, respectively, with $f = 64$ and $T = 16$, (g)-(i) the results of (d)-(f) after AMF, respectively.

Although this chapter aims at a basic comparison, we also provide a quick guide based on all the above results:

- (i) For low-frequency phase unwrapping, all the TPU methods have a good result with the help of AMF. However, increasing the frequency to achieve higher accuracy is often adopted in measurement.

- (ii) In high-frequency FPP, for static measurement where the accuracy is more concerned than the speed, GC performs best in aFPP and thus suggested.
- (iii) Compared with aFPP, bFPP significantly enhances the measurement speed but also causes more E^N in PC and GC due to defocusing. cFPP not only further enhances the speed by reducing the required patterns, but also greatly reduces E^U in TF and GC and completely removes E^N in TF. As a result, when high speed is assumed, TF in cFPP performs best and is suggested.

5.5 Conclusion

This chapter compares the performance of TF, PC, and GC phase unwrapping methods under different situations, including aFPP, bFPP, and cFPP. The principles and noise models of these methods are analyzed in detail. Four types of phase unwrapping errors are caused by different error sources: E^A , E^B , E^C , and E^D , which are further classified into E^U and E^N according to the error distribution. In aFPP, E^U is more noticeable in TF and PC than GC when the fringe frequency is high ($f \geq 16$), and E^N is around left and right image borders in TF, while it is around 2π phase jumps in PC and GC. In general, GC has the best performance in aFPP. In bFPP, the overall performances of the three methods are similar to those in aFPP, but there are more E^N in PC and GC than aFPP, and thus TF turns to be competitive with GC. In cFPP, compared with bFPP, E^U in TF and PC is reduced greatly and E^N in TF is completely removed, and thus TF performs best. Based on our simulations and experiments, we find that, for low-frequency phase unwrapping, all the three TPU methods have a good result, while for high-frequency phase unwrapping which is desired in high-quality measurement, GC in aFPP and TF in cFPP outperform their peers. Therefore, for measurement where accuracy is more concerned than the speed GC in aFPP is preferred, while for dynamic measurement pursuing extreme high speed, TF in cFPP is suggested.

Chapter 6 Conclusions and Future Works

6.1 Conclusions

Fringe projection profilometry (FPP) as an active optical measurement technique is widely applied in various fields due to its high accuracy, high speed, high resolution, and low cost. An important process in FPP is recovering the unwrapped phase of the fringe pattern(s), which is further converted to the 3D profile of the objects. Phase-shifting (PS) algorithm with temporal phase unwrapping (TPU) has proved a good performance for the unwrapped phase recovery of objects with discontinuities, thus has been well developed and widely used in recent works. However, different TPU methods have different characteristics, thus, it is difficult to select a proper TPU method for a given application. There are many factors/parameters that influence the performance of the TPU method, such as the number of codes, the order of the codes, and the coding strategies. This thesis has provided a systematic study of TPU methods for high-accuracy and high-speed measurement, in which some algorithms are developed for performance enhancement and suggestions are given for TPU selection.

First, a novel gray code (GC) phase unwrapping framework is proposed for binary defocused FPP (bFPP). In the proposed framework, a weighted optimization algorithm is developed for binary pattern design, a normalization-denoising-clustering process is used for gray code reconstruction, and geometry/continuity constrained phase unwrapping is incorporated to further reduce the gray code pattern. Compared with the previous GC methods used in bFPP, the proposed framework is more flexible which can be easily applied on GC with different numbers of codes (n), even with $n > 3$. Therefore, the proposed framework enables more codes to be coded and decoded with binary defocused patterns than the binary and ternary GC methods, which reduced the required patterns and improve the efficiency. It is noticed that, for a wrapped phase with a typical frequency of 64, the required patterns for phase unwrapping are constantly reduced when we increase n for 2 to 4. However, the reduction speed slows down when $n > 4$, and a new reduction of the patterns happen until $n = 8$. Experiments show that GC reconstruction becomes more

difficult when $n = 8$ compared with that when $n = 4$ due to the unavoidable noise and the defocusing effect, and there are many obvious phase unwrapping errors when $n = 8$ even after filtering. Thus, $n = 4$ is suggested for high-speed measurement with GC phase unwrapping, which unwraps the phase with fewer patterns and with good accuracy.

Subsequently, we comprehensively compare the simple code (SC) method with the GC method to discuss the influence of different orders of codes on phase unwrapping. Compared with GC, SC has more phase unwrapping errors around the 2π phase jumps, which cannot be easily removed by the existing filters. As a result, SC is obviously inferior to GC in accuracy. To suppress the errors in the SC method, an identification and binary classification (IBC) algorithm is proposed, with which SC is comparable with GC in theory, and also the further comprehensive comparison towards real measurement is more meaningful. In our comparison, different factors are considered, including the strategies on using IBC and adaptive median filtering (AMF) for error removal, the number of codes n , the step heights and invalid regions on the measured objects, defocusing level and noise level of the system. From the simulations and experiments, we observe that (i) both IBC and AMF are insensitive to the defocusing level and noise; (ii) SC + IBC + AMF and GC + AMF have excellent performance with only a few errors around step heights; (iii) when n increases, the number of the errors increase and the number of required patterns decrease; (iv) SC + IBC + AMF can use fewer patterns than GC + AMF when incorporated with continuity/geometry constraints without the loss of accuracy. Therefore, in general, both SC + IBC + AMF and GC + AMF can be used; when accuracy is critical, binary SC/GC + AMF should be used; when speed is critical, quaternary SC + IBC + AMF + continuity/geometry constraints should be used.

Last, we evaluate and compare the accuracy of TPU methods using different coding strategies theoretically and experimentally, including two-frequency (TF), phase coding (PC), and GC phase unwrapping methods. Three situations are considered, traditional 8-bit focused FPP (aFPP), high-speed binary defocused FPP (bFPP), and geometry/continuity constrained binary defocused FPP (cFPP). The accuracy comparison of the three TPU methods is based on their sensitivities to noise, for which noise models of the PC and GC methods are built and compared with the existing noise model of the TF method. Four types of phase unwrapping errors are caused by different error sources: E^A , E^B , E^C , and E^D ,

which are further classified into E^U and E^N according to the error distribution. In aFPP, E^U is more noticeable in TF and PC than GC when the fringe frequency is high ($f \geq 16$), and E^N is around left and right image borders in TF, while it is around 2π phase jumps in PC and GC. In general, GC has the best performance in aFPP. In bFPP, the overall performances of the three methods are similar to those in aFPP, but there are more E^N in PC and GC than aFPP, and thus TF turns to be competitive with GC. In cFPP, compared with bFPP, E^U in TF and PC is reduced greatly and E^N in TF is completely removed, and thus TF performs best. Based on our simulations and experiments, we find that, for low-frequency phase unwrapping, all the three TPU methods have a good result, while for high-frequency phase unwrapping which is desired in high-quality measurement, GC in aFPP and TF in cFPP outperform their peers. Thus, for static measurement where the accuracy is more concerned than the speed, GC in aFPP is preferred, while for dynamic measurement where extreme high speed is required, TF in cFPP is suggested.

6.2 Future works

6.2.1 Temporal phase unwrapping comparison towards high accuracy

In the comparison of the TPU methods (TF, PC, and GC) in Chapter 5, the methods using the least number of patterns are selected for the interest of high-speed measurement. However, in more general applications, the accuracy is valued more than the speed, and minimizing the number of required patterns is not the top priority. Thus, it is worthy to investigate the performance of the different TPU methods when we ease the restriction of the number of required patterns, which motivates us to propose the comparison of TPU towards high accuracy as our future research direction.

As the previous simulations and experiments shows the inferiority of the PC method in Chapter 5, we will concentrate on the two/multi-frequency and GC methods in the new comparison towards high accuracy. The noise models of multi-frequency (MF) and binary GC will be built and compared first. Based on the noise models, the error distributions and error rates of the two methods under different systems (aFPP, bFPP, and cFPP) will be then analyzed. Simulations and experiments will be also conducted to verify the noise models and the theoretical analysis. A guide for TPU selection towards high-accuracy

measurement will be provided based on the observations in the simulations and experiments.

6.2.2 Gray code extraction using deep learning

As aforementioned in Chapter 3, it becomes difficult to decode the GC from the captured defocused binary patterns when we increase the number of codes to 8. In addition, as discussed in Section 5.5.2, the GC cannot be derived correctly when over-exposure occurs on the object. Even the proposed- normalization-denoising-clustering process cannot solve these problems. Therefore, a more robust GC reconstruction algorithm should be proposed.

Recently, many deep-learning-based methods have been applied in FPP for the enhancement of accuracy and speed. For example, a fringe pattern analysis using deep learning is proposed to recover the wrapped phase using a single fringe pattern [108]. Compared with the existing single-frame methods, such as Fourier transform and windowed Fourier transform methods, this deep-learning-based method provides a better performance around the edges of the object. Besides, deep learning is also used to improve the accuracy of TF in high-frequency phase unwrapping [94]. These works inspire us to deep learning to extract the gray code from the defocused binary patterns.

The gray code extraction can be regarded as an image segmentation problem, in which the gray code patterns should be segmented according to the gray levels on the object and a proper code should be allotted to each segment. To ensure a high success rate of gray code extraction, different neural networks for image segmentation will be tested.

6.2.3 Temporal phase unwrapping with one additional pattern

In the traditional TPU, usually at least 2 additional patterns are required for phase unwrapping. When we reduce the number of additional patterns to 1, the accuracy of phase unwrapping will dramatically decrease. With the development of deep learning, it becomes possible to unwrap the phase with only 1 additional pattern. Some potential patterns, such as a sinusoidal pattern with unit frequency, a pattern with a linear intensity, a pattern with a stepped structure, and a random binary pattern, can be used for single-frame phase

unwrapping with deep learning are shown in Fig. 6.1. We will build up different neural networks for the different single pattern used for phase unwrapping, and the accuracy of these neural networks will be evaluated and compared.

Figure 6.1 Some potential patterns can be used for single-frame phase unwrapping with deep learning. (a) A sinusoidal pattern with unit frequency, (b) a pattern with a linear intensity, (c) a pattern with a stepped structure, (d) a random binary pattern.

Author's Publications

Journals

- X. He, D. Zheng, Q. Kemaio, and G. Christopoulos, “Quaternary gray-code phase unwrapping for binary fringe projection profilometry,” *Opt. Lasers Eng.*, vol. 121, no. May, pp. 358–368, 2019, doi: 10.1016/j.optlaseng.2019.04.009.
- X. He and Q. Kemaio, “A comparison of n-ary simple code and n-ary gray code phase unwrapping in high-speed fringe projection profilometry,” *Opt. Lasers Eng.*, vol. 128, no. January, p. 106046, 2020, doi: 10.1016/j.optlaseng.2020.106046.
- X. He and Q. Kemaio, “A comparative study on temporal phase unwrapping methods in high-speed fringe projection profilometry,” *Opt. Lasers Eng.*, vol. 142, no. February, p. 106613, 2021, doi: 10.1016/j.optlaseng.2021.106613.

Conference papers & presentations

- X. He, D. Zheng, and Q. Kemaio, “Octal gray-code phase unwrapping for high-speed fringe projection profilometry,” presented at *the 2019 Joint International Workshop IWAIT-IFMIA*, 2019, Singapore.
- J. Zhao, X. He, and Q. Kemaio. “Automatic body measurement by neural networks,” in *Proceedings of the 2019 2nd International Conference on Robot Systems and Applications*, 2019, pp. 9-14, doi:10.1145/3378891.3378897.
- X. He. “Investigation of gray code phase unwrapping for high-speed fringe projection profilometry,” presented at the *First International Forum on 3D Optical Sensing and Applications*, 2020, online.
- X. He and Q. Kemaio. “Fringe projection profilometry with absolute measurement: a discussion,” presented at the *International Workshop on Advanced Image Technology*, 2021, online.

References

- [1] Z. Song, *Handbook of 3D Machine Vision*. CRC press, Taylor & Francis Group LLC, 2013.
- [2] J.-A. Beraldin, B. Carrier, D. MacKinnon, and L. Cournoyer, “Characterization of Triangulation-Based 3D Imaging Systems Using Certified Artifacts,” *NCSLI Meas.*, vol. 7, no. 4, pp. 50–60, 2012, doi: 10.1080/19315775.2012.11721620.
- [3] A. G. Marrugo, F. Gao, and S. Zhang, “State-of-the-art active optical techniques for three-dimensional surface metrology: a review [Invited],” *J. Opt. Soc. Am. A*, vol. 37, no. 9, p. B60, 2020, doi: 10.1364/josaa.398644.
- [4] S. Van der Jeught and J. J. J. Dirckx, “Real-time structured light profilometry: A review,” *Opt. Lasers Eng.*, vol. 87, no. 2016, pp. 18–31, 2015, doi: 10.1016/j.optlaseng.2016.01.011.
- [5] J. Geng, “Structured-light 3D surface imaging: a tutorial,” *Adv. Opt. Photonics*, vol. 3, no. 2, p. 128, 2011, doi: 10.1364/aop.3.000128.
- [6] W. Linder, *Digital photogrammetry: A practical course*. 2009.
- [7] F. Remondino, S. F. El-Hakim, A. Gruen, and L. Zhang, “Turning images into 3-D models,” *IEEE Signal Process. Mag.*, vol. 25, no. 4, pp. 55–65, Jul. 2008, doi: 10.1109/MSP.2008.923093.
- [8] J. Baqersad, P. Poozesh, C. Niezrecki, and P. Avitabile, “Photogrammetry and optical methods in structural dynamics – A review,” *Mech. Syst. Signal Process.*, vol. 86, no. August, pp. 17–34, 2017, doi: 10.1016/j.ymssp.2016.02.011.
- [9] M. J. Westoby, J. Brasington, N. F. Glasser, M. J. Hambrey, and J. M. Reynolds, “‘Structure-from-Motion’ photogrammetry: A low-cost, effective tool for geoscience applications,” *Geomorphology*, vol. 179, pp. 300–314, 2012, doi: 10.1016/j.geomorph.2012.08.021.
- [10] A. Gruen, “Development and Status of Image Matching in Photogrammetry,” *Photogramm. Rec.*, vol. 27, no. 137, pp. 36–57, 2012, doi: 10.1111/j.1477-9730.2011.00671.x.
- [11] A. Zaharescu, E. Boyer, K. Varanasi, and R. Horaud, “Surface feature detection and description with applications to mesh matching,” *2009 IEEE Comput. Soc. Conf.*

- Comput. Vis. Pattern Recognit. Work. CVPR Work. 2009*, vol. 2009 IEEE, pp. 373–380, 2009, doi: 10.1109/CVPRW.2009.5206748.
- [12] R. A. Newcombe *et al.*, “KinectFusion: Real-time dense surface mapping and tracking,” *2011 10th IEEE Int. Symp. Mix. Augment. Reality, ISMAR 2011*, pp. 127–136, 2011, doi: 10.1109/ISMAR.2011.6092378.
 - [13] R. A. Newcombe, S. J. Lovegrove, and A. J. Davison, “DTAM: Dense tracking and mapping in real-time,” *Proc. IEEE Int. Conf. Comput. Vis.*, pp. 2320–2327, 2011, doi: 10.1109/ICCV.2011.6126513.
 - [14] P. Ondruska, P. Kohli, and S. Izadi, “MobileFusion: Real-Time Volumetric Surface Reconstruction and Dense Tracking on Mobile Phones,” *IEEE Trans. Vis. Comput. Graph.*, vol. 21, no. 11, pp. 1251–1258, 2015, doi: 10.1109/TVCG.2015.2459902.
 - [15] J. C. Wyant and V. P. Bennett, “Using Computer Generated Holograms to Test Aspheric Wavefronts,” *Appl. Opt.*, vol. 11, no. 12, pp. 2833–2839, Dec. 1972, doi: 10.1364/AO.11.002833.
 - [16] Y.-Y. Cheng and J. C. Wyant, “Two-wavelength phase shifting interferometry,” *Appl. Opt.*, vol. 23, no. 24, p. 4539, 1984, doi: 10.1364/ao.23.004539.
 - [17] K. Creath, “V Phase-Measurement Interferometry Techniques,” vol. 26, E. Wolf, Ed. Elsevier, 1988, pp. 349–393.
 - [18] R. Jones, C. Wykes, and J. Wykes, *Holographic and speckle interferometry*, vol. 6. Cambridge university press, 1989.
 - [19] L. Deck and P. de Groot, “High-speed noncontact profiler based on scanning white-light interferometry,” *Appl. Opt.*, vol. 33, no. 31, pp. 7334–7338, Nov. 1994, doi: 10.1364/AO.33.007334.
 - [20] A. Fusiello, E. Trucco, and A. Verri, “Compact algorithm for rectification of stereo pairs,” *Mach. Vis. Appl.*, vol. 12, no. 1, pp. 16–22, 2000, doi: 10.1007/s001380050120.
 - [21] S. Zhang and P. S. Huang, “Novel method for structured light system calibration,” *Opt. Eng.*, vol. 45, no. 8, pp. 1–8, Aug. 2006, doi: 10.1117/1.2336196.
 - [22] M. Hansard, S. Lee, O. Choi, and R. P. Horaud, *Time-of-flight cameras: principles, methods and applications*. Springer Science & Business Media, 2012.
 - [23] A. Hata and D. Wolf, “Road marking detection using LIDAR reflective intensity data and its application to vehicle localization,” *2014 17th IEEE Int. Conf. Intell. Transp. Syst. ITSC 2014*, pp. 584–589, 2014, doi: 10.1109/ITSC.2014.6957753.

- [24] J. Hecht, "Lidar for self-driving cars," *Opt. Photonics News*, vol. 29, no. 1, pp. 26–33, 2018.
- [25] S. May, D. Droeschel, S. Fuchs, D. Holz, and A. Nüchter, "Robust 3D-mapping with time-of-flight cameras," in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2009, pp. 1673–1678, doi: 10.1109/IROS.2009.5354684.
- [26] P. Fankhauser, M. Bloesch, D. Rodriguez, R. Kaestner, M. Hutter, and R. Siegwart, "Kinect v2 for mobile robot navigation: Evaluation and modeling," in *2015 International Conference on Advanced Robotics (ICAR)*, 2015, pp. 388–394, doi: 10.1109/ICAR.2015.7251485.
- [27] J. Xu and S. Zhang, "Status, challenges, and future perspectives of fringe projection profilometry," *Opt. Lasers Eng.*, no. September 2019, 2020, doi: 10.1016/j.optlaseng.2020.106193.
- [28] L. C. Chen and C. C. Huang, "Miniaturized 3D surface profilometer using digital fringe projection," *Meas. Sci. Technol.*, vol. 16, no. 5, pp. 1061–1068, 2005, doi: 10.1088/0957-0233/16/5/003.
- [29] K. Genovese and C. Pappalettere, "Whole 3D shape reconstruction of vascular segments under pressure via fringe projection techniques," *Opt. Lasers Eng.*, vol. 44, no. 12, pp. 1311–1323, 2006, doi: 10.1016/j.optlaseng.2005.12.005.
- [30] F. Berryman, P. Pynsent, J. Fairbank, and S. Disney, "A new system for measuring three-dimensional back shape in scoliosis," *Eur. Spine J.*, vol. 17, no. 5, pp. 663–672, 2008, doi: 10.1007/s00586-007-0581-x.
- [31] J. I. Laughner, S. Zhang, H. Li, C. C. Shao, and I. R. Efimov, "Mapping cardiac surface mechanics with structured light imaging," *Am. J. Physiol. - Hear. Circ. Physiol.*, vol. 303, no. 6, pp. 712–720, 2012, doi: 10.1152/ajpheart.00269.2012.
- [32] J. Burke, T. Bothe, W. Osten, and C. F. Hess, "Reverse engineering by fringe projection," *Interferom. XI Appl.*, vol. 4778, no. June 2002, p. 312, 2002, doi: 10.1117/12.473547.
- [33] B. Zhang, J. Ziegert, F. Farahi, and A. Davies, "In situ surface topography of laser powder bed fusion using fringe projection," *Addit. Manuf.*, vol. 12, pp. 100–107, 2016, doi: 10.1016/j.addma.2016.08.001.
- [34] H. N. Yen, D. M. Tsai, and J. Y. Yang, "Full-field 3-D measurement of solder pastes using LCD-based phase shifting techniques," *IEEE Trans. Electron. Packag. Manuf.*, vol. 29, no. 1, pp. 50–57, 2006, doi: 10.1109/TEPM.2005.862632.

- [35] T. W. Hui and G. K. H. Pang, "Solder paste inspection using region-based defect detection," *Int. J. Adv. Manuf. Technol.*, vol. 42, no. 7–8, pp. 725–734, 2009, doi: 10.1007/s00170-008-1639-6.
- [36] F. Deng, Y. Ding, K. Peng, J. Xi, Y. Yin, and Z. Zhu, "Three-dimensional surface inspection for semiconductor components with fringe projection profilometry," *Opt. Metrol. Insp. Ind. Appl. IV*, vol. 10023, no. November 2016, p. 100230Y, 2016, doi: 10.1117/12.2246094.
- [37] G. Zhou, Z. Li, C. Wang, and Y. Shi, "A Novel Method for Human Expression Rapid Reconstruction," *Tsinghua Sci. Technol.*, vol. 14, no. SUPPL. 1, pp. 62–65, 2009, doi: 10.1016/S1007-0214(09)70008-2.
- [38] J. Xue, Q. Zhang, C. Li, W. Lang, M. Wang, and Y. Hu, "3D face profilometry based on galvanometer scanner with infrared fringe projection in high speed," *Appl. Sci.*, vol. 9, no. 7, 2019, doi: 10.3390/app9071458.
- [39] G. S. Spagnolo, D. Ambrosini, and D. Paoletti, "Low-cost optoelectronic system for three-dimensional artwork texture measurement," *IEEE Trans. Image Process.*, vol. 13, no. 3, pp. 390–396, 2004, doi: 10.1109/TIP.2003.821116.
- [40] G. Sansoni and F. Docchio, "3-D optical measurements in the field of cultural heritage: The case of the Vittoria Alata of Brescia," *IEEE Trans. Instrum. Meas.*, vol. 54, no. 1, pp. 359–368, 2005, doi: 10.1109/TIM.2004.838915.
- [41] T. Zaman, P. Jonker, B. Lenseigne, and J. Dik, "Simultaneous capture of the color and topography of paintings using fringe encoded stereo vision," *Herit. Sci.*, vol. 2, no. 1, pp. 1–10, 2014, doi: 10.1186/s40494-014-0023-0.
- [42] C. Zuo, S. Feng, L. Huang, T. Tao, W. Yin, and Q. Chen, "Phase shifting algorithms for fringe projection profilometry: A review," *Opt. Lasers Eng.*, vol. 109, no. August, pp. 23–59, 2018, doi: 10.1016/j.optlaseng.2018.04.019.
- [43] P. S. Huang, "Trapezoidal phase-shifting method for three-dimensional shape measurement," *Opt. Eng.*, vol. 44, no. 12, p. 123601, 2005, doi: 10.1117/1.2147311.
- [44] J. Kofman, "Two-step triangular-pattern phase-shifting method for three-dimensional object-shape measurement," *Opt. Eng.*, vol. 46, no. 8, p. 083201, 2007, doi: 10.1117/1.2768616.
- [45] S. Lei and S. Zhang, "Flexible 3-D shape measurement using projector defocusing," *Opt. Lett.*, vol. 34, no. 20, p. 3080, 2009, doi: 10.1364/ol.34.003080.
- [46] G. Rao, L. Song, S. Zhang, X. Yang, K. Chen, and J. Xu, "Depth-driven variable-

- frequency sinusoidal fringe pattern for accuracy improvement in fringe projection profilometry,” *Opt. Express*, vol. 26, no. 16, p. 19986, 2018, doi: 10.1364/oe.26.019986.
- [47] M. Zhang *et al.*, “Robust and efficient multi-frequency temporal phase unwrapping: optimal fringe frequency and pattern sequence selection,” *Opt. Express*, vol. 25, no. 17, p. 20381, 2017, doi: 10.1364/oe.25.020381.
 - [48] V. Srinivasan, H. C. Liu, and M. Halioua, “Automated phase-measuring profilometry of 3-D diffuse objects,” *Appl. Opt.*, vol. 23, no. 18, p. 3105, 1984, doi: 10.1364/ao.23.003105.
 - [49] S. Zhang and S.-T. Yau, “High-resolution, real-time 3D absolute coordinate measurement based on a phase-shifting method,” *Opt. Express*, vol. 14, no. 7, p. 2644, 2006, doi: 10.1364/oe.14.002644.
 - [50] M. Takeda and K. Mutoh, “Fourier transform profilometry for the automatic measurement of 3-D object shapes,” *Appl. Opt.*, vol. 22, no. 24, p. 3977, 1983, doi: 10.1364/ao.22.003977.
 - [51] X. Su and W. Chen, “Fourier transform profilometry: a review,” *Opt. Lasers Eng.*, vol. 35, no. 5, pp. 263–284, 2001, doi: [https://doi.org/10.1016/S0143-8166\(01\)00023-9](https://doi.org/10.1016/S0143-8166(01)00023-9).
 - [52] X. Su and Q. Zhang, “Dynamic 3-D shape measurement method: A review,” *Opt. Lasers Eng.*, vol. 48, no. 2, pp. 191–204, 2010, doi: 10.1016/j.optlaseng.2009.03.012.
 - [53] D. C. Ghiglia and M. D. Pritt, *Two-dimensional phase unwrapping: theory, algorithms, and software*, vol. 4. Wiley New York, 1998.
 - [54] X. Su and W. Chen, “Reliability-guided phase unwrapping algorithm: A review,” *Opt. Lasers Eng.*, vol. 42, no. 3, pp. 245–261, 2004, doi: 10.1016/j.optlaseng.2003.11.002.
 - [55] J. M. Huntley and H. Saldner, “Temporal phase-unwrapping algorithm for automated interferogram analysis,” *Appl. Opt.*, vol. 32, no. 17, p. 3047, 1993, doi: 10.1364/ao.32.003047.
 - [56] S. Zhang, “Absolute phase retrieval methods for digital fringe projection profilometry: A review,” *Opt. Lasers Eng.*, vol. 107, no. March, pp. 28–37, 2018, doi: 10.1016/j.optlaseng.2018.03.003.
 - [57] H. Guo, H. He, Y. Yu, and M. Chen, “Least-squares calibration method for fringe projection profilometry,” *Opt. Eng.*, vol. 44, no. 3, pp. 1–9, Mar. 2005, doi:

10.1117/1.1871832.

- [58] L. Huang, P. S. K. Chua, and A. Asundi, “Least-squares calibration method for fringe projection profilometry considering camera lens distortion,” *Appl. Opt.*, vol. 49, no. 9, pp. 1539–1548, 2010, doi: 10.1364/AO.49.001539.
- [59] Z. Zhang, H. Ma, T. Guo, S. Zhang, and J. Chen, “Simple, flexible calibration of phase calculation-based three-dimensional imaging system,” *Opt. Lett.*, vol. 36, no. 7, p. 1257, 2011, doi: 10.1364/ol.36.001257.
- [60] J. Lu, R. Mo, H. Sun, and Z. Chang, “Flexible calibration of phase-to-height conversion in fringe projection profilometry,” *Appl. Opt.*, vol. 55, no. 23, p. 6381, 2016, doi: 10.1364/ao.55.006381.
- [61] Y. Yin, X. Peng, A. Li, X. Liu, and B. Z. Gao, “Calibration of fringe projection profilometry with bundle adjustment strategy,” *Opt. Lett.*, vol. 37, no. 4, p. 542, 2012, doi: 10.1364/ol.37.000542.
- [62] W. Zhang, W. Li, L. Yu, H. Luo, H. Zhao, and H. Xia, “Sub-pixel projector calibration method for fringe projection profilometry,” *Opt. Express*, vol. 25, no. 16, p. 19158, 2017, doi: 10.1364/oe.25.019158.
- [63] K. Qian, “Windowed Fourier transform for fringe pattern analysis,” *Appl. Opt.*, vol. 43, no. 13, pp. 2695–2702, 2004, doi: 10.1364/AO.43.002695.
- [64] Q. Kemao, “Two-dimensional windowed Fourier transform for fringe pattern analysis: Principles, applications and implementations,” *Opt. Lasers Eng.*, vol. 45, no. 2, pp. 304–317, 2007, doi: 10.1016/j.optlaseng.2005.10.012.
- [65] J. Zhong and J. Weng, “Spatial carrier-fringe pattern analysis by means of wavelet transform: Wavelet transform profilometry,” *Appl. Opt.*, vol. 43, no. 26, pp. 4993–4998, 2004, doi: 10.1364/AO.43.004993.
- [66] M. A. Gdeisat, D. R. Burton, and M. J. P. Lalor, “Spatial carrier fringe pattern demodulation by use of a two-dimensional continuous wavelet transform,” *Appl. Opt.*, vol. 45, no. 34, pp. 8722–8732, 2006, doi: 10.1364/AO.45.008722.
- [67] M. A. Gdeisat *et al.*, “Spatial and temporal carrier fringe pattern demodulation using the one-dimensional continuous wavelet transform: Recent progress, challenges, and suggested developments,” *Opt. Lasers Eng.*, vol. 47, no. 12, pp. 1348–1361, 2009, doi: 10.1016/j.optlaseng.2009.07.009.
- [68] L. Huang, Q. Kemao, B. Pan, and A. K. Asundi, “Comparison of Fourier transform, windowed Fourier transform, and wavelet transform methods for phase extraction

- from a single fringe pattern in fringe projection profilometry,” *Opt. Lasers Eng.*, vol. 48, no. 2, pp. 141–148, 2010, doi: 10.1016/j.optlaseng.2009.04.003.
- [69] K. Creath, “V Phase-Measurement Interferometry Techniques,” *Prog. Opt.*, vol. 26, no. C, pp. 349–393, Jan. 1988, doi: 10.1016/S0079-6638(08)70178-1.
- [70] J. E. Greivenkamp, “Generalized Data Reduction For Heterodyne Interferometry,” *Opt. Eng.*, vol. 23, no. 4, pp. 350–352, 1984, doi: 10.1117/12.7973298.
- [71] P. Carré, “Installation et utilisation du comparateur photoélectrique et interférentiel du Bureau International des Poids et Mesures,” *Metrologia*, vol. 2, no. 1, pp. 13–23, 1966, doi: 10.1088/0026-1394/2/1/005.
- [72] P. Hariharan, B. F. Oreb, and T. Eiju, “Digital phase-shifting interferometry: a simple error-compensating phase calculation algorithm,” *Appl. Opt.*, vol. 26, no. 13, pp. 2504–2506, 1987.
- [73] J. Li, L. G. Hassebrook, and C. Guan, “Optimized two-frequency phase-measuring-profilometry light-sensor temporal-noise sensitivity,” *J. Opt. Soc. Am. A*, vol. 20, no. 1, p. 106, 2003, doi: 10.1364/josaa.20.000106.
- [74] M. Zhao, L. Huang, Q. Zhang, X. Su, A. Asundi, and Q. Kemao, “Quality-guided phase unwrapping technique: Comparison of quality maps and guiding strategies,” *Appl. Opt.*, vol. 50, no. 33, pp. 6214–6224, 2011, doi: 10.1364/AO.50.006214.
- [75] B. Budianto, P. K. D. Lun, and T.-C. Hsung, “Marker encoded fringe projection profilometry for efficient 3D model acquisition,” *Appl. Opt.*, vol. 53, no. 31, pp. 7442–7453, 2014, doi: 10.1364/AO.53.007442.
- [76] P. Cong, Z. Xiong, Y. Zhang, S. Zhao, and F. Wu, “Accurate Dynamic 3D Sensing With Fourier-Assisted Phase Shifting,” *IEEE J. Sel. Top. Signal Process.*, vol. 9, no. 3, pp. 396–408, 2015, doi: 10.1109/JSTSP.2014.2378217.
- [77] B. Li, Y. Wang, J. Dai, W. Lohry, and S. Zhang, “Some recent advances on superfast 3D shape measurement with digital binary defocusing techniques,” *Opt. Lasers Eng.*, vol. 54, pp. 236–246, 2014, doi: 10.1016/j.optlaseng.2013.07.010.
- [78] D. Zheng, Q. Kemao, F. Da, and H. S. Seah, “Ternary Gray code-based phase unwrapping for 3D measurement using binary patterns with projector defocusing,” *Appl. Opt.*, vol. 56, no. 13, p. 3660, 2017, doi: 10.1364/ao.56.003660.
- [79] S. Zhang and S.-T. Yau, “High-speed three-dimensional shape measurement system using a modified two-plus-one phase-shifting algorithm,” *Opt. Eng.*, vol. 46, no. 11, p. 113603, 2007, doi: 10.1117/1.2802546.

- [80] C. Zuo *et al.*, “High-speed three-dimensional shape measurement for dynamic scenes using bi-frequency tripolar pulse-width-modulation fringe projection,” *Opt. Lasers Eng.*, vol. 51, no. 8, pp. 953–960, 2013, doi: 10.1016/j.optlaseng.2013.02.012.
- [81] J. S. Hyun and S. Zhang, “Superfast 3D absolute shape measurement using five binary patterns,” *Opt. Lasers Eng.*, vol. 90, no. July 2016, pp. 217–224, 2017, doi: 10.1016/j.optlaseng.2016.10.017.
- [82] J.-S. Hyun and S. Zhang, “Enhanced two-frequency phase-shifting method,” *Appl. Opt.*, vol. 55, no. 16, pp. 4395–4401, 2016, doi: <https://doi.org/10.1364/AO.55.004395>.
- [83] C. Zhou, T. Liu, S. Si, J. Xu, Y. Liu, and Z. Lei, “An improved stair phase encoding method for absolute phase retrieval,” *Opt. Lasers Eng.*, vol. 66, pp. 269–278, 2015, doi: 10.1016/j.optlaseng.2014.09.011.
- [84] Y. Xing, C. Quan, and C. J. Tay, “Multi-subzone algorithm for absolute phase retrieval in digital fringe projection profilometry,” *Appl. Opt.*, vol. 55, no. 34, p. 9774, 2016, doi: 10.1364/ao.55.009774.
- [85] S. Zhang, “Flexible 3D shape measurement using projector defocusing: extended measurement range,” *Opt. Lett.*, vol. 35, no. 7, p. 934, 2010, doi: 10.1364/ol.35.000934.
- [86] Y. Wang, S. Zhang, and J. H. Oliver, “3D shape measurement technique for multiple rapidly moving objects,” *Opt. Express*, vol. 19, no. 9, p. 8539, 2011, doi: 10.1364/oe.19.008539.
- [87] D. Zheng and F. Da, “Self-correction phase unwrapping method based on Gray-code light,” *Opt. Lasers Eng.*, vol. 50, no. 8, pp. 1130–1139, 2012, doi: 10.1016/j.optlaseng.2012.01.019.
- [88] D. Zheng, F. Da, Q. Kemao, and H. S. Seah, “Phase-shifting profilometry combined with Gray-code patterns projection: unwrapping error removal by an adaptive median filter,” *Opt. Express*, vol. 25, no. 5, p. 4700, 2017, doi: 10.1364/oe.25.004700.
- [89] Q. Zhang, X. Su, L. Xiang, and X. Sun, “3-D shape measurement based on complementary Gray-code light,” *Opt. Lasers Eng.*, vol. 50, no. 4, pp. 574–579, 2012, doi: 10.1016/j.optlaseng.2011.06.024.
- [90] Z. Wu, W. Guo, Y. Li, Y. Liu, and Q. Zhang, “High-speed and high-efficiency three-dimensional shape measurement based on Gray-coded light,” *Photon. Res.*, vol. 8,

- no. 6, pp. 819–829, Jun. 2020, doi: 10.1364/PRJ.389076.
- [91] C. Zuo, L. Huang, M. Zhang, Q. Chen, and A. Asundi, “Temporal phase unwrapping algorithms for fringe projection profilometry: A comparative review,” *Opt. Lasers Eng.*, vol. 85, pp. 84–103, 2016, doi: 10.1016/j.optlaseng.2016.04.022.
 - [92] H. Zhao, W. Chen, and Y. Tan, “Phase-unwrapping algorithm for the measurement of three-dimensional object shapes,” *Appl. Opt.*, vol. 33, no. 20, pp. 4497–4500, Jul. 1994, doi: 10.1364/AO.33.004497.
 - [93] Z. Wang, D. A. Nguyen, and J. C. Barnes, “Some practical considerations in fringe projection profilometry,” *Opt. Lasers Eng.*, vol. 48, no. 2, pp. 218–225, 2010, doi: 10.1016/j.optlaseng.2009.06.005.
 - [94] W. Yin *et al.*, “Temporal phase unwrapping using deep learning,” *Sci. Rep.*, vol. 9, no. 1, pp. 1–12, 2019, doi: 10.1038/s41598-019-56222-3.
 - [95] Y.-Y. Cheng and J. C. Wyant, “Multiple-wavelength phase-shifting interferometry,” *Appl. Opt.*, vol. 24, no. 6, pp. 804–807, 1985, doi: 10.1364/AO.24.000804.
 - [96] Y. Wang and S. Zhang, “Novel phase-coding method for absolute phase retrieval,” *Opt. Lett.*, vol. 37, no. 11, p. 2067, 2012, doi: 10.1364/ol.37.002067.
 - [97] D. Zheng and F. Da, “Phase coding method for absolute phase retrieval with a large number of codewords,” *Opt. Express*, vol. 20, no. 22, p. 24139, 2012, doi: 10.1364/oe.20.024139.
 - [98] Y. Xing, C. Quan, and C. J. Tay, “A modified phase-coding method for absolute phase retrieval,” *Opt. Lasers Eng.*, vol. 87, pp. 97–102, 2016, doi: 10.1016/j.optlaseng.2016.03.018.
 - [99] B. Li, Y. Fu, J. Zhang, H. Wu, G. Xia, and G. Jiang, “A fast three-dimensional shape measurement method based on color phase coding,” *Optik (Stuttg.)*, vol. 127, no. 3, pp. 1011–1015, 2016, doi: 10.1016/j.ijleo.2015.10.036.
 - [100] Y. An and S. Zhang, “Three-dimensional absolute shape measurement by combining binary statistical pattern matching with phase-shifting methods,” *Appl. Opt.*, vol. 56, no. 19, p. 5418, 2017, doi: 10.1364/ao.56.005418.
 - [101] W. Lohry, V. Chen, and S. Zhang, “Absolute three-dimensional shape measurement using coded fringe patterns without phase unwrapping or projector calibration,” *Opt. Express*, vol. 22, no. 2, p. 1287, 2014, doi: 10.1364/oe.22.001287.
 - [102] K. Liu, Y. Wang, D. L. Lau, Q. Hao, and L. G. Hassebrook, “Dual-frequency pattern scheme for high-speed 3-D shape measurement,” *Opt. Express*, vol. 18, no. 5, p.

- 5229, 2010, doi: 10.1364/oe.18.005229.
- [103] Y. Wang, K. Liu, Q. Hao, D. L. Lau, and L. G. Hassebrook, "Period coded phase shifting strategy for real-time 3-D structured light illumination," *IEEE Trans. Image Process.*, vol. 20, no. 11, pp. 3001–3013, 2011, doi: 10.1109/TIP.2011.2155072.
 - [104] W. Yin *et al.*, "High-speed 3D shape measurement using the optimized composite fringe patterns and stereo-assisted structured light system," *Opt. Express*, vol. 27, no. 3, p. 2411, 2019, doi: 10.1364/oe.27.002411.
 - [105] W. Lohry and S. Zhang, "High-speed absolute three-dimensional shape measurement using three binary dithered patterns," *Opt. Express*, vol. 22, no. 22, p. 26752, 2014, doi: 10.1364/oe.22.026752.
 - [106] T. Tao, Q. Chen, J. Da, S. Feng, Y. Hu, and C. Zuo, "Real-time 3-D shape measurement with composite phase-shifting fringes and multi-view system," *Opt. Express*, vol. 24, no. 18, p. 20253, 2016, doi: 10.1364/oe.24.020253.
 - [107] G. Wu, Y. Wu, L. Li, and F. Liu, "High-resolution few-pattern method for 3D optical measurement," *Opt. Lett.*, vol. 44, no. 14, p. 3602, 2019, doi: 10.1364/ol.44.003602.
 - [108] S. Feng *et al.*, "Fringe pattern analysis using deep learning," *Adv. Photonics*, vol. 1, no. 02, p. 1, 2019, doi: 10.1117/1.ap.1.2.025001.
 - [109] S. Feng, C. Zuo, W. Yin, G. Gu, and Q. Chen, "Micro deep learning profilometry for high-speed 3D surface imaging," *Opt. Lasers Eng.*, vol. 121, no. April, pp. 416–427, 2019, doi: 10.1016/j.optlaseng.2019.04.020.
 - [110] J. Qian *et al.*, "Single-shot absolute 3D shape measurement with deep-learning-based color fringe projection profilometry," *Opt. Lett.*, vol. 45, no. 7, p. 1842, 2020, doi: 10.1364/ol.388994.
 - [111] S. Zhang, D. Van Der Weide, and J. Oliver, "Superfast phase-shifting method for 3-D shape measurement," *Opt. Express*, vol. 18, no. 9, p. 9684, 2010, doi: 10.1364/oe.18.009684.
 - [112] Y. Gong and S. Zhang, "Ultrafast 3-D shape measurement with an off-the-shelf DLP projector," *Opt. Express*, vol. 18, no. 19, p. 19743, 2010, doi: 10.1364/oe.18.019743.
 - [113] S. Zhang, "Recent progresses on real-time 3D shape measurement using digital fringe projection techniques," *Opt. Lasers Eng.*, vol. 48, no. 2, pp. 149–158, 2010, doi: 10.1016/j.optlaseng.2009.03.008.
 - [114] D. Zheng, F. Da, Q. Kemao, and H. S. Seah, "Phase error analysis and compensation for phase shifting profilometry with projector defocusing," *Appl. Opt.*, vol. 55, no.

- 21, p. 5721, 2016, doi: 10.1364/ao.55.005721.
- [115] G. A. Ayubi, J. A. Ayubi, J. M. Di Martino, and J. A. Ferrari, "Pulse-width modulation in defocused three-dimensional fringe projection," *Opt. Lett.*, vol. 35, no. 21, p. 3682, 2010, doi: 10.1364/ol.35.003682.
 - [116] Y. Wang and S. Zhang, "Optimal pulse width modulation for sinusoidal fringe generation with projector defocusing," *Opt. Lett.*, vol. 35, no. 24, p. 4121, 2010, doi: 10.1364/ol.35.004121.
 - [117] C. Zuo, Q. Chen, S. Feng, F. Feng, G. Gu, and X. Sui, "Optimized pulse width modulation pattern strategy for three-dimensional profilometry with projector defocusing," *Appl. Opt.*, vol. 51, no. 19, pp. 4477–4490, 2012, doi: 10.1364/AO.51.004477.
 - [118] Y. Wang and S. Zhang, "Superfast multifrequency phase-shifting technique with optimal pulse width modulation," *Opt. Express*, vol. 19, no. 6, p. 5149, 2011, doi: 10.1364/oe.19.005149.
 - [119] B. E. BAYER, "An optimum method for two-level rendition of continuous tone pictures," *IEEE Int. Conf. Commun. June, 1973*, vol. "26-11"-"2, 1973, [Online]. Available: <https://ci.nii.ac.jp/naid/20000818242/en/>.
 - [120] R. W. Floyd, "An adaptive algorithm for spatial gray-scale," in *Proc. Soc. Inf. Disp.*, 1976, vol. 17, pp. 75–77.
 - [121] P. Stucki, "MECCA: a multiple-error correction computation algorithm for bi-level image hardcopy reproduction." Verlag nicht ermittelbar, 1981.
 - [122] J. Dai, B. Li, and S. Zhang, "High-quality fringe pattern generation using binary pattern optimization through symmetry and periodicity," *Opt. Lasers Eng.*, vol. 52, no. 1, pp. 195–200, 2014, doi: 10.1016/j.optlaseng.2013.06.010.
 - [123] W. Lohry and S. Zhang, "Genetic method to optimize binary dithering technique for high-quality fringe generation," *Opt. Lett.*, vol. 38, no. 4, p. 540, 2013, doi: 10.1364/ol.38.000540.
 - [124] J. Dai and S. Zhang, "Phase-optimized dithering technique for high-quality 3D shape measurement," *Opt. Lasers Eng.*, vol. 51, no. 6, pp. 790–795, 2013, doi: 10.1016/j.optlaseng.2013.02.003.
 - [125] Y. An, J.-S. Hyun, and S. Zhang, "Pixel-wise absolute phase unwrapping using geometric constraints of structured light system," *Opt. Express*, vol. 24, no. 16, pp. 18445–18459, Aug. 2016, doi: 10.1364/OE.24.018445.

- [126] W. Lohry and S. Zhang, “3D shape measurement with 2D area modulated binary patterns,” *Opt. Lasers Eng.*, vol. 50, no. 7, pp. 917–921, 2012, doi: 10.1016/j.optlaseng.2012.03.002.
- [127] H. Wang, Q. Kemao, and S. H. Soon, “Valid point detection in fringe projection profilometry,” *Opt. Express*, vol. 23, no. 6, p. 7535, 2015, doi: 10.1364/oe.23.007535.
- [128] X. He, D. Zheng, Q. Kemao, and G. Christopoulos, “Quaternary gray-code phase unwrapping for binary fringe projection profilometry,” *Opt. Lasers Eng.*, vol. 121, no. May, pp. 358–368, 2019, doi: 10.1016/j.optlaseng.2019.04.009.
- [129] R. C. Gonzalez and R. E. Woods, *Digital Image Processing (3rd Edition)*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 2006.
- [130] Y. Wang and S. Zhang, “Three-dimensional shape measurement with binary dithered patterns,” *Appl. Opt.*, vol. 51, no. 27, pp. 6631–6636, 2012, doi: 10.1364/AO.51.006631.
- [131] G. Sansoni, M. Carocci, and R. Rodella, “Three-dimensional vision based on a combination of gray-code and phase-shift light projection: analysis and compensation of the systematic errors,” *Appl. Opt.*, vol. 38, no. 31, p. 6565, 1999, doi: 10.1364/ao.38.006565.
- [132] X. He, D. Zheng, Q. Kemao, and G. Christopoulos, “Quaternary gray-code phase unwrapping for binary fringe projection profilometry,” *Opt. Lasers Eng.*, vol. 121, no. April, pp. 358–368, 2019, doi: 10.1016/j.optlaseng.2019.04.009.
- [133] X. He and Q. Kemao, “A comparison of n-ary simple code and n-ary gray code phase unwrapping in high-speed fringe projection profilometry,” *Opt. Lasers Eng.*, vol. 128, no. January, p. 106046, 2020, doi: 10.1016/j.optlaseng.2020.106046.
- [134] H. Guo, H. He, and M. Chen, “Gamma correction for digital fringe projection profilometry,” *Appl. Opt.*, vol. 43, no. 14, pp. 2906–2914, 2004, doi: 10.1364/AO.43.002906.