

# Advanced phase shifting algorithms: development and evaluation

Chen, Yuchi

2022

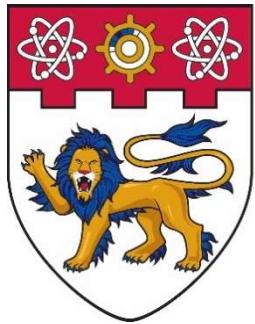
Chen, Y. (2022). Advanced phase shifting algorithms: development and evaluation.  
Doctoral thesis, Nanyang Technological University, Singapore.  
<https://hdl.handle.net/10356/159437>

<https://hdl.handle.net/10356/159437>

---

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

*Downloaded on 14 Oct 2024 16:16:06 SGT*



**NANYANG  
TECHNOLOGICAL  
UNIVERSITY**  
**SINGAPORE**

**ADVANCED PHASE SHIFTING ALGORITHMS: DEVELOPMENT AND  
EVALUATION**

**CHEN YUCHI**

**SCHOOL OF COMPUTER SCIENCE AND ENGINEERING**

**2022**

**ADVANCED PHASE SHIFTING ALGORITHMS:  
DEVELOPMENT AND EVALUATION**

**CHEN YUCHI**

**2022**

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

**2022**

## **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.

11/ Jan. /2022

.....  
Date

.....  
陈宇驰  
.....

Chen Yuchi

## **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.

11/ Jan. /2022

TU NTU NTU NTU NTU NTU NTU NTU  
NTU NTU NTU NTU NTU NTU NTU NTU NTU  
NTU NTU NTU NTU NTU NTU NTU NTU NTU  
NTU NTU NTU NTU NTU NTU NTU NTU NTU



.....

Date

.....

Qian Kemao

# Authorship Attribution Statement

This thesis contains material from 4 paper(s) published in the following peer-reviewed journal(s) where I was the first and/or corresponding author.

Chapter 3 is published as Y. Chen and Q. Kema, "Advanced iterative algorithm for phase extraction: performance evaluation and enhancement," *Optics express*, vol. 27, no. 26, pp. 37634-37651, 2019.

The contributions of the co-authors are as follows:

- Prof. Qian Kema provided the initial project direction.
- I prepared the manuscript drafts. Prof. Qian Kema and I reviewed and revised the manuscript.
- Prof. Qian Kema and I proposed an enhanced advanced iterative algorithm to improve the accuracy and the robustness of the advanced iterative algorithm.
- I conducted simulations and experiments for the evaluation of the advanced iterative algorithm and the enhanced advanced iterative algorithm.

Chapter 4 is published as Y. Chen and Q. Kema, "General iterative algorithm for phase extraction from fringe patterns with random phase-shifts, intensity harmonics, and non-uniform phase-shift distribution," *Optics Express*, vol. 29, no. 19, pp. 30905-30926, 2021.

The contributions of the co-authors are as follows:

- Prof. Qian Kema supervised me on the overall research direction.
- I prepared the manuscript drafts. Prof. Qian Kema and I reviewed and revised the manuscript.
- Prof. Qian Kema and I proposed a general iterative algorithm which is immune to the common error sources.
- I conducted simulations and experiments to compare the performance of the general iterative algorithm with other phase-shifting algorithms.

Chapter 5 is published as Y. Chen and Q. Kemao, "Phase-shifting algorithms with known and unknown phase shifts: comparison and hybrid,". Optics Express, vol. 30, no. 5, pp. 8275-8302, 2022.

The contributions of the co-authors are as follows:

- Prof. Qian Kemao supervised me on the overall research direction.
  - I prepared the manuscript drafts. Prof. Qian Kemao and I reviewed and revised the manuscript.
  - Prof. Qian Kemao and I proposed a hybrid kPSA-GIA which combines the advantages of different phase-shifting algorithms.
  - I conducted simulations and experiments to compare the performance of different phase-shifting algorithms and to confirm the good performance of the kPSA-GIA.

Chapter 6 is published as Y. Chen, T. Wang, and Q. Kemao, "Parallel advanced iterative algorithm for phase extraction with unknown phase-shifts," *Optics and Lasers in Engineering*, vol. 138, p. 106408, 2021.

The contributions of the co-authors are as follows:

- Prof. Qian Kemao and Doctor Wang Tianyi supervised me on the overall research direction.
  - I prepared the manuscript drafts. Prof. Qian Kemao, Doctor Wang Tianyi, and I reviewed and revised the manuscript.
  - Prof. Qian Kemao, Doctor Wang Tianyi, and I proposed a fully parallelized GPU-accelerated advanced iterative algorithm which improves the speed of the advanced iterative algorithm.
  - I conducted experiments to confirm the fast speed and good accuracy of the fully parallelized GPU-accelerated advanced iterative algorithm.

11/Jan. /2022

Date

Chen Yuchi

# Acknowledgments

Firstly, I am greatly honored that I can have the opportunity to study at Nanyang Technological University (NTU) as a Ph.D candidate. I am grateful to my supervisor Professor Qian Kemaor for his monitoring and support on my Ph.D study and research. Professor Qian is an innovative, patient, and motivating person who is passionate about research. These qualities profoundly affect me. This thesis cannot be finished without his valuable advice and guidance.

Secondly, I would like to express my appreciation to my co-supervisor, Doctor Chen Dengpeng for his consistent support and encouragement. It is not possible for me to study at NTU as a Ph.D candidate without his great effort. By working with Doctor Chen Dengpeng, I have gained a lot of experience. His high requirement for systemic thinking has had a great impact on me. Besides, I would like to thank Economic Development Board for providing the financial support.

Thirdly, I am grateful to all my thesis advisory committee members, Professor Lap-Pui Chau and Assistant Professor Zhang Hanwang for their kind support, constructive comments, and insightful questions on my research work. I genuinely cherish the opportunity to be guided by such a wonderful Thesis Advisory Committee team. Meanwhile, special thanks to my senior Doctor Wang Tianyi for his guidance and help in my academic life.

Fourthly, I would like to express my sincere appreciation to the examiners for their efforts and patience.

Last but not least, I would like to thank my parents for their belief and emotional support. Meanwhile, I would like to thank my wife, Wang Yuzheng, for her accompany along my Ph.D journey.



# Contents

Abstract .....	v
List of Figures .....	viii
List of Table .....	xi
Chapter 1 Introduction .....	1
1.1. Optical metrology .....	1
1.2. Laser interferometry .....	2
1.2.1. Fringe skeletonizing technique .....	4
1.2.2. Phase-shifting technique .....	5
1.2.3. Fourier transform technique .....	6
1.2.4. Phase unwrapping .....	8
1.3. Possible error sources and more phase-shifting algorithms .....	9
1.3.1. Possible error sources in the fringe patterns .....	9
1.3.2. More phase-shifting algorithms .....	11
1.4. Motivation, contributions, and thesis organization .....	13
1.4.1. Motivations .....	13
1.4.2. Contributions .....	14
1.4.3. Organization .....	15
Chapter 2 Literature review .....	17
2.1. The phase-shifting algorithm with known phase shifts (kPSA) .....	17
2.1.1. Principle of the kPSAs .....	17
2.1.2. Evaluation and issues of the kPSAs .....	18
2.2. Phase-shifting algorithms with unknown phase shifts (uPSAs) .....	21
2.2.1. Trigonometric identities based uPSAs .....	21
2.2.2. Decomposition based uPSAs .....	22
2.2.3. Optimization based uPSAs .....	24
Chapter 3 Advanced iterative algorithm for phase extraction: performance evaluation and enhancement .....	37
3.1. Background .....	37
3.2. Condition numbers for phase and phase shifts characterization .....	38

3.3. Simulation strategy and AIA's robustness .....	39
3.3.1. Simulation strategy .....	39
3.3.2. Fringe density requirement .....	41
3.4. AIA's accuracy performance .....	42
3.4.1. Influence of the phase shifts .....	42
3.4.2. Influence of frame number .....	43
3.4.3. Influence of noise .....	44
3.4.4. The AIA's phase error model .....	44
3.5. AIA's improving practices .....	45
3.5.1. Controlling the phase shifts .....	45
3.5.2. Controlling the frame number .....	46
3.5.3. Suppressing the noise .....	47
3.6. Error estimation.....	48
3.6.1. Phase shifts .....	48
3.6.2. Fringe amplitude.....	49
3.6.3. Noise amplitude .....	49
3.6.4. Phase error .....	49
3.7. The Enhanced AIA (eAIA) .....	50
3.8. Experimental verification.....	53
3.9. Summary .....	55
Chapter 4 General iterative algorithm for phase extraction from fringe patterns with random phase shifts, intensity harmonics and nonuniform phase-shift distribution.....	57
4.1. Background .....	57
4.2. General fringe pattern model.....	58
4.3. The GIA.....	58
4.3.1. The phase extraction strategy of the AIA .....	58
4.3.2. The implementation of the GIA.....	61
4.4. Simulation evaluation.....	63
4.4.1. Algorithm specifications.....	63
4.4.2. Simulated fringe pattern .....	63
4.4.3. Example with random bias phase shifts .....	66

4.4.4. Example with regular bias phase shifts .....	66
4.5. Experimental verification.....	67
4.5.1. Set up and ground truth .....	68
4.5.2. Results .....	69
4.6. Summary .....	70
Chapter 5 Evaluation of phase-shifting algorithm with known and unknown phase shifts.....	73
5.1. Background .....	73
5.2. Codes, fringe patterns and error measures .....	73
5.3. The pre-requisites of the uPSAs.....	75
5.3.1. On the phase .....	75
5.3.2. On the phase-shift distribution .....	78
5.3.3. On the background intensity and the fringe amplitude.....	79
5.3.4. Summary.....	81
5.4. Within the uPSAs: the performance with unknown phase shifts .....	82
5.4.1. Phase-shift errors .....	82
5.4.2. Intensity harmonics.....	83
5.4.3. Phase-shift nonuniformities .....	85
5.4.4. Random noise .....	86
5.4.5. More tests and the summary.....	87
5.5. Between the uPSAs and the kPSAs: the performance with known phase shifts.....	88
5.5.1. Phase-shift errors .....	89
5.5.2. Intensity harmonics.....	90
5.5.3. Phase-shift nonuniformities .....	91
5.5.4. Random noise .....	92
5.5.5. More tests and the summary.....	93
5.6. Hybrid kPSA-GIA .....	93
5.7. Experimental verification.....	96
5.8. Summary .....	98
Chapter 6 Parallel phase extraction algorithm for phase-shifted fringe patterns with unknown phase shifts.....	101
6.1. Background .....	101

6.2. The AIA's parallelization.....	102
6.2.1. The AIA's structure .....	102
6.2.2. Data cropping and the risk.....	103
6.2.3. Parallel decomposition of the AIA .....	104
6.3. gAIA's CUDA implementation .....	105
6.3.1. Mapping between CUDA and a NVIDIA's GPU .....	106
6.3.2. gAIA's flowchart and implementation .....	106
6.4. Experimental verification.....	110
6.4.1. Algorithm implementations and data collection for comparison .....	110
6.4.2. The accuracy.....	111
6.4.3. The speed .....	112
6.5. Summary .....	113
Chapter 7 Conclusion, and future works.....	115
7.1. Conclusions .....	115
7.2. Future work .....	116
7.2.1. Improve the computation efficiency of the GIA .....	116
7.2.2. Further improvement of the GIA's accuracy.....	117
7.2.3. Comparison between the kPSA, the uPSA and deep learning based phase-shifting algorithms .....	117
7.2.4. Optimization based uPSA with global optimization techniques .....	118
Author's Publications.....	119
Reference .....	121

# Abstract

Optical metrology has diverse applications for quality control and geometry evaluation of optical components, semiconductor components, and mechanical objects. Among the various techniques of optical metrology, phase-shifting interferometry (PSI) has achieved great success in various engineering fields due to its merits of high sensitivity full field measurement, and easiness to use. In the PSI, the measured quantities are encoded into the phase of phase-shifted fringe patterns. Thus, the phase is expected to be extracted accurately. However, in practice, phase errors are inevitable due to the following common error sources in a PSI system including (i) the phase-shift errors, (ii) the intensity harmonics, (iii) the nonuniform phase-shift distributions and (iv) the random intensity noise. Hence, to suppress the impact of these error sources, researchers have proposed many phase-shifting algorithms (PSAs). These PSAs are either with known phase shifts and based on simple triangular identities (abbreviated as kPSA) or with unknown phase shifts and based on advanced techniques (abbreviated as uPSA). Subsequently, many development and evaluation studies focusing on the robustness, the accuracy and the speed of the PSAs were conducted. However, these evaluation and development works are mainly for the kPSAs and only a few are for the uPSAs, which makes the overall picture of the PSAs unclear. Therefore, we conducted a systematic study on the developments and evaluations of the uPSAs in this thesis.

Firstly, we focused on the evaluation and enhancement of the uPSAs' robustness and accuracy. We started with an elegant and well-recognized uPSA, the advanced iterative algorithm (AIA). Although the AIA has good performance shown in literature, it has not been sufficiently investigated on its robustness and accuracy with different types of fringe patterns, which leads to our systematic evaluation. One of the difficulties in the AIA's evaluation is the characterization of the phase and phase shifts, for which we used the AIA's least-squares matrix's condition number. Subsequently, for the robustness evaluation, theoretical evaluations based on the condition numbers and a large number of simulations were carried out. Based on the theoretical evaluations and simulations, the restriction on the fringe density for the AIA's success was confirmed. To evaluate the AIA's accuracy, simulations were carried out to examine the influence of the phase shifts, the frame numbers and the noise levels. Based on the simulations' results, a phase error model was established for the AIA and three individual improving practices, *i.e.*, controlling the

phase shifts, controlling the frame numbers and suppressing noise, were proposed. Moreover, by combining the findings in the AIA's robustness and accuracy evaluation, an enhanced AIA (eAIA) was proposed for the enhancement of the AIA's robustness and accuracy. Meanwhile, the eAIA can also predict the phase error based on the AIA's phase error model and the proposed practical estimation methods.

Secondly, we focused on the accuracy improvement of the uPSAs in the condition that all the common error sources are presented in the fringe patterns. Although different uPSAs have been proposed to suppress the impact of different error sources, these uPSAs restricted their fringe pattern model in different ways, thus none of them are immune to all the common error sources. Hence, we proposed a general iterative algorithm (GIA) which ambitiously incorporates all the concerned error sources into its fringe pattern model and estimates the errors together with the phase and the phase shifts. The unknowns in the GIA were divided into three groups based on their different natures which are (i) the background intensity and the fringe amplitudes, (ii) the phase, and (iii) the phase-shift related parameters. Subsequently, we optimized the three groups of unknowns alternatively by using the Levenberg-Marquart method. In addition, the background intensity and the fringe amplitudes were estimated within a small window in the GIA for better immunity to random intensity noise.

Thirdly, we focused on the robustness evaluation of more uPSAs and the accuracy comparison among all categories of PSAs including both the uPSAs and the typical kPSAs. To fulfill this task, we comprehensively evaluated the restrictions of four representative uPSAs, the AIA, the algorithm based on the principal component analysis, the algorithm based on VU factorization, and the GIA. We then compared the accuracy of these uPSAs among themselves and also with twelve benchmarking kPSAs through simulations. Our simulations with random phase shifts demonstrated the outstanding accuracy performance of the GIA. Meanwhile, our simulations with regular phase shifts showcased the advantages and disadvantages of both the kPSAs and the uPSAs. The kPSAs have fast speed and robust performance. Besides, it is always possible to find a suitable algorithm to tackle the presenting error source(s). However, selecting a proper kPSA needs a deep understanding about the kPSAs. Meanwhile, the uPSAs with better accuracy, especially the GIA, are very strong competitors and can be powerful aids to the kPSAs if the restrictions are not violated. By combining the advantages of both the kPSAs and uPSAs, a hybrid kPSA-GIA was proposed to improve the accuracy of the kPSAs and to relieve the restrictions of the GIA.

Finally, we focused on the slow computation issue of the AIA. The AIA alternatively and iteratively estimates the phase and the phase shifts. Hence, the AIA is much slower compared to the triangular identities based kPSAs. Currently, there is only one attempt for accelerating the AIA by using a hybrid GPU-CPU implementation. This attempt accelerates the phase calculation by GPU cores and keeps the phase shifts calculation on the CPU, which has not realized the full potential of the GPU based acceleration. Benefiting from an explicit evaluation on the AIA's structure, we were able to propose a fully parallelized GPU-accelerated AIA (gAIA) in which a pointwise pattern was used to accelerate the phase estimation and a parallel reduction pattern was used to break the bottleneck of the AIA in phase shifts estimation. The gAIA can achieve similar accuracy performance and  $500\times$  speedup compared to the AIA which is sequentially executed on a single-core CPU. Meanwhile, compared to the hybrid GPU-CPU implemented AIA which has a risk of divergence,  $10\times$  speedup is achieved by the gAIA. With a normal GPU, the gAIA achieved real-time phase extraction, *i.e.*, only 26.55 ms is needed for the gAIA for extracting phase from 13 frames of fringe patterns with  $2048\times 2048$  pixels per frame.

**Keywords:** Phase-shifting interferometry, phase-shifting algorithm, development and evaluation, unknown phase shifts.

# List of Figures

Figure 1.1.Accuracy and measurement range of various optical metrology techniques. Image was recreated based on the image in [10]. .....	2
Figure 1.2.Typical Michelson interferometer [108]. .....	3
Figure 1.3.One simulated fringe pattern. ....	4
Figure 1.4.Fringe patterns with different phase shifts. ....	5
Figure 1.5.Wrapped phase. ....	6
Figure 1.6.One simulated fringe pattern with a spatial carrier. ....	7
Figure 1.7.The wrapped phase map. ....	8
Figure 1.8.Unwrapped phase map. ....	9
Figure 1.9.Fringe pattern with intensity harmonics. ....	10
Figure 3.1.Fringe patterns used in simulations. ....	40
Figure 3.2.Performance with different densities. ....	41
Figure 3.3.A fringe pattern with complex phase, $N_p=10$ , $\omega_c=0.25$ and random noise with $\sigma/B=10\%$ .....	42
Figure 3.4.RMSEs of the phases extracted from fringe patterns with different phase shifts. ....	43
Figure 3.5.RMSEs of the phases extracted from fringe patterns with different frame numbers..	43
Figure 3.6.RMSEs of the phases extracted from fringe patterns with different noise levels. ....	44
Figure 3.7.RMSEs of the phases extracted from fringe patterns with different frame numbers and different noise levels. ....	45
Figure 3.8.Improvement by controlling the phase shifts. ....	46
Figure 3.9. Performance with frame number control.....	47
Figure 3.10.Improvement by suppressing the noise. ....	47
Figure 3.11.Prediction of the AIA's accuracy. ....	50
Figure 3.12.Flowchart of eAIA.....	52
Figure 3.13.Experiment data.....	53
Figure 3.14.Performance of eAIA. ....	54
Figure 3.15.Extracted phases. ....	54
Figure 4.1.Pseudocode of LM algorithm .....	60
Figure 4.2.Flowchart of GIA. ....	61

Figure 4.3.Parameters for simulation.....	65
Figure 4.4.Phase errors of example with random bias phase shifts.....	66
Figure 4.5.Phase errors of example with regular bias phase shifts.....	67
Figure 4.6.Sketch of the Fizeau interferometer .....	68
Figure 4.7.Experimental parameters.....	68
Figure 4.8.Experiment results (unit: rads). ....	69
Figure 4.9.Difference between the measured intensity and fringe pattern generated by.....	70
Figure 4.10.Phase error results from the experiment.....	70
Figure 5.1.Simulation parameters of the common ground. ....	74
Figure 5.2.Fringe patterns with different fringe densities.....	76
Figure 5.3.Phase errors from fringe patterns with different fringe densities. ....	76
Figure 5.4.Simulation parameters of concentrated fringe pattern.....	77
Figure 5.5.Phase errors from concentrated fringe patterns. ....	77
Figure 5.6.Phase errors from fringe patterns with different phase shifts.....	78
Figure 5.7.Simulated parameters with background intensity and fringe amplitude nonuniformities. ....	79
Figure 5.8.Phase errors from fringe patterns with different nonuniform background intensities and fringe amplitudes.....	79
Figure 5.9.Simulated parameters with discontinues background intensity and fringe amplitude.80	80
Figure 5.10.Phase error map from fringe patterns with discontinues background intensity and fringe amplitude. ....	80
Figure 5.11.Middle column of phase error map of the uPSAs from fringe patterns with discontinues background intensity and fringe amplitude.....	81
Figure 5.12.Phase errors from fringe patterns with different amplitude of the phase-shift errors. .....	83
Figure 5.13.The fringe patterns with intensity harmonics. ....	83
Figure 5.14.Phase errors from fringe patterns with random phase shifts and different amplitude of the intensity harmonics.....	84
Figure 5.15.Phase errors from fringe patterns with regular phase shifts and different amplitude of the intensity harmonics. ....	84
Figure 5.16.Phase shifts with nonuniformities. ....	85

Figure 5.17. Phase errors from fringe patterns with different levels of phase-shift nonuniformities.	86
Figure 5.18. The fringe patterns with random noise.	86
Figure 5.19. Phase errors from fringe patterns with different levels of random noise.	87
Figure 5.20. Phase errors from fringe patterns with different frame numbers and random noise.	87
Figure 5.21. Phase errors from fringe patterns with phase-shift errors.	90
Figure 5.22. Phase errors from fringe patterns with gamma.	91
Figure 5.23. Phase errors from fringe patterns with phase-shift nonuniformities.	92
Figure 5.24. Phase errors from fringe patterns with random noise.	92
Figure 5.25. Simulated fringe patterns.	95
Figure 5.26. Phase errors from fringe patterns with intensity harmonics.	95
Figure 5.27. The first fringe patterns.	96
Figure 5.28. Sketch of the Fizeau interferometer [207].	96
Figure 5.29. Fringe patterns.	97
Figure 5.30. The extracted phases.	98
Figure 6.1. Flowchart of the AIA.	103
Figure 6.2. Accuracy performance of the AIA with fringe pattern cropping.	104
Figure 6.3. Mapping from CUDA kernel to GPU.	106
Figure 6.4. Flowchart of the gAIA.	107
Figure 6.5. Schematic model of CUDA's parallel reduction kernel.	109
Figure 6.6. The phase-shifted fringe patterns.	111
Figure 6.7. Accuracy performance of different implementations of the AIA.	112
Figure 7.1. Fringe pattern with noise.	117

# List of Table

Table 2.1.The selected kPSA examples .....	20
Table 4.1.Settings of different algorithms. ....	63
Table 4.2.Simulated fringe patterns. ....	65
Table 4.3.RMSEs from example with random bias phase shifts (Unit: rad). ....	66
Table 4.4.RMSEs from example with regular bias phase shifts (Unit: rad). ....	67
Table 4.5.RMSEs from the experiment (Unit: rad). ....	70
Table 5.1.The pre-requisites of the uPSAs on fringe parameters. ....	82
Table 5.2.Standard deviation value of the phase noise (Unit: rad). ....	97
Table 6.1.Mapping from the steps of the AIA to the parallel patterns .....	105
Table 6.2.Time consumption of the five implementations of the AIA (unit: ms). ....	113



# **Chapter 1 Introduction**

## **1.1.Optical metrology**

Metrology, as the science of measurement concerning both the theoretical and practical aspects, is critical in all industrial branches and enables faster and more accurate manufacturing with a higher standard and cheaper cost. Among different metrology techniques, the optical metrology [1-10] which concerns measurement with light is playing a significant role in various application fields including the semiconductor manufacturing [11-16], the mechanical engineering [17-20], the optical shop testing [21-25], the material engineering [26-31], the bio-engineer [32-37], etc. The interferometry based techniques [38-54], the time-of-flight based techniques [55-57] and the triangulation based techniques [58-61] are some typical examples of optical metrology.

The interferometry based techniques are the most accurate techniques in optical metrology. They are widely used in the measurements of the wavefronts [62-64], the shapes [65-68], the deformations [69-72] and the refractive indexes [73-76]. Besides, they were also used for defining the length of one meter [77] and for detecting gravitational waves [78]. Based on the mechanism of information extraction, the interferometry based techniques can be further divided into the laser interferometry [79-82], the phase-shifting interferometry [83-86], the digital holography [87-90], the shearography [91-94] and the white light interferometry [95-97]. In these techniques, the phenomenon of interference is used to extract the information. Two coherent light beams combine based on the superposition principle [98]. The combined beam has some meaningful properties relating to the measured quantity. The interferometry based techniques can provide nanometer level sensitivities in trade of the measurement range.

The time-of-flight based techniques are typically used to measure distance or 3D coordinate information for a large field of view. Based on the time lag between the emitted signal and the reflected signal, the distance from the testing object to the detector can be measured. Typically, lasers or LEDs are used as the light sources of the time-of-flight based techniques. Due to the advantage of a large measurement range, these techniques are widely used in the navigation of drones [99], robots [100] and automotive vehicles [101]. However, the accuracy and resolution are the limitations of the time-of-flight based techniques.

The triangulation based techniques have accuracy and a measurement range between the interferometry techniques and time-of-flight based techniques. The triangulation based techniques extract the shape or distance through correspondences between the generated structured patterns and the captured images. Based on the pre-calibrated 3D positions of the structured patterns emitter and the camera, the object's shape is recovered based on triangulation. Dot patterns [102], line patterns [103], and fringe patterns [104] are used in these techniques for face recognition [105], body shape measurement [106] and surface roughness analysis [107].

Based on the review above, the accuracy and measurement range of the different optical metrology techniques are summarized in Figure 1.1. Compared to the other techniques, the interferometry based techniques are with excellent accuracy, hence are widely used in different application fields.

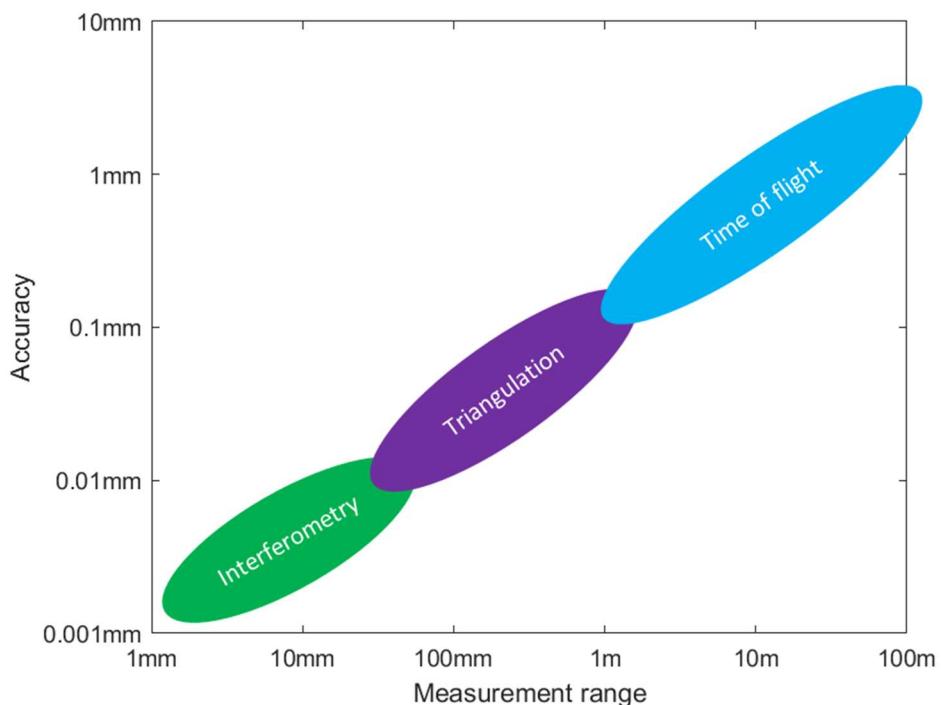


Figure 1.1. Accuracy and measurement range of various optical metrology techniques. Image was recreated based on the image in [10].

## 1.2.Laser interferometry

Laser interferometry has been intensely studied in the past century. The most famous configuration of the interferometry, the Michelson interferometer is shown in Figure 1.2. In the Michelson

interferometer, a beam splitter splits the source beam into the two arms. The reference surface and the testing surface reflect the beams in the reference arm and the test arm, respectively. When the testing beam meets the reference beam at the beam splitter, the amplitudes of the two beams are combined based on the superposition principle [98]. Since the two beams are coherent, a fringe pattern with bright and dark intervals appears. Subsequently, instead of directing back to the source, the resulting fringe pattern is directed to a detector which is typically a camera or a photoelectric detector.

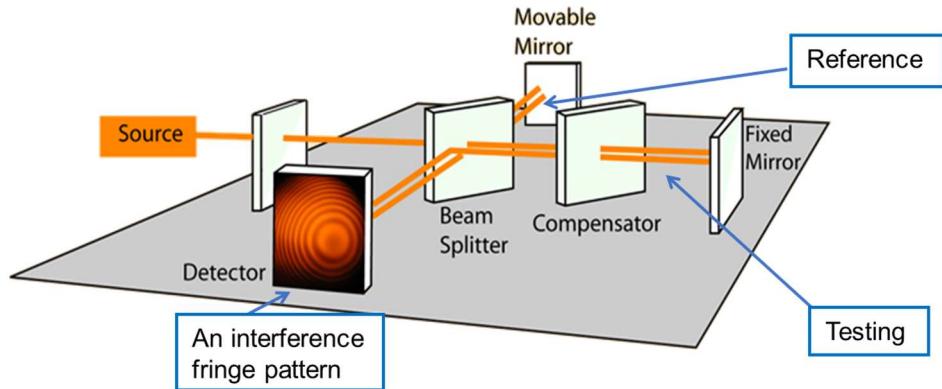


Figure 1.2.Typical Michelson interferometer [108].

It is interesting to realize that, by observing the fringe pattern, we can detect nanometer scale defects or shape errors of the specimen. The fringe pattern generated by interferometers can be represented as,

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

where  $x$  and  $y$  are the spatial coordinates with  $x=0, 1, \dots, N_x-1$  and  $y=0, 1, \dots, N_y-1$ ;  $N_x$  is the total number of rows;  $N_y$  is the total number of columns;  $A$  and  $B$  are the background intensity and fringe amplitude, respectively;  $I$  is the fringe pattern intensity; and  $\varphi$  is the phase information which is proportional to the measured quantity and should be extracted. It is worth mentioning that the background intensity and fringe amplitude of the fringe patterns are related to the amplitude of the light beams. As the amplitude of the light beams mostly related to the reflectivity of the reference and testing surfaces, the background intensity and fringe amplitude of the fringe patterns barely change with respect to time. For illustration, Figure 1.3 shows one fringe pattern simulated based on Eq. (1.1) with  $N_x=N_y=512$ , and a phase of

$$\varphi(x, y) = \frac{x^2 + y^2}{10000}. \quad (1.2)$$

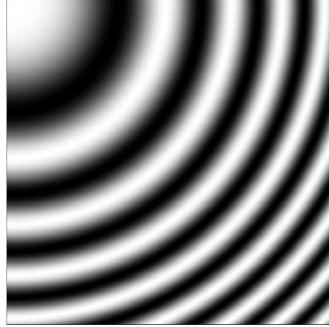


Figure 1.3. One simulated fringe pattern.

Meanwhile, it is worth mentioning that the interferometry is designed for the optical smooth surface whose surface height variation within the diffraction limited lateral resolution of the observation is smaller than a quarter of the wavelength [109].

### 1.2.1. Fringe skeletonizing technique

Many phase extraction techniques have been developed. The fringe skeletonizing technique is one of the early attempts of phase extraction. It locates the skeletons of the fringe pattern and interpolates the phase based on the extracted skeletons. This technique generally consists of four steps [110, 111]: (i) the fringe pattern filtering, (ii) the skeletons identification, (iii) the fringe order number assignment, and (iv) the phase result interpolation.

The fringe pattern filtering step uses spatial filters to reduce the noise in the fringe pattern [112]. Typically, a low-pass filter can remove the high frequency noise including the random noise and/or the speckle noise. A high-pass filter can remove low frequency noise such as nonuniform illuminations.

The skeletons form by lines with one pixel wide which are the local maximum of the fringe pattern. The skeletons can be identified by different methods including the adaptive digital filter [113], the least-squares fitting [114], the nonlinear regression [115], the fast Fourier transform [116], the zero crossing algorithm [117] or the Yatagai matrix[118].

The fringe order number assignment step segments the pixels based on the skeletons and assigns fringe numbers to these pixels. The fringe order number assignment is an important step in the fringe skeletonizing technique and its result will be used in the later interpolation step. A wrongly assigned fringe order number can lead to errors all over the phase map. When there is no close

fringe in the fringe pattern, it is easy to assign the fringe number [119]. When there are close fringe patterns, human intervention might be needed [120-122].

After the fringe order numbers have been assigned for all the pixels, interpolations between the skeletons are needed to generate the phase map. The phase map can be interpolated either globally or locally. The global approach is typically based on two-dimensional Zernike polynomials fitting [123, 124]. The local approach is with different methods including the spline interpolation [125], the bilinear interpolation [125] and the triangular interpolation [125].

The fringe skeletonizing technique has many issues including unsatisfactory accuracy and nonautomatic measurement. Hence, this technique is less used in the recent two decades.

### 1.2.2.Phase-shifting technique

Clearly, the fringe skeletonizing technique has its limitations. To make accurate and automatic phase extraction feasible, phase-shifting interferometry (PSI) was proposed. In the PSI, phase shifts are introduced to generate phase-shifted fringe patterns. As a result, the phase is encoded into the intensity variations of the phase-shifted fringe patterns, which enables phase extraction in a pixel by pixel manner. In the PSI, the phase-shifted fringe patterns can be represented as:

$$I(x, y; i) = A(x, y; i) + B(x, y; i) \cos[\varphi(x, y) + \delta(i)], \quad (1.3)$$

where  $i=0,1,\dots, F-1$  is the frame number index,  $F$  is the total frame number and  $\delta$  is the phase shifts. For illustration, phase-shifted fringe patterns simulated based on Eq. (1.3) with a phase simulated based on Eq. (1.2),  $F=4$  and  $\delta(i)=i\pi/2$  are shown in Figure 1.4.

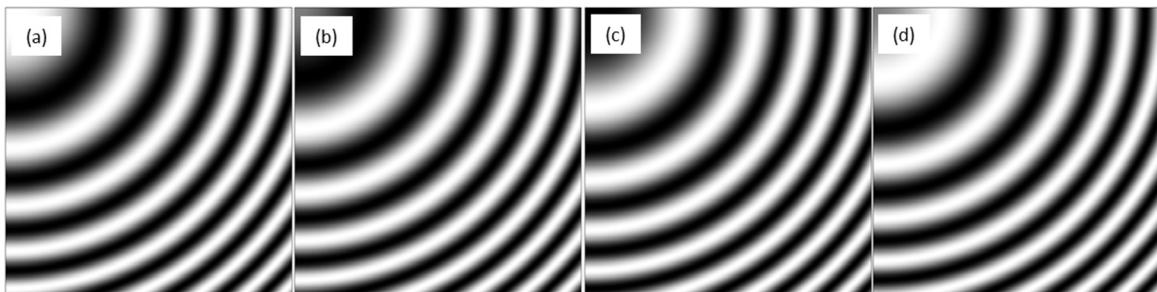


Figure 1.4.Fringe patterns with different phase shifts.

(a) The first; (b) The Second; (c) The third; (d) the fourth.

With the phase-shifted fringe patterns, the phase can be extracted by a phase-shifting algorithm (PSA). For illustration, a 4-step PSA [126] which can extract the phase from the fringe patterns in Figure 1.4 is used. When we know that  $F=4$  and  $\delta(i)=i\pi/2$ , it is not difficult to derive that

$$S(x, y) = [I(x, y; 3) - I(x, y; 1)] = 2B(x, y) \sin[\varphi(x, y)], \quad (1.4)$$

$$C(x, y) = [I(x, y; 0) - I(x, y; 2)] = 2B(x, y) \cos[\varphi(x, y)]. \quad (1.5)$$

Based on trigonometric identities, the phase can be calculated as

$$\varphi(x, y) = \text{atan2}[S(x, y), C(x, y)], \quad (1.6)$$

where  $\text{atan2}$  is the arctangent function which can be found in the popular programming software. It is worth mentioning that the phase extracted is the principal value of the arctangent function which is wrapped within  $(-\pi, \pi]$ . Figure 1.5 shows the phase extracted based on the 4-step PSA from the fringe patterns in Figure 1.4.

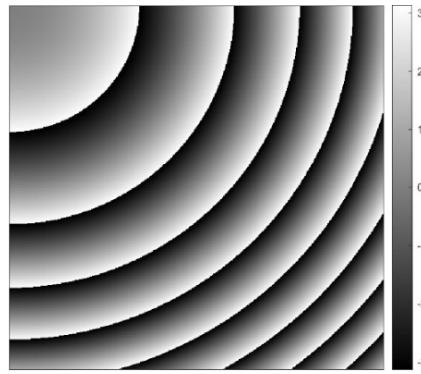


Figure 1.5. Wrapped phase.

### 1.2.3. Fourier transform technique

A special successful technique for phase extraction is the Fourier transform technique. In this technique, one frame of fringe pattern with spatial carriers is used and the phase is extracted by Fourier transform [127]. With the introduced spatial carriers, the fringe pattern can be represented as

$$I(x, y) = A(x, y) + B(x, y) \cos[\varphi(x, y) + \omega_{cx}x + \omega_{cy}y], \quad (1.7)$$

where  $\omega_{cx}$  and  $\omega_{cy}$  are the spatial carrier frequencies in row and column, respectively. Figure 1.6 shows one frame of fringe pattern with spatial carriers simulated based on Eq. with  $N_x=N_y=512$ ,  $\omega_{cx}=0.4$ ,  $\omega_{cy}=0$  and a phase simulated according to Eq. (1.2).

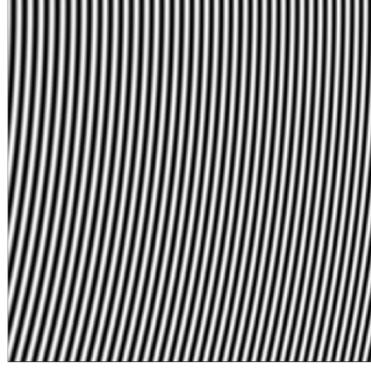


Figure 1.6. One simulated fringe pattern with a spatial carrier.

Based on Euler's formula, Eq. (1.7) can be rewritten as

$$I(x, y) = A(x, y) + C(x, y) + C^*(x, y), \quad (1.8)$$

with

$$C(x, y) = \frac{1}{2} B(x, y) \exp\left\{j[\varphi(x, y) + \omega_{cx}x + \omega_{cy}y]\right\}, \quad (1.9)$$

where the superscript \* represents a complex conjugate and  $j$  is the imaginary unit. To get the phase information, we need to separate  $C(x, y)$  from  $C^*(x, y)$  and  $A(x, y)$ . The Fourier transform is perfect for this task. The 2D Fourier transform of Eq. (1.8) can be calculated as

$$\begin{aligned} F[I(\xi_x, \xi_y)] &= F[A(\xi_x, \xi_y)] + F[C(\xi_x - \omega_{cx}, \xi_y - \omega_{cy})] \\ &\quad + F[C^*(\xi_x + \omega_{cx}, \xi_y + \omega_{cy})], \end{aligned} \quad (1.10)$$

where  $\xi_x$  and  $\xi_y$  are the frequency coordinates in row and column, respectively. The three components in the righthand side of Eq. (1.10) are concentrated around  $(0, 0)$ ,  $(\omega_{cx}, \omega_{cy})$  and  $(-\omega_{cx}, -\omega_{cy})$ , respectively. When the  $\omega_{cx}$  and/or  $\omega_{cy}$  are large enough, the three components are with no overlaps. By using a proper bandpass filter, the spectrum of  $F[C(\xi_x, \xi_y)]$  can be extracted. The inverse Fourier transform will help us recover  $C(x, y)$ . Subsequently, the phase is extracted as

$$\varphi(x, y) = \text{atan2}\{\text{Im}[C(x, y)], \text{Re}[C(x, y)]\}, \quad (1.11)$$

where  $\text{Re}(\cdot)$  and  $\text{Im}(\cdot)$  output the real part and the imaginary part of a complex number, respectively. The  $\tan^{-1}$  function returns again the principal value. For illustration, Figure 1.7 shows the wrapped phase extracted based on the Fourier transform technique from the fringe pattern in Figure 1.6.

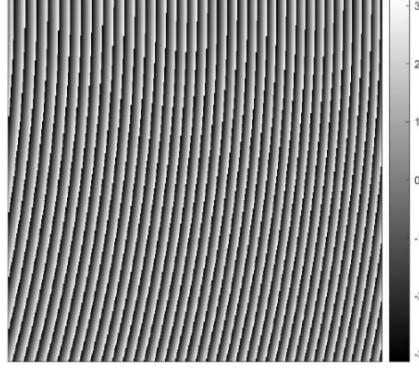


Figure 1.7.The wrapped phase map.

#### 1.2.4. Phase unwrapping

The phases extracted in the phase-shifting technique and Fourier transform technique are all the principal values of the arctangent function, thus  $2\pi$  phase jumps will be noticed in the phase maps. Hence, phase unwrapping is needed to flatten the phase map. The phase is generally unwrapped based on the following relationship [128, 129]

$$\phi(x, y) = \varphi(x, y) + 2k_\varphi(x, y)\pi, \quad (1.12)$$

where  $\phi$  represents the unwrapped phase and  $k_\varphi(x, y) \in \mathbf{Z}$ .

When the unwrapped phase at  $(x-1, y-1)$ ,  $\phi(x-1, y-1)$ , is obtained, the neighboring pixel's  $\phi(x, y)$  is extracted with the assumption that the phase is spatially continuous as

$$-\pi < \phi(x, y) - \phi(x-1, y-1) < \pi. \quad (1.13)$$

By substituting Eq. (1.12) into Eq. (1.13), we get

$$-\pi < \varphi(x, y) + 2k_\varphi(x, y)\pi - \varphi(x-1, y-1) < \pi, \quad (1.14)$$

which can be rearranged as

$$-0.5 + \frac{\phi(x-1, y-1) - \varphi(x, y)}{2\pi} < k_\varphi(x, y) < 0.5 + \frac{\phi(x-1, y-1) - \varphi(x, y)}{2\pi}. \quad (1.15)$$

Subsequently, the unwrapped phase  $\phi(x, y)$  can be extracted as

$$\Phi(x, y) = \varphi(x, y) + 2\pi \times \text{round}\left[\frac{\phi(x-1, y-1) - \varphi(x, y)}{2\pi}\right], \quad (1.16)$$

where  $\text{round}(\cdot)$  returns the input's nearest integer. Figure 1.8 shows the unwrapped phase map calculated based on the above theory from the phase map in Figure 1.7.

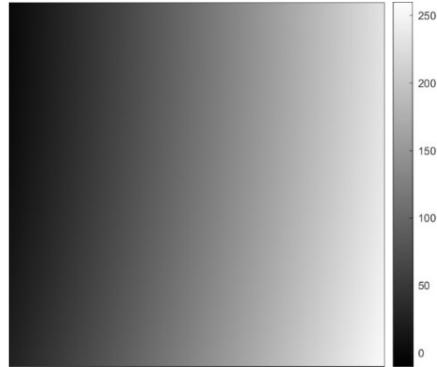


Figure 1.8.Unwrapped phase map.

Although the theory of phase unwrapping is simple, it is still challenging in practice due to error propagation. If a certain pixel's unwrapped phase is estimated wrongly, the unwrapped phases of the subsequent pixels are likely to be wrong. Hence, many phase unwrapping algorithms were proposed including the quality guided algorithms [130-133], the cut based algorithms [134, 135], the partitioning algorithm [136] and the cellular automata algorithm [137].

### **1.3.Possible error sources and more phase-shifting algorithms**

#### **1.3.1.Possible error sources in the fringe patterns**

Clearly, the phase-shifting technique has its advantages, thus, will be focused on in this thesis. Meanwhile, the phase-shifting technique also faced the challenge of different error sources [138]. Some error sources will lead to a systematic error in the phase which is a consistent or proportional difference between the extracted phase and ground true phase, including

- (i) Phase-shift errors: In most scenarios, the phase extraction relies on one assumption that the phase shifts are introduced in a known manner. If the phase shifts are not introduced as designed, the measured intensities will not precisely map to the designed sinusoidal signal. Thus, errors in fringe patterns will occur. The phase-shift errors might be induced by incorrect calibration, the nonlinearity of the phase-shifter or a combination thereof [138].
- (ii) Intensity harmonics: The intensity harmonics make the captured phase-shifted fringe patterns distorted. The intensity harmonics are normally induced by the detector's nonlinearity [139] or multi-beam reflections [140, 141]. Detector nonlinearity induced intensity harmonics are more significant if the fringe patterns' intensities are high, especially when at or near saturation conditions [138]. Multi-beam reflections induced

intensity harmonics are more significant when measuring transparent elements with parallel surfaces [141]. For better illustration, intensity harmonics are added a line of the fringe pattern simulated based on fringe pattern simulated based on Eq. (1.1) with  $N_x=512$ , and a phase of  $\phi(x)=4\pi x/512$  by using a gamma model, *i.e.*  $I'$  represents detector's nonlinearity [139] with  $\gamma=1.5$ . The noiseless fringe pattern and fringe pattern with intensity harmonics are shown in Figure 1.9. Clearly, the intensity harmonics will distort the fringe pattern and lead to non-perfect sine wave.

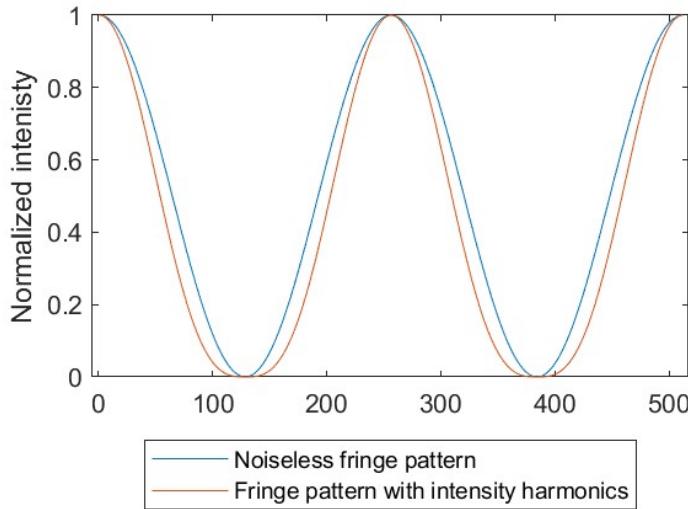


Figure 1.9.Fringe pattern with intensity harmonics.

(iii)Nonuniform phase-shift distribution: The fringe pattern model in Eq. (1.3) assumes a spatial independent phase shift for each frame. This assumption will be violated in measurements with vibration [142] and/or air turbulence [143]. Both the vibration and air turbulence may change the optical path difference and induce the spatial dependency of the phase shifts.

While another error source will lead to a random error in the phase which is a change difference between the extracted phase and ground true phase, which is

(iv)Random noise: The random noise in the fringe patterns is mostly additive white gaussian noise which is an independently distributed random variable drawn from a normal distribution and will occur in each pixel of each frame. The random noise usually comes from the readout noise of CMOS and CCD due to the analog to digital converter [144].

All these error sources will impact the extracted phase's accuracy. Thus, suppressing the influences of these error sources is very important for measurements with PSI.

### 1.3.2.More phase-shifting algorithms

For accurate phase extraction from error contaminated phase-shifted fringe patterns, many phase-shifting algorithms (PSAs) have been developed [145-147]. These PSAs can be divided into two categories: (i) those with known phase shifts and (ii) those with unknown phase shifts.

The majority of PSAs adopt known phase shifts, extract the phase based on trigonometric identities [138, 145-148], and are abbreviated as kPSAs for convenience. They are simple and widely used. The 3-step algorithm [126], the 4-step algorithm [126], the 5-step algorithm [149], the digital Fourier transform algorithms [126], the least-squares algorithms [150, 151], the averaging algorithms [149, 152, 153], the (N+1)-bucket algorithms [154, 155], the windowed digital Fourier transform algorithms [156] and the general error-compensation algorithms [157] are such examples.

The other PSAs, which are abbreviated as uPSAs, adopt unknown phase shifts and thus are naturally immune to phase-shift errors. They could be further classified into three sub-categories.

The first sub-category uses trigonometric identities for derivation, with the help of a regular phase shifts assumption, *e.g.*, Carré's algorithm [158] and the Stoilov's algorithm [159], or the statistical properties of fringe patterns, *e.g.*, the Lissajous figure method [160, 161], the phase statistics method [162] and the Euclidean matrix norm method [163].

The second sub-category decomposes the fringe patterns into different components through linear algebra manipulations, such as the principal component analysis [164], the Gram–Schmidt orthonormalization [165], the hyper-accurate ellipse fitting in subspace [166] and the VU factorization [167].

The third sub-category, which is with the greatest number of uPSAs, adopts an optimization approach. Okada *et al.* proposed their algorithm using an alternative linear least-squares fitting framework to iteratively estimate the phase and the phase shifts [168]. The advanced iterative algorithm [169] improved Okada *et al.*'s algorithm to make the phase extraction simple, robust and accurate. Xu *et al.* and Hoang *et al.* focused on phase extractions from fringe patterns with intensity harmonics by using linear least-squares fitting and exhaustive search, respectively [170,

171]. Duan *et al.* and Chen *et al.* focused on phase extractions from fringe patterns with nonuniform phase shifts by using linear least-squares fitting and nonlinear least-squares fitting, respectively [172, 173]. Deck proposed a model based phase-shifting interferometry [174] to minimize the impact from both the intensity harmonics and the phase-shift nonuniformities.

With a large number of PSAs available, examining their performance becomes critical for proper algorithm selection and performance boosting. Many works focusing on accuracy, robustness and speed have been carried out.

There are three main approaches to evaluating the kPSAs' accuracy. In the first approach, the performance of some selected kPSAs with respect to various error sources is directly compared through simulations, including Creath's early study [148] and Buytaert *et al.*'s comprehensive comparison of 84 algorithms [145], among others [138, 145-148]. In the second approach, Freischlad and Koliopoulos proposed the Fourier description method to evaluate the immunity of kPSAs to the phase-shift errors, the intensity harmonics and the random noise [175]. The Fourier description method was further generalized by the Frequency Transfer Function method [176, 177]. In the third approach, the kPSA's analysis and design are seamlessly combined through Larkin *et al.*'s [155] and de Groot's [178] frequency domain evaluation, Surrel's characteristic polynomials [156, 179] and Hibino *et al.*'s equation solving [157, 180-182], making the systematic algorithm design of error-compensation algorithms possible.

Besides, the robustness and the speed of the kPSAs were also focused on. The robustness of the kPSAs has been evaluated in different works [110, 183, 184], in which they confirmed that the performance of the kPSAs is not affected by the background intensity, the fringe amplitude and the phase distributions. Besides, the fast speed of the kPSAs is confirmed in [185]. Meanwhile, the GPU acceleration of the kPSAs has been considered in [186].

As for the uPSA, Kemao *et al.*'s evaluation of the accuracy of Carré's algorithm [187]; Vargas *et al.*'s theoretical evaluation of PCA's restrictions [188]; and Zhu *et al.*'s hybrid GPU-CPU implementation of the AIA [189] can be seen as examples of the evaluation and development works of the uPSAs' accuracy, robustness and speed, respectively. However, such works are much fewer than those for the kPSA.

## **1.4.Motivation, contributions, and thesis organization**

As mentioned, many PSAs with both known and unknown phase shifts have been proposed and lots of studies have been carried out for the evaluation and enhancement of the PSAs. However, these studies are mainly for the kPSAs and only a few are for the uPSAs which makes the overall picture of the PSAs unclear.

### **1.4.1.Motivations**

Thus, we are motivated to conduct a systematic evaluation and development study focusing on the uPSAs. Specifically, we are with the following motivations:

- (i) The advanced iterative algorithm (AIA) is the most classic and well-recognized uPSA. Even with a simple algorithm structure, the AIA could accurately extract phase from fringe patterns with arbitrary frame number ( $\geq 3$ ) and random phase shifts. Hence, there are many applications used the AIA for phase extraction. However, as an iterative algorithm, robustness is a major concern for the AIA. Meanwhile, the accuracy performance of the AIA is also not clear. These leave uncertainties in the utilization of the AIA.
- (ii) Many uPSAs have been proposed to minimize the impact of different common error sources. However, none of these algorithms is immune to all the error sources. The accuracy of the uPSA is not optimized, especially when all the common error sources are presented. Thus, we are motivated to propose a simple, accurate and robust uPSA which is immune to all the mentioned error sources.
- (iii) Except for the AIA, the robustness and the accuracy of other uPSAs are also not clear. Meanwhile, the uPSAs are never compared with the kPSAs. These leave an unclear full picture of the PSAs. More specifically, the following aspects are not clear and need to be answered: (i) Are there any restrictions on the fringe patterns when we use the uPSAs; (ii) Which PSA is with the best performance when the fringe patterns are with errors and random/regular phase shifts; and (iii) Is it possible to further improve the PSAs by combining the merits of the kPSAs and the uPSAs?
- (iv) Compared to the kPSAs, the optimization based uPSAs which iteratively estimate the phase and phase shifts, have much higher computational costs. For example, we test the AIA with

13 frames of fringe patterns. For each fringe pattern, the image size is 2048×2048. The AIA needs 3.13 minutes to extract the phase by using a MATLAB implementation. The time cost could reduce to 19.6 seconds if a C++ implementation is used, but the computation cost is still high. Hence, faster AIA is needed. Currently, there is only one acceleration work for the AIA though using a hybrid GPU-CPU implementation [189], which parallels and accelerates the phase calculation by a GPU, but the phase shifts calculation remains on the CPU and is accelerated by processing only the center 2% fringe patterns. However, cropping the fringe patterns may lead to divergence of the AIA. Meanwhile, the hybrid GPU-CPU implementation sacrifices the speed of phase shifts estimation which should also be accelerated by a GPU.

#### 1.4.2. Contributions

To address the mentioned motivation, we conducted four different works respectively. The contributions of this thesis are summarized below.

- (i) The major difficulty in the evaluation of the AIA is how to characterize the phase and phase shifts. We proposed to use the condition numbers of the least-squares matrices in the AIA for the characterization of the phase and phase shifts. We carried out large amounts of simulations and condition numbers based on theoretical elevations to confirm the AIA's robustness. Subsequently, a fringe pattern requirement was confirmed. To evaluate the AIA's accuracy, simulations were carried out to investigate the influence of the phase shifts, the frame numbers and the noise levels. Besides, a phase error model was established for the AIA and three individual improving practices, *i.e.*, controlling the phase shifts, controlling the frame number and suppressing the noise, were proposed. Moreover, an enhanced AIA (eAIA) was proposed for the enhancement of the AIA's robustness and accuracy.
- (ii) A general iterative algorithm (GIA) was proposed to suppress the impact from all the mentioned common error sources. The GIA incorporates all the concerned error sources into the fringe model and estimates them together with the phase. Based on their different natures, the unknowns in the GIA are classified into three groups: (i) the background intensity and the fringe amplitude, (ii) the phase and (iii) the phase-shift related parameters. Subsequently, these unknowns are optimized alternatively by the Levenberg-Marquart

method, which has demonstrated excellent convergence and accuracy. In addition, the background intensity and the fringe amplitude are separately estimated within a small window for better immunity to random noise.

- (iii) A comprehensive evaluation of the uPSAs' robustness and accuracy is carried out. In the evaluation, we selected the AIA [169] and the algorithm based on principal component analysis (PCA) [164] due to their fundamental ideas, and the GIA and the algorithm based on VU factorization [167] due to their outstanding accuracy. We confirmed the requirements when using these uPSAs. Besides, we compared the accuracy of the uPSAs among themselves and also with twelve benchmarking kPSAs to showcase their advantages and disadvantages. Meanwhile, we proposed a hybrid kPSA-GIA which uses the kPSAs to initialize the GIA and uses the GIA to extract the phase. The hybrid kPSA-GIA both improves the accuracy of the kPSAs, and reliefs the GIA's restriction on fringe density. Besides, the hybrid kPSA-GIA can reconstruct the fringe patterns. The reconstructed fringe patterns can be compared with the input fringe patterns for performance evaluation.
- (iv) With a detailed evaluation of the AIA's structure, we proposed a fully GPU-accelerated AIA (gAIA) which is with similar accuracy performance compared to a CPU implementation of the AIA and reaches  $500\times$  speedup. In the gAIA, we used a pointwise pattern for the phase extraction and used a reduction pattern for the phase shifts estimation to break the bottlenecks of the AIA. Besides, we also evaluated the possible convergence issue of fringe pattern cropping which has been used in the hybrid GPU-CPU implementation of the AIA. Moreover, compared to the hybrid GPU-CPU implementation, the gAIA is  $10\times$  faster.

#### 1.4.3.Organization

There are seven chapters in this thesis. Chapter 1 introduces the background of the PSAs and summarizes the motivations and contributions of this thesis. Chapter 2 reviews the PSAs, including both the kPSAs and the uPSAs. Chapter 3 investigates the AIA's performance thoroughly and integrates the findings of AIA into an enhanced AIA. Chapter 4 proposes the GIA which is immune to all the mentioned common error sources. Chapter 5 shows the detailed accuracy and robustness evaluation of the four well performing uPSAs, compares the performance of the uPSAs with the kPSAs and proposes a hybrid kPSA-GIA which combines the advantages of the kPSAs and the

GIA. Chapter 6 proposes a fast fully GPU-accelerated AIA (gAIA) and studies the performance of the gAIA in detail. Chapter 7 provides some future works and gives the conclusion.

# Chapter 2 Literature review

## 2.1.The phase-shifting algorithm with known phase shifts (kPSA)

### 2.1.1.Principle of the kPSAs

The fringe pattern model used in kPSAs follows Eq. (1.3) with an assumption on the phase shifts as

$$\delta(i) = (i - \Gamma)\omega, \quad (2.1)$$

where  $\omega$  is the phase-shift increment, *i.e.*, the phase-shift difference between two consecutive frames, and  $\Gamma$  is the parameter which controls the phase origin, *i.e.*, there is no shift of phase at the  $\Gamma$ th frame. We call this special frame the frame of phase origin [147]. Most kPSAs use the first frame (*i.e.*,  $\Gamma=0$ ) or the middle frame (*i.e.*,  $\Gamma=(F-1)/2$ ) as the frame of phase origin.

Based on trigonometric identities and with an assumption of known phase shifts, the phase is extracted by a weighted combination of phase-shifted fringe patterns as [147]

$$\varphi(x, y) = \text{atan2}\left[S(x, y), C(x, y)\right] = \text{atan2}\left[\sum_{i=0}^{F-1} s(i) I(x, y; i), \sum_{i=0}^{F-1} c(i) I(x, y; i)\right], \quad (2.2)$$

where  $S$  and  $C$  are the arctangent function's numerator and denominator, respectively; and  $s$  and  $c$  are the weights for the combinations. Over decades, many kPSAs have been proposed. Since the kPSAs have been well developed and evaluated, only the following typical ones are selected to show in Table 2.1 [126, 149, 152, 154, 156, 157, 180, 190, 191]:

- (i) The classic kPSAs with small step numbers ( $F=3\sim 5$ ) which are convenient to use. We select the 3-step algorithm by Bruning et al. [126], the 4-step algorithm by Bruning et al. [126] and the 5-step algorithm by Schwider et al. [149]. These kPSAs are listed as the first three algorithms in Table 2.1.
- (ii) The popular families of the kPSAs specified with a median step number of  $F=7$ . The noticeable families included the digital Fourier transform algorithms (DFT) [126], the averaging algorithms (AVE) [152], the  $(N+1)$ -bucket algorithms  $(N+1)$  [154], the windowed digital Fourier transform algorithms (WDFT) [156] and the general error-compensation algorithms (GEC) [157]. These kPSAs are listed as the next five algorithms in Table 2.1.

(iii) The kPSAs with good accuracy and large step numbers ( $F=9\sim15$ ). We select the 9-step algorithm by Hibino et al [157] which is with good immunity to phase-shift nonuniformities, the 10-step algorithm by Surrel [156] and the 11-step algorithm by Hibino et al. [180, 190] (with a minus sign as indicated by [156]) which are both with good immunity to up to the 4th order intensity harmonics with the presence of linear phase-shift errors, and the 15-step algorithm by De Groot et al. [191] which is with good immunity to even order intensity harmonics and random noise. These kPSAs are listed as the last four algorithms in Table 2.1.

### 2.1.2. Evaluation and issues of the kPSAs

As discussed in chapter 1.3.2, the kPSAs can robustly extract the phase from fringe patterns with different phases, different background intensities and different fringe amplitudes. Besides, the fast speed of the kPSAs enables their wide adoption in industrial applications. Meanwhile, many accuracy evaluation works have been carried out for the kPSAs including the simulation evaluations carried out by Creath [148] and Buytaert *et al.* [145], the Fourier description based theoretical evaluation by Freischlad and Koliopoulos [175], the Frequency Transfer Function based theoretical evaluation by Servin *et al.* [176, 177], the characteristic polynomial based theoretical evaluation by Surrel [156, 179] and the equation solving based theoretical evaluation by Hibino *et al.* [157, 180-182].

The characteristic polynomial method is discussed as an example. In this method, the kPSAs expressed by Eq. (2.2) are rewritten as

$$\begin{aligned}\varphi(x, y) &= \arg \left\{ \sum_{i=0}^{F-1} [s(i)\mathbf{j} + c(i)] I(x, y; i) \right\}, \\ &= \arg \left\{ \sum_{i=0}^{F-1} \gamma(i) I(x, y; i) \right\},\end{aligned}\tag{2.3}$$

where  $\arg$  calculates the angle of a complex number.

Meanwhile, by treating the fringe patterns as a periodic function, we can express the  $I(x, y; i)$  by a Fourier series as

$$I(x, y; i) = \sum_{k=-\infty}^{\infty} \varsigma_k \exp[jk\varphi(x, y)] \exp[jk\delta(i)],\tag{2.4}$$

where  $\varsigma_k$  is the coefficient of the Fourier series. Since the phase shifts follow Eq. (2.1), we can get

$$\begin{aligned} I(x, y; i) &= \sum_{k=-\infty}^{\infty} \varsigma_k \exp[jk\varphi(x, y)] \exp[jk(i-\Gamma)\omega], \\ &= \sum_{k=-\infty}^{\infty} \varsigma_k \exp[jk\varphi(x, y) - jk\Gamma\omega] \exp(jik\omega). \end{aligned} \quad (2.5)$$

By substituting Eq. (2.5) into Eq. (2.3), we get

$$\varphi(x, y) = \arg \left\{ \sum_{k=-\infty}^{\infty} \varsigma_k \exp[jk\varphi(x, y) - jk\Gamma\omega] P(z) \right\}, \quad (2.6)$$

where

$$P(z) = \sum_{i=0}^{F-1} Y(i) z^i, \quad (2.7)$$

with

$$z = \exp(j\omega)^k. \quad (2.8)$$

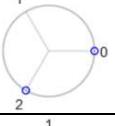
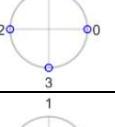
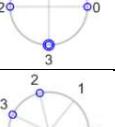
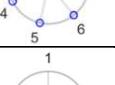
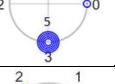
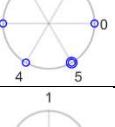
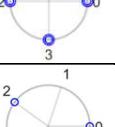
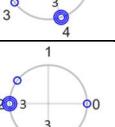
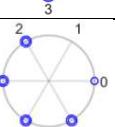
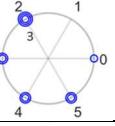
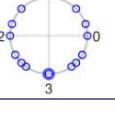
The complex function  $P(z)$  is known as the characteristic polynomial. The roots of the equation  $P(z)=0$  showcase the properties of a kPSA. Some of the conclusions include:

- (i) A kPSA is insensitive to the  $m$ th order intensity harmonic if there are roots at both  $z=\pm m$ .
- (ii) A kPSA is insensitive to the  $n$ th order phase-shift errors if there are  $(n+1)$  roots at  $z=-1$ .
- (iii) A kPSA is insensitive to both the  $m$ th order intensity harmonic and the  $n$ th order phase-shift errors if there are  $(n+1)$  roots at both  $z=\pm m$ .

Besides, these roots could be represented geographically. Hence, the characteristic polynomials' roots of the kPSAs in Table 2.1 are also shown in the last column of Table 2.1.

It is worth mentioning that, within the kPSAs, many evaluation and performance comparison works have been carried out. However, few works focused on the comparison between the kPSAs and the uPSAs which will be comprehensively discussed in chapter 5.

Table 2.1.The selected kPSA examples

Name	F	$\omega$ (rad)	$\Gamma$	Parameter $s$ and $c$	CP plot *
3-step [126]	3	$2\pi/3$	1	$\frac{\sqrt{3} \times [1,0,-1]}{-1,2,-1}$	
4-step [126]	4	$\pi/2$	0	$\frac{0,-1,0,1}{1,0,-1,0}$	
5-step [149]	5	$\pi/2$	0	$\frac{0,-2,0,2,0}{1,0,-2,0,1}$	
DFT [126]	7	$2\pi/7$	0	$\frac{\sin(i \times \frac{2\pi}{7})(i=0-6)}{\cos(i \times \frac{2\pi}{7})(i=0-6)}$	
AVE [152]	7	$\pi/2$	0	$\frac{1,-6,-5,20,-5,-6,1}{1,4,-15,0,15,-4,-1}$	
N+1 [154]	7	$\pi/3$	0	$\frac{1,-3,-3,0,3,3,-1}{\sqrt{3} \times [1,1,-1,-2,-1,1,1]}$	
WDFT [156]	7	$\pi/2$	3	$\frac{-1,0,3,0,-3,0,1}{0,-2,0,4,0,-2,0}$	
GEC [157]	7	$2\pi/5$	3	$\frac{-0.263,0.325,0.951,0,-0.951,-0.325,0.263}{-0.0854,-0.724,0.309,1,0.309,-0.724,-0.0854}$	
9-step [157]	9	$\pi/2$	4	$\frac{-1,2,14,18,0,-18,-14,-2,1}{-2,-8,-8,8,20,8,-8,-8,-2}$	
10-step [156]	10	$\pi/3$	0	$\frac{\sqrt{3} \times [-1,-3,-3,1,6,6,1,-3,-3,-1]}{1,-1,-7,-11,-6,6,11,7,1,-1}$	
11-step [180, 190]	11	$\pi/3$	5	$\frac{-\sqrt{3} \times [0,-1,-4,-7,-6,0,6,7,4,1,0]}{-2,-5,-6,-1,8,12,8,-1,-6,-5,-2}$	
15-step [191]	15	$\pi/2$	7	$\frac{-1,0,9,0,-21,0,29,0,-29,0,21,0,-9,0,1}{0,-4,0,15,0,-26,0,30,0,-26,0,15,0,-4,0}$	

\* Numbers inside the unit circle indicate the number of roots for clarity.

## 2.2. Phase-shifting algorithms with unknown phase shifts (uPSAs)

Other than the kPSAs, the uPSAs are also playing important roles in phase-shifting techniques. In this thesis, we introduce a few classic or well performing uPSAs. As discussed in chapter 1.3.2, these uPSAs could be divided into three sub-categories, (i) the trigonometric identities based uPSAs (ii) the decomposition based uPSAs and (iii) the optimization based uPSAs.

### 2.2.1. Trigonometric identities based uPSAs

In this sub-chapter, the trigonometric identities based uPSA with the longest history, Carré algorithm is discussed.

#### 2.2.1.1. Principle of Carré algorithm [158]

The fringe pattern model of Carré's algorithm is following Eq. (1.3) with  $F=4$  and regular phase shifts as

$$\delta(i) = (i - 1.5)\omega. \quad (2.9)$$

Based on triangular identities, and assuming that  $\omega \in [0, \pi]$ , we can derive [192]

$$\varphi(x, y) = \text{atan}2 \left\{ \begin{array}{l} \left\{ 3[I(x, y; 2) - I(x, y; 3)] - [I(x, y; 1) - I(x, y; 4)] \right\} \\ \sqrt{[I(x, y; 2) - I(x, y; 3)] + [I(x, y; 1) - I(x, y; 4)]} \\ , [I(x, y; 2) + I(x, y; 3)] - [I(x, y; 1) + I(x, y; 4)] \end{array} \right\}. \quad (2.10)$$

Noting that, since the phase shifts are unknown, Carré's algorithm is a uPSA. However, Carré's algorithm is always treated as one of the kPSAs for convenience due to the following reasons: (i) Comparing Eq. (2.10) and Eq. (2.2) shows that the form of Carré's algorithm is similar to the other kPSAs; (ii) Furthermore, although the phase-shift increment  $\omega$  is unknown, it remains constant; and (iii) Finally, Carré's algorithm is a 4-step algorithm, unlike other uPSAs which can be applied to any sufficient numbers of frames.

#### 2.2.1.2. Evaluation and issues of the trigonometric identities based uPSAs

The accuracy performance of Carré's algorithm with respect to different error sources was evaluated by Kemao *et al.* quantitatively [187] through linear approximation of the phase function. In the evaluation, the best phase increment of  $\omega=1.92$  rads ( $110^\circ$ ) is confirmed [187]. Meanwhile,

$\omega$  is always set as  $\pi/2$  rad in practice for simplicity. Carré's algorithm was well evaluated, but its performance was never compared with other uPSAs which will be conducted in chapter 5.

### 2.2.2. Decomposition based uPSAs

The decomposition based uPSAs decompose the fringe patterns into sine and cosine components and extract the phase Subsequently. We select the uPSA based on the principal component analysis (PCA) due to its fundamental idea [164] and the uPSA based on VU factorization (VU) due to its outstanding accuracy [167] for discussion in this sub-chapter.

#### 2.2.2.1. Principle of the PCA [164]

The PCA uses a fringe pattern model as

$$I(i; j) = A(i; j) + B(i; j) \cos[\varphi(j) + \delta(i)]. \quad (2.11)$$

where  $j=0, 1, \dots, N$  is the pixel index which is connected with the pixel's location  $(x, y)$  through  $j_{x,y} = xN_y + y$ . The PCA first removes the background intensity by subtracting the temporal average intensity, *i.e.*,  $A(j) \approx \sum_{i=0}^{F-1} I(i; j)/F$ . Furthermore, if the following approximation is well satisfied,

$$\sum_{j=0}^{N-1} B \cos[\varphi(j)] B \sin[\varphi(j)] \approx 0, \quad (2.12)$$

then, viewing from the entire image,  $B\sin(\varphi)$  and  $B\cos(\varphi)$  are orthogonal to each other and can be considered as two most significant principal components of the background-intensity-removed fringe patterns [164]. Thus, the task of phase extraction becomes the estimation of these two principal components, which can be easily achieved by the principal component analysis.

To do so, the background-intensity-removed fringe pattern intensities of the  $i$ th frame are taken column-wisely to form a row vector with a length of  $N$ . Next, the  $F$  row vectors from the  $F$  fringe patterns are stacked to form an  $F \times N$  matrix  $\mathbf{D}$  whose entries are  $D_{ij} = I(i; j) - A(j)$ . Subsequently, the following covariance matrix  $\mathbf{C} = \mathbf{D} \times \mathbf{D}^T$  where the superscript  $T$  represents matrix transpose is calculated. The size of  $\mathbf{C}$  is  $F \times F$  and the  $\mathbf{C}$  can be diagonalized through a singular value decomposition as

$$\mathbf{\Pi} = \mathbf{A} \mathbf{C} \mathbf{A}^T, \quad (2.13)$$

where  $\mathbf{\Pi}$  is a diagonal matrix whose size is  $F$  by  $F$ , and  $\mathbf{A}$  is the transformation matrix whose size is  $F$  by  $F$ . Thus, based on the Hotelling transform, the principal components of  $\mathbf{D}$  can be extracted as

$$\mathbf{N} = \mathbf{AD}, \quad (2.14)$$

where each column of  $\mathbf{N}$  is a principal component of  $\mathbf{D}$  whose size is  $N$ . The two principal components with the biggest eigenvalues,  $\mathbf{N}_{1,:}$  and  $\mathbf{N}_{2,:}$  which are also the first the second column of the matrix  $\mathbf{N}$ , are corresponding to the sine and cosine terms of phase. Hence, the phase could be obtained as

$$\varphi = \text{atan2}\left(\mathbf{N}_{1,:}, \mathbf{N}_{2,:}\right). \quad (2.15)$$

### 2.2.2.2.Principle of the VU [167]

The fringe pattern model used in the VU is similar to the ones of the PCA. Meanwhile, the VU did not remove the background intensity. As such, the VU rewrites the fringe pattern model with reshaped matrices as follows,

$$\mathbf{I} = \mathbf{VU}^T = (\mathbf{a} \quad \mathbf{c} \quad -\mathbf{s}) \begin{pmatrix} \mathbf{1}^T \\ \mathbf{u}^T \\ \mathbf{v}^T \end{pmatrix}, \quad (2.16)$$

where the matrix  $\mathbf{V}$  has a dimension of  $N \times 3$ ; the three vectors  $\mathbf{a}$ ,  $\mathbf{c}$  and  $\mathbf{s}$  have respective elements  $a_j = A(j)$ ,  $c_j = B(j)\cos[\varphi(j)]$  and  $s_j = B(j)\sin[\varphi(j)]$ , where  $j$  has the same definition as the one in the PCA; the matrix  $\mathbf{U}$  has a dimension of  $3 \times F$ ; the vectors  $\mathbf{u}$  and  $\mathbf{v}$  have respective elements of  $u_i = \cos[\delta(i)]$  and  $v_i = \sin[\delta(i)]$ .

Assuming the matrix  $\mathbf{U}$  is known, the matrix  $\mathbf{V}$  can be calculated as

$$\mathbf{V} = \mathbf{IU}(\mathbf{UU}^T)^{-1}, \quad (2.17)$$

where the superscript  $-1$  denotes the matrix inversion. Next, to introduce a perturbation to the iterations, nonlinear boosting is carried out with the following steps (i) calculating the phase as

$$\varphi = \text{atan2}\left(\mathbf{V}_{:,3}, \mathbf{V}_{:,2}\right). \quad (2.18)$$

where the notation  $\mathbf{V}_{:,i}$  means the  $i$ th row of matrix  $\mathbf{V}$ . (ii) And updating two components of the matrix  $\mathbf{V}$  as

$$\mathbf{V}_{:,2} = \cos(\varphi), \mathbf{V}_{:,3} = -\sin(\varphi). \quad (2.19)$$

Subsequently, the matrix  $\mathbf{U}$  could be updated as

$$\mathbf{U} = (\mathbf{V}^T \mathbf{V})^{-1} \mathbf{V}^T \mathbf{I}. \quad (2.20)$$

Clearly, Eq. (2.17) and Eq. (2.20) are dual forms. Without the perturbation from the nonlinear boosting, the iterations will enter into a repeating cycle.

The matrix  $\mathbf{U}$  calculation, nonlinear boosting and matrix  $\mathbf{V}$  calculation are iteratively carried out. The VU starts the matrix  $\mathbf{U}$  initialized by random or pre-defined phase shifts and stops when the mean value of the change in phase is small enough, *i.e.*,

$$\frac{\sum_{j=0}^{N-1} [\varphi^m(j) - \varphi^{m-1}(j)]^2}{N} < \varepsilon_1, \quad (2.21)$$

or when the iteration number exceeds the higher limit  $M$ ,

$$m > M, \quad (2.22)$$

where  $m$  is the iteration index and  $\varepsilon_1$  is the pre-set error tolerance of the VU. In [167], settings of  $M=200$  and  $\varepsilon_1=1\times 10^{-8}$  rad were suggested.

### 2.2.2.3. Evaluation and issues of the decompositions based uPSAs

The PCA has two restrictions as indicated in its evaluation [188]: (i) There should be more than one fringe in the image so that the orthogonality assumption in Eq. (2.12) is valid; and (ii) the phase shifts must be well allocated in the  $[0, 2\pi]$  interval range so that the temporal averaging well estimates the background intensity and the principal components can be well decomposed [188]. These two restrictions will be evaluated through simulations in chapter 5 in which we will show more insights. Besides, the performance of the PCA will also be systematically compared with other uPSAs in chapter 5.

For the VU, there are very few evaluations. Hence, we will evaluate the restrictions when using the VU and will compare the performance of the VU with other uPSAs and also the kPSAs in chapter 5.

### 2.2.3. Optimization based uPSAs

The optimization based uPSAs use an alternative optimization framework to iteratively estimate the phase and the phase shifts by minimizing the sum of the squares of the residuals between the theoretical and measured fringe pattern. We select the advanced iterative algorithm (AIA) [169], Xu's algorithm [170], Hoang's algorithm [171], the phase-tilt iteration algorithm (PTI) [172] and

the model based phase-shifting interferometry algorithm (MPSI) [174] for discussion in this sub-chapter.

### 2.2.3.1.Principle of the AIA [169]

The AIA uses a fringe pattern as

$$I(i; j) = A(i; j) + B(i; j)\cos[\varphi(j) + \delta(i)]. \quad (2.23)$$

which is the same as the one used in the PCA. There are two steps iteratively executed in the AIA which are (i) the pixel by pixel phase calculation and (ii) the frame by frame phase shifts calculation.

In the pixel by pixel phase calculation, the AIA assumes that the background intensity and fringe amplitude are uniform across the frames. With the known phase shifts, Eq. (2.23) is rewritten as

$$I(i; j) = a(j) + b(j)\cos[\delta(i)] + c(j)\sin[\delta(i)], \quad (2.24)$$

where  $a(j)=A(j)$ ;  $b(j)=B(j)\cos[\varphi(j)]$  and  $c(j)=-B(j)\sin[\varphi(j)]$ . The three unknowns  $a(j)$ ,  $b(j)$ , and  $c(j)$  are determined through linear least-squares optimization by minimizing the sum of the squares of the residuals between the theoretical and measured fringe pattern as

$$\mathbf{X}_p(j) = \mathbf{A}_p^{-1} \mathbf{B}_p(j), \quad (2.25)$$

where

$$\mathbf{X}_p(j) = [a(j) \ b(j) \ c(j)]^T, \quad (2.26)$$

$$\mathbf{A}_p = \begin{bmatrix} F & \sum_{i=0}^{F-1} \cos \delta(i) & \sum_{i=0}^{F-1} \sin \delta(i) \\ \sum_{i=0}^{F-1} \cos \delta(i) & \sum_{i=0}^{F-1} \cos^2 \delta(i) & \sum_{i=0}^{F-1} \cos \delta(i) \sin \delta(i) \\ \sum_{i=0}^{F-1} \sin \delta(i) & \sum_{i=0}^{F-1} \cos \delta(i) \sin \delta(i) & \sum_{i=0}^{F-1} \sin^2 \delta(i) \end{bmatrix}, \quad (2.27)$$

$$\mathbf{B}_p(j) = [\sum_{i=0}^{F-1} I(i, j) \ \sum_{i=0}^{F-1} I(i, j) \cos \delta(i) \ \sum_{i=0}^{F-1} I(i, j) \sin \delta(i)]^T, \quad (2.28)$$

and the superscript  $-1$  in Eq. (2.25) indicates matrix inversion. Next, the phase can be extracted as

$$\varphi(j) = \text{atan2}[-c(j), b(j)]. \quad (2.29)$$

In the frame by frame phase shifts calculation, the AIA assumes that the background intensity and fringe amplitude are uniform across the pixels. With the known phase shifts, Eq. (2.23) is rewritten as

$$I(i, j) = a'(i) + b'(i) \cos[\varphi(j)] + c'(i) \sin[\varphi(j)], \quad (2.30)$$

where  $a'(i)=A(i)$ ;  $b'(i)=B(i)\cos[\delta(i)]$  and  $c'(i)=-B(i)\sin[\delta(i)]$ . Similarly, the three unknowns  $a'(j)$ ,  $b'(j)$ , and  $c'(j)$  are calculated through linear least-squares optimization by minimizing the sum of the squares of the residuals between the theoretical and measured fringe pattern as

$$\mathbf{X}_{\text{ps}}(i) = \mathbf{A}_{\text{ps}}^{-1} \mathbf{B}_{\text{ps}}(i), \quad (2.31)$$

where

$$\mathbf{X}_{\text{ps}}(i) = [a'(i) \ b'(i) \ c'(i)]^T, \quad (2.32)$$

$$\mathbf{A}_{\text{ps}} = \begin{bmatrix} N & \sum_{j=0}^{N-1} \cos \varphi(j) & \sum_{j=0}^{N-1} \sin \varphi(j) \\ \sum_{j=0}^{N-1} \cos \varphi(j) & \sum_{j=0}^{N-1} \cos^2 \varphi(j) & \sum_{j=0}^{N-1} \cos \varphi(j) \sin \varphi(j) \\ \sum_{j=0}^{N-1} \sin \varphi(j) & \sum_{j=0}^{N-1} \cos \varphi(j) \sin \varphi(j) & \sum_{j=0}^{N-1} \sin^2 \varphi(j) \end{bmatrix}, \quad (2.33)$$

$$\mathbf{B}_{\text{ps}}(i) = \left[ \sum_{j=0}^{N-1} I(i, j) \ \sum_{j=0}^{N-1} I(i, j) \cos \varphi(j) \ \sum_{j=0}^{N-1} I(i, j) \sin \varphi(j) \right]^T. \quad (2.34)$$

Next, the phase shifts could be extracted as

$$\delta(i) = \text{atan2}[-c'(i), b'(i)]. \quad (2.35)$$

To start the iterations, the AIA initializes the phase shifts based on foreknowledge or simply initializes the phase shifts as random values. The phase and phase shifts estimation are iteratively executed until the change in phase shifts are small enough

$$\left| [\delta^m(i) - \delta^m(1)] - [\delta^{m-1}(i) - \delta^{m-1}(1)] \right| < \varepsilon_1, i = 0, 1, \dots, F-1, \quad (2.36)$$

or when the iteration number exceeds the higher limit  $M$ ,

$$m > M, \quad (2.37)$$

where the superscript  $m$  is the index of the iterations,  $|.|$  outputs the input's absolute value, and  $\varepsilon_1$  is the error tolerance of the AIA. In [169], settings of  $M=200$  and  $\varepsilon_1=1\times 10^{-4}$  rad were suggested.

### 2.2.3.2.Principle of Xu's algorithm [170]

For phase extraction from fringe patterns with intensity harmonics, Xu *et al.* use a fringe pattern model as

$$I(i; j) = A(i; j) + \sum_{k=1}^P B_k(i; j) \cos[k\varphi(j) + k\delta(i)]. \quad (2.38)$$

where  $B_k$  is the fringe amplitudes of different orders of intensity harmonics.

Firstly, Xu's algorithm assumes known phase shifts and frame independent background intensity and fringe amplitudes and rewrites Eq. (2.38) as

$$I(i; j) = \sum_{q=0}^{2P} X_q(j) S_q(i), \quad (2.39)$$

where  $X_0(j) = A(j)$ ,  $X_{2k-1}(j) = B_k(j) \cos[k\varphi(j)]$ ,  $X_{2k}(j) = -B_k(j) \sin[k\varphi(j)]$ ,  $S_0(i) = 1$ ,  $S_{2k-1}(i) = \cos[k\delta(i)]$ , and  $S_{2k}(i) = \sin[k\delta(i)]$ . This representation directly extends the idea in the AIA. The solution is also similar,

$$\mathbf{X}_p(j) = \mathbf{A}_p^{-1} \mathbf{B}_p(j), \quad (2.40)$$

where

$$\mathbf{X}_p(j) = [X_0(j) \ X_1(j) \ \dots \ X_{2P-1}(j) \ X_{2P}(j)]^T, \quad (2.41)$$

$\mathbf{A}_p$  is a  $(2P+1) \times (2P+1)$  matrix with each entry represented as

$$(\mathbf{A}_p)_{v,u} = \sum_{i=0}^{F-1} S_v(i) S_u(i), v, u = 0, 1, \dots, 2P, \quad (2.42)$$

and  $\mathbf{B}_p(j)$  is a vector with  $(2P+1)$  terms,

$$[\mathbf{B}_p(j)]_v = \sum_{i=0}^{F-1} I(i, j) S_v(i), v = 0, 1, \dots, 2P. \quad (2.43)$$

Subsequently, the phase is calculated as

$$\varphi(j) = \text{atan2}[-X_2(j), X_1(j)]. \quad (2.44)$$

In the second step, Xu's algorithm assumes known phase and pixel independent background intensity and fringe amplitudes and rewrites Eq. (2.38) as

$$I(i, j) = \sum_{q=0}^{2P} X'_q(i) S'_q(j), \quad (2.45)$$

where  $X'_0(i) = A(i)$ ,  $X'_{2k-1}(i) = B_k(i) \cos[k\delta(i)]$ ,  $X'_{2k}(i) = -B_k(i) \sin[k\delta(i)]$ ,  $S'_0(j) = 1$ ,  $S'_{2k-1}(j) = \cos[k\varphi(j)]$ , and  $S'_{2k}(j) = \sin[k\varphi(j)]$ . The least-squares solution gives

$$\mathbf{X}_{\text{ps}}(i) = \mathbf{A}_{\text{ps}}^{-1} \mathbf{B}_{\text{ps}}(i), \quad (2.46)$$

where

$$\mathbf{X}_{\text{ps}}(i) = [X'_0(i) \ X'_1(i) \ \dots \ X'_{2P-1}(i) \ X'_{2P}(i)]^T, \quad (2.47)$$

$\mathbf{A}_{\text{ps}}$  is a  $(2P+1) \times (2P+1)$  matrix with each entry represented as

$$(\mathbf{A}_{\text{ps}})_{v,u} = \sum_{j=0}^{N-1} S'_v(j) S'_u(j), v, u = 0, 1, \dots, 2P, \quad (2.48)$$

and  $\mathbf{B}_{\text{ps}}(i)$  is a vector with  $(2P+1)$  terms,

$$[\mathbf{B}_{\text{ps}}(i)]_v = \sum_{j=0}^{N-1} I(i, j) S'_v(j), v = 0, 1, \dots, 2P. \quad (2.49)$$

Subsequently, the phase shifts are calculated as

$$\delta(i) = \text{atan2}[-X'_2(i), X'_1(i)]. \quad (2.50)$$

Xu's algorithm iteratively executes the phase estimation and the phase shifts estimation. Besides, Xu's algorithm and the AIA have the same starting settings and stopping criteria.

#### 2.2.3.3. Principle of Hoang's algorithm [171]

Hoang's algorithm is also designed for phase extraction from fringe patterns with intensity harmonics, thus its fringe pattern model is the same as the one used in Xu's algorithm.

In the first step, Hoang's algorithm assumes pixel independent background intensity and fringe amplitudes. Subsequently, with the known phase and phase shifts, the background intensity and fringe amplitudes are calculated frame-wisely as

$$\mathbf{X}_{\text{ps\_b}}(i) = \mathbf{A}_{\text{ps\_b}}^{-1}(i) \mathbf{B}_{\text{ps\_b}}(i), \quad (2.51)$$

where

$$\mathbf{X}_{\text{ps\_b}}(i) = [A(i) \ B_1(i) \ \dots \ B_{P-1}(i) \ B_P(i)]^T, \quad (2.52)$$

$\mathbf{A}_{\text{ps\_b}}(i)$  is a  $(P+1) \times (P+1)$  matrix with each element represented as

$$[\mathbf{A}_{\text{ps\_b}}(i)]_{v,u} = \sum_{j=0}^{N-1} \{\cos[v\varphi(j) + v\delta(i)] \cos[u\varphi(j) + u\delta(i)]\}, v, u = 0, 1, \dots, P, \quad (2.53)$$

and  $\mathbf{B}_{\text{ps\_b}}(i)$  is a vector with  $(P+1)$  terms,

$$[\mathbf{B}_{\text{ps\_b}}(i)]_v = \sum_{j=0}^{N-1} \{I(i, j) \cos[v\varphi(j) + v\delta(i)]\}, v = 0, 1, \dots, P. \quad (2.54)$$

With the calculated background intensity and fringe amplitudes, and the existing phase values, the phase shift of every frame is updated through an exhaustive search by minimizing the sum of the squares of the residuals between the theoretical and measured fringe pattern. As suggested by the creators of this algorithm [171], the phase shift of every frame will be searched with a step of  $10^{-4}$  rad in a range of  $\pi$  around the old phase shift. The purpose of this fine search is for the convergence of the phase shifts.

Secondly, Hoang's algorithm assumes frame independent background intensity and fringe amplitudes. Subsequently, with the known phase and the updated phase shifts, the background intensity and fringe amplitude of every order are calculated pixel-wisely as

$$\mathbf{X}_{p\_b}(j) = \mathbf{A}_{p\_b}^{-1}(j) \mathbf{B}_{p\_b}(j), \quad (2.55)$$

where

$$\mathbf{X}_{p\_b}(j) = [A(j) \ B_1(j) \ \dots \ B_{P-1}(j) \ B_P(j)]^T, \quad (2.56)$$

$\mathbf{A}_{p\_b}(j)$  is a  $(P+1) \times (P+1)$  matrix with each element represented as

$$[\mathbf{A}_{p\_b}(j)]_{v,u} = \sum_{i=0}^{F-1} \left\{ \cos[v\varphi(j) + v\delta(i)] \cos[u\varphi(j) + u\delta(i)] \right\}, \quad v, u = 0, 1, \dots, P, \quad (2.57)$$

$\mathbf{B}_{p\_b}(j)$  is a vector with  $(P+1)$  terms,

$$[\mathbf{B}_{p\_b}(j)]_v = \sum_{i=0}^{F-1} \left\{ I(i, j) \cos[v\varphi(j) + v\delta(i)] \right\}, \quad v = 0, 1, \dots, P. \quad (2.58)$$

Subsequently, the phase is updated through an exhaustive search by minimizing the sum of the squares of the residuals between the theoretical and measured fringe pattern. The searching range and the searching step are set as  $\pi$  and  $0.0001$  rad, respectively.

Hoang's algorithm uses the AIA to initialize the phase and phase shifts. Meanwhile, Hoang's algorithm's stopping criteria is the same as the ones of the AIA.

#### 2.2.3.4. Principle of the PTI [172]

The PTI minimizes the influence of the phase-shift tilts [172] and uses a fringe pattern model as

$$\begin{aligned} I(i; j) &= A(j) + B(j) \cos[\varphi(j) + \delta(i; j)] \\ &= A(j) + B(j) \cos[\varphi(j) + \delta_0(i) + \alpha_1(i)x + \beta_1(i)y], \end{aligned} \quad (2.59)$$

where  $\delta_0$ ,  $\alpha_1$  and  $\beta_1$  are the phase-shift tilts' parameters.

Firstly, with the known  $\delta(i; j)$ , Eq. (2.59) can be rewritten as

$$I(i; j) = a(j) + b(j)\cos[\delta(i; j)] + c(j)\sin[\delta(i; j)], \quad (2.60)$$

where  $a(j)=A(j)$ ;  $b(j)=B(j)\cos[\varphi(j)]$ ; and  $c(j)=-B(j)\sin[\varphi(j)]$ . The three unknowns  $a(j)$ ,  $b(j)$ , and  $c(j)$  are determined as

$$\mathbf{X}_p(j) = \mathbf{A}_p^{-1}(j)\mathbf{B}_p(j), \quad (2.61)$$

where

$$\mathbf{X}_p(j) = [a(j) \ b(j) \ c(j)]^T, \quad (2.62)$$

$$\mathbf{A}_p(j) = \begin{bmatrix} F & \sum_{i=0}^{F-1} \cos \delta(i; j) & \sum_{i=0}^{F-1} \sin \delta(i; j) \\ \sum_{i=0}^{F-1} \cos \delta(i; j) & \sum_{i=0}^{F-1} \cos^2 \delta(i; j) & \sum_{i=0}^{F-1} \frac{\cos \delta(i; j)}{\sin \delta(i; j)} \\ \sum_{i=0}^{F-1} \sin \delta(i; j) & \sum_{i=0}^{F-1} \frac{\cos \delta(i; j)}{\sin \delta(i; j)} & \sum_{i=0}^{F-1} \sin^2 \delta(i; j) \end{bmatrix}, \quad (2.63)$$

and

$$\mathbf{B}_p(j) = [\sum_{i=0}^{F-1} I(i; j) \ \sum_{i=0}^{F-1} I(i; j)\cos \delta(i; j) \ \sum_{i=0}^{F-1} I(i; j)\sin \delta(i; j)]^T. \quad (2.64)$$

Subsequently, the phase, background intensity and fringe amplitude are calculated as

$$\varphi(j) = \text{atan2}[-c(j), b(j)], \quad (2.65)$$

$$A(j) = a(j), \quad (2.66)$$

and

$$B(j) = \sqrt{b(j)^2 + c(j)^2}. \quad (2.67)$$

Secondly, the phase-shift tilts' parameters in the row direction are estimated based on the fringe patterns in the  $Y$ th column. Eq. (2.59) is rewritten as

$$\begin{aligned} I(i; j_{x,Y}) &= A(j_{x,Y}) + B(j_{x,Y})\cos[\varphi(j_{x,Y}) + \delta_{0Y}(i) + \alpha_1(i)x] \\ &= A(j_{x,Y}) + B(j_{x,Y})\cos[\Phi_r(i; j_{x,Y}) + p_r(i)], \end{aligned} \quad (2.68)$$

where  $j_{x,Y}$  is the index of pixels in the  $Y$ th column;  $\delta_{0Y}$  is the mean phase shifts calculated based on the  $Y$ th column,  $\Phi_r(i; j_{x,Y}) = \varphi(j_{x,Y}) + \alpha_1(i)(x-r)$ ,  $p_r(i) = \delta_{0Y}(i) + \alpha_1(i)r$  and  $r=0, 1, 2, \dots, R$ . Subsequently, Eq. (2.68) is rewritten as

$$I(i; j_{x,Y}) = A(j_{x,Y}) + B(j_{x,Y}) \left\{ \cos[\Phi_r(i; j_{x,Y})] b'_r(i) + \sin[\Phi_r(i; j_{x,Y})] c'_r(i) \right\}, \quad (2.69)$$

where  $b'_r(i) = \cos[p_r(i)]$  and  $c'_r(i) = -\sin[p_r(i)]$ . Based on the calculated  $\varphi$ ,  $A$  and  $B$  together with the known  $\delta_0$ ,  $\alpha_1$  and  $\beta_1$  from the last iteration, the two unknowns  $b'_r(i)$  and  $c'_r(i)$  are determined as

$$\begin{bmatrix} b'_r(i) & c'_r(i) \end{bmatrix}^T = \mathbf{A}_{ps\_r}^{-1}(i) \mathbf{B}_{ps\_r}(i), \quad (2.70)$$

where

$$\mathbf{A}_{ps\_r}^{-1}(i) = \begin{bmatrix} \sum_{x=0}^{N_x-1} B(j_{x,Y})^2 & B(j_{x,Y})^2 \\ \sum_{x=0}^{N_x-1} \cos^2[\Phi_r(i; j_{x,Y})] & \sum_{x=0}^{N_x-1} \cos[\Phi_r(i; j_{x,Y})] \\ B(j_{x,Y})^2 & \sin[\Phi_r(i; j_{x,Y})] \\ \sum_{x=0}^{N_x-1} \cos[\Phi_r(i; j_{x,Y})] & \sum_{x=0}^{N_x-1} B(j_{x,Y})^2 \\ \sin[\Phi_r(i; j_{x,Y})] & \sin^2[\Phi_r(i; j_{x,Y})] \end{bmatrix}, \quad (2.71)$$

and

$$\mathbf{B}_{ps\_r}(i) = \begin{bmatrix} \sum_{x=0}^{N_x-1} [I(i; j_{x,Y}) - A(j_{x,Y})] B(j_{x,Y}) \cos[\Phi_r(i; j_{x,Y})] \\ \sum_{x=0}^{N_x-1} [I(i; j_{x,Y}) - A(j_{x,Y})] B(j_{x,Y}) \sin[\Phi_r(i; j_{x,Y})] \end{bmatrix}. \quad (2.72)$$

Subsequently, the  $p_r(i)$  could be calculated as

$$p_r(i) = \text{atan2}\left[-c'_r(i), b'_r(i)\right]. \quad (2.73)$$

Next, with the  $R$  calculated  $p_r(i)$ , the phase-shift tilts' parameters  $\delta_0$  and  $\alpha_1$  could be updated as

$$\alpha(i) = \frac{1}{R} \sum_{r=1}^R \tan^{-1} \left[ \frac{\sin[p_r(i) - p_{r-1}(i)]}{\cos[p_r(i) - p_{r-1}(i)]} \right], \quad (2.74)$$

and

$$\delta_{0Y}(i) = p_0(i). \quad (2.75)$$

In the third step, the phase-shift tilts' parameter in the row direction  $\beta_1$  and the mean phase shifts based on the  $X$ th row  $\delta_{0X}$  are calculated through a similar method. Subsequently, the mean phase shift of the entire frame could be calculated as

$$\delta_0(i) = \frac{\delta_{0X} - \beta_1(i)Y}{2} + \frac{\delta_{0Y} - \alpha_1(i)X}{2}. \quad (2.76)$$

At the starting point, the PTI uses the phase shifts calculated by the AIA from small regions of the fringe patterns to linearly fit  $\delta_0$ ,  $\alpha_1$  and  $\beta_1$ . These parameters are used to initialize the PTI. The PTI stops when the iteration number exceeds the higher limit  $M$  as shown in Eq. (2.37), or when the change in  $\delta_0$ ,  $\alpha_1$  and  $\beta_1$  of all the frames are sufficiently small, *i.e.*,

$$\left[ \delta_0^{m+1}(i) - \delta_0^m(0) \right] - \left[ \delta_0^m(i) - \delta_0^m(0) \right] < \varepsilon_1, i = 0, 1, \dots, F-1, \quad (2.77)$$

$$\left[ \alpha_1^{m+1}(i) - \alpha_1^m(0) \right] - \left[ \alpha_1^m(i) - \alpha_1^m(0) \right] < \varepsilon_2, i = 0, 1, \dots, F-1, \quad (2.78)$$

and

$$\left[ \beta_1^{m+1}(i) - \beta_1^m(0) \right] - \left[ \beta_1^m(i) - \beta_1^m(0) \right] < \varepsilon_2, i = 0, 1, \dots, F-1. \quad (2.79)$$

In [172], settings of  $M=200$ ,  $\varepsilon_1=1\times 10^{-4}$  rad and  $\varepsilon_2=1\times 10^{-4}$  were suggested.

#### 2.2.3.5. Principle of the MPSI [174]

The MPSI incorporates both the intensity harmonics and the phase-shift tilts. In design, the MPSI is intended for phase extraction of the Fizeau interferometer. Meanwhile, phase extraction with the MPSI from general phase-shifted fringe patterns could also be achieved and will be discussed in this sub-chapter. In the MPSI, a fringe pattern model with intensity harmonics induced by multi-reflections and phase-shift tilts is considered,

$$I(i; j) = A(j) + \sum_{k=1}^P B(j)(-g)^{k-1} \cos\left\{k[\varphi(j) + \delta_0(i) + \alpha_1(i)x + \beta_1(i)y]\right\}, \quad (2.80)$$

where  $g$  represents the intensity harmonics caused by multiple reflections.

Firstly, with the known background intensity  $A$ , fringe amplitude  $B$  and phase  $\varphi$ , the fringe pattern model could be linearized to the first order in the deviations of  $\delta_0$ ,  $\alpha_1$  and  $\beta_1$  as,

$$I(i; j) = A(j) + \sum_{k=1}^P B(j)(-g)^{k-1} \cos\left\{k[\varphi(j) + k\delta(i; j)]\right\} + [\delta'_0(i) + \alpha'(i)x + \beta'(i)y]H(i; j), \quad (2.81)$$

with

$$H(i; j) = -\sum_{k=1}^P B(j)(-g)^{k-1} \cos\left\{k[\varphi(j) + k[\delta_0(i) + \alpha_1(i)x + \beta_1(i)y]]\right\}. \quad (2.82)$$

where the prime represents a small deviation from the current value. Subsequently, the deviation of phase-shift tilts related parameters are estimated through linear regression by minimizing the sum of the squares of the residuals between the theoretical and measured fringe pattern.

In the second step, with the phase shifts calculated in the first step, Eq. (2.80) is rewritten as

$$I(i;j) = \sum_{q=0}^{2P} X_q(j) S_q(i), \quad (2.83)$$

where  $X_0(j)=A(j)$ ,  $X_{2k-1}(j)=B(j)(-g)^{k-1}\cos[k\varphi(j)]$ ,  $X_{2k}(j)=-B(j)(-g)^{k-1}\sin[k\varphi(j)]$ ,  $S_0(i;j)=1$ ,  $S_{2k-1}(i)=\cos[k\delta(i;j)]$ , and  $S_{2k}(i)=\sin[k\delta(i;j)]$ . The solution of the linear least-squares optimization is similar to the AIA,

$$\mathbf{X}_p(j) = \mathbf{A}_p^{-1}(j) \mathbf{B}_p(j), \quad (2.84)$$

where

$$\mathbf{X}_p(j) = [X_0(j) \ X_1(j) \ \dots \ X_{2P-1}(j) \ X_{2P}(j)]^T, \quad (2.85)$$

$\mathbf{A}_p(j)$  is a  $(2P+1) \times (2P+1)$  matrix with each entry represented as

$$[\mathbf{A}_p(j)]_{v,u} = \sum_{i=0}^{F-1} S_v(i;j) S_u(i;j), v, u = 0, 1, \dots, 2P, \quad (2.86)$$

and  $\mathbf{B}_p(j)$  is a vector with  $(2P+1)$  terms,

$$[\mathbf{B}_p(j)]_v = \sum_{i=0}^{F-1} I(i;j) S_v(i;j), v = 0, 1, \dots, 2P. \quad (2.87)$$

Subsequently, the phase  $\varphi$ , background intensity  $A$ , fringe amplitude  $B$  and intensity harmonic parameter  $g$  are calculated as

$$\varphi(j) = \text{atan2}[-X_2(j), X_1(j)], \quad (2.88)$$

$$A(j) = X_0(j), \quad (2.89)$$

$$B(j) = \sqrt{X_1(j)^2 + X_2(j)^2}, \quad (2.90)$$

and

$$g = \frac{\sum_{j=0}^{N-1} \sqrt{X_3(j)^2 + X_4(j)^2}}{\sum_{j=0}^{N-1} B(j)}. \quad (2.91)$$

The above iterations stop when the iteration number exceeds the higher limit  $M$  as shown in Eq. (2.37) or when the change in the sum of the squares of the residuals between the theoretical and measured fringe pattern is smaller than  $\zeta$ . The MPSI starts with pre-calibrated phase shifts, phase extracted with a kPSA [193] together with background intensity and fringe amplitude initialized as

$$A(j) = \frac{\max_i[I(i;j)] + \min_i[I(i;j)]}{2}, \quad (2.92)$$

and

$$B(j) = \frac{\max_i [I(i; j)] - \min_i [I(i; j)]}{2}. \quad (2.93)$$

where max and min return the maximum and minimum value of the input, respectively. In [174], settings of  $M=200$  and  $\zeta=0.01\%$  were suggested.

#### 2.2.3.6. Evaluations and issues of the optimization based uPSAs

In our evaluation, we noticed that although these optimization based uPSAs have demonstrated good accuracy performance, they also have their limitations including:

- (i) The AIA's robustness with respect to different types of fringe patterns and the AIA's accuracy with respect to different fringe parameters are not clear. These aspects will be evaluated in chapter 3. Meanwhile, although the AIA is the most well-recognized uPSA, its performance was never comprehensively compared with the kPSAs and the other uPSAs. This comparison will be carried out in chapter 5.
- (ii) The computation speed of the AIA is much slower compared to the kPSAs due to the iterations. This inefficacy problem becomes much more serious when the frame number and/or pixel number are large. Hence, the acceleration of the AIA will be discussed in chapter 6.
- (iii) Many uPSAs are designed to minimize the impact of different error sources, *i.e.*, the AIA is designed to suppress the impact of the phase-shift errors; Xu's and Hoang's algorithms are designed to suppress the impact of the intensity harmonics, the PTI is designed to suppress the impact of the phase-shift tilts; and the MPSI is designed to suppress the impact of the intensity harmonics due to the multiple reflections and the phase-shift tilts. However, none of them can deal with all the error sources mentioned in chapter 1.3.1. Thus, a novel uPSA is proposed in chapter 4 for accurate phase extraction from fringe patterns with all the concerned error sources.
- (iv) Compared to the AIA, Xu's algorithm, Hoang's algorithm and the MPSI all have worse error immunities to random noise. This is due to the fact that the least-squares matrices used in these algorithms are with higher rank compared to the ones used in the AIA, *i.e.* Xu's algorithm and the MPSI used  $(2P+1) \times (2P+1)$  matrices, Hoang's algorithm used

$(P+1) \times (P+1)$  matrices and the AIA used  $3 \times 3$  matrices. A higher rank least-squares matrix will lead to bad error immunity to random noise [194]. This phenomenon will be detailly demonstrated in Chapter 5.



# **Chapter 3 Advanced iterative algorithm for phase extraction: performance evaluation and enhancement<sup>\*</sup>**

## **3.1. Background**

In this chapter, we focus on the evaluation and enhancement of the robustness and the accuracy of the uPSAs. As a starting point, we evaluate and enhance the most well-recognized uPSA, the AIA. As discussed in chapter 2.2.3, the AIA uses an alternative optimization framework to iteratively estimate the phase and phase shifts. The AIA can extract phase accurately from fringe patterns with arbitrary frame number ( $\geq 3$ ) and random phase shifts with a simple and elegant algorithm structure. However, compared to the well-evaluated kPSAs, the AIA's robustness and accuracy were not systematically evaluated, which leads to our evaluations. Firstly, one major difficulty in the evaluation of the AIA is the characterization of the phase and phase shifts, for which we use the condition numbers of the least-squares matrices used in the AIA. Subsequently, the fringe density requirement for the AIA's success is confirmed. Secondly, we conducted simulations to evaluate the accuracy of the phase extracted by the AIA with respect to different phase shifts, frame numbers and noise levels. Subsequently, we propose controlling the phase shifts, controlling the frame number and suppressing the noise level, as three accuracy improving practices for the AIA. Thirdly, a phase error model is established for the AIA. Meanwhile, a practical method to estimate the phase errors before the phase extraction is proposed. Benefiting from the evaluations and the proposed improving practices, an enhanced AIA (eAIA) is proposed for the AIA's success and for achieving the desired phase accuracy. To demonstrate the effectiveness of the AIA, an optical experiment is carried out with a Fizeau interferometer.

---

<sup>\*</sup> The work in this chapter has been published as Y. Chen and Q. Kemao, "Advanced iterative algorithm for phase extraction: performance evaluation and enhancement," *Optics express*, vol. 27, no. 26, pp. 37634-37651, 2019.

### 3.2. Condition numbers for phase and phase shifts characterization

The AIA has been detailly reviewed in chapter 2.2.3.1. Clearly, the two least-squares matrices  $\mathbf{A}_p$  and  $\mathbf{A}_{ps}$  are critical to both the accuracy and robustness of the AIA. If  $\mathbf{A}_p$  or  $\mathbf{A}_{ps}$  is singular or near singular, the convergence of the AIA cannot be guaranteed. Hence, we adopt the condition number to qualify these matrices as the condition number directly indicates the singularity of a matrix. Meanwhile, as the two least-squares matrices  $\mathbf{A}_p$  and  $\mathbf{A}_{ps}$  are directly related to the phase and the phase shifts, the condition numbers of these matrices can also be used to characterize the phase and phase shifts. For a certain matrix  $\mathbf{H}$ , the condition number can be calculated as the ratio between its maximum singular value  $\sigma_{\max}(\mathbf{H})$  and minimum singular value  $\sigma_{\min}(\mathbf{H})$  [195],

$$\kappa(\mathbf{H}) = \frac{\sigma_{\max}(\mathbf{H})}{\sigma_{\min}(\mathbf{H})}. \quad (3.1)$$

We start with the evaluation of the matrix  $\mathbf{A}_{ps}$ . With a real vector  $\mathbf{x}$ , we can calculate the inner product of the two vectors  $\mathbf{A}_{ps}\mathbf{x}$  and  $\mathbf{x}$  as

$$\langle \mathbf{A}_{ps}\mathbf{x}, \mathbf{x} \rangle = \mathbf{x}^T \mathbf{A}_{ps} \mathbf{x} = \sum_{j=1}^N \left\{ x_1 + \cos[\varphi(j)]x_2 + \sin[\varphi(j)]x_3 \right\}^2 \geq 0, \quad (3.2)$$

which means that the matrix  $\mathbf{A}_{ps}$  is positive semidefinite [195]. Meanwhile, as shown in Eq. (2.33), the matrix  $\mathbf{A}_{ps}$  is also real and symmetrical. Hence, the eigenvalues of  $\mathbf{A}_{ps}$  equal to the singular values of  $\mathbf{A}_{ps}$  [195]. Subsequently, following the property of the Rayleigh quotient [196], it is not hard to derive that

$$\sigma_{\min}(\mathbf{A}_{ps}) \leq \frac{\mathbf{x}^T \mathbf{A}_{ps} \mathbf{x}}{\mathbf{x}^T \mathbf{x}} \leq \sigma_{\max}(\mathbf{A}_{ps}), \quad (3.3)$$

for vector  $\mathbf{x}$  with nonzero components.

Considering some special values of vector  $\mathbf{x}$ , e.g.  $[1 \ 0 \ 0]^T$ ,  $[0 \ 1 \ 0]^T$  and  $[0 \ 0 \ 1]^T$ , we can get

$$\sigma_{\max}(\mathbf{A}_{ps}) \geq N, \quad (3.4)$$

$$\sigma_{\min}(\mathbf{A}_{ps}) \leq \min\left(\sum_{j=0}^{N-1} \cos^2[\varphi(j)], \sum_{j=0}^{N-1} \sin^2[\varphi(j)]\right). \quad (3.5)$$

Thus following Eq. (3.1), we have

$$\kappa(\mathbf{A}_{ps}) \geq \frac{N}{\min\left(\sum_{j=0}^{N-1} \sin^2[\varphi(j)], \sum_{j=0}^{N-1} \cos^2[\varphi(j)]\right)}. \quad (3.6)$$

With the help of the Pythagorean trigonometric identity,

$$\cos^2[\varphi(j)] + \sin^2[\varphi(j)] = 1, \quad (3.7)$$

we can get

$$\min\left(\sum_{j=1}^N \sin^2[\varphi(j)], \sum_{j=1}^N \cos^2[\varphi(j)]\right) \leq \frac{N}{2}. \quad (3.8)$$

Hence the inequality (3.6) becomes

$$\kappa(\mathbf{A}_{ps}) \geq 2. \quad (3.9)$$

The case  $\kappa(\mathbf{A}_{ps})=2$  is achievable when  $\sigma_{\min}(\mathbf{A}_{ps}) = \sum_{j=0}^{N-1} \sin^2[\varphi(j)] = \sum_{j=0}^{N-1} \cos^2[\varphi(j)] = N/2$ . This is the scenario that the phase is evenly allocated within  $[0, 2\pi]$ . Meanwhile,  $\kappa(\mathbf{A}_{ps})$  could reach infinity, *i.e.*,  $\mathbf{A}_{ps}$  is singular, in some special cases, such as when  $\varphi(j)=0$ , for all the pixels. Hence, in summary, we have  $2 \leq \kappa(\mathbf{A}_{ps}) < \infty$ . Following a similar computation, we can compute that  $2 \leq \kappa(\mathbf{A}_p) < \infty$ , where  $\kappa(\mathbf{A}_p)=2$  can be achieved when  $\delta(i)=2i\pi/F$ .

### 3.3. Simulation strategy and AIA's robustness

With the condition numbers, we could characterize the phase and the phase shifts. Hence, the robustness and the accuracy of the AIA could be evaluated. In this chapter, the fringe patterns used for the AIA's evaluation are introduced first. With a large among of simulations, we conclude a requirement on fringe density when using the AIA.

#### 3.3.1. Simulation strategy

To evaluate the performance of the AIA, many simulations have been carried out. By using simulations, we can compare the extracted phases with the ground true phase, and we can repeat the tests thousands of times. The AIA is coded according to chapter 2.2.3.1. In the simulations, we noticed that the mean values of the phase errors are comparable to the standard deviation values of the phase errors. Hence, we use the root-mean-square error (RMSE) as the phase error's indicator in this chapter.

The fringe patterns used for comparison are simulated as

$$I(x, y; i) = A(x, y; i) + B(x, y; i) \cos[\varphi(x, y) + \delta(i)] + n(x, y; i), \quad (3.10)$$

where  $n$  is the noise. The settings are as:

- (i) The frame number is set as  $F=3\sim40$ ;
- (ii) The image size is set as  $N=N_x\times N_y=256\times 256$ ;
- (iii) three phase distributions are simulated including (a) a linear phase ( $\varphi^1$ ), (b) a quadratic phase ( $\varphi^2$ ), and (c) a complex phase ( $\varphi^3$ ), as,

$$\varphi^1(x, y) = \frac{2N_p\pi}{N_x}x, \quad (3.11)$$

$$\varphi^2(x, y) = \frac{2N_p\pi}{\left(\frac{N_x}{2}\right)^2 + \left(\frac{N_y}{2}\right)^2} r^2, \quad (3.12)$$

$$\varphi^3(x, y) = N_p \times \text{peaks\_s}(N_x, N_y) + \omega_c x \quad (3.13)$$

where  $N_p$  is number of fringes over the frame,  $r^2=(x-N_x/2)^2+(y-N_y/2)^2$ , the “peaks\_s” function linearly scales the MATLAB’s “peaks” function into  $[0, 2\pi]$  and  $\omega_c$  is a carrier frequency;

- (iv) the phase shifts are either simulated to be random values within  $[0, 2\pi)$  or to be regular values as  $\delta(i)=2i\pi/F$  so that  $\kappa(A_{ps})$  reaches its minimal;
- (v) The background intensity  $A$  is assumed to be without pixel-to-pixel and frame-to-frame variations; The fringe amplitude  $B$  is set as  $B=A$ ;
- (vi) The noise in the fringe patterns is simulated as additive white gaussian noise which has a mean value of zero and different standard deviation values of  $\sigma/B=0\%\sim100\%$ .

Three examples of the fringe patterns used in the simulations are shown in Figure 3.1.

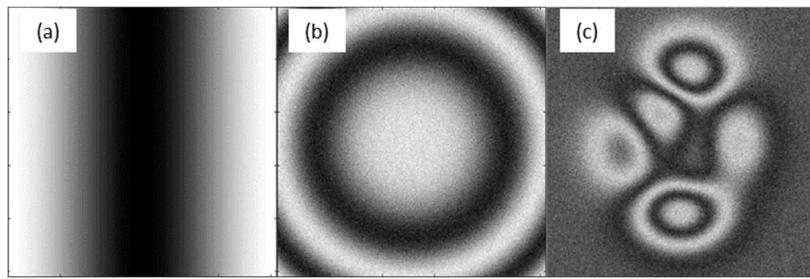


Figure 3.1.Fringe patterns used in simulations.

- (a) A fringe pattern with linear phase,  $N_p=1$  and no noise; (b) A fringe pattern with quadratic phase,  $N_p=2$  and random noise with  $\sigma/B=10\%$ ; (c) A fringe pattern with complex phase,  $N_p=3$ ,  $\omega_c=0$  and random noise with  $\sigma/B=20\%$

### 3.3.2.Fringe density requirement

In our simulations, we noticed that the phase distribution is the major influencing factor of the AIA's robustness. In the case that there are not sufficient variations of the phase, *i.e.*, the fringe density is low,  $\mathbf{A}_{ps}$  will be nearly singular. Meanwhile, in the extreme case that there is no variation in the phase,  $\mathbf{A}_{ps}$  will be singular. In these cases, the AIA will fail. Thus, we need to find the requirement on the fringe density for the AIA's success.

Simulations are carried out. The fringe patterns are simulated according to chapter 3.3.1 with special settings of  $F=4$  and  $\delta(i)=2i\pi/F$ . The three types of phase distributions are all tested with  $N_p=0.1\sim 3$  and  $\omega_c=0$ . One group of noiseless fringe patterns with  $\sigma/B=0\%$  and one group of noisy fringe patterns with  $\sigma/B=10\%$  are used in the simulations. We repeat the simulations 1,000 times for each group to minimize the impact of the AIA's initialization with random phase shifts. We plot  $\kappa(\mathbf{A}_{ps})$  of these fringe patterns with respect to  $N_p$  in Figure 3.2(a). The maximum RMSEs of the phases are calculated and plotted with respect to  $N_p$  in Figure 3.2(b) and Figure 3.2(c) for the noiseless and noisy cases, respectively. The RMSEs are high for both the noiseless and noisy cases when  $N_p<0.5$  as  $\mathbf{A}_{ps}$  is nearly singular. Thus,  $N_p=0.5$  is a critical point for the AIA's performance. Conservatively, we suggest a fringe density requirement of more than one fringe over the frame (*i.e.*,  $N_p\geq 1$ ) so that  $\kappa(\mathbf{A}_{ps})$  is close to 2 and a good RMSE result could be achieved. This fringe density requirement is further confirmed by simulations with 60,000 sets of fringe patterns. When  $N_p\geq 1$ , all tests converge with the AIA. This fringe density requirement will be satisfied in the rest of the chapter.

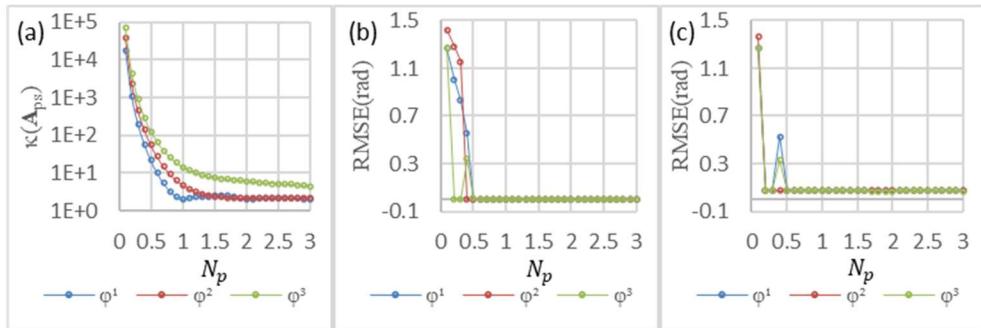


Figure 3.2.Performance with different densities.

- (a)  $\kappa(\mathbf{A}_{ps})$  of different phase distributions; (b) RMSEs from noiseless fringe patterns with different phase distributions; (c) RMSEs from noisy fringe patterns with different phase distributions.

### 3.4.AIA's accuracy performance

With the robustness of the AIA evaluated, we now move to the accuracy evaluation. In this chapter, the accuracy of the AIA with respect to different fringe parameters is evaluated. Based on the findings of the evaluations, an AIA's phase error model is integrated.

By fulfilling the fringe density requirement discussed in chapter 3.2, three types of phase distributions are used to test the accuracy of the AIA including (i) a linear phase  $\varphi^1$  with  $N_p=1$ , (ii) a quadratic phase  $\varphi^2$  with  $N_p=2$ , and (iii) a complex phase  $\varphi^3$  with  $N_p=10$  and  $\omega_c=0.25$ . One noisy fringe pattern is shown in Figure 3.3 for illustration.

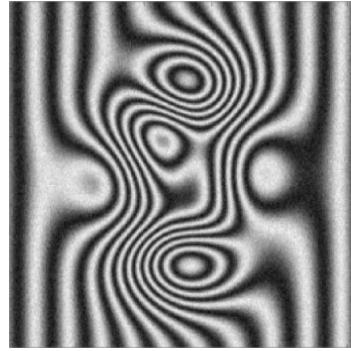


Figure 3.3.A fringe pattern with complex phase,  $N_p=10$ ,  $\omega_c=0.25$  and random noise with  $\sigma/B=10\%$

In our tests, we noticed that even with different types of phases, the accuracy performance of the AIA is very similar as long as other fringe parameters are the same. Hence, in later chapters, we only show the result with  $\varphi^3$ , even though all three phase distributions are evaluated.

#### 3.4.1.Influence of the phase shifts

As discussed in chapter 3.2, we can use  $\kappa(\mathbf{A}_p)$  to characterize the phase shifts. Hence, our task here could be simplified as evaluating  $\kappa(\mathbf{A}_p)$ 's influence on the AIA's accuracy. Simulations are carried out according to chapter 3.3.1. The fringe patterns are simulated with  $F= 4$ ; random phase shifts  $\kappa(\mathbf{A}_p)=2\sim 500$ ; and a fixed noise level  $\sigma/B=10\%$ .

We compare the AIA with the least-squares algorithm (LSA) where we assume the simulated phase shifts are known [150]. The RMSEs of the phases extracted by the AIA and the LSA are plotted against  $\kappa(\mathbf{A}_p)^{1/2}$  in Figure 3.4(a). Meanwhile, we also group  $\kappa(\mathbf{A}_p)^{1/2}$  into different bins with size of 1. For each bin, we calculate the mean value of the RMSEs which is plotted against the

mean value of each bin's  $\kappa(\mathbf{A}_p)^{1/2}$  in Figure 3.4(b). A linear correlation is observed for both the AIA and the LSA. A smaller  $\kappa(\mathbf{A}_p)$  will result in a smaller RMSE for the AIA as expected. Although, the AIA is with no knowledge of the phase shifts and the LSA is with the precise knowledge of the phase shifts, the AIA's accuracy performance is comparable to the LSA's accuracy performance. In the case  $\kappa(\mathbf{A}_p)^{1/2} > 10$ , the AIA's RMSEs are even slightly smaller.

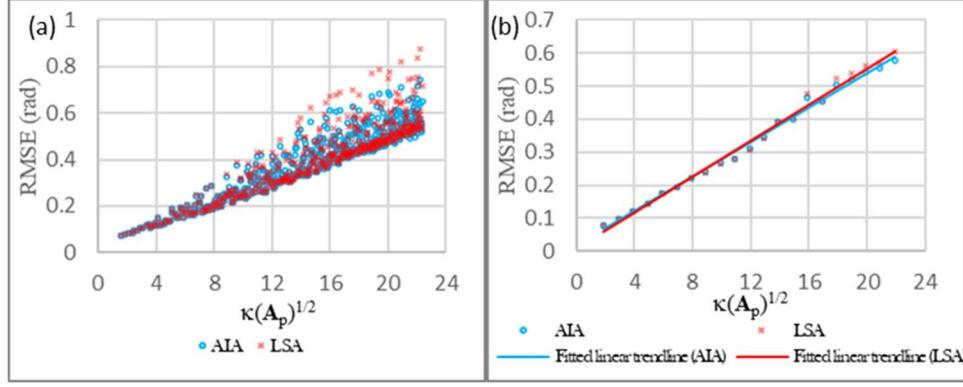


Figure 3.4.RMSEs of the phases extracted from fringe patterns with different phase shifts.

(a) with respect to  $\kappa(\mathbf{A}_p)^{1/2}$ ; (b) with respect to  $\kappa(\mathbf{A}_p)^{1/2}$ .

### 3.4.2.Influence of frame number

The frame number  $F$  is another factor which may affect the accuracy of the AIA. Simulations are carried out according to chapter 3.3.1. The simulated fringe patterns are with regular phase shifts  $\delta(i)=2i\pi/F$  and a fixed noise level  $\sigma/B=10\%$ . The AIA is used to extract the phases. The RMSEs of the phases are plotted against  $F$  and  $F^{-1/2}$  in Figure 3.5(a) and Figure 3.5(b), respectively. Clearly, the phase error of the AIA is smaller when the frame number is larger. Meanwhile, the RMSEs of the phases extracted by the AIA are linearly correlated to  $F^{-1/2}$ .

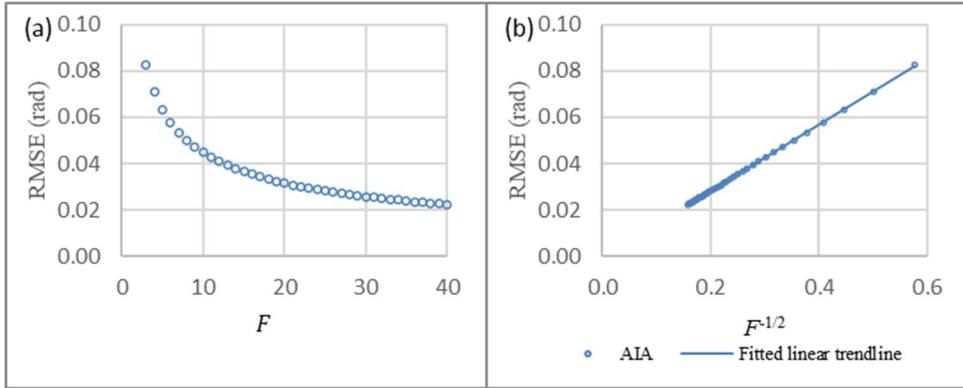


Figure 3.5.RMSEs of the phases extracted from fringe patterns with different frame numbers.

(a) with respect to  $F$ ; (b) with respect to  $F^{-1/2}$ .

### 3.4.3.Influence of noise

The noise will affect the accuracy of  $\mathbf{B}_p$  and  $\mathbf{B}_{ps}$  and thus affect the accuracy of the AIA. For verification, simulations are carried out according to chapter 3.3.1. The fringe patterns are simulated with  $F=4$ ,  $\delta(i)=2i\pi/F$  and different values of  $\sigma$ . The AIA is used to extract the phases. The RMSEs of the phases are plotted against  $\sigma/B$  as shown in Figure 3.6. Clearly, the RMSE is linearly related to  $\sigma$ .

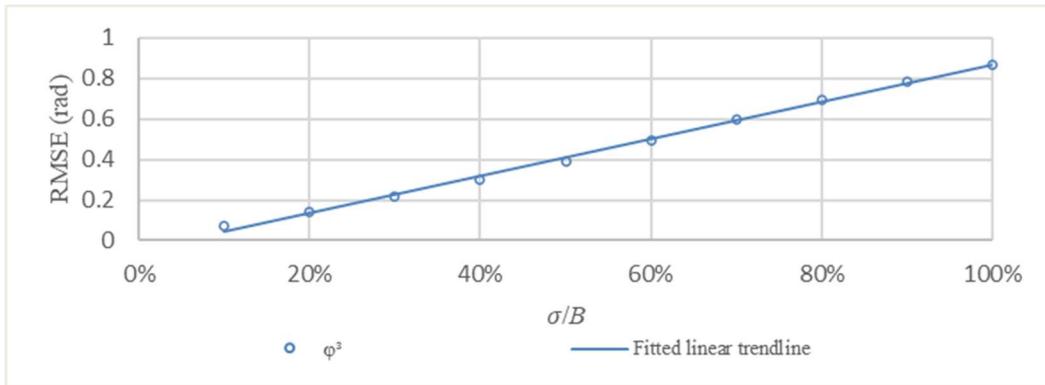


Figure 3.6.RMSEs of the phases extracted from fringe patterns with different noise levels.

### 3.4.4.The AIA's phase error model

The relationship between the phase errors and the  $F$  and the noise amplitude  $\sigma$  was not only for the AIA but was also found for the kPSAs [177, 179, 197]. Based on our evaluations on the relationship between the AIA's RMSE and  $\kappa(\mathbf{A}_p)$ ,  $F$  and  $\sigma$ , a general RMSE model is fitted for the AIA as

$$E_{\text{AIA}} = 0.42 \left( \sqrt{\kappa(\mathbf{A}_p)} + 2 \right) \times \frac{\sigma / B}{\sqrt{F}}. \quad (3.14)$$

For verification, simulations are carried out according to chapter 3.3.1. The fringe patterns are simulated with  $F=3\sim 10$ ; different random phase shifts with  $\kappa(\mathbf{A}_p)=2\sim 50$ ; and  $\sigma/B=10\%\sim 50\%$ . The AIA is used to extract the phases. The RMSEs of the phases are plotted against  $E_{\text{AIA}}$  in Figure 3.7. Clearly, the RMSEs have a great agreement with  $E_{\text{AIA}}$ . Hence, it is possible to use  $E_{\text{AIA}}$  to represent the error in the phase extracted by the AIA.

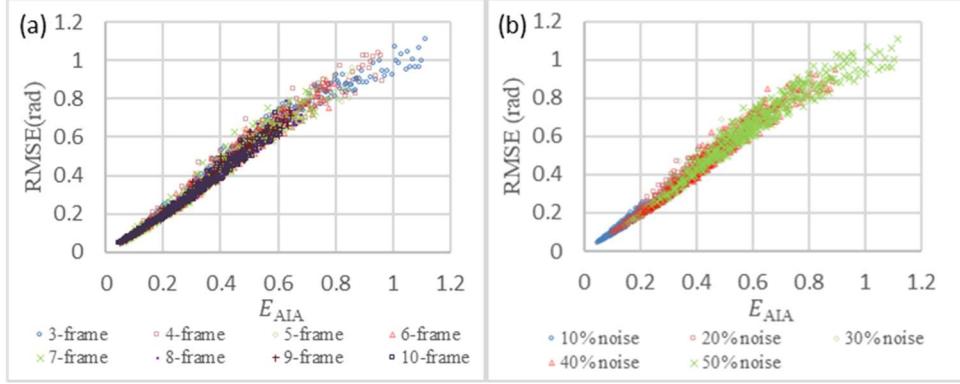


Figure 3.7.RMSEs of the phases extracted from fringe patterns with different frame numbers and different noise levels.

(a) with respect to  $E_{AIA}$  with different  $F$ ; (b) with respect to  $E_{AIA}$  with different  $\sigma/B$ .

### 3.5.AIA's improving practices

Based on the evaluations, we proposed three improving practices.

#### 3.5.1.Controlling the phase shifts

One of the major advantages of the AIA is its capability for phase extraction from fringe patterns with random phase shifts. Hence, controlling the phase shifts is seldomly considered when using the AIA. However, as shown in chapter 3.4.1, smaller  $\kappa(\mathbf{A}_p)$ , *i.e.*, phase shifts which are evenly or nearly evenly allocated among  $[0, 2\pi]$ , will lead to a smaller RMSE for the AIA. Thus, controlling the phase shifts, which is common in phase-shifting interferometry with kPSAs [148], can be utilized as the first improving practice for the AIA's accuracy. Meanwhile, the phase shifts for the AIA can be roughly controlled which contrasts with the traditional kPSAs.

To verify the proposed improving practice of controlling the phase shifts, simulations are carried out according to chapter 3.4.1. Two groups of fringe patterns are used in the simulations. For the first group, the phase shifts are simulated randomly. For the other group, the phase shifts are simulated as  $\delta(i)=2i\pi/F+n_\delta$  where  $n_\delta$  is the phase-shift errors picked randomly from a normal distribution  $N(0, 25)$ , *i.e.*, the standard deviation value of the phase-shift error is 5 degree. In this case,  $\kappa(\mathbf{A}_p)\approx 2$ . We simulated 1,000 sets of fringe patterns for each group. The AIA is used to extract the phases from the two groups of fringe patterns. The RMSEs of the phases are shown in

Figure 3.8 in which the y-axis is on a log scale. Clearly, the accuracy of the AIA is improved by controlling the phase shifts.

Furthermore, we also use the LSA to extract the phase from the fringe patterns in the controlled group. For the LSA, we used the nominal values of phase shifts,  $\delta(i)=2i\pi/F$ . The RMSEs are also shown in Figure 3.8. The LSA's RMSEs are larger than the ones of the AIA. It clearly shows the AIA's advantage on imprecisely controlled phase shifts [198].

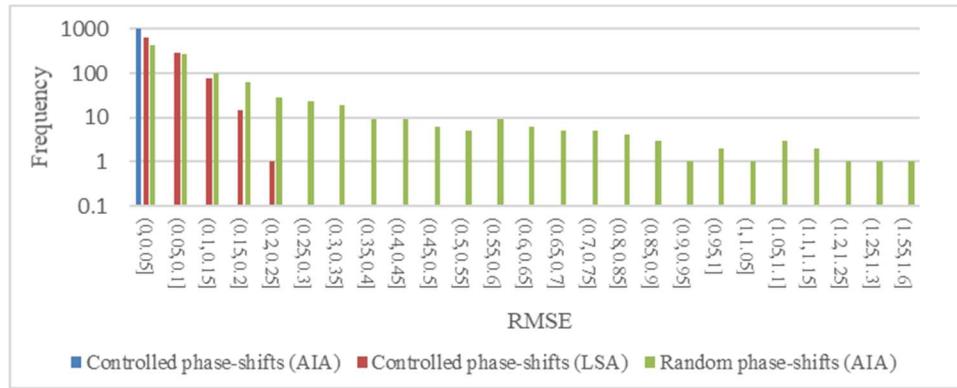


Figure 3.8. Improvement by controlling the phase shifts.

### 3.5.2. Controlling the frame number

When controlling phase shifts are not feasible, random phase shifts should be expected. We can increase the frame number to reduce  $\kappa(\mathbf{A}_p)$ . This is considered as the second improving practice of the AIA. For verification, simulations are carried out according to chapter 3.4.2. The fringe patterns are simulated with  $F=4\sim20$  and random phase shifts. For each frame number, 1,000 sets of fringe patterns are simulated. The mean values of  $\kappa(\mathbf{A}_p)$  of the fringe patterns are plotted in Figure 3.9(a). The AIA is used to extract the phases. The mean values of the RMSEs of the phases are plotted against  $F$  in Figure 3.9(b). Both the mean values of  $\kappa(\mathbf{A}_p)$  and the mean values of the RMSEs reduce with respect to  $F$ . Besides, if regular phase shifts are used, the RMSEs will also reduce with the increased frame number as shown in chapter 3.4.2.

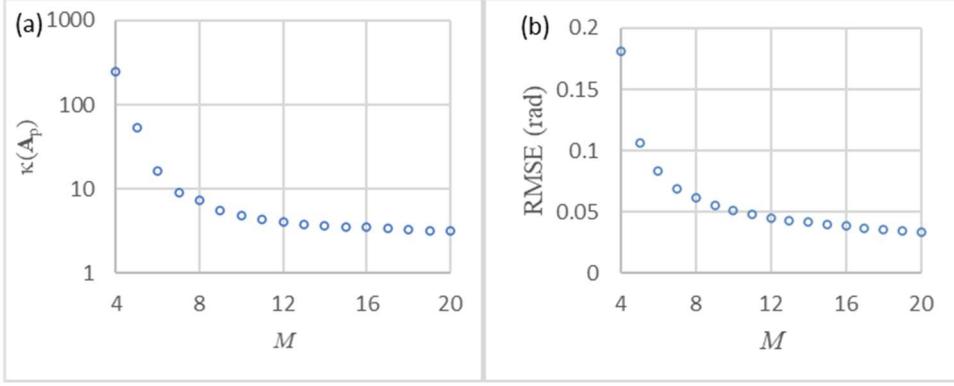


Figure 3.9. Performance with frame number control.

(a) Mean values of  $\kappa(\mathbf{A}_p)$  with respect to  $F$ . (b) Mean values of RMSEs with respect to  $F$ .

### 3.5.3. Suppressing the noise

As shown in chapter 3.4.3, the AIA prefers a lower amplitude of noise. Hence, we suggest using fringe pattern filters before the phase extraction as the third improving practice of the AIA. Meanwhile, the fringe pattern filters should preserve the details of the fringe patterns. Thus, two types of filters are selected. A  $3 \times 3$  median filter (MF) is selected as it is simple and effective for general images. A windowed Fourier filter (WFF) is selected as it is effective for fringe patterns [199]. For verification, simulations are carried out according to chapter 3.4.3. However, before the phase extraction with the AIA, the MF or the WFF are used to denoise the fringe patterns. The RMSEs of the phases extracted from the filtered and unfiltered fringe patterns are plotted against  $\sigma/B$  in Figure 3.10. Both the MF and the WFF improve the accuracy significantly, while the WFF performs better than the MF.

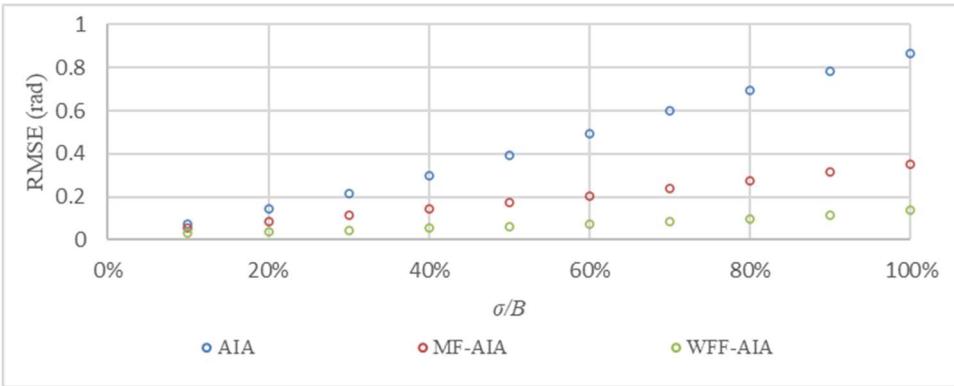


Figure 3.10. Improvement by suppressing the noise.

## 3.6.Error estimation

In chapter 3.4.4, we introduced a general phase error model of the AIA,  $E_{\text{AIA}}$ , which can represent the phase error of the AIA. However, in real applications, the phase shifts, the noise amplitude and the fringe amplitude are not available. In this sub-chapter, methods for estimating these parameters and the phase error are discussed.

### 3.6.1.Phase shifts

Phase shifts estimation is not a difficult task. Different methods such as the ellipse fitting based method [160] and the windowed Fourier transform based method [200] have been proposed. In this sub-chapter, a Fourier transform based phase shifts estimation method is used.

Firstly, we rewrite the fringe pattern shown in Eq. (2.23) as

$$I(i; j) = A(i; j) + C(i; j) + C^*(i; j), \quad (3.15)$$

with

$$C(i; j) = \frac{1}{2} B(i; j) \exp\left\{j[\varphi(j) + \delta(i)]\right\} = \frac{1}{2} B(i; j) \exp[j\delta(i)] \exp[j\varphi(j)], \quad (3.16)$$

where the superscript \* represents a complex conjugate and  $\mathbf{j}$  is the imaginary unit. As  $\delta(0)=0$ , we can get

$$C(i; j) = C(0; j) \exp[j\delta(i)], \quad (3.17)$$

which means that  $C(i; j)$  is directly related to the  $\delta(i)$ .

Next, we need to separate  $C(i; j)$  from  $C^*(i; j)$  and  $A(i; j)$ . The 2D Fourier transform of Eq. (3.15) is

$$F[I(\xi_x, \xi_y; i)] = F[A] + F[C(\xi_x, \xi_y; i)] + F[C^*(\xi_x, \xi_y; i)], \quad (3.18)$$

where  $\xi_x$  and  $\xi_y$  are the frequency coordinates in row and column directions, respectively. The  $C(i; j)$  could be separated from the  $C^*(i; j)$  and  $A(i; j)$  by identifying the spectrum lobe ridge of  $F[C(\xi_x, \xi_y; i)]$  as

$$(u, v) = \arg \max_{\xi_x, \xi_y} \sum_{i=0}^{F-1} |F[C(\xi_x, \xi_y; i)]|. \quad (3.19)$$

From Eq. (3.17), we can estimate the phase shifts as

$$\hat{\delta}(i) = \angle F[C(u, v; i)] - \angle F[C(u, v; 0)]. \quad (3.20)$$

Subsequently, the estimated phase shifts are used to form the least-squares matrix for phase calculation ( $\hat{\mathbf{A}}_p$ ). The condition number of this matrix,  $\kappa(\hat{\mathbf{A}}_p)$ , can be calculated.

### 3.6.2.Fringe amplitude

For the estimation of the fringe amplitude, we directly use the LSA [151]. In the last step, the phase shifts are estimated, thus, we can calculate  $\mathbf{A}_p$  and  $\mathbf{B}_p$  used in Eq. (2.25) and the fringe amplitude of every pixel can be estimated as

$$\hat{B}(j) = \sqrt{b^2(j) + c^2(j)}. \quad (3.21)$$

By averaging over all the pixels, we can get and the mean fringe amplitude  $\hat{B}$  as,

$$\hat{B} = \frac{1}{N} \sum_{j=0}^{N-1} \hat{B}(j). \quad (3.22)$$

### 3.6.3.Noise amplitude

To estimate the noise amplitude, we use a  $3 \times 3$  Laplacian mask based estimation algorithm [201]. This noise estimating algorithm can be used for unfiltered and MF filtered fringe patterns. The noise level of each fringe pattern is estimated as

$$\hat{\sigma}(i) = \sqrt{\frac{2}{\pi}} \frac{1}{6(N_x - 2)(N_y - 2)} \sum_{I_i^d} |I^d(i) \otimes N|, \quad (3.23)$$

where  $| \cdot |$  calculates the absolute value,  $\otimes$  represents the convolution operator,

$$N = \begin{bmatrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{bmatrix}, \quad (3.24)$$

and  $I^d$  is a down sampled version of the native fringe pattern to minimize the influence on neighboring pixels due to the MF. Subsequently, the noise of the set of fringe patterns can be calculated as

$$\hat{\sigma} = \frac{1}{F} \sum_{i=0}^{F-1} \hat{\sigma}(i). \quad (3.25)$$

### 3.6.4.Phase error

Now with all the fringe parameters estimated, the phase error of the AIA can be estimated as

$$\hat{E}_{\text{AIA}} = 0.42 \left( \sqrt{\kappa(\hat{\mathbf{A}}_p)} + 2 \right) \times \frac{\hat{\sigma} / \hat{B}}{\sqrt{F}}. \quad (3.26)$$

For verification, simulations are carried out according to chapter 3.4.4. Meanwhile, instead of computing  $E_{\text{AIA}}$ ,  $\hat{E}_{\text{AIA}}$  is calculated for each set of fringe patterns. The RMSEs of the phases extracted by the AIA are plotted against  $\hat{E}_{\text{AIA}}$  in Figure 3.11. Excellent agreement is observed.

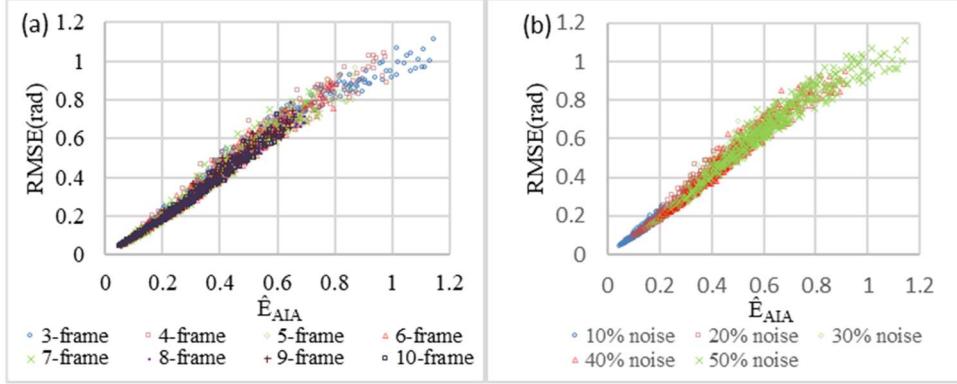


Figure 3.11. Prediction of the AIA's accuracy.

(a) RMSEs with respect to  $\hat{E}_{\text{AIA}}$  with different  $F$ ; (b) RMSEs with respect to  $\hat{E}_{\text{AIA}}$  with different  $\sigma$ .

The  $\hat{E}_{\text{AIA}}$  can also be used to control the AIA extracted phase's accuracy through controlling the fringe parameters. When controlling the phase shifts is feasible,  $\kappa(\mathbf{A}_p) \approx 2$ . We can use the phase error model to estimate the required frame number for the desired phase accuracy as

$$F = \text{ceil} \left[ 2 \left( \frac{\hat{\sigma} / \hat{B}}{\tau} \right)^2 \right], \quad (3.27)$$

where  $\tau$  is the tolerance for the RMSE of the extracted phase.

### 3.7.The Enhanced AIA (eAIA)

In previous sub-chapters, the fringe density requirement of the AIA, the improving practices of the AIA and the practical phase error estimation method of the AIA are discussed. By combining these findings, we propose an eAIA.

Two major issues of the AIA are that it may fail in practice when the fringe density is low, and it has uncertain accuracy. To solve these problems, the eAIA is proposed by combing fringe

patterns acquisition with the AIA based phase extraction. The eAIA's flowchart is shown in Figure 3.12. The detailed highlights are as follows:

- (i) When the fringe density is low, the AIA may fail. Thus, in the eAIA, we confirm the fringe pattern is dense enough for the AIA's success by hardware adjustments such as tilting the reference or test surface;
- (ii) Controlling the phase shifts is encouraged in the eAIA for phase extraction with better accuracy and fewer frames. It is worth mentioning that the phase shifts could be roughly controlled, thus, the implementation is not hard;
- (iii) Adaptive fringe pattern acquisition is implemented in the eAIA for the desired accuracy. The error model in Eq. (3.26) is used to predict the accuracy of the phase after each fringe pattern acquisition and the eAIA will stop only when the desired accuracy is reached. It is worth mentioning that this process is automatic;
- (iv) In the eAIA, any type of fringe pattern filter could be used as long as it can effectively remove the noise and the noise level can be estimated after the filtering. For the eAIA shown in Figure 3.12, we used a  $3 \times 3$  MF.

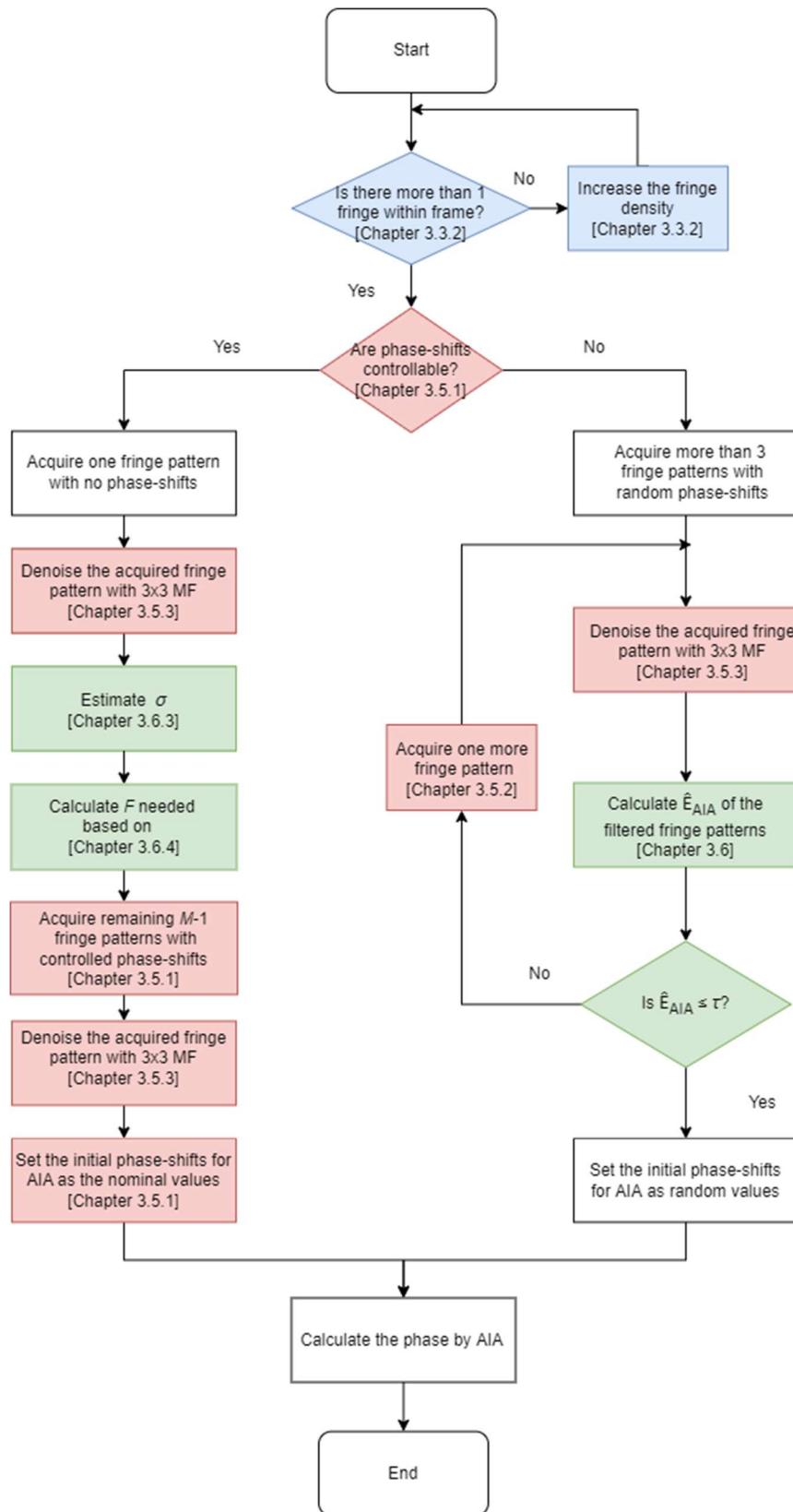


Figure 3.12.Flowchart of eAIA.

### 3.8.Experimental verification

For verification of the eAIA's effectiveness, optical experiments with a Fizeau interferometer were conducted. In this Fizeau interferometer, both the reference and test object are wedged optical windows. We change the wavelength of the laser to introduce phase shifts [202]. We firstly collect 160 fringe patterns. The image size is  $N=N_x \times N_y = 1000 \times 992$  and the phase shifts are random values. The AIA is firstly applied to the set of 160 fringe patterns. This phase is used as the ground true phase of the experiment. The first frame of the fringe patterns is shown in Figure 3.13(a) and the ground true phase is shown in Figure 3.13(b).

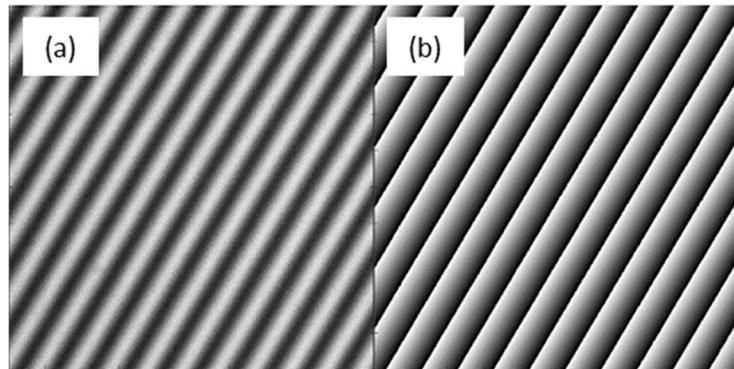


Figure 3.13.Experiment data.

(a) First frame of fringe patterns; (b) The ground-true phase.

The accuracy threshold for this experiment is set as 0.04 rad, *i.e.*,  $\tau=0.04$ . We use the first frame of the fringe patterns and two randomly selected fringe patterns to start the eAIA and randomly select one more fringe pattern if the accuracy threshold is not satisfied. In our experiment, seven fringe patterns are needed for the desired phase accuracy. Following the flowchart of the eAIA, the phase should only be extracted once after the collection of all the seven fringe patterns. Meanwhile, for monitoring the phase error reduction, the phase extraction with the AIA is repeatedly carried out. For each frame number, both  $\hat{E}_{\text{AIA}}$  and the RMSEs of the phase are calculated and plotted against the frame number in Figure 3.14(a). The phase shifts estimated by the AIA from the seven fringe patterns are shown in Figure 3.14(b) by plotting complex numbers whose length is 1 and angle is the phase shifts estimated by the AIA from the seven fringe patterns on a unit circle where the bold numbers represent the fringe pattern index.

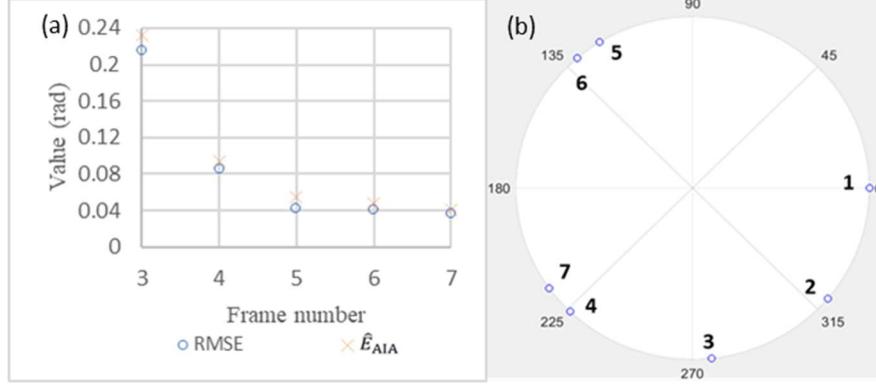


Figure 3.14.Performance of eAIA.

(a) RMSE of phases and  $\hat{E}_{\text{AIA}}$  with respect to frame number; (b) Estimated phase shifts of the 7 frames.

A few observations are noticed

- (i) The  $\hat{E}_{\text{AIA}}$  can effectively predict the RMSEs of phase extracted by the AIA;
- (ii) As indicated by the influence of the frame 4 and frame 5, if the phase shift of a fringe pattern makes the phase-shift distribution more evenly allocated among  $[0, 2\pi]$ , this fringe pattern can reduce the AIA's phase error significantly;
- (iii) On the contrary, as indicated by the influence of the frame 6 and frame 7, if the phase shift of a fringe pattern does not improve the phase-shift distribution's evenness, this fringe pattern will mildly reduce the phase error.

These observations are confirmed by more experiments with different sets of fringe patterns. Meanwhile, for the experiment discussed in Figure 3.14, the phases extracted by the AIA from three and seven frames of fringe patterns are shown in Figure 3.15(a) and Figure 3.15(b), respectively.

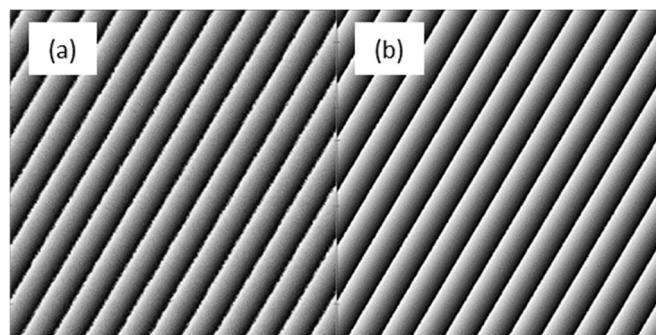


Figure 3.15.Extracted phases.

- (a) Phase extracted with 3 fringe patterns; (b) Phase extracted with 7 fringe patterns.

### **3.9. Summary**

In this chapter, we focused on the evaluation and enhancement of the robustness and the accuracy of the AIA. To evaluate the AIA, a large number of simulations and theoretical evaluations were carried out. To characterize the phase and phase shifts, condition numbers of the least-squares matrices,  $\kappa(\mathbf{A}_{ps})$  and  $\kappa(\mathbf{A}_p)$ , were utilized. Based on the simulations and condition number based theoretical evaluations, we confirmed that more than one fringe over the frame should be maintained for the AIA's success. Besides, the simulations showed that the RMSEs of the AIA are linearly correlated to  $\kappa(\mathbf{A}_p)^{1/2}$ ,  $F^{-1/2}$  and  $\sigma$ . With our practical parameter estimation methods and the phase error model summarized from the simulations, we can predict the accuracy of the AIA. Finally, by combining fringe patterns acquisition with the AIA based phase extraction, we proposed an eAIA for AIA's accuracy and robustness enhancement.



# **Chapter 4 General iterative algorithm for phase extraction from fringe patterns with random phase shifts, intensity harmonics and nonuniform phase-shift distribution\***

## **4.1. Background**

The robustness and the accuracy performance of the AIA have been evaluated in detail in the last chapter. Meanwhile, an enhanced AIA has been proposed for better robustness and better accuracy when there is random noise in the fringe pattern. However, the phase extraction process is also challenged by other error sources such as the intensity harmonics and/or phase-shift nonuniformities. To suppress the impact of these error sources, different algorithms have been proposed as discussed in chapter 2.2.3. However, none of these algorithms considers all the common error sources in its fringe patterns. Thus, these algorithms will have compromised phase accuracy in certain conditions. To achieve a good phase extraction accuracy with the presence of all the common error sources mentioned in chapter 1.3.1, we proposed a general iterative algorithm (GIA). Firstly, we established a fringe pattern model incorporating all the mentioned errors; Next, we divided the unknowns into three groups, *i.e.*, (i) the background intensity and the fringe amplitude, (ii) the phase, and (iii) the phase shifts related parameters, and then alternatively optimized them through Levenberg-Marquart method. Finally, to improve the immunity of the GIA to random noise, neighborhood pixels are also considered in the background intensity and fringe amplitude estimation step. We carried out simulations and optical experiments to compare the accuracy of the GIA and the other optimization based uPSAs discussed in chapter 2.2.3.

---

\* The work in this chapter has been published as Y. Chen and Q. Kemo, "General iterative algorithm for phase-extraction from fringe patterns with random phase-shifts, intensity harmonics and non-uniform phase-shift distribution," *Optics Express*, vol. 29, no. 19, pp. 30905-30926, 2021.

## 4.2.General fringe pattern model

The fringe pattern models used for different algorithms have been discussed in Chapter 2. However, none of these fringe pattern model considers all the error sources. In this sub-chapter, we set up the general fringe pattern model for the GIA.

In real measurements, random noise is inevitable. Thus, we assume that phase-shifted fringe patterns consisting two parts, a noiseless part  $I^t$  and a noise part  $n$ , as [192]

$$I(x, y; i) = I^t(x, y; i) + n(x, y; i). \quad (4.1)$$

The noiseless part considers the intensity harmonics and the phase-shift nonuniformities as

$$I^t(x, y; i) = A(x, y; i) + \sum_{k=1}^P B_k(x, y; i) \cos[k\varphi(x, y) + k\delta(x, y; i)], \quad (4.2)$$

where  $P$  represents the maximum order of the harmonics,  $k$  represents the intensity harmonics order index and  $B_k$  represents the fringe amplitude of the  $k$ th order intensity harmonic. In this model, the phase shifts are assumed to be with pixel-to-pixel variations due to the phase-shift nonuniformities. We use the Maclaurin polynomial to represent the pixel-frame-dependent phase shifts as

$$\delta(x, y; i) = \sum_{u=0}^Q \sum_{v=0}^u \frac{\alpha_{uv}(i) x^v y^{u-v}}{(N_x - 1)^v (N_y - 1)^{u-v}}, \quad (4.3)$$

where  $v$  and  $u-v$  are the order of the 2D polynomials,  $Q$  is the highest order of the polynomials and  $\alpha_{uv}$  are the coefficients of the 2D polynomials.

## 4.3.The GIA

With the fringe pattern model, we now focus on phase extraction.

### 4.3.1.The phase extraction strategy of the AIA

The fringe pattern model considers all the common error sources. However, this model is not usable in phase extraction. For  $F$  frames of fringe patterns with  $N$  pixels, we have  $FN$  known fringe pattern intensities but also have more than  $(PF+F+1) \times N$  unknowns (for the phase, background intensity and fringe amplitudes) in the model. As did in all the algorithms reviewed in chapter 2,

we continue to assume that there is no frame to frame variations in the background intensity and fringe amplitudes. This assumption is reasonable in practice. Hence, we can rewrite Eq. (4.2) as

$$I^t(x, y; i) = A(x, y) + \sum_{k=1}^P B_k(x, y) \cos[k\varphi(x, y) + k\delta(x, y; i)]. \quad (4.4)$$

Thus, the optimization problem becomes a well-posed problem when  $F > (P+2)$ .

When focusing on phase extraction, the major challenge is how to estimate the many highly coupled unknowns reliably and accurately from the highly complex and nonlinear fringe model. We propose the GIA to solve this problem, with the following novel ingredients:

- (i) Due to the complexity of the model and the unknown phase shifts, phase extraction with simple trigonometric calculations will not be feasible; furthermore, a linear least-squares based iterative algorithm similar to the AIA is also not suitable. Instead, we propose to adopt a general optimization strategy and set a straightforward objective function as,

$$E = \sum_{i,j} [I^t(x, y; i) - I(x, y; i)]^2, \quad (4.5)$$

- (ii) Clearly, the above optimization problem must deal with many unknowns including  $(P+1) \times N$  fringe amplitude related parameters,  $N$  phase parameters and  $(2Q+1) \times (F-1)$  phase parameters in total  $(P+2) \times N + (2Q+1) \times (F-1)$  parameters. Optimizing all the unknowns simultaneously will be sensitive to random noise and easy to fail due to the long size optimization vector [194]. Thus, it is not suggested in practice. Instead, we propose to divide the unknowns into groups and to solve the optimization problem group by group. Based on a careful examination of the fringe pattern model, we divide the unknowns into three groups: (a) group **B** which contains the background intensity  $A(x, y)$  and fringe amplitudes  $B_k(x, y)$ , (b) group **P** which contains phase  $\varphi(x, y)$ , and (c) group **S** which contain phase-shift related parameters  $\alpha_{uv}(i)$ . Next, each group of unknowns will be optimized with the assumption the other two groups of unknowns are known and fixed.

- (iii) In the optimization of group **B**, for improving the immunity of the GIA to the random noise, we include neighboring pixels to enhance the signal-to-noise ratio. Hence, the objective function in Eq.(4.5) will be used in this step. Meanwhile, in the optimization of the phase, neighboring pixels are not considered as only parameters in group **B** have slow spatial variation.

- (iv) In the optimization of group **P**, each  $\varphi(x, y)$  at a different pixel contributes to the objective function independently, thus, it simplifies the objective functions as

$$E_p(x, y) = \sum_i [I^t(x, y; i) - I(x, y; i)]^2. \quad (4.6)$$

(v) In the optimization of group  $\mathbf{S}$ , the  $\alpha_{uv}(i)$  are independent frame-wisely, which simplifies the objective function as

$$E_F(i) = \sum_{x,y} [I^t(x, y; i) - I(x, y; i)]^2. \quad (4.7)$$

(vi) For the optimization of each group, the Levenberg-Marquart method (LM) is used for its good accuracy and robustness. The pseudocode of an LM optimization example is shown in Figure 4.1. In this example, we optimize the unknown vector  $\mathbf{x}$  whose entries are  $x_q$ , by minimizing the optimization function of

$$f(\mathbf{x}) = \frac{1}{2} \|\mathbf{r}(\mathbf{x})\|^2 = \frac{1}{2} \sum_p [r_p(\mathbf{x})]^2, \quad (4.8)$$

where  $\mathbf{r}(\mathbf{x})$  is the residual vector whose entries are  $r_p(\mathbf{x})$ .

```

Begin:
    Initialize  $M_{LM}, \tau, \eta$ 
     $m_{LM} := 0; v := 2;$ 
     $[J(\mathbf{x})]_{pq} = \frac{\partial r_p}{\partial x_q}(\mathbf{x}); \mathbf{A}_{LM} := \mathbf{J}^T(\mathbf{x})\mathbf{J}(\mathbf{x}); g := \mathbf{J}^T(\mathbf{x})\mathbf{r}(\mathbf{x});$ 
     $\mu := \max[(\mathbf{A}_{LM})_{ii}];$ 
WHILE (not found) AND ( $m_{LM} < M_{LM}$ )
    Solve  $(\mathbf{A}_{LM} + \mu) \times h_{LM} = -g;$ 
     $\mathbf{x}_{new} := \mathbf{x} + h_{LM};$ 
     $Q := \frac{[\mathbf{r}(\mathbf{x}) - \mathbf{r}_{new}(\mathbf{x})] \times [\mathbf{r}(\mathbf{x}) + \mathbf{r}_{new}(\mathbf{x})]}{h_{LM} \times (\mu h_{LM} - g)};$ 
    IF  $Q > 0$ 
         $\mathbf{x} := \mathbf{x}_{new}; \mu := \mu \times \max\left[\frac{1}{3}, 1 - (2Q - 1)^3\right]; v := 2;$ 
         $found := \|\mathbf{J}^T(\mathbf{x})\mathbf{r}(\mathbf{x})\|_\infty < \eta;$ 
    ELSE
         $\mu := \mu \times v; v := 2 \times v;$ 
    END IF
     $[J(\mathbf{x})]_{pq} = \frac{\partial r_p}{\partial x_q}(\mathbf{x}); \mathbf{A}_{LM} := \mathbf{J}^T(\mathbf{x})\mathbf{J}(\mathbf{x}); g := \mathbf{J}^T(\mathbf{x})\mathbf{r}(\mathbf{x});$ 
     $m_{LM} := m_{LM} + 1;$ 
END WHILE
END
```

Figure 4.1.Pseudocode of LM algorithm

### 4.3.2.The implementation of the GIA

In order to make the GIA programable, the overall flowchart is shown in Figure 4.2 followed by details of the algorithm.

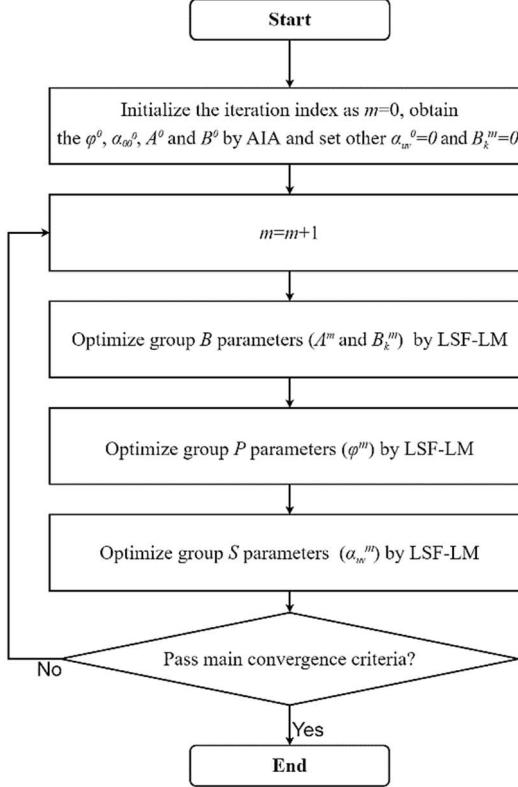


Figure 4.2.Flowchart of GIA.

#### 4.3.2.1.Optimizing group B

Firstly, we assume that the fringe amplitudes have no pixel to pixel variation within a  $(2w+1) \times (2w+1)$  window which centered at the target pixel  $(x,y)$  as

$$A(x-w, y-w) = \dots = A(x, y) = \dots = A(x+w, y+w), \quad (4.9)$$

$$B_k(x-w, y+w) = \dots = B_k(x, y) = \dots = B_k(x+w, y+w). \quad (4.10)$$

Next, we use the LM to estimate background intensity and fringe amplitudes of the target pixel  $(x,y)$  with the known phase and phase shifts. In this step, the vector of unknown, the vector of residual and the Jacobian matrix  $\mathbf{J}(\mathbf{x})$  are set up as

$$x_k = \begin{cases} A(x, y) & \text{if } k = 0, \\ B_k(x, y) & \text{if } k = 1, \dots, P, \end{cases} \quad (4.11)$$

$$r_{(2w+1) \times (2w+1) \times i+l}(\mathbf{x}) = I^t(x+s, y+t; i) - I(x+s, y+t; i), \quad (4.12)$$

$$J_{(2w+1) \times (2w+1) \times i+l, k}(\mathbf{x}) = \cos[k\varphi(x+s, y+t) + k\delta(x+s, y+t; i)], \quad (4.13)$$

where  $l=0, 1, \dots, (2w+1) \times (2w+1)-1$  represents the index of the pixels in the window.

#### 4.3.2.2.Optimizing group P

Secondly, with the calculated fringe amplitude and the known phase shifts, the phase of each pixel is updated by the LM. In this step, the vector of unknown, the vector of residual and the Jacobian matrix  $\mathbf{J}(\mathbf{x})$  are set up as

$$\mathbf{x} = \varphi(x, y), \quad (4.14)$$

$$r_i(\mathbf{x}) = I^t(x, y; i) - I(x, y; i), \quad (4.15)$$

$$J_{i,1}(\mathbf{x}) = \frac{\partial r_i(\mathbf{x})}{\partial \mathbf{x}} = \sum_{k=1}^P -kB_k(j) \sin[k\varphi(x, y) + k\delta(x, y; i)], \quad (4.16)$$

with  $i=0, 1, \dots, F-1$ .

#### 4.3.2.3.Optimizing group S

Thirdly, with the calculated fringe amplitude and phase, the phase-shift related parameters are updated frame-wisely by the LM. In this step, the vector of unknown, the vector of residual and the Jacobian matrix  $\mathbf{J}(\mathbf{x})$  are set up as

$$x_{u(u+1)/2+v} = \alpha_{uv}(i), \quad (4.17)$$

$$r_j(\mathbf{x}) = I^t(x, y; i) - I(x, y; i), \quad (4.18)$$

$$J_{j,u(u+1)/2+v}(\mathbf{x}) = \frac{\partial r_j(\mathbf{x})}{\partial x_{u(u+1)/2+v}} = \sum_{k=1}^P \frac{-kx^v y^{u-v} B_k(x, y) \sin[k\varphi(x, y) + k\delta(x, y; i)]}{(N_x - 1)^v (N_y - 1)^{u-v}}, \quad (4.19)$$

with  $j=0, 1, \dots, N-1$ ,  $u=0, 1, \dots, Q$  and  $v=0, 1, \dots, u$ .

#### 4.3.2.4.Convergence check

The optimizations of the three groups are iteratively executed. The GIA initializes the background intensity, fringe amplitude, phase and phase shifts by the AIA with fringe amplitudes of all higher

order harmonics and phase-shift nonuniformities parameters  $\alpha_{uv}$  ( $u \neq v \neq 0$ ) set to zero. Meanwhile, the GIA stops when the iteration number exceeds the higher limit  $M$ , *i.e.*,

$$m > M, \quad (4.20)$$

or when the absolute change of the objective function shown in Eq. (4.5) is sufficiently small, *i.e.*

$$|E^{m+1} - E^m| / E^m < \zeta, \quad (4.21)$$

where superscript  $m$  is the index of the iteration.

## 4.4.Simulation evaluation

Fringe patterns with different errors are simulated to compare the performance of the GIA, the AIA, Xu's algorithm, Hoang's algorithm, the PTT and the MPSI. In this sub-chapter, the fringe pattern simulation strategies are introduced first. Next, the accuracy performance of the GIA is compared with the other algorithms.

### 4.4.1.Algorithm specifications

The existing optimization based uPSAs are coded according to the description in chapter 2.2.3. The GIA is coded according to the description in chapter 4.3. The settings of the algorithms are shown in Table 4.1. Meanwhile, we set (i)  $\eta=1\times10^{-4}$  (ii)  $M_{LM}=200$  (iii)  $w=8$  and (iv)  $Q=4$  for the GIA.

Table 4.1.Settings of different algorithms.

	AIA	Xu's algorithm	Hoang's algorithm	PTI	MPSI	GIA
M	200					
$\varepsilon$ (rad)	$1\times10^{-4}$	$1\times10^{-4}$	$1\times10^{-4}$	-	-	-
$\zeta$	-	-	-	-	0.01%	0.01%
P	-	3	3	-	3	3
$\varepsilon_1 \varepsilon_2$ (rad)	-	-	-	$1\times10^{-5}$	-	-

### 4.4.2.Simulated fringe pattern

In the simulation comparison, we will discuss the performance of two examples, the random phase shifts example and the regular phase shifts example. In each example, ten sets of fringe patterns are simulated. For Set I, which is the first set of fringe patterns, phase-shifted fringe patterns are simulated ideally as,

$$I_o(x, y; i) = A(x, y) + B(x, y) \cos[\varphi(x, y) + \delta_0(i)], \quad (4.22)$$

where  $\delta_0$  is called as bias phase shifts which represent phase shifts without tilts or nonuniformities,  $A(x, y)=0.5$ ,  $B(x, y)=0.4$  and the phase

$$\varphi(x, y) = 10 \times \text{peaks\_s}(N_x, N_y) + 0.5y, \quad (4.23)$$

Here, the `peaks_s` scales the “peaks” function in MATLAB into  $[0, 2\pi]$ .

Next, we simulate different sets of noisy fringe patterns in Set II to Set IX. The error sources added are shown in For both examples, we set the image size as  $N=1024 \times 1024$ , *i.e.*,  $N_x=N_y=1024$  and set the frame number as eight, *i.e.*,  $F=8$ . Meanwhile, one example is with random bias phase shifts as

$$\delta_0 = [0 \ 0.440 \ 0.795 \ 1.246 \ 1.601 \ 2.665 \ 2.695 \ 3.449]. \quad (4.25)$$

The other example is with regular bias phase shifts as

$$\delta_0(i) = i\pi / 4. \quad (4.26)$$

These simulated bias phase shifts and the simulated phase will be provided to the MPSI for initialization.

Table 4.2, where a tick corresponds to the presence of the error source. The random noise is added as additive white gaussian noise with a zero mean value and a standard deviation value of  $\sigma_r=0.04$ . The intensity harmonics are added using a gamma model, *i.e.*  $I_o^\gamma$  represents the fringe patterns with nonlinearities [203] with  $\gamma=1.5$ . The phase shifts with tilt or nonuniformities are simulated according to Eq. (4.3) with  $Q=1$  or  $Q=4$  and  $\alpha_{vu}$  are simulated as random values with a zero mean value and a standard deviation value of  $\sigma_\alpha=0.5$  with  $\alpha_{00}=\delta_0$ . Meanwhile, spatial variations for the background intensity and fringe amplitude are added to the Set IX for the generation of Set X as

$$A(x, y) = 0.5 - 0.25 \left[ \frac{(x - N_x / 2)^2 + (y - N_y / 2)^2}{(N_x / 2)^2 + (N_y / 2)^2} \right], \quad (4.24)$$

and  $B(x, y)=0.8 \times A(x, y)$ . For these ten sets of fringe patterns, we classified them into four categories which are Category (A) without intensity harmonics or phase-shift nonuniformities, Category (B) with intensity harmonics only, Category (C) with phase-shift nonuniformities only, and Category (D) with both intensity harmonics and phase-shift nonuniformities.

For both examples, we set the image size as  $N=1024\times 1024$ , i.e.,  $N_x=N_y=1024$  and set the frame number as eight, i.e.,  $F=8$ . Meanwhile, one example is with random bias phase shifts as

$$\delta_0 = [0 \ 0.440 \ 0.795 \ 1.246 \ 1.601 \ 2.665 \ 2.695 \ 3.449]. \quad (4.25)$$

The other example is with regular bias phase shifts as

$$\delta_0(i) = i\pi/4. \quad (4.26)$$

These simulated bias phase shifts and the simulated phase will be provided to the MPSI for initialization.

Table 4.2.Simulated fringe patterns.

Category	Set	Phase-shift errors			
		Random noise	Harmonics	Tilt (Q=1)	Nonlinear (Q=4)
A	I				
	II	✓			
B	III		✓		
	IV	✓	✓		
C	V			✓	
	VI			✓	
D	VII	✓		✓	
	VIII		✓	✓	
D	IX	✓	✓	✓	
	X	✓	✓	✓	

For illustration, some of the parameters used for the simulations are illustrated in Figure 4.3.

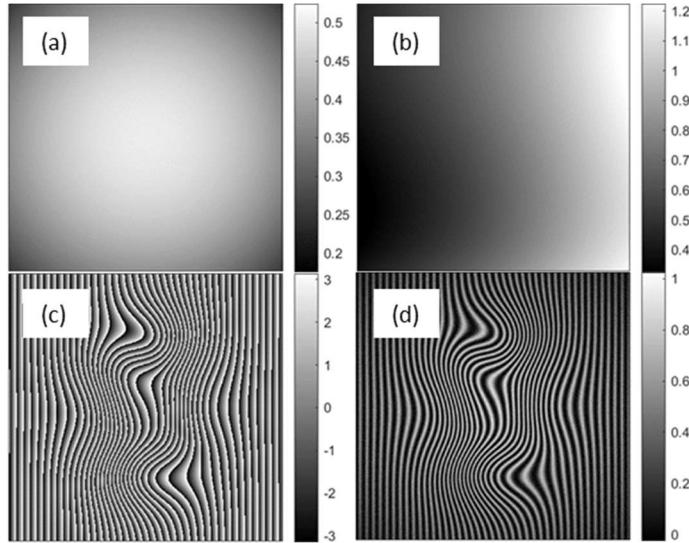


Figure 4.3.Parameters for simulation.

(a) Background intensity  $A$ ; (b) Second frame's phase shift; (c) Ground-true phase; (d) First fringe pattern from Set X.

#### 4.4.3.Example with random bias phase shifts

The phases are extracted by the mentioned algorithms from the ten sets of fringe patterns. The root means square error (RMSE) is used to represent the phase error. The RMSEs of the phases are shown in Figure 4.4 and numerically in Table 4.3. The four categories of simulations are color coded as Category (A) with pink color, Category (B) with green color, Category (C) with blue color and Category (D) with yellow color.

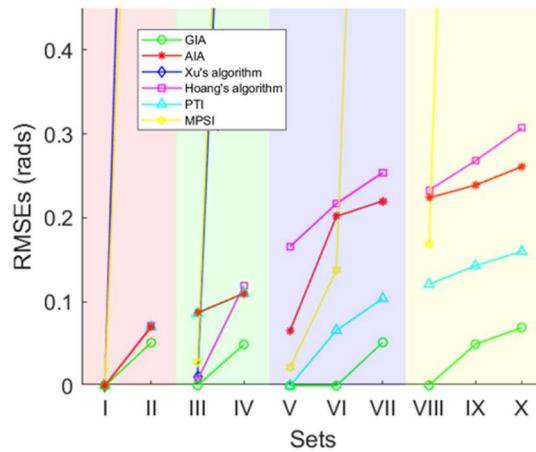


Figure 4.4.Phase errors of example with random bias phase shifts.

Table 4.3.RMSEs from example with random bias phase shifts (Unit: rad).

Sets	I	II	III	IV	V	VI	VII	VIII	IX	X
GIA	$9.9 \times 10^{-8}$	0.051	$4.3 \times 10^{-4}$	0.049	$3.3 \times 10^{-5}$	$3.2 \times 10^{-5}$	0.052	$6.5 \times 10^{-4}$	0.049	0.069
AIA	$1.4 \times 10^{-5}$	0.070	0.087	0.110	0.066	0.202	0.220	0.224	0.239	0.261
Xu's algorithm	$3.4 \times 10^{-6}$	1.796	0.011	1.253	1.621	1.243	1.862	1.795	1.584	1.726
Hoang's algorithm	$2.9 \times 10^{-5}$	0.071	0.007	0.119	0.166	0.217	0.254	0.233	0.268	0.307
PTI	$8.8 \times 10^{-6}$	0.070	0.086	0.110	$6.4 \times 10^{-5}$	0.066	0.104	0.121	0.143	0.160
MPSI	$6.5 \times 10^{-5}$	1.400	0.028	1.366	0.022	0.138	1.620	0.169	1.792	1.651

For category (A) with neither intensity harmonic nor phase-shift nonuniformity, the performances of all the algorithms are similar for ideal fringe patterns; while if the random noise is presented, Xu's algorithm, Hoang's algorithm and the MPSI result large phase error which agrees with the discussion in chapter 2.2.3.6. For category (B), Xu's algorithm, Hoang's algorithm and the MPSI perform better than the AIA and the PTI for random noiseless case, but such improvement diminishes when noise is added. For category (C), the accuracy of the PTI is better than the AIA, which shows the effectiveness of the phase-tilts immune algorithms. The MPSI also shows good immunity to phase-shift tilts if random noise is not presented. For category (D) with

both intensity harmonics and phase-shift nonuniformities, all algorithms' RMSEs increase. For all the categories, the GIA is with the most outstanding performance.

#### 4.4.4.Example with regular bias phase shifts

The bias phase shifts used in the above example are random. Meanwhile, in lots of conditions the bias phase shifts could be calibrated to be regular values. [174]. Thus, the example with regular bias phase shifts is carried out. The RMSEs of the phase extracted are shown in Figure 4.5 and Table 4.4. We conclude a few new findings including (i) Regular phase shifts will lead to small RMSEs for all algorithms; (ii) the MPSI performs well when the phase shifts are regular, as claimed in [174]; and (iii) the GIA is with the smallest RMSEs.

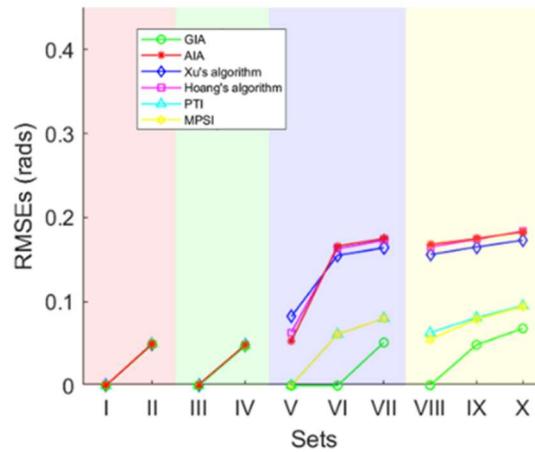


Figure 4.5.Phase errors of example with regular bias phase shifts.

Table 4.4.RMSEs from example with regular bias phase shifts (Unit: rad).

Sets	I	II	III	IV	V	VI	VII	VIII	IX	X
GIA	$6.6 \times 10^{-8}$	0.049	$6.4 \times 10^{-6}$	0.047	$1.4 \times 10^{-6}$	$7.3 \times 10^{-6}$	0.056	$4.3 \times 10^{-4}$	0.049	0.068
AIA	$1.2 \times 10^{-5}$	0.050	0.001	0.048	0.053	0.166	0.175	0.168	0.175	0.183
Xu's algorithm	$7.8 \times 10^{-7}$	0.050	$2.0 \times 10^{-5}$	0.048	0.082	0.155	0.164	0.156	0.165	0.173
Hoang's algorithm	$5.1 \times 10^{-5}$	0.050	$4.5 \times 10^{-5}$	0.048	0.063	0.163	0.173	0.165	0.174	0.184
PTI	$2.4 \times 10^{-6}$	0.050	$2.0 \times 10^{-5}$	0.048	$1.6 \times 10^{-4}$	0.061	0.080	0.063	0.081	0.095
MPSI	$1.3 \times 10^{-5}$	0.050	$6.6 \times 10^{-4}$	0.048	$3.0 \times 10^{-5}$	0.061	0.080	0.055	0.079	0.094

#### 4.5.Experimental verification

To furtherly confirm the performance of the GIA, optical experiments are carried out. The accuracy of the phase extracted by the GIA is compared with the phases extracted by other algorithms. In

the experiments, the intensity harmonics and the phase-shift nonuniformities are proposedly introduced.

#### 4.5.1. Set up and ground truth

We carried out experiments with a Fizeau interferometer as shown Figure 4.6. We change the wavelength of the laser to introduce phase shifts [202]. The distance between the wedged window and the wafer is 25mm and the wavelength of the laser is 635nm.

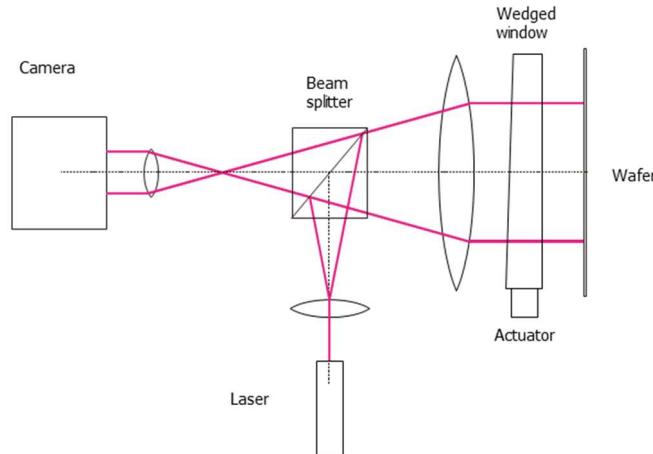


Figure 4.6. Sketch of the Fizeau interferometer.

Seventeen phase-shifted fringe patterns with image sizes of  $N=1024\times 1024$  are first collected with phase shifts of  $\pi i/4$ . Thus, the wavelength change between two adjacent frames is about 1pm. The first fringe pattern is shown in Figure 4.7(a). We use the AIA to extract the phase from the seventeen fringe patterns. This phase is shown in Figure 4.7(b) and is treated as the ground truth of the phase.

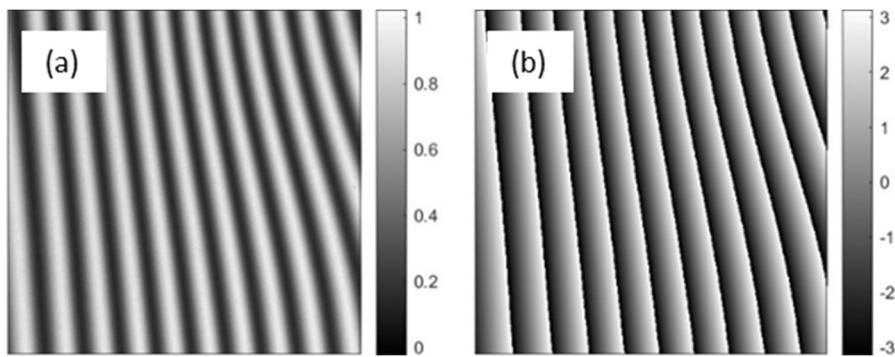


Figure 4.7. Experimental parameters.

(a) First fringe pattern; (b) The phase ground truth.

#### 4.5.2.Results

Subsequently, two experiment examples with either regular phase shifts or random phase shifts are carried out. In each experiment example, we collect four sets of fringe patterns according to the Sets II, IV, VII and IX. Other sets are not acquirable as there is no mechanism to eliminate the random noise or to introduce nonuniformities in the background intensity and fringe amplitude.

In the experiment, we change the camera's gamma setting to  $\gamma=1.5$  for the generation of the intensity harmonics [204], and we use a motor to vibrate the reference for the generation of the phase-shift nonuniformities [142]. As we don't have the ground true bias phase shifts, we use the phase shifts extracted by the AIA for the initialization of the MPSI. The phases are extracted by different algorithms. Meanwhile, we also extract the phase shifts with nonuniformities by using the GIA. The phase and one frame's phase shift extracted by the GIA from the fringe patterns Set IV with random bias phase shifts are shown in Figure 4.8.

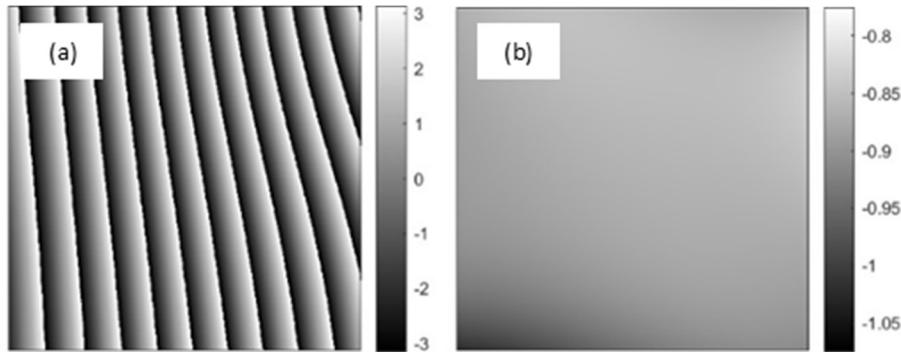


Figure 4.8.Experiment results (unit: rads).

(a) Phase extracted by the GIA; (b) One phase shift extracted by the GIA.

All the fringe pattern parameters including the phase, non-uniform phase shifts, background intensity and fringe amplitudes of different order of harmonics, can be extracted by the different uPSAs. Thus, we can use them to reconstruct the fringe patterns and then compare them with the captured fringe pattern to check the royalty of the reconstruction. A smaller difference indicates better reconstruction and hence more accurate extraction of the parameters. The differences using all the algorithms are shown in Figure 4.9, where the least difference is generated by GIA.

We compare the phases with the phase ground truth and calculate the RMSEs of the phases. The results are shown in Figure 4.10 and Table 4.5. From the result, we notice that: (i) The RMSE results in the experiments and simulations show the same trend; (ii) Regular bias phase shifts are beneficial for all the algorithms; (iii) The GIA is with the most outstanding accuracy.

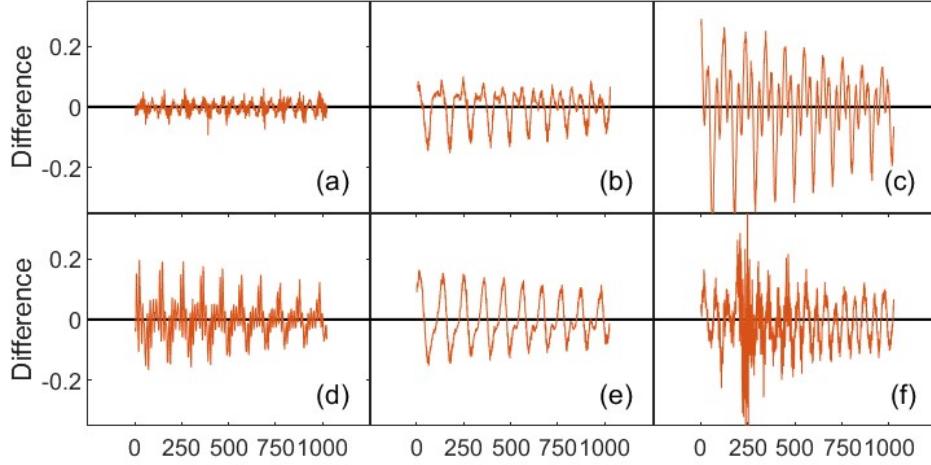


Figure 4.9.Difference between the measured intensity and fringe pattern generated by  
 (a) GIA; (b) AIA; (c) Xu's algorithm; (d) Hoang's algorithm; (e) PTI; (f) MPSI.

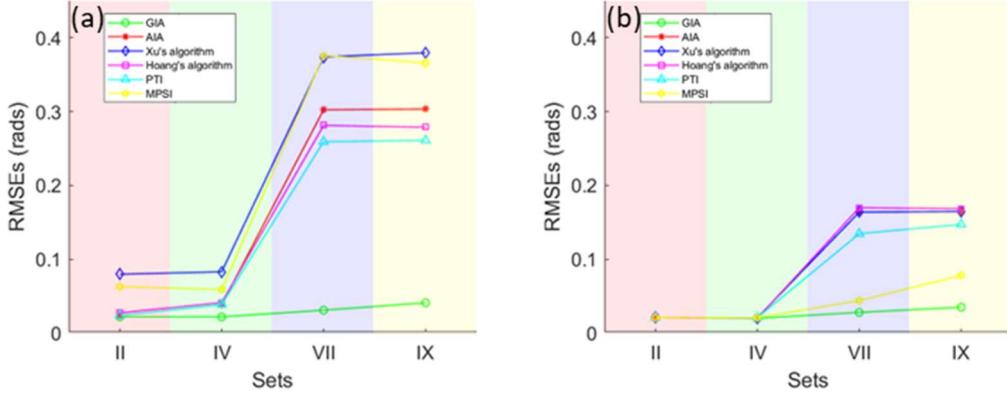


Figure 4.10.Phase error results from the experiment.

(a) with random bias phase shifts; (b) with regular bias phase shifts.

Table 4.5.RMSEs from the experiment (Unit: rad).

Sets	Random phase shifts				Regular phase shifts			
	II	IV	VII	IX	II	IV	VII	IX
GIA	0.022	0.022	0.031	0.041	0.021	0.020	0.028	0.035
AIA	0.024	0.039	0.302	0.303	0.021	0.021	0.164	0.165
Xu's algorithm	0.080	0.083	0.374	0.380	0.021	0.021	0.164	0.165
Hoang's algorithm	0.027	0.041	0.281	0.279	0.021	0.020	0.170	0.168
PTI	0.024	0.039	0.259	0.261	0.021	0.021	0.135	0.147
MPSI	0.063	0.059	0.376	0.366	0.021	0.021	0.044	0.078

## 4.6.Summary

In this chapter, to suppress the influence of different error sources on the uPSAs, we proposed a GIA which considers all the error sources in its model. Based on their different natures, the

unknowns in the GIA are classified into three groups: (i) the background intensity and the fringe amplitude, (ii) the phase and (iii) the phase-shift related parameters. Subsequently, these three groups of unknowns are optimized alternatively by the Levenberg-Marquart method, which has demonstrated excellent convergence and accuracy performance. In addition, the background intensity and the fringe amplitude are estimated within a small window, and thus the error immunity of the GIA to the random noise is improved. The accuracy of the GIA is compared with other optimization based uPSAs in where the GIA shows the best accuracy performance.



# **Chapter 5 Evaluation of phase-shifting algorithm with known and unknown phase shifts\***

## **5.1. Background**

As discussed in Chapter 2, a large pool of PSAs has been developed, including both the uPSAs and the kPSAs. Numerous studies on PSA's evaluation followed. However, these evaluation works mainly are for the kPSAs and very few are for the uPSAs. Although, evaluations on the AIA and accuracy comparison among the optimization based uPSAs have been carried out in chapter 3 and chapter 4. The overall picture of the PSAs is still unclear. Specifically, there is a lack of: (i) robustness analysis of more uPSAs, (ii) accuracy comparison among different categories of uPSAs, (iii) accuracy comparison between the uPSAs and the kPSAs, and (iv) accuracy improvement of PSA by considering the merits of both the uPSAs and the kPSAs. To solve these problems, we conduct a detailed performance evaluation of five uPSAs from the three categories: Carré's algorithm, the advanced iterative algorithm (AIA), the general iterative algorithm (GIA), the algorithm based on the principal component analysis and the algorithm based on VU factorization. To confirm the performance, merits and restrictions of different PSAs, simulations are carried out to compare these uPSAs among themselves and also with twelve benchmarking kPSAs which are reviewed in chapter 2.1.1 and shown in Table 2.1. By combining the merits of the kPSAs and one well performing uPSA, the GIA, we proposed a hybrid kPSA-GIA.

## **5.2. Codes, fringe patterns and error measures**

The main approach for our PSA evaluation is through simulations, which have been widely used and well-recognized. The implementation of the kPSAs is according to Eq. (2.2) and the weight parameters in Table 2.1, which is straightforward. Carré's algorithm is implemented following Eq. (2.10) and Refs. [158, 192]. The AIA and the GIA are coded following the descriptions in chapter 2.2.3.1 and chapter 4.3, respectively. The codes of the PCA and the VU were made available by

---

\* The work in this chapter has been submitted as Y. Chen and Q. Kemao, "Phase-shifting algorithms with known and unknown phase shifts: comparison and hybrid,". Optics Express, vol. 30, no. 5, pp. 8275-8302, 2022.

the respective original authors [164, 167], which are directly adopted in this chapter. The settings of the parameters in the AIA, the GIA and the VU follow the descriptions in chapter 2.2.

Due to the complexity of our evaluation, we start with the following default specifications on the fringe patterns as common ground: (i) The frame number is eight; (ii) The image size is  $N_x=N_y=1024$ ; (iii) The background intensity and the fringe amplitude are  $A=0.5$  and  $B=0.4$ ; (iv) The phase is simulated as

$$\varphi(x, y) = \omega_p \times \text{peaks\_s}(N_x, N_y) + \omega_x x + \omega_y y, \quad (5.1)$$

with  $\omega_p=10$ ,  $\omega_x=0$  and  $\omega_y=0.25$  where `peaks_s` scales the “peaks” function in MATLAB into  $[0, 2\pi]$ ; (v) The phase shifts are set randomly as

$$\delta_0 = [0 \ 0.440 \ 0.795 \ 1.246 \ 1.601 \ 2.665 \ 2.695 \ 3.449]. \quad (5.2)$$

The phase map and the first fringe patterns of the common ground are shown in Figure 5.1 for illustration.

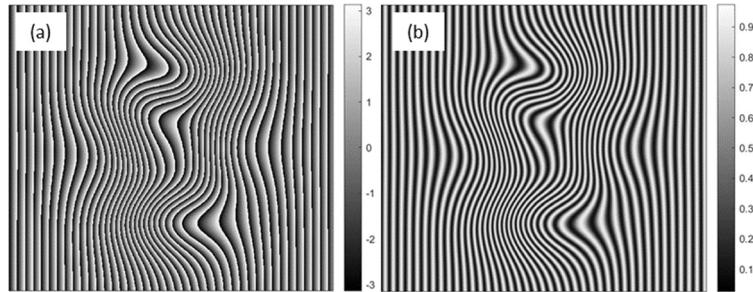


Figure 5.1. Simulation parameters of the common ground.

(a) The phase; (b) The first fringe pattern.

With this default setting, we will vary one particular parameter within an interval while keeping the others at their default values, to evaluate how this particular parameter affects the uPSAs. Note that the phase shifts are purposely set the same as in chapter 4.4.3 for cross comparison. However, we have extensively tried other random phase shifts, which give consistent results.

The mean value of a phase error (ME) and the peak and valley of the phase error (PVE) are selected as two error measures. Later on, when random noise presents, the standard deviation error (STDE) will also be used together with the ME. Given the ground true value of  $\varphi_0$  and the extracted phase  $\varphi$ , we denote the phase error as  $\Delta\varphi=\varphi-\varphi_0$ . The error measures are then defined as

$$ME = \frac{\sum_{x,y} [\Delta\varphi(x,y)]}{N_x N_y}, \quad (5.3)$$

$$PVE = \max_{x,y} [\Delta\varphi(x,y)] - \min_{x,y} [\Delta\varphi(x,y)], \quad (5.4)$$

$$STDE = \sqrt{\frac{1}{N_x N_y - 1} \sum_{x,y} [\Delta\varphi(x,y) - ME]^2}. \quad (5.5)$$

### 5.3.The pre-requisites of the uPSAs

A significant merit of the uPSAs is that they can extract the phase from fringe patterns with random unknown phase shifts. However, this advantage is not achieved at no cost. In fact, the uPSAs may result in significant phase errors due to specific phase, phase-shift distribution, background intensity or fringe amplitude. In contrast, the kPSAs' performance and the kPSA-like Carré's algorithm's performance are not affected. Thus, it is necessary to understand these pre-requisites of the AIA, the GIA, the PCA and the VU on fringe parameters. We seek these clarifications in this sub-chapter where all the error sources do not present.

#### 5.3.1.On the phase

The phase distribution affects the performance of the uPSAs. Firstly, we focus on the global variation of the phase in the entire frame, which can also be represented by the total number of fringes (*i.e.*,  $2\pi$  periods) per frame and is called as fringe density. In our simulation following chapter 5.2 with  $\omega_p=10$  and  $\omega_x=0$ , we further express the parameter  $\omega_y$  in the phase distribution in Eq. (5.1) as,

$$\omega_y = \frac{2\pi\tau_\varphi}{N_y - 1}, \quad (5.6)$$

where  $\tau_\varphi$  indicates the fringe density, *i.e.*, there are  $\tau_\varphi$  fringes in the entire frame. We vary  $\omega_y$  from a fixed default value of 0.25 to an interval characterized by  $\tau_\varphi \in (0, 3]$  to study its influence. Here in Figure 5.2, we show three fringe patterns simulated with  $\tau_\varphi=0.5$ ,  $\tau_\varphi=1$  and  $\tau_\varphi=2$  for better illustration.

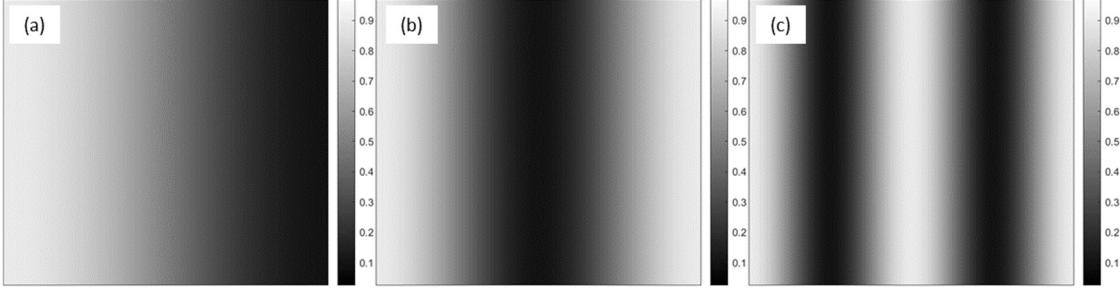


Figure 5.2.Fringe patterns with different fringe densities.

(a) With  $\tau_\phi=0.5$ ; (b) With  $\tau_\phi=1$ ; (c) With  $\tau_\phi=2$ .

The MEs and the PVEs of the phase extracted by the uPSAs are plotted against  $\tau_\phi$  in Figure 5.3. The MEs of the AIA and the GIA are insignificant, but those of the PCA and the VU are very large, which has been observed earlier in [205]. According to the PVEs, the fringe density should be higher than half a fringe in the frame ( $\tau_\phi \geq 0.5$ ) in order for the AIA, the GIA and the VU to perform accurately and stably. For the PCA, a fringe density of at least two fringes in the frame ( $\tau_\phi \geq 2$ ) is required so that Eq. (2.12) can be satisfied to achieve a high accuracy, which is more stringent than the pre-requisite ( $\tau_\phi \geq 1$ ) given in [188].

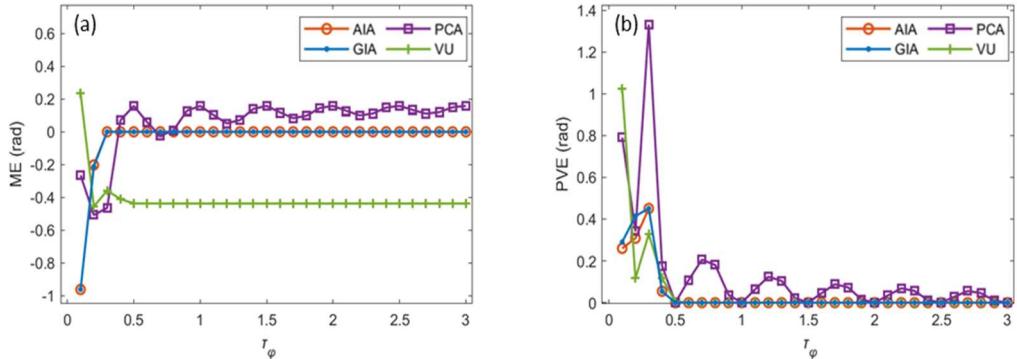


Figure 5.3.Phase errors from fringe patterns with different fringe densities.

(a) The MEs; (b) the PVEs.

We further study the distribution of the fringe, *i.e.*, the situation where the fringe is unevenly gathered in the frame. For this purpose, in the above simulation, we modify the phase as

$$\varphi(x, y) = \begin{cases} \pi y / \tau_\phi (N_y - 1), & \text{if } y \leq \tau_\phi (N_y - 1) \\ 0, & \text{if } y > \tau_\phi (N_y - 1) \end{cases} \quad (5.7)$$

where  $\tau_\phi$  controls the area of the half a fringe and  $\tau_\phi \in [0.01, 1]$ . We set  $\tau_\phi=0.5$  for the AIA, GIA, and the VU and  $\tau_\phi=2$  for the PCA. In Figure 5.4, we show the phase and the first phase-shifted

fringe pattern with  $\tau_\phi=0.5$   $\tau_\phi=0.05$ , *i.e.*, the half a fringe is concentrated within the left 5% area of the frame.

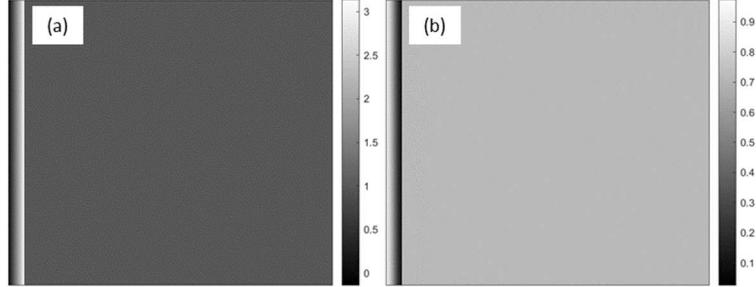


Figure 5.4.Simulation parameters of concentrated fringe pattern.

(a) The phase; (b) The first fringe pattern.

The MEs and the PVEs of the phase extracted by the uPSAs are plotted against  $\tau_\phi$  in Figure 5.5. Even when the half a fringe is concentrated within 1% of the frame, the AIA, the GIA, and the VU can perform stably. However, the PVE of the PCA is large which demonstrates that the PCA performance badly when the fringe patterns are concentrated.

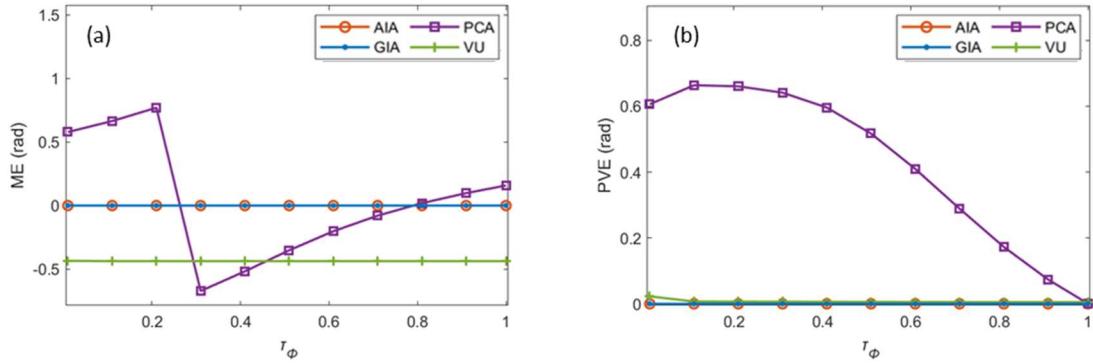


Figure 5.5.Phase errors from concentrated fringe patterns.

(a) The MEs; (b) the PVEs.

There are a few remarks on the above results: (i) we also test the fringes with circular and other complex phases and obtain good consistencies; (ii) the GIA's requirement for  $\tau_\phi<0.5$  comes from that of the AIA because the AIA result is used to initialize the GIA. If a kPSA is used for initialization, then this pre-requisite is not necessary; (iii) as a byproduct, we notice that the phase in Eq. (5.7) is discontinuous between the two parts, none of the AIA, the GIA and the VU perform weirdly, especially around the discontinuity area. Meanwhile, we then carry out more simulations with different discontinuous phases fulfilling the fringe density requirement of  $\tau_\phi>0.5$  for the AIA,

GIA, and the VU and  $\tau_\phi > 2$  for the PCA, from which, it is confirmed that all these uPSAs are indeed insensitive to the phase discontinuities.

We now mention that, if a large ME from a particular algorithm is a concern in a certain measurement, then this algorithm should be used with care. However, in many phase measurements, MEs are less important than PVEs [206]. We will thus discuss more on PVEs in the rest of the chapter, while providing all the ME plots for reference.

### 5.3.2. On the phase-shift distribution

To see the influence of the phase-shift distribution, the fringe patterns are simulated according to chapter 5.2, but the phase shifts are set as  $\delta(i) = (i\pi/4) \times \tau_\delta$  ( $i=0 \sim 7$ ), *i.e.*, the phase shifts are evenly allocated within an interval of  $[0, 2\pi \times \tau_\delta]$ . For example, when  $\tau_\delta$  is 0.5, the phase shifts cover  $[0, \pi]$ ; when  $\tau_\delta$  is 1, they cover  $[0, 2\pi]$ ; when  $\tau_\delta$  is further increased to 2, they cover  $[0, 4\pi]$ , or in other words, they cover  $[0, 2\pi]$  twice. The MEs and the PVEs of the uPSAs are plotted against  $\tau_\delta$  as shown in Figure 5.6. All uPSAs' PVEs are in the order of  $10^{-3}$  rad, showing that they are insensitive to the phase-shift distributions. Although the phase shifts used in Figure 5.4 are all regular, random phase shifts are also tested in which we notice that all the uPSAs are insensitive to phase-shift distributions, as well expected. We highlight here that no error sources are presented in our prerequisite study. In fact, when there are errors in the fringe patterns, phase-shift distributions evenly allocated within  $[0, 2\pi)$  are preferred as what will be discussed later.

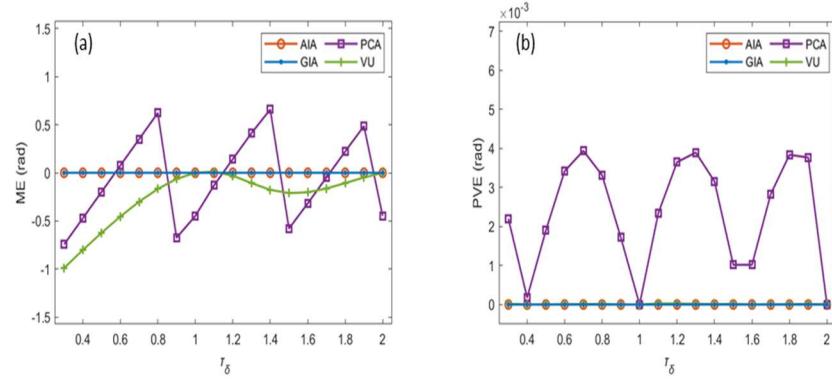


Figure 5.6. Phase errors from fringe patterns with different phase shifts.

(a) The MEs; (b) the PVEs.

### 5.3.3.On the background intensity and the fringe amplitude

To study the influence of the background intensity and the fringe amplitude, we first use the following model to make it spatially nonuniform,

$$A(x, y) = 0.5 - \frac{\tau_B y}{N_y - 1}, \quad (5.8)$$

where  $\tau_B$  controls the nonuniformity level. Subsequently, the fringe amplitude is set as  $B(x, y) = 0.8 \times A(x, y)$ . The background intensity with  $\tau_B=0.4$  and the first fringe pattern with  $\tau_B=0.4$  are shown in Figure 5.7.

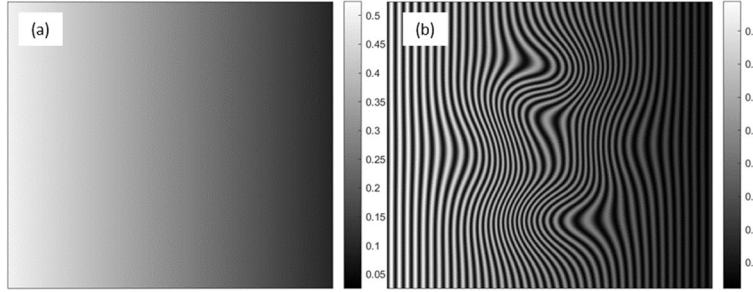


Figure 5.7.Simulated parameters with background intensity and fringe amplitude nonuniformities.

(a) The background intensity; (b) the first fringe pattern.

The MEs and the PVEs of uPSAs are plotted against  $\tau_B$  as shown in Figure 5.8. The GIA is least sensitive to the background intensity and the fringe amplitude non-uniformity. Meanwhile, although the PCA and the VU have no assumption on the background intensity and the fringe amplitude, in theory, the performance is influenced.

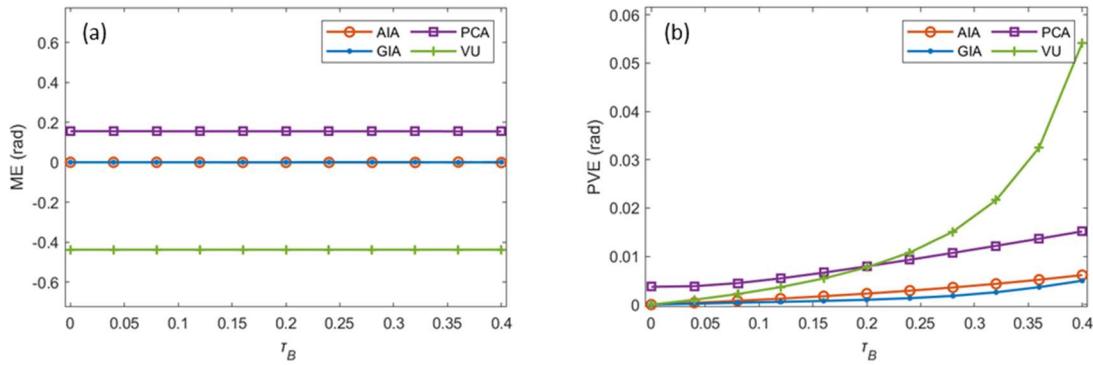


Figure 5.8.Phase errors from fringe patterns with different nonuniform background intensities and fringe amplitudes.

(a) The MEs; (b) the PVEs.

Secondly, background intensity and fringe amplitude with a horizontal discontinuity are simulated as

$$A(x, y) = \begin{cases} 0.5 & \text{if } x \leq N_x / 2, \\ 0.25 & \text{if } x > N_x / 2, \end{cases} \quad (5.9)$$

and  $B(x, y) = 0.8 \times A(x, y)$ . The background intensity with discontinuity and the first fringe pattern is shown in Figure 5.9.

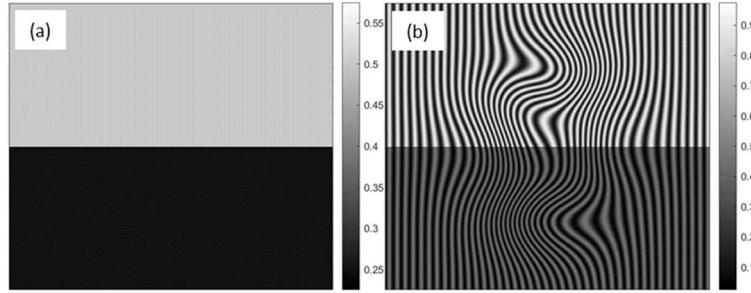


Figure 5.9.Simulated parameters with discontinuous background intensity and fringe amplitude.

(a) The background intensity; (b) the first fringe pattern.

The phase error map of the uPSAs is shown in Figure 5.10. The middle columns of the phase error results from the uPSAs are shown in Figure 5.11.

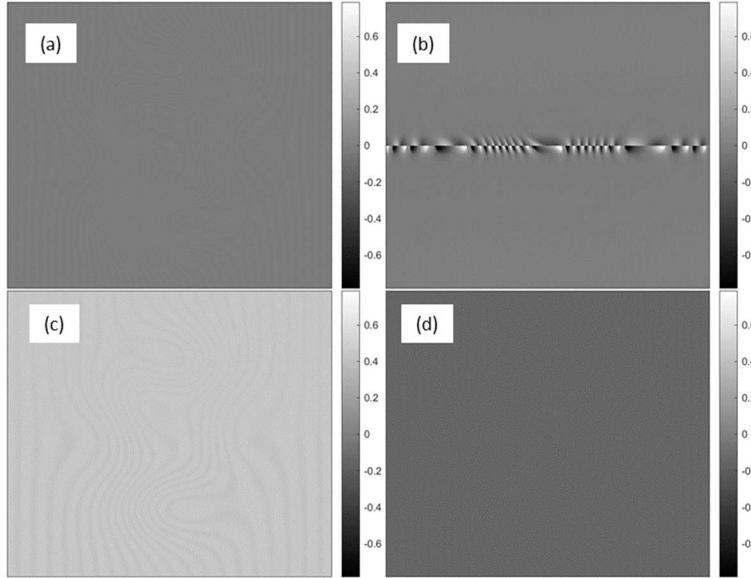


Figure 5.10.Phase error map from fringe patterns with discontinuous background intensity and fringe amplitude.

(a) the AIA; (b) the GIA; (c) the PCA; (d) the VU.

The GIA has an error at the discontinuity line because it assumes the uniformity of the background intensity and the fringe amplitude. The other uPSAs are not sensitive to discontinuity

in the background intensity and the fringe amplitude. It is worth mentioning that fringe pattern segmentations could be easily achieved for this simulation case based on the background intensity shown in Figure 5.9(a). Meanwhile, since the AIA can accurately estimate the background intensity and the fringe amplitude, it is possible to split the simulated phase-shifted fringe patterns up into two groups of phase-shifted fringe patterns with uniform background intensity. The phases extracted by the GIA from the two groups of phase-shifted fringe patterns with uniform background intensity can be stacked together to generate a phase map. This stacked phase map will be accurate, which has been confirmed by simulations. This practice of combining the fringe pattern segmentation and the GIA can be considered when it is feasible, and the background intensity and fringe amplitude are discontinuous. However, when there are errors in the fringe pattern or there are too many segments in the background intensity, fringe pattern segmentations might be difficult, thus more complex segmentation algorithms might be needed.

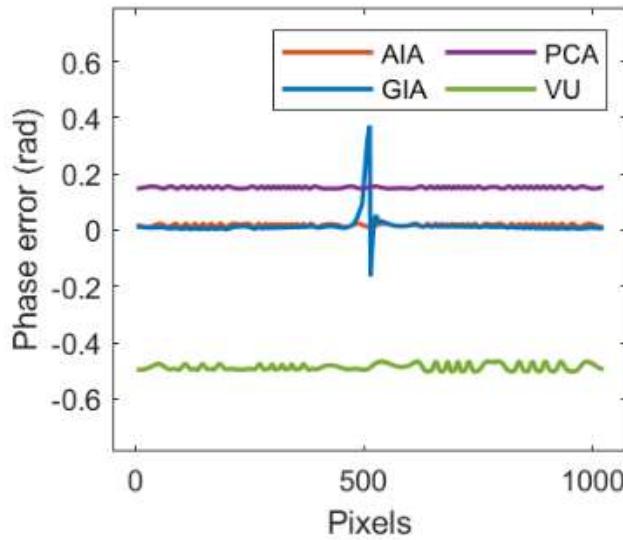


Figure 5.11. Middle column of phase error map of the uPSAs from fringe patterns with discontinuous background intensity and fringe amplitude.

#### 5.3.4. Summary

We summarize the pre-requisite of the uPSAs on fringe parameters and recommendations in Table 5.1. Two conditions affect the uPSAs' performance. The first is when the fringe density is low, and the second (affecting the GIA only) is when the background intensity and fringe amplitude are discontinuous. Meanwhile, many fringe patterns, such as those from the optical element and wafer

testing, do not fall into these two conditions. Bearing these in mind, we now proceed to detailed comparisons when various error sources are presented.

Table 5.1.The pre-requisites of the uPSAs on fringe parameters.

Fringe parameters		AIA	GIA	PCA	VU
Phase	Fringe density	$\geq 0.5$	$\geq 0.5$ (initialized by AIA); Free (initialized the kPSAs)	$\geq 2$	$\geq 0.5$
Background intensity and fringe amplitudes	Nonuniformities			Slightly sensitive	Slightly sensitive
	Discontinuities		Sensitive (in general); Insensitive (if segmented)		

## 5.4. Within the uPSAs: the performance with unknown phase shifts

Based on the pre-requisite study above, in later simulations, we set up two fringe pattern simulation restrictions: (i) The background intensity and the fringe amplitude are continuous; and (ii) Fringe density is higher than two fringes per image. To evaluate the performance of the uPSAs with various error sources, one error source at a time will be added into the fringe pattern simulated based on the default specifications discussed in chapter 5.2, for detailed study. More tests with different fringe patterns will be carried out afterward to support the study.

### 5.4.1. Phase-shift errors

In the kPSAs, the phase shifts are added actively, but the introduced values may deviate from their respective nominal values, *i.e.*, phase-shift errors occur. Typically, the mismatch between the true and nominal values includes both a pixel-independent part and a pixel-dependent part. The former is normally referred to as a phase-shift error whose impact on the uPSAs is discussed in this sub-chapter, while the latter is normally referred as phase-shift nonuniformity whose impact on the uPSAs will be discussed in chapter 5.4.3.

The phase-shift errors are introduced as additive noise with a mean of zero and a standard deviation of  $\sigma_\delta=0\sim 2\pi/4$ . The MEs and the PVEs of the uPSAs are plotted against  $\sigma_\delta$  in Figure 5.12. When  $\sigma_\delta > \pi/4$ , the phase-shift errors are considered serious, but the obtained PVEs of all uPSAs are smaller than  $5\times 10^{-3}$  rad, which is natural as the uPSAs are designed for unknown random phase shifts, and thus are insensitive to phase-shift errors.

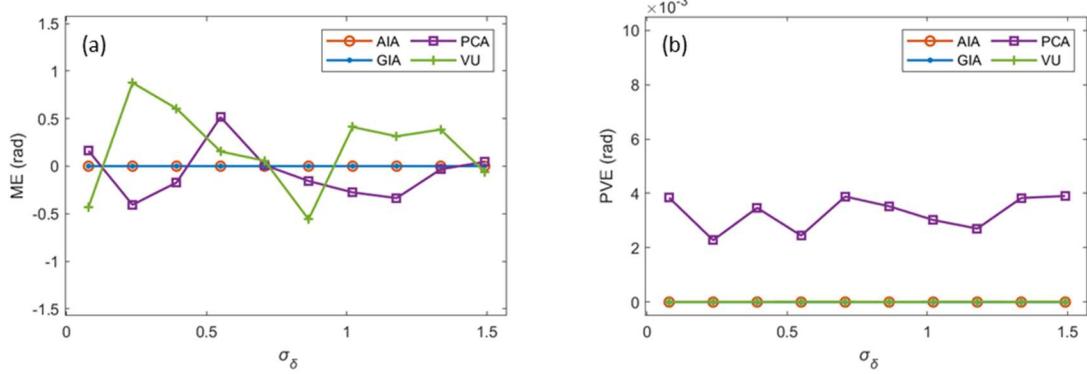


Figure 5.12. Phase errors from fringe patterns with different amplitude of the phase-shift errors.

(a) the MEs; (b) the PVEs.

#### 5.4.2. Intensity harmonics

The intensity harmonics are added by using a gamma model, *i.e.*,  $I'$  is used to represent the nonlinear fringe intensity with different  $\gamma=1\sim 3$  [203]. For better illustration, the fringe patterns with  $\gamma=1$ ,  $\gamma=1.5$  and  $\gamma=3$  are shown in Figure 5.13.

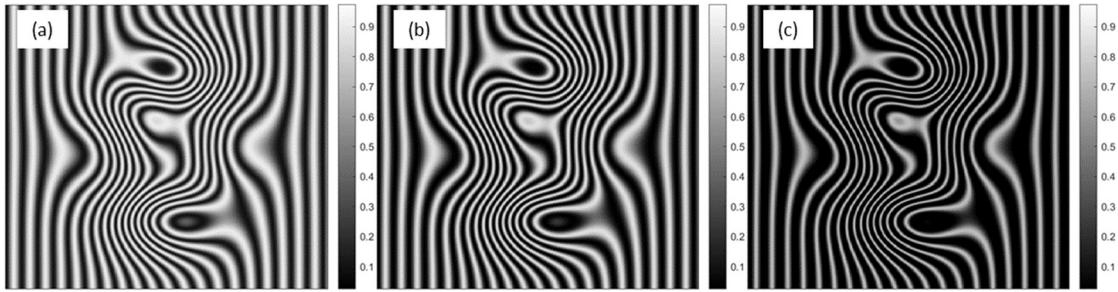


Figure 5.13. The fringe patterns with intensity harmonics.

(a) with  $\gamma=1$ ; (b) with  $\gamma=1.5$ ; (c) with  $\gamma=3$ .

The MEs and the PVEs of the uPSAs are plotted against  $\gamma$  in Figure 5.14. With the increment of  $\gamma$ , the PVEs of all uPSAs increase. The GIA shows the best immunity to the intensity harmonic followed by the AIA, the PCA and the VU.

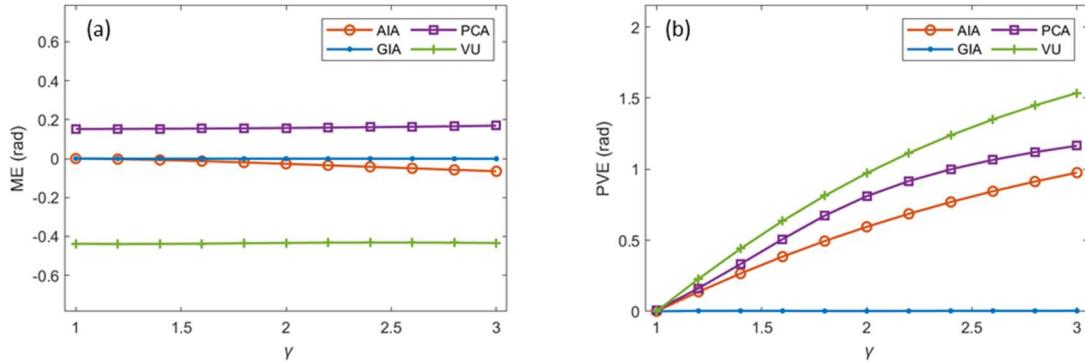


Figure 5.14. Phase errors from fringe patterns with random phase shifts and different amplitude of the intensity harmonics.

(a) The MEs; (b) the PVEs.

Besides, with the presence of intensity harmonics, the phase-shift distribution plays a more significant role. The MEs and the PVEs of simulations according to chapter 5.3.2 with intensity harmonics of  $\gamma=1.5$  are shown in Figure 5.15. It is clear now that the phase shifts should be evenly allocated within  $[0, 2\pi)$ . The GIA is seen as an exception. In general, making the phase shifts cover at least a full  $2\pi$  interval is recommended for common practice.

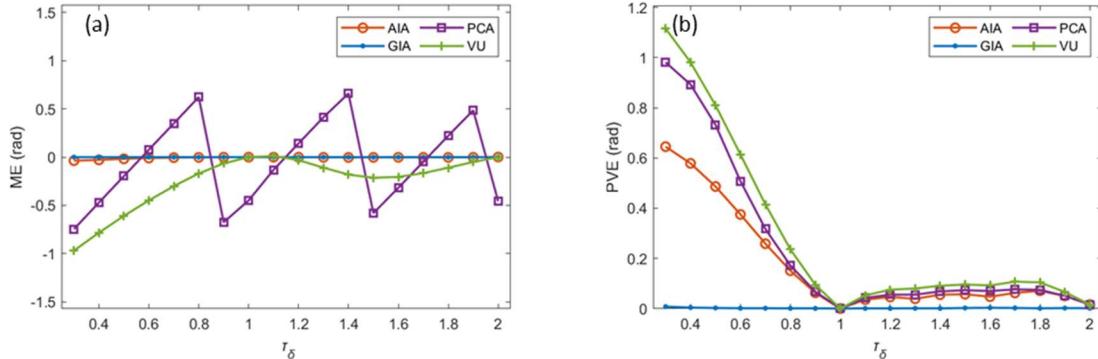


Figure 5.15. Phase errors from fringe patterns with regular phase shifts and different amplitude of the intensity harmonics.

(a) The MEs; (b) the PVEs.

Finally, we note that, our intention is to study the random phase-shift distributions, but the simulated phase shifts are regular. Nevertheless, when the random phase shifts approach a regular distribution which is nearly evenly allocated within  $[0, 2\pi)$ , the error measures are better than the completely random phase shifts case. In other words, when the random phase shifts are introduced, we should make them cover a full  $2\pi$  interval with nearly even distribution, if possible, for better

error immunity to intensity harmonics. A similar preference for the phase shifts is also noticed in simulations with random noise.

### 5.4.3. Phase-shift nonuniformities

Different phase-shift nonuniformities are simulated as

$$\delta(x, y; i) = \sum_{u=0}^Q \sum_{v=0}^u \frac{\alpha_{uv}(i) x^v y^{u-v}}{(N_x - 1)^v (N_y - 1)^{u-v}}, \quad (5.10)$$

with different  $Q$  and different  $\alpha_{uv}$ . For demonstration, first, a group of base  $\bar{\alpha}_{uv}$  with  $u=0, \dots, Q$ ,  $v=0, \dots, u$  and  $Q=4$  is simulated as random values with a zero mean and a standard deviation value of  $\sigma_\alpha=1$ . Next, the different amplitudes of phase-shift nonuniformities are simulated based on Eq. (5.10), with

$$\alpha_{uv} = \tau_\alpha \bar{\alpha}_{uv}, \quad (5.11)$$

and  $\alpha_{00}=\delta_0$  in Eq. (5.2). The phase shifts of the second and last frame of fringe when  $\tau_\alpha=0.15$  are shown in Figure 5.16.

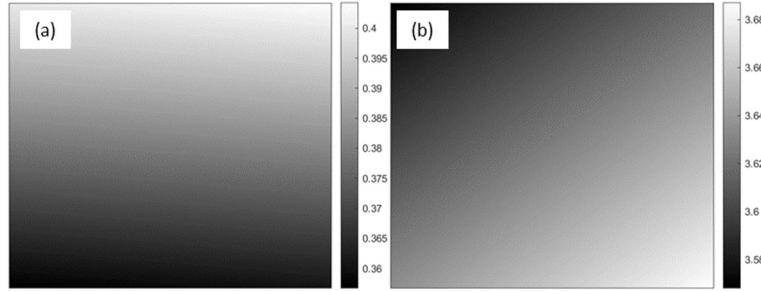


Figure 5.16. Phase shifts with nonuniformities.

(a) The second frame; (b) the last frame.

The MEs and the PVEs of the uPSAs are plotted against  $\tau_\alpha$  in Figure 5.17. The GIA as the only algorithm considers the phase-shift nonuniformities has the best performance. The other three uPSAs have similar PVEs which increase significantly with respect to  $\tau_\alpha$ .

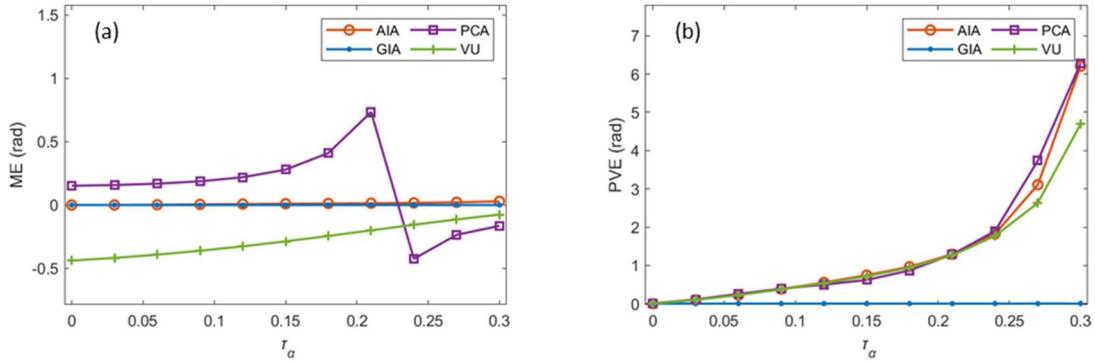


Figure 5.17. Phase errors from fringe patterns with different levels of phase-shift nonuniformities.

(a) The MEs; (b) the STDEs.

#### 5.4.4. Random noise

Random noise cannot be avoided in a real measurement. It is simulated as additive white Gaussian noise which has a zero mean and a standard deviation value of  $\sigma_r$ . Different amplitudes of random noise are simulated as  $\sigma_r=0\sim0.2$  (corresponding to 0%~50% of fringe amplitude). The first fringe patterns with different levels of random noise of  $\sigma_r=0$ ,  $\sigma_r=0.04$  and  $\sigma_r=0.20$  are shown in Figure 5.18.

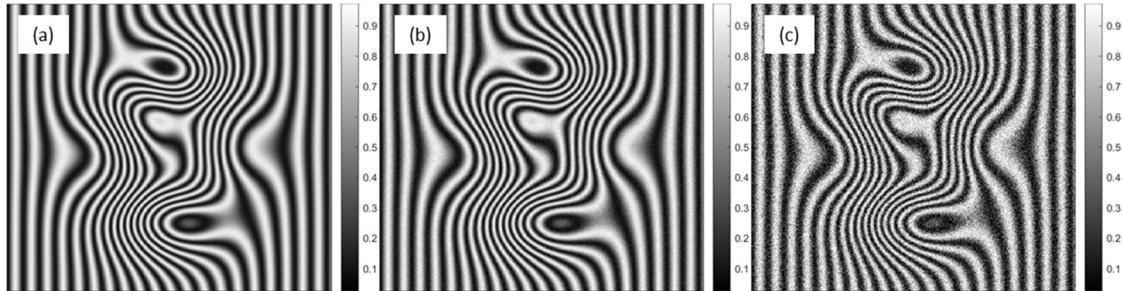


Figure 5.18. The fringe patterns with random noise.

(a) with  $\sigma_r=0$ ; (b) with  $\sigma_r=0.04$ ; (c) with  $\sigma_r=0.20$ .

The MEs and the STDEs of the uPSAs are plotted against  $\sigma_r$  in Figure 5.19. The STDEs are linearly related to the noise level for all the uPSAs. When the noise is minor, the performance of all the uPSAs is similar. However, when the noise becomes heavier, the GIA shows a better immunity, while the other three uPSAs have a similar STDE performance due to their similar least-squares nature. The better performance of the GIA is due to the averaging effect in the window based background intensity and fringe amplitude estimation.

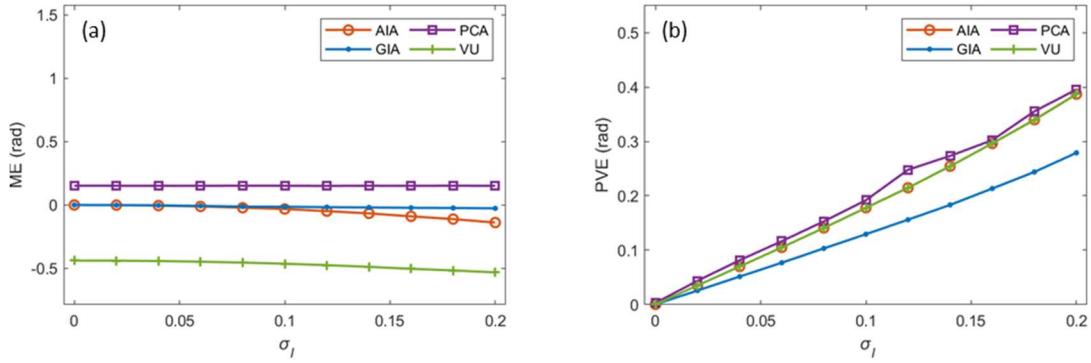


Figure 5.19. Phase errors from fringe patterns with different levels of random noise.

(a) The MEs; (b) the STDEs.

We also study the influence of the frame number with the presence of random noise. Simulations are carried out according to chapter 5.2 with a random noise  $\sigma_I=0.04$ , regular phase shifts of  $\delta(i)=2i\pi/F$ , and different frame numbers  $F$ . The MEs and the STDEs of the phases extracted by the uPSAs are plotted against  $F$  as shown in Figure 5.20. Clearly, more fringe patterns should be acquired if higher accuracy is desired. In addition, the preference for phase shifts which are evenly allocated within  $[0, 2\pi]$  is also noticed in the simulation with random noise.

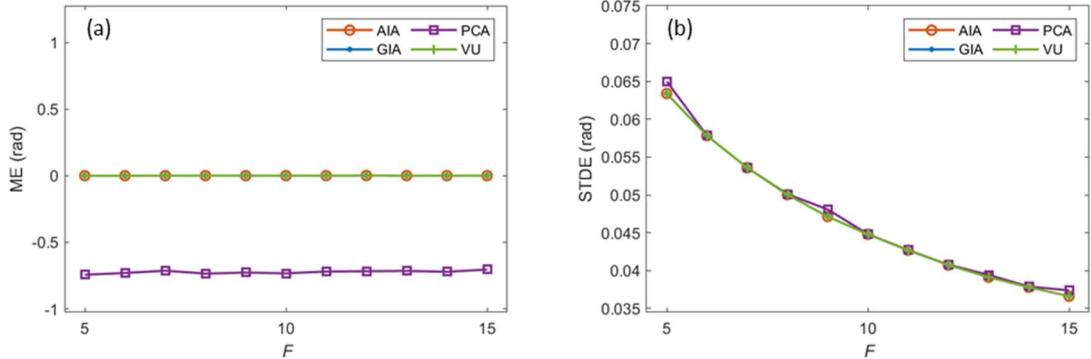


Figure 5.20. Phase errors from fringe patterns with different frame numbers and random noise.

(a) The MEs; (b) the STDEs.

#### 5.4.5. More tests and the summary

In the evaluation above, we focus on the default specifications discussed in chapter 5.2. We further test in total of 250 base sets of fringe patterns with: (i) different phase distributions simulated based on Eq. (5.1) with  $\omega_p=0\sim 20$ ,  $\omega_x=0\sim 0.5$  and  $\omega_y=0\sim 0.5$ ; (ii) different background intensity simulated based on Eq. (5.8) with  $\tau_B=0\sim 0.4$  and fringe amplitude as  $B(x,y)=0.8\times A(x,y)$ ; (iii) Different frame number as  $F=3\sim 20$ ; (iv) Different phase shifts simulated as random value cast within  $[0, 2\pi)$  or

specially set as  $\delta(i)=2i\pi/F\times\tau_\delta$ . The error sources are added to them one by one, and then the performance of the uPSAs is tested. The previous results are typical. The main conclusions are as follows:

- (i) In general, the performance of the uPSAs with random phase shifts and regular phase shifts is similar when phase-shift errors and phase-shift nonuniformities are percented, as expected. However, with other error sources present, regular phase shifts evenly allocated within  $[0, 2\pi)$  are preferred and recommended.
- (ii) All the uPSAs prefer lower amplitude intensity harmonics, phase-shift nonuniformities and random noise. Thus, minimizing these error sources during testing, if possible, is recommended.
- (iii) All the uPSAs have better noise suppression with more fringe patterns, and thus, capturing more fringe patterns is recommended, if time consumption is not an issue.
- (iv) The GIA shows an outstanding performance due to the fact that it explicitly incorporates all the mentioned error sources into the fringe model and then explicitly estimates through optimization. Nevertheless, it suffers from the discontinuity problem.

## **5.5.Between the uPSAs and the kPSAs: the performance with known phase shifts**

The uPSAs were developed for random unknown phase shifts. However, given the actively controlled known phase-shift values, the uPSAs can still work by treating them as unknown values. This provides a common scenario that uPSAs can be compared with the benchmarking kPSAs. We continue to use simulations to compare the uPSAs and the kPSAs. An experiment confirmation is given as well. The default specifications on the fringe patterns are similar to the ones described in chapter 5.2 except that the frame numbers and phase shifts are simulated based on the restrictions of the kPSAs and Carré algorithm. For the kPSAs, the frame numbers and phase shifts follow Table 1; for Carré, the frame number is set as  $F=4$  and the phase-shift increment  $\omega$  is set as  $\pi/2$  rad.

### 5.5.1. Phase-shift errors

Following the requirement of the kPSAs, the phase shifts are simulated according to Eq. (2.1). Furthermore, the phase shifts with mis-calibration/detuning error are included as [145]

$$\delta(i) = (i - \Gamma)\omega + \zeta(i), \quad (5.12)$$

with

$$\zeta(i) = \varepsilon_1(i - \Gamma)\omega + \varepsilon_2(i - \Gamma)^2\omega^2, \quad (5.13)$$

where  $\zeta(i)$  is the phase-shift error of the  $i$ th frame,  $\varepsilon_1$  and  $\varepsilon_2$  are linear and quadratic error coefficients, respectively. Two cases are simulated as follows: (i) linear miscalibration with  $\varepsilon_1=0.05$  and  $\varepsilon_2=0$  and (ii) quadratic miscalibration with  $\varepsilon_1=0$  and  $\varepsilon_2=0.025$ . The MEs and the PVEs of all the selected kPSAs and the uPSAs are shown in Figure 5.21, where one kPSA only works for one particular frame number, while a uPSA is applicable for all frame numbers. The following are observed:

- For the kPSAs, different algorithms give different PVEs, and thus choosing a proper algorithm is important, as has been well-understood and practiced. Most kPSAs, except the 3-step, the 4-step and the DFT, have good immunity to the linear phase-shift error, but only the AVE, the GEC and the 15-step have good immunity to the quadratic phase-shift error. These observations are consistent with evaluation through the equation solving method [157, 180-182]. If one wishes to select a proper kPSA systematically, Surrel's characteristic polynomials [156, 179] and Hibino et al.'s equation solving [157, 180-182] are recommended.
- For the uPSAs, the AIA, the GIA, the PCA and the VU have small PVEs, as expected. All the uPSAs perform much better than the kPSAs.

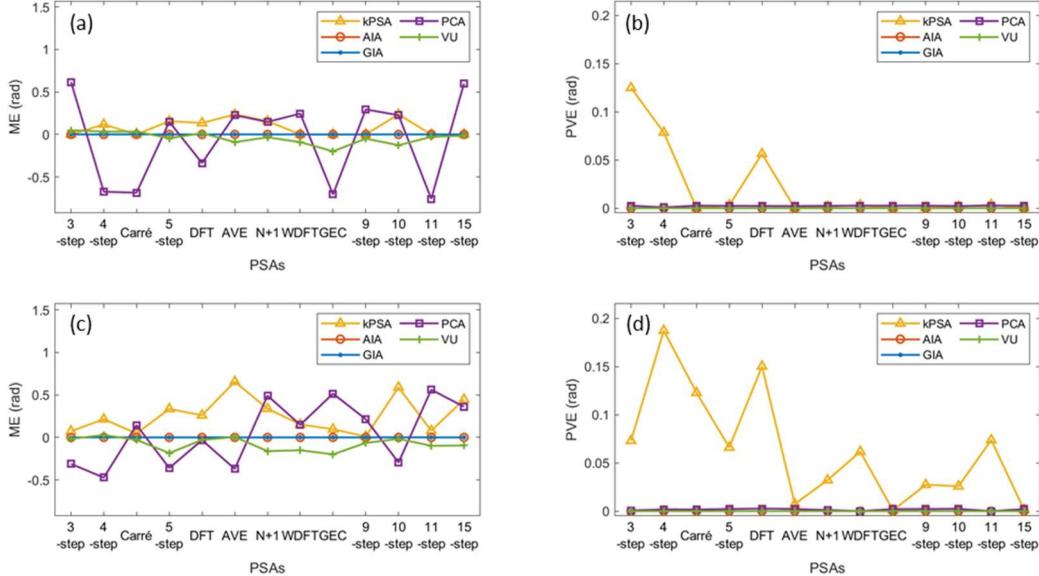


Figure 5.21. Phase errors from fringe patterns with phase-shift errors.

- (a) The MEs with linear phase-shift errors; (b) the PVEs with linear phase-shift errors; (c) the MEs with quadratic phase-shift errors; (d) the PVEs with quadratic phase-shift errors.

### 5.5.2. Intensity harmonics

The intensity harmonics are added by using a gamma model similar to the simulations in chapter 5.4.2. We use  $\gamma=1.5$  as an example. Due to this nonlinearity, the second and third order of harmonics appear, with their respective amplitudes about 10% and less than 1% of that of the first harmonic. There are also higher order harmonics with even smaller amplitudes. The MEs and the PVEs of the kPSAs and the uPSAs are shown in Figure 5.22, from which the following are observed:

- For the kPSA, based on the theoretical evaluation through the characteristic polynomials [156], the DFT, the N+1, the GEC, the 10-step and the 11-step have good immunity to both the second and third order intensity harmonics, thus have the smallest PVEs; the 4-step, the 5-step, the 9-step, the WDFT and the 15-step have immunity only to the second order intensity harmonic, thus, have the second tier PVEs; the 3-step and the AVE have no immunity to the second and third order intensity harmonics at all, thus, have the worst PVEs. Carré is close to the worst tier.
- For the uPSAs, the AIA, the PCA and the VU have much worse performance than the GIA and the well-performing kPSAs. Besides, when the frame number and the phase shifts are

simulated based on the requirement of the DFT, *i.e.*, the phase shifts of the seven fringe patterns are evenly allocated within  $[0, 2\pi]$ , the PVEs of all uPSAs reach their minimum like what has been demonstrated in chapter 5.4.2.

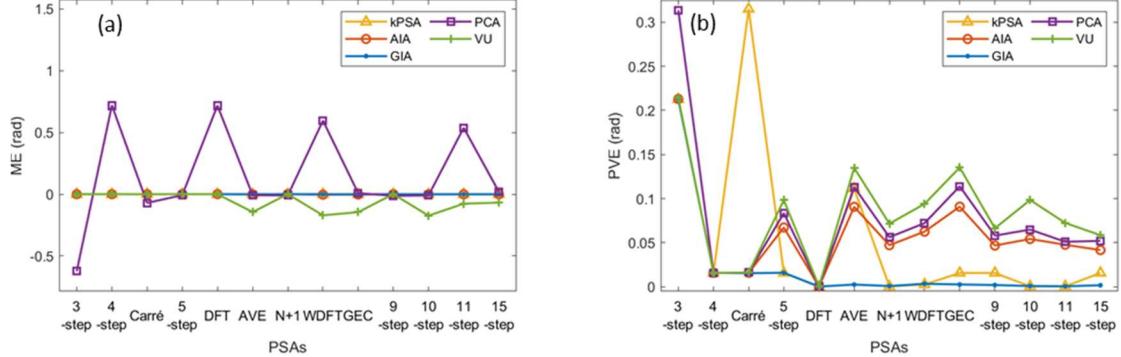


Figure 5.22. Phase errors from fringe patterns with gamma.

(a) The MEs; (b) the PVEs.

### 5.5.3. Phase-shift nonuniformities

To add the phase-shift nonuniformities, the phase shifts are simulated as [157]

$$\delta(x, y; i) = (i - \Gamma) \omega + \varepsilon_1(x, y)(i - \Gamma) \omega + \varepsilon_2(x, y)(i - \Gamma)^2 \omega^2. \quad (5.14)$$

This phase shift model has its similarity to the one used in chapter 5.5.1 except that  $\varepsilon_1$  and  $\varepsilon_2$  are pixel-dependent now. The  $\varepsilon_1$  and  $\varepsilon_2$  are simulated as

$$\varepsilon_1(x, y) = \alpha_{\varepsilon 11} \left[ \frac{x}{(N_x - 1)} \right]^2 + \alpha_{\varepsilon 12} \left[ \frac{x}{(N_x - 1)} \right] \left[ \frac{y}{(N_y - 1)} \right] + \alpha_{\varepsilon 13} \left[ \frac{y}{(N_y - 1)} \right]^2, \quad (5.15)$$

$$\varepsilon_2(x, y) = \alpha_{\varepsilon 21} \left[ \frac{x}{(N_x - 1)} \right]^2 + \alpha_{\varepsilon 22} \left[ \frac{x}{(N_x - 1)} \right] \left[ \frac{y}{(N_y - 1)} \right] + \alpha_{\varepsilon 23} \left[ \frac{y}{(N_y - 1)} \right]^2. \quad (5.16)$$

where  $\alpha_{\varepsilon 11}$ ,  $\alpha_{\varepsilon 12}$ ,  $\alpha_{\varepsilon 13}$ ,  $\alpha_{\varepsilon 21}$ ,  $\alpha_{\varepsilon 22}$  and  $\alpha_{\varepsilon 23}$  are set as random values with a zero mean and a standard deviation value of 0.01. The MEs and the PVEs of the kPSAs and the uPSAs are shown in Figure 5.23, from which the following are observed:

- For the kPSAs, the 9-step is with the best immunity to phase-shift nonuniformities which is consistent with the evaluation in [157].
- For the uPSAs, the GIA performs the best due to its explicit consideration of the phase-shift nonuniformity in the algorithm design.

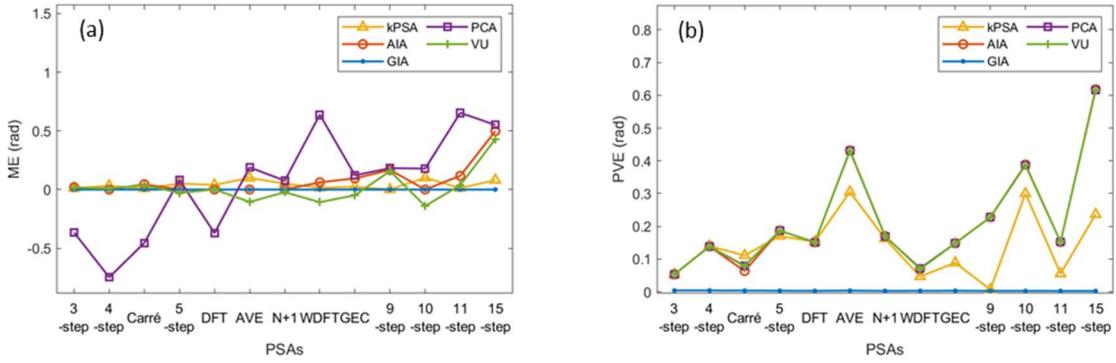


Figure 5.23. Phase errors from fringe patterns with phase-shift nonuniformities.

(a) The MEs; (b) the PVEs.

#### 5.5.4. Random noise

Random noise is added as additive white Gaussian noise which is similar to the simulation in chapter 5.4.4. We set  $\sigma_i=0.04$  (10% of fringe amplitude). The MEs and the STDEs of the kPSAs and the uPSAs are shown in Figure 5.24, from which the following are observed:

- For the kPSAs, the STDEs are slightly larger compared to those of the uPSAs. The STDE reduces with the increasing frame number. The DFT has the best immunity to random noise among those with the same frame number, which is consistent with the evaluation in [181].
- For the uPSAs, all the algorithms have very similar performance, due to their common least-squares nature. However, as has been shown in Figure 5.19 in chapter 5.4.4, when noise is high, the GIA shows slightly better noise suppression.

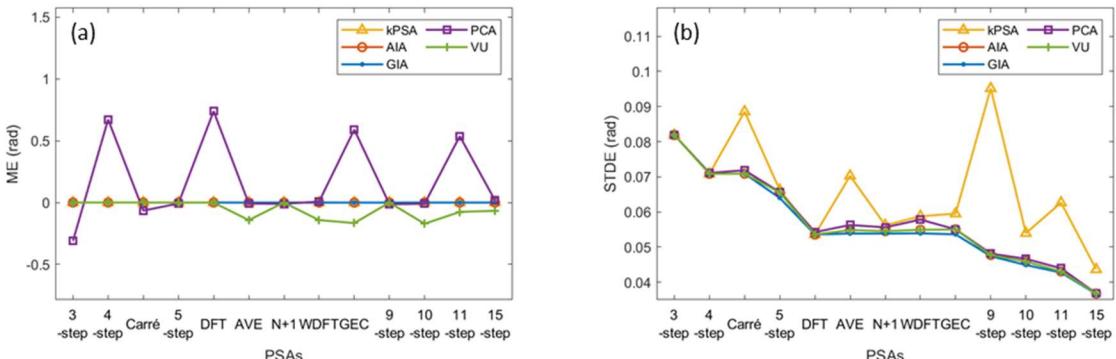


Figure 5.24. Phase errors from fringe patterns with random noise.

(a) The MEs; (b) the STDEs.

### 5.5.5.More tests and the summary

More tests with different phases, background intensities and fringe amplitudes, as described in chapter 5.4.5, are carried out. The previous results shown in chapter 5.4.1~5.4.4 are typical. Based on the evaluation, we summarize a few conclusions:

- (i) The kPSAs, in general, do not show good immunity to the tested error sources. However, one can always find a suitable algorithm to tackle the presenting error source(s). For example, the DFT has the best error immunity to the random noise when the frame number is fixed. Meanwhile, it is always possible to increase the frame number for better immunity to random noise. Similarly, the AVE and the GEC show good immunity to the phase-shift errors up to quadratic order; The DFT, the N+1, the GEC, the 10-step and the 11-step have good immunity to the second and third order intensity harmonics; The 9-step is insensitive to phase-shift nonuniformities. These results hint the importance of a proper algorithm selection, which may require a deep understanding about the kPSAs.
- (ii) The GIA consistently shows good immunity to all four tested error sources, as it is the only PSA incorporating these error sources. Besides, other uPSAs also show good immunity to the phase-shift error and the random noise.
- (iii) The uPSAs have pre-requisites as shown in chapter 5.3.4, which poses undesired uncertainty in measurement or requires extra controls to satisfy these pre-requisites during the measurement. However, this paper sends a clear message that the uPSAs are very strong competitors and alternatives to the kPSAs. In particular, (a) when the pre-requires are satisfied, the uPSAs can perform better; (b) after a kPSA extracts phase, one may further use this result to initialize and trigger a uPSA, for a better result as what will be demonstrated in chapter 5.6; and (c) in the adversary situation and environment where controlling the phase shifts is difficult, a high phase extraction accuracy can still be expected by a uPSA.

## 5.6.Hybrid kPSA-GIA

Based on the above extensive evaluations, we further propose a hybrid kPSA-GIA, which uses a kPSA to extract the phase to initialize the GIA and uses the GIA to extract the phase again. This

hybrid algorithm can be interpreted from two viewpoints. From the first point of view, even when a kPSA is not properly selected and does not well-compensate a certain error source, the subsequent GIA can serve as a refinement to improve the phase accuracy. From the other point of view, since in most phase-shifting interferometers, the phase shifts can be well controlled and known, a kPSA can be used to replace the AIA to initialize the GIA. By doing so, the pre-requisite of the fringe density in the GIA is removed, which further improves the applicability of the GIA. In summary, the hybrid kPSA-GIA alleviates the difficulty of kPSA selection, improves the phase extraction accuracy of a kPSA, and removes the fringe density pre-requisite in the GIA. One may choose any kPSAs suitable for the fringe patterns for initialization to form hybrid algorithms such as AVE-GIA or WDFT-GIA.

To validate the advantages, we test all the simulation cases in chapter 5.5.1~5.5.4 again with the hybrid kPSA-GIA. The phase error measures of the hybrid kPSA-GIA are similar to the AIA initialized GIA's error measures and are smaller than the original kPSAs' results. Besides, we also tested low density fringe patterns with the phase distribution simulated according to Eq. (5.1), Eq. (5.6), and with  $\tau_\phi < 0.5$ . For all simulation conditions, the hybrid kPSA-GIA performs stably even with  $\tau_\phi = 0$ . These results further confirm the effectiveness of the hybrid kPSA-GIA. Meanwhile, it is also worth mentioning that when the fringe density is very low, the error immunities of the hybrid kPSA-GIA are not as good as the results of  $\tau_\phi \geq 0.5$  cases because there is not enough intensity variation within the window for GIA's accurate estimation of the background intensity and the fringe amplitude. However, in such a case, it is reasonable to use a larger window size for a better accuracy performance.

To validate the effectiveness of the hybrid kPSA-GIA and its insensitivity to the fringe density, two sets of phase-shifted fringe patterns are specially discussed. The first set is generated according to chapter 5.5.2 with phase distribution according to Eq. (5.1),  $\omega_p=10$ ,  $\omega_x=0$  and  $\omega_y=0.25$  and serves as a case with normal fringe density (about 40 fringes per frame). The second set has a phase distribution according to Eq. (5.1), Eq. (5.6), and  $\tau_\phi=0.3$ , and serves as a case with low fringe density (0.3 fringes per frame). The first fringe patterns simulated under these conditions are shown in Figure 5.25.

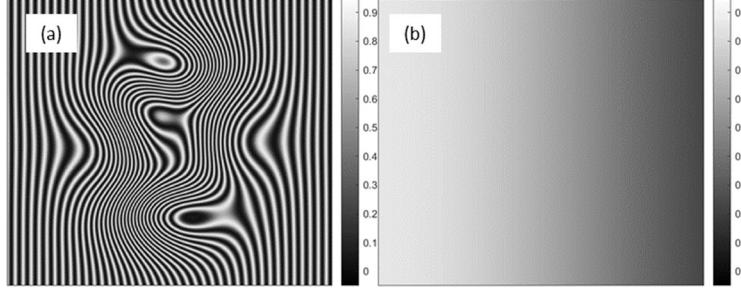


Figure 5.25. Simulated fringe patterns.

(a) With normal fringe density; (b) with low fringe density.

The PVEs of the kPSAs, the GIA and the hybrid kPSA-GIA of these two cases are shown in Figure 5.26(a) and Figure 5.26(b), respectively. In both figures, the PVEs of the kPSA (in yellow) have been successfully suppressed by the hybrid kPSA-GIA (in black). Furthermore, The GIA's PVEs in Figure 5.26(b) are much worse than that in Figure 5.26(a) because of the low fringe density, but it is successfully boosted by the initialization with kPSAs.

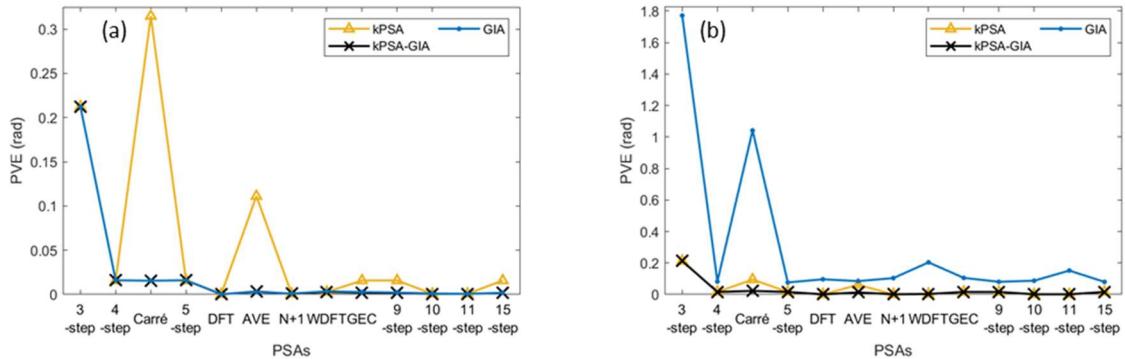


Figure 5.26. Phase errors from fringe patterns with intensity harmonics.

(a) The PVEs with normal fringe density; (b) The PVEs with low fringe density.

Meanwhile, the GIA has an interesting capability and advantage that it can reconstruct a noiseless fringe pattern following the model in chapter 4, because the GIA estimates all the parameters in this model. The difference between the reconstructed fringe pattern and the original fringe pattern can be used as an indicator of the GIA's performance on convergence and accuracy. The kPSA-GIA inherits this advantage. As an example, we simulate fringe patterns according to chapter 5.5.2 with  $F=7$ ,  $\omega=\pi/2$ ,  $I=0$ ,  $\omega_p=10$ ,  $\omega_x=0$  and  $\omega_y=0.25$ , and use WDFT-GIA for phase extraction and fringe reconstruction. The first simulated fringe pattern, the first reconstructed fringe pattern and the difference between them are shown in Figure 5.27. Clearly, the difference

between the simulated pattern and the reconstructed pattern has an order of magnitude of  $10^{-4}$ , which indicates the good performance of the hybrid kPSA-GIA.

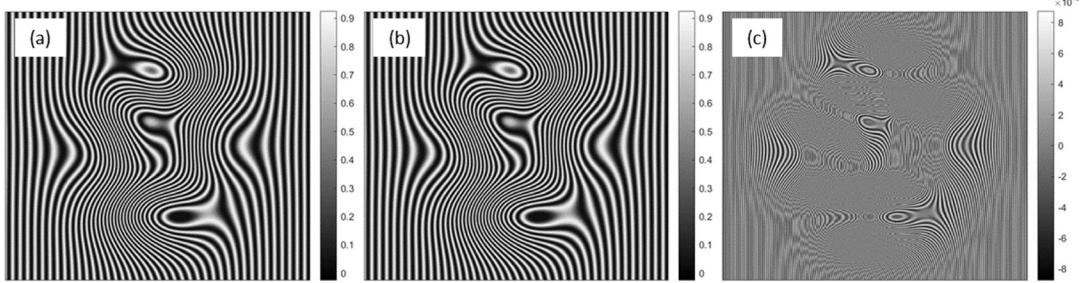


Figure 5.27. The first fringe patterns.

(a) The simulated; (b) The reconstructed; (c) The difference.

To ease the process of the kPSA selection, we further propose to use the least-squares algorithm as the initializing kPSA for the hybrid kPSA-GIA as the least-squares algorithm can be used for any frame number and any phase shifts as long as they are known, which leads to an LSA-GIA. We tested the LSA-GIA again with the simulation case discussed in Figure 5.26 and obtain similar error measures as those from the kPSA-GIA in that figure.

## 5.7.Experimental verification

To verify the conclusions drawn in the above chapters, we conducted an optical testing experiment with the Fizeau interferometer sketched in Figure 5.28. We change the wavelength of the laser to introduce phase shifts [202]. In the experiments, the image size is  $N_x=N_y=1024$ . The distance between the wedged window and the wafer is 25mm and the wavelength of the laser is 635nm.

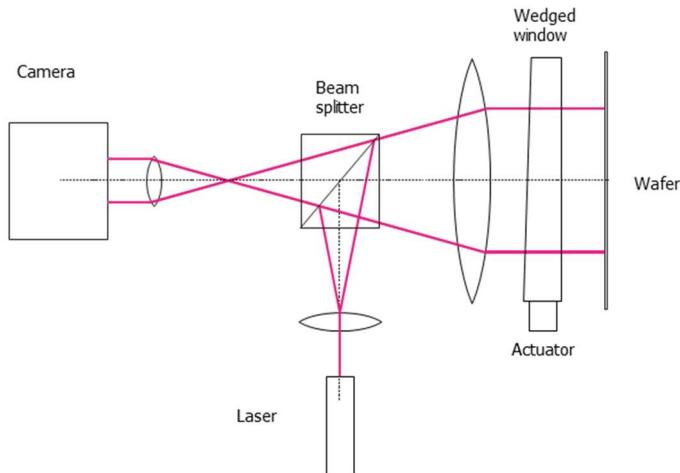


Figure 5.28.Sketch of the Fizeau interferometer [207].

The AVE and the WDFT are selected to compare with the uPSAs and the kPSA-GIA due to their unique performance. Thus, we capture a set of seven fringe patterns with phase shifts of  $i\pi/2$  ( $0 \leq i \leq 6$ ). Thus, the wavelength change between two adjacent frames is about 2pm. Meanwhile, in this experiment, the intensity harmonics and the phase-shift non-uniformities are purposely introduced by controlling the nonlinear response of the camera with  $\gamma=1.5$  [204] and exciting the mounting of the interferometer's reference by a mechanical actuator [142], respectively. The first, the fourth and the seventh frames are shown in Figure 5.29.

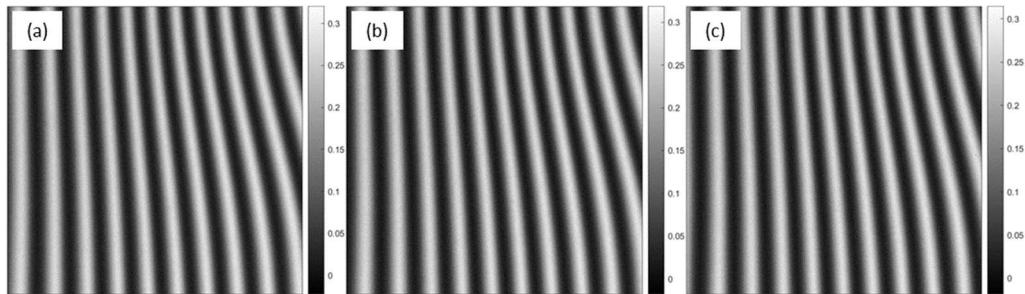


Figure 5.29.Fringe patterns.

(a) The first; (b) The fourth; (c) The seventh.

The AVE and the WDFT are used to form the hybrid algorithms of the AVE-GIA and the WDFT-GIA. These two hybrid algorithms together with the AVE, the WDFT, the AIA, the GIA, the PCA and the VU are used to process the fringe patterns. The middle rows of the phases extracted by these PSAs are shown in blue in Figure 5.30. The phase results are increasing from left to right, so a common linear part is removed. Then, these phase results are filtered by a Gaussian-weighted moving average filter with a filter length of 20 pixels and a standard deviation of 4 pixels, as overlaid in red in Figure 5.30. After we subtract the filtered values from the raw phase in Figure 5.30, we obtain the noise-like residuals whose standard deviation values are calculated and shown in Table 5.2. It is seen that the phase extracted by all the PSAs are similar in this particular experiment without the presence of serious error sources. The uPSAs and their respective hybrid kPSA-uPSAs, although initialized differently, give the same standard deviations, which are slightly smaller than those from the kPSAs.

Table 5.2.Standard deviation value of the phase noise (Unit: rad).

	AVE	WDFT	AIA	GIA	PCA	VU	AVE-GIA	WDFT-GIA
STDV	0.098	0.087	0.060	0.054	0.060	0.065	0.054	0.054

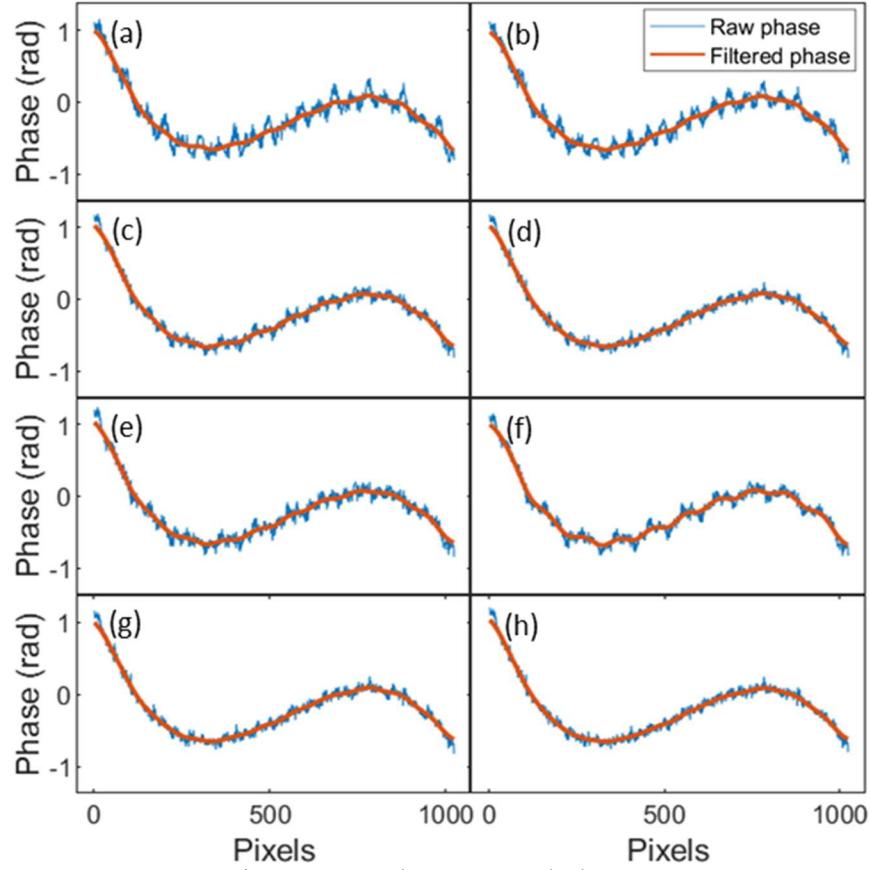


Figure 5.30. The extracted phases.

(a) by the AVE; (b) by the WDFT; (c) by the AIA; (d) by the GIA; (e) by the PCA; (f) by the VU; (g) by the AVE-GIA; (h) by the WDFT-GIA.

## 5.8. Summary

Phase shifting is an important technique in optical metrology. Many PSAs with known and unknown phase shifts have been proposed to extract the phase accurately from the phase-shifted fringe patterns. Although many performance evaluation and comparison works have been intensively carried out for the kPSAs, very few were done for the uPSAs. In this chapter, we examine the performance of four well-performing uPSAs including the AIA, the GIA, the PCA and the VU. Our simulations with random phase shifts evaluate the uPSAs' pre-requisites on fringe parameters and compared the performance of the uPSAs' with different error sources. Based on it, two restrictions on fringe patterns when using the uPSAs are summarized; and the best error immunity of the GIA is also noticed. Our simulations with regular phase shifts compared kPSAs with the uPSAs, in which we noticed it is possible to find a suitable algorithm to tackle the

presenting error source(s). However, a proposer kPSA selection requires the user to have good mastering of the kPSAs. Meanwhile, the uPSAs with better accuracy are very strong competitors and can be powerful aids to the kPSAs. Based on this principle, a hybrid kPSA-GIA, which uses a kPSA to initialize the GIA, and uses the GIA to extract the phase is proposed. The hybrid kPSA-GIA both improves the accuracy of the kPSAs and reliefs the GIA's restriction on fringe density. Besides, the hybrid kPSA-GIA provides another advantage that it can reconstruct the fringe patterns to compare with the input fringe patterns for performance evaluation.



# **Chapter 6 Parallel phase extraction algorithm for phase-shifted fringe patterns with unknown phase shifts\***

## **6.1. Background**

Now, we focus on the slow speed of the optimization based uPSAs. As the AIA is the most recognized uPSA and most optimization based uPSAs have a similar structure to the AIA, we accelerate the AIA in this chapter. As mentioned, the AIA iteratively estimates the phase and the phase shifts. These iterations enable the AIA's phase extraction without prior knowledge of the phase shifts and also cause a much higher computation cost compared to the trigonometric identities based kPSAs. In modern precision metrology applications, the number of frames could reach more than ten [208] and the number of pixels per fringe pattern could reach more than tens of millions [209]. Hence, large frame numbers and/or large pixel number make the inefficiency problem in phase extraction much more serious for the AIA. For example, we test the AIA with 13 frames of fringe patterns. For each fringe pattern, the image size is  $2048 \times 2048$ . The AIA needs 3.13 minutes to extract the phase by using a MATLAB implementation. The time cost could reduce to 19.6 seconds if a C++ implementation is used. However, the speed is still not fast enough for industrial applications requiring real-time phase extractions. GPU based parallel computing [189, 210] has been considered in one implementation of the AIA [189]. In that implementation, the phase extraction is moved to the GPU and accelerated by GPU based parallel computing; while the phase shifts estimation is kept on the CPU and is accelerated by processing only the central 2% of the fringe patterns. However, when there are insufficient phase variations over the area for phase shifts estimation, the AIA may not converge as shown in chapter 3. Besides, the hybrid GPU-CPU implementation needs frequent data transfer between the CPU and GPU which leads to unoptimized speed. With a detailed evaluation of the AIA's algorithm structure, we accelerate the AIA by fully GPU based parallel computing and call this implementation of the AIA as gAIA. Compared to a sequential implementation of the AIA which is executed on a single-core-CPU, the

---

\* The work in this chapter has been published as Y. Chen, T. Wang, and Q. Kemao, "Parallel advanced iterative algorithm for phase extraction with unknown phase-shifts," *Optics and Lasers in Engineering*, vol. 138, p. 106408, 2021.

gAIA achieves  $500\times$  speedup without accuracy scarification. Compared to the hybrid GPU-CPU implementation of the AIA,  $10\times$  speedup is reached. With a normal NVIDIA RTX 2080 Ti GPU, the gAIA can reach real-time phase extraction, *i.e.*, only 26.55 ms is needed for the gAIA to extract phase from the fringe patterns with  $F=13$  and  $N=N_x\times N_y=2048\times 2048$ .

## 6.2.The AIA's parallelization

In this sub-chapter, the AIA's structure is evaluated in detail for acceleration. Besides, the convergence issues which might be induced by the cropping of the fringe patterns are discussed. Meanwhile, the AIA's parallel decomposition is discussed and analyzed.

### 6.2.1.The AIA's structure

The principle of the AIA is introduced in chapter 2.2.3.1 in detail. A flowchart of the AIA is shown in Figure 6.1 where we numbered the steps of the AIA. From the flowchart, a few remarks are made:

- (i) The AIA is sequential. The next iteration can only start when the current iteration is finished. These iterations are implemented by a while loop (Step 2). It is worth mentioning that Step 2 is only the shell of the loop while Steps 3-6 are not included in Step 2.
- (ii) Step 4 corresponds to the pixel-wise phase extraction. Step 5 corresponds to the frame-wise phase shifts extraction. These two steps are the most time-consuming steps of the AIA. In our test, for one iteration, Step 4 takes 1.3s and Step 5 takes 1.1s when fringe patterns with  $F=13$  and  $N=N_x\times N_y=2048\times 2048$  are processed by a single-CPU-core sequential C++ implementation of the AIA.
- (iii) Meanwhile, other steps of the AIA are computational trivial, and their time consumptions are negligible. Thus, Step 4 and Step 5 are the bottlenecks of the AIA.

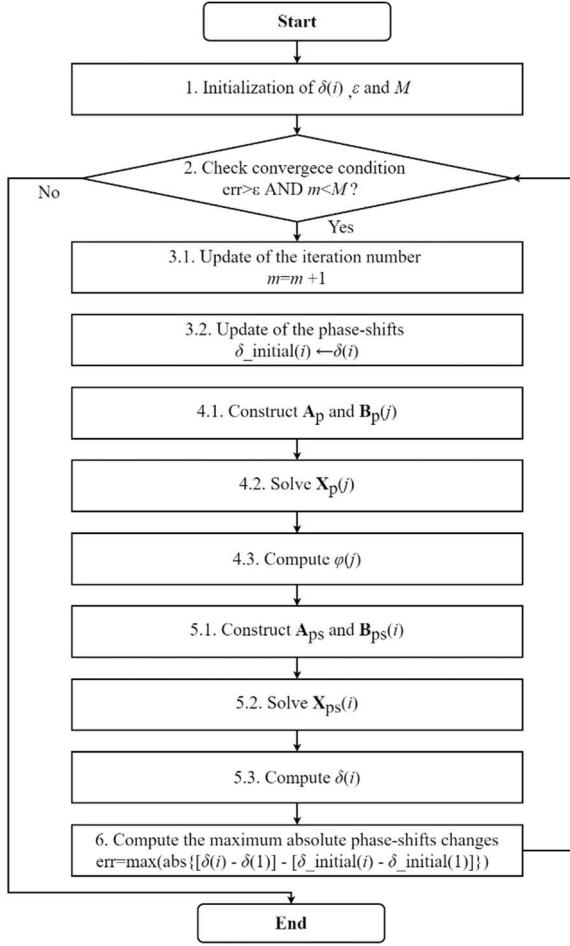


Figure 6.1.Flowchart of the AIA.

### 6.2.2.Data cropping and the risk

As shown in Eq. (2.31), the matrix  $\mathbf{A}_{ps}$  and the vector  $\mathbf{B}_{ps}(i)$  are needed for the determination of the phase shifts. The computations of  $\mathbf{A}_{ps}$  and  $\mathbf{B}_{ps}(i)$  need to sum across all the pixels. In the hybrid GPU-CPU implementation of the AIA [189], the phase shifts estimation is carried out in the central 2% of fringe patterns for acceleration. This idea is feasible as the phase shift of each frame is assumed to be constant. Meanwhile, as shown in chapter 3.3.2, when the area for phase shifts estimation does not have enough fringe density variations, the AIA will fail, since  $\mathbf{A}_{ps}$  is near singular. Cropping of the fringe patterns will reduce the fringe density. Thus, it may lead to divergence of the AIA. We carry out a simulation with four frames of fringe patterns whose image size is  $N=256\times 256$ . The phase shifts are set as  $\delta(i)=i\pi/2$ , and the fringe density is set as one fringe over the frame. The first fringe pattern is shown in Figure 6.2(a). Fringe cropping is used in the phase shifts estimation. Different cropping percentages are tested. The root-mean-square errors

(RMSE) of the phases are calculated and plotted against the cropping percentages in Figure 6.2(b). When the cropping percentage is lower than 25%, *i.e.*, less than half fringe over the area, the RMSEs of the AIA are large. Hence, cropping the fringe patterns is not a wise idea when the fringe density of the fringe patterns is not high enough.

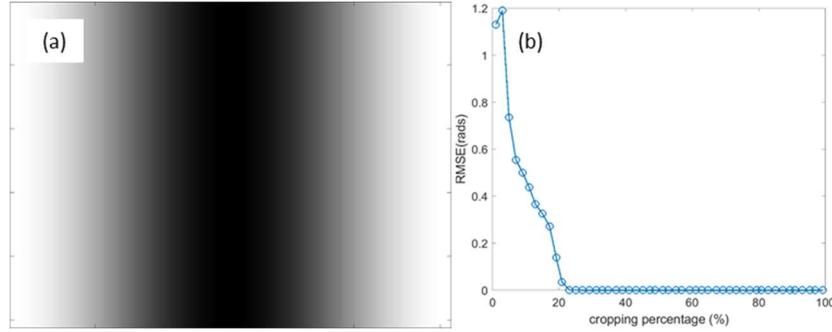


Figure 6.2. Accuracy performance of the AIA with fringe pattern cropping.

(a) First frame of the linear fringe patterns; (b) RMSEs of the phase.

### 6.2.3. Parallel decomposition of the AIA

GPU based parallel computing has diverse applications in optical metrology [210-214]. In where, four common parallel patterns have been adopted [210] including: (i) the pointwise pattern for operations which are identical, independent and need to execute a massive number of times; (ii) the tiling pattern for operations which involve the neighboring pixels; (iii) the divide-and-conquer pattern for operations which can be divided into different smaller parallelizable tasks; and (iv) the rendering and interpolation pattern for operations which visualize the 3D scenes from 2D images.

As discussed in chapter 6.2.1, the bottlenecks of the AIA are Step 4 and Step 5. With a detailed evaluation of the AIA, we found that both Step 4 and Step 5 can be parallelized for GPU acceleration. In the meantime, almost all the other steps can also be moved to GPU by using the four common parallel patterns. In Table 6.1, each sub-step of the AIA is matched to a parallel pattern. Here are some explanations.

- (i) In Step 4, the phase is extracted by three sub-steps, (4.1)  $\mathbf{A}_p$  and  $\mathbf{B}_p$  construction, (4.2)  $\mathbf{X}_p$  computation, and (4.3) phase computations. Each sub-step in Step 4 can be parallelized by a pointwise pattern as they are all identical and pixel independent operations. Besides, since the pixel number is large, using the pointwise pattern could fully utilize the potential of a GPU.

- (ii) In Step 5, there are also three sub-steps, (5.1)  $\mathbf{A}_{ps}$  and  $\mathbf{B}_{ps}$  construction, (5.2)  $\mathbf{X}_{ps}$  computation, and (5.3) phase shifts computations. Typically, the frame numbers of the fringe patterns used in the AIA are between three [169] and a hundred [215]. The inter-frame computational burden is not heavy. However, the intra-frame operations, such as the constructions of  $\mathbf{A}_{ps}$  and  $\mathbf{B}_{ps}$  which include the sum across all the pixels in a frame, are computationally expensive. Thus, Sub-step 5.1 is the bottleneck of Step 5 and is not suitable for using the pointwise pattern. Instead, the reduction pattern which is a special divide-and-conquer pattern is used for the long vector summations. To keep Sub-steps 5.2 and 5.3 on GPU for avoiding the time-consuming large size data transfer, these two sub-steps are executed with the pointwise pattern.
- (iii) Step 3 and Step 5 are with light computational cost. To avoid data transfer between the CPU and GPU, both Step 3 and Step 5 (except Sub-step 3.1) are moved to GPU using suitable parallel patterns.
- (iv) With such a parallel decomposition, the AIA is almost 100% parallelized. Thus, almost all the steps of the AIA could be accelerated by a GPU.

Table 6.1. Mapping from the steps of the AIA to the parallel patterns

Steps/Sub-steps of the AIA	Implementation pattern
1. Initialize $\delta(i), \varepsilon, M$	CPU (one time)
2. While loop	CPU (sequential)
3.1. Update the iteration number	CPU (sequential)
3.2. Update the $\delta(i)$ values	Pointwise (pixel-wise)
4.1. Construct $\mathbf{A}_p$ and $\mathbf{B}_p(j)$	Pointwise (pixel-wise)
4.2. Solve $\mathbf{X}_p(j)$	Pointwise (pixel-wise)
4.3. Compute $\phi(j)$	Pointwise (pixel-wise)
5.1. Construct $\mathbf{A}_{ps}$ and $\mathbf{B}_{ps}(i)$	Divide-and-conquer (reduction pattern)
5.2. Solve $\mathbf{X}_{ps}(i)$	Pointwise (frame-wise)
5.3. Compute $\delta(i)$	Pointwise (frame-wise)
6. Compute error	Pointwise (frame-wise)

### 6.3.gAIA's CUDA implementation

To eliminate the bottlenecks of the AIA, the gAIA is proposed. We select the CUDA programming model (CUDA) and a NVIDIA's GPU for the implementation of the gAIA due to the simple integration into existing C/C++ programs and excellent parallelism [216]. The mapping from

CUDA to a NVIDIA's GPU and the implementation of the gAIA will be discussed in this sub-chapter.

### 6.3.1.Mapping between CUDA and a NVIDIA's GPU

CUDA is designed and developed by NVIDIA [216]. The mapping from CUDA to a NVIDIA's GPU is schematically illustrated in Figure 6.3. In CUDA, a parallelizable problem is divided into smaller sub-problems and assigned to grids of blocks. In every block, there are up to 1024 threads, which are the smallest execution unit of CUDA. The CUDA blocks will be mapped to the GPU's Streaming Multiprocessors (SMs) and the smaller sub-problems are processed on the SMs. Generally, a GPU core's performance is much worse than a CPU core. However, a GPU has many cores, thus, if all the cores are used for processing at the same time, the GPU's performance is much better than the CPU [210]. Meanwhile, it is worth mentioning that in CUDA a CPU is still needed as a "host" to assign the works to the GPU which is also known as the "device".

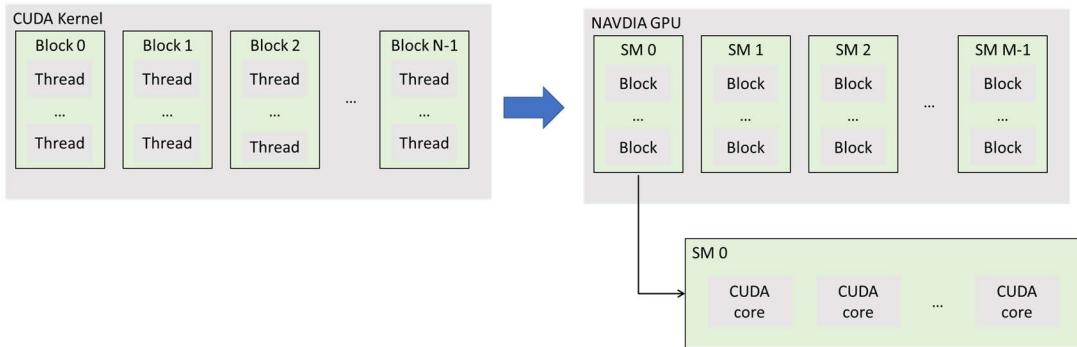


Figure 6.3.Mapping from CUDA kernel to GPU.

### 6.3.2.gAIA's flowchart and implementation

We show the flowchart of the gAIA in Figure 6.4 with detailed explanations in the following sub-chapters.

#### 6.3.2.1.Data transfer from CPU to GPU

At the starting point, the gAIA needs to transfer the fringe patterns from the CPU to the GPU's global memory as shown in Sub-step T.1. Meanwhile, Sub-step T.2 is the memory transfer of *err* from the device to the host. At the end, the extracted phase should be transferred from the device back to the host.

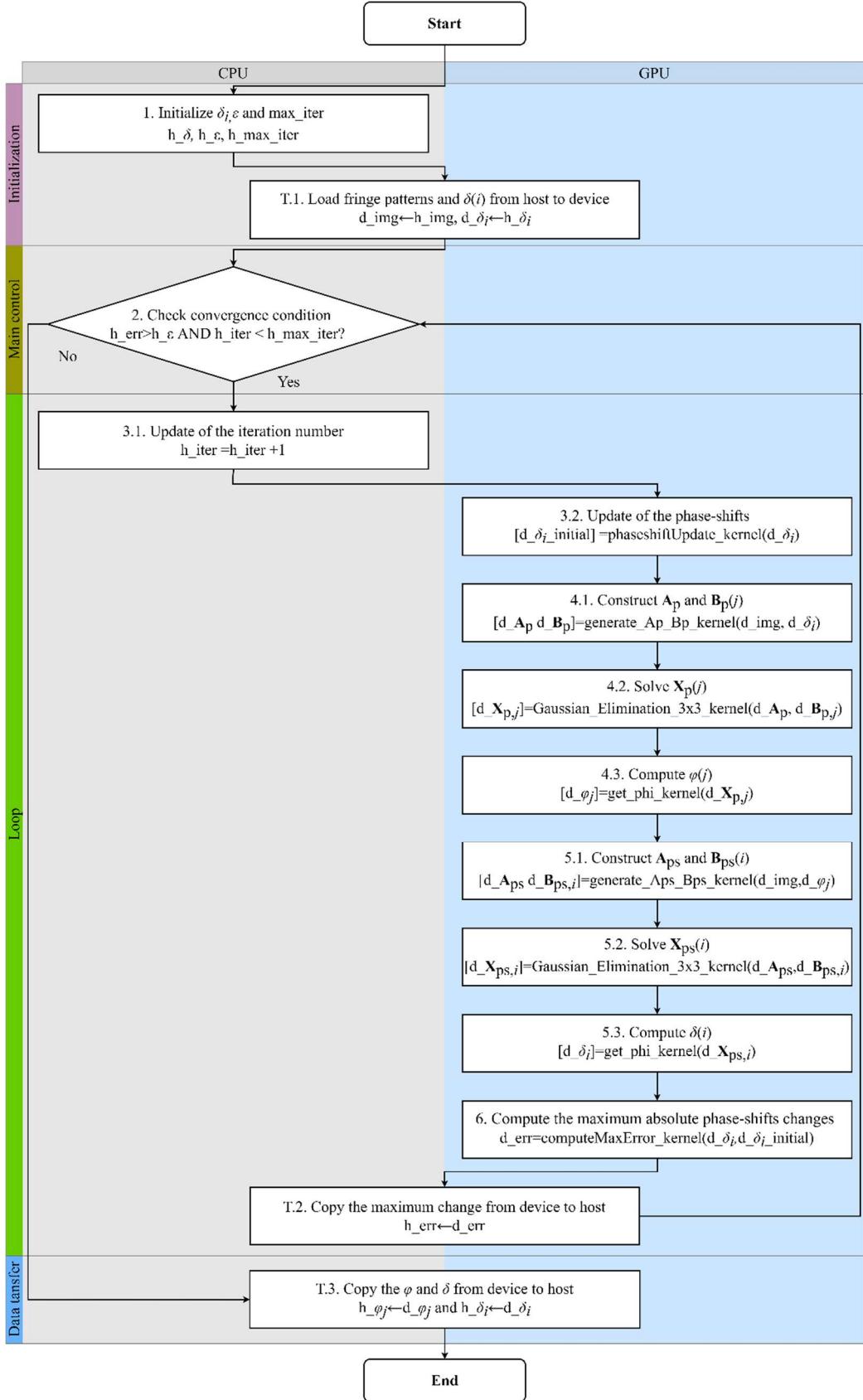


Figure 6.4. Flowchart of the gAIA.

### 6.3.2.2. Pointwise pattern

The pointwise pattern is used multiple times in both the phase extraction and the phase shifts estimation. Sub-step 4.3 is used as an example to introduce the implementation details. The phase calculation applies to the  $N$  pixels within a frame, which consumes a long time when the image size is large. Furthermore, the process of each pixel is independent. Hence, GPU acceleration is suitable for this step by allocating  $N$  CUDA threads for the  $N$  pixels' computation. The other steps which could be parallelized by the pointwise pattern are implemented in similar manners.

### 6.3.2.3. Reduction pattern.

Sub-step 5.1 in the flowchart of the gAIA is the parallel reduction applied to construct  $\mathbf{A}_{ps}$  and  $\mathbf{B}_{ps}(i)$ . There are different levels of reductions to carry out the entire parallel reduction algorithm in CUDA, including a grid-level reduction in global memory, block-level reduction in a block's shared memory, and thread-level reduction in per-thread local registers [217, 218]. We use the computation of  $\sum_{j=0}^{N-1} I(i; j)$ , *i.e.*, the computation of the first element in  $\mathbf{B}_{ps}(i)$ , as an example for better explanation. In the block-level reduction, multiple CUDA threads are spawned. For each thread, to avoid the possible bank conflict in the block's shared memory, the addition of the two elements located at  $id \times n + i$  and  $id \times n + i + n/2$  is carried out as shown in a schematic model of adding  $n = \text{floor}(N/k)$  (where  $k$  represents the number of blocks) elements in one CUDA block in Figure 6.5. The next round of summation uses a half number of the used threads for the summation of the previous results. This process is repeated until the final element is obtained. This final element is the partial sum of this block. After that, the partial sums from all the blocks will be summed up at the grid-level for the final result.

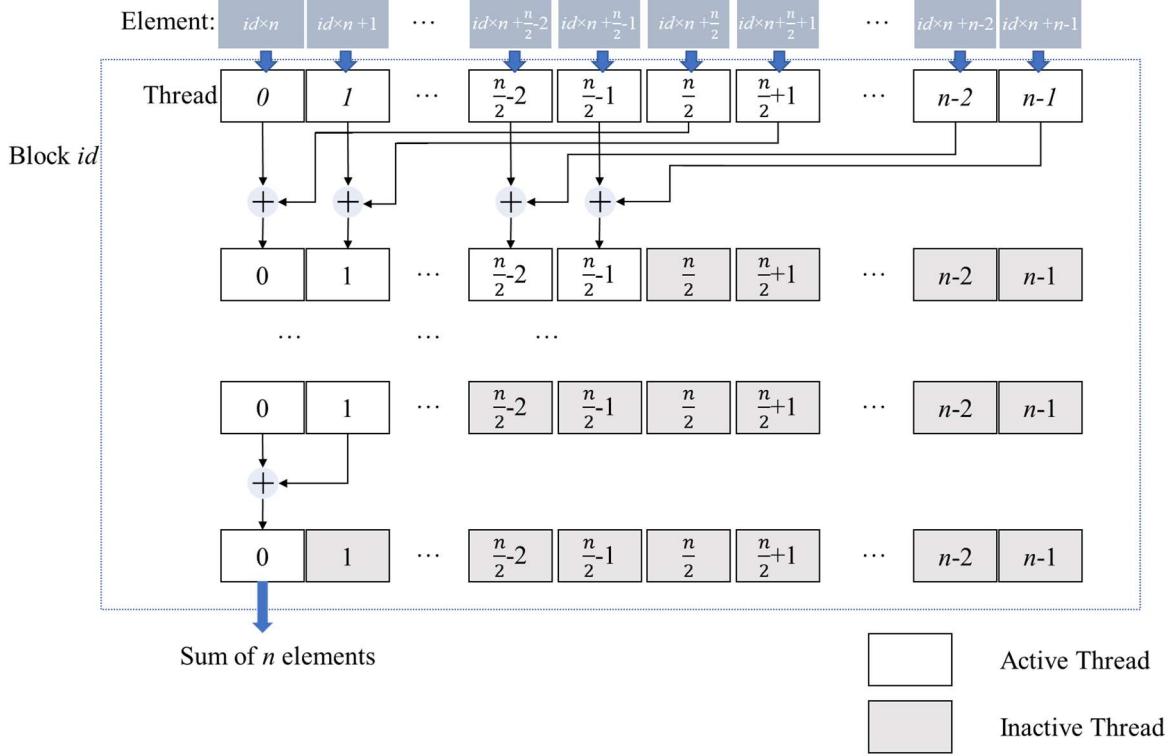


Figure 6.5.Schematic model of CUDA's parallel reduction kernel.

#### 6.3.2.4.Linear system solver on GPU cores

As shown in Sub-steps 4.2 and 5.2, each pixel or each frame has a  $3 \times 3$  linear equation to be solved. For these sub-steps, we first select the cuSOLVER as the solver for the linear systems [216, 219]. However, cuSOLVER is designed for general linear systems and the problem scale should be large enough to benefit from the speedup by CUDA. Therefore, in order to use cuSOLVER in Sub-step 4.2, a sparse matrix whose size is  $3N \times 3N$  needs to be formed by stacking the  $3 \times 3$  matrices  $\mathbf{A}_p$  along the diagonal direction. Besides,  $\mathbf{B}_p$  needs to be rearranged into a single long vector with  $3N$  elements. However, the speed of using cuSOLVER in such a manner is even slower than the sequential CPU solver.

In fact, solving a  $3 \times 3$  linear equation is not a heavy computational burden for a CUDA thread. Thus, each CUDA thread can be assigned to solve one  $3 \times 3$  linear equation. Since  $\mathbf{A}_p$  is symmetric and nonsingular in general, Cholesky decomposition, LU factorization and Gaussian elimination with partial pivoting (GEPP) are potential solvers [220]. The time complexities of the three solvers are all  $[O(n^3)]$  as only one vector will be processed in each CUDA core. Therefore, the simpler GEPP is used as the linear equation solver for each thread.

## 6.4.Experimental verification

Different implementations of the AIA are coded and compared with the gAIA. In this sub-chapter, the accuracy and the speed of the different implementations of the AIA are evaluated.

### 6.4.1.Algorithm implementations and data collection for comparison

The following implementations of the AIA are tested including:

- (i) Single-CPU-core: Following the flowchart in Figure 6.1, a sequential implementation of the AIA is coded and will be executed on a single-CPU-core.
- (ii) Multi-CPU-core: A parallel implementation of the AIA is coded and will be executed on multiple CPU cores. This implementation is with the following optimizations: (a) the pointwise and reduction patterns operations are parallelized by the OpenMP [221]; (b) The linear equations are solved by the Intel's math kernel library optimized LAPACK routine ?sysv [222]; and (c) we enabled the AVX2 instruction;
- (iii) Hybrid GPU-CPU cropping AIA: The existing Hybrid GPU-CPU AIA is coded based on its Ref. [189];
- (iv) Hybrid GPU-CPU: The existing Hybrid GPU-CPU AIA but avoided the data cropping as suggested in chapter 6.2.2.
- (v) gAIA: The proposed gAIA is coded based on the description in chapter 6.3.

The Microsoft Visual Studio C++ 2017 is used as the compiler for all the algorithms. For CUDA, we use version 10.2. The tests are executed on a workstation with an NVIDIA GeForce RTX 2080 Ti (68 SMs with 4352 CUDA cores and 11 GB RAM), an Intel® Xeon® E5-2699 v3 processors (18 cores, 2.3Ghz main frequency) and 256 GB RAM.

We used a Fizeau interferometer to collect the testing fringe patterns. In the interferometer, we used a wedged optical window as the reference and use a silicon wafer as the test object. We changed the wavelength of the interferometer to introduce the phase shifts. [202]. The distance between the wedged window and the wafer is 25mm and the wavelength of the laser is 635nm. We collected thirteen frames of phase-shifted fringe patterns ( $N=4096 \times 4096$ ) whose nominal phase shifts are  $\delta(i)=i\pi/4$ . Thus, the wavelength change between two adjacent frames is about 1pm. Three frames of fringe patterns are shown in Figure 6.6, for illustration.

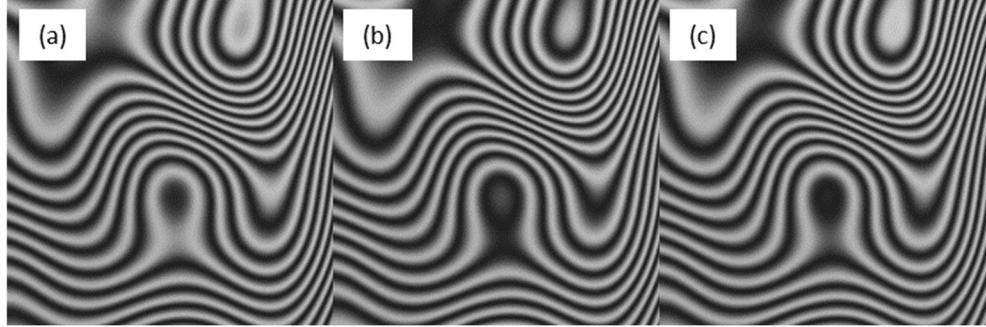


Figure 6.6. The phase-shifted fringe patterns.

(a) 1st; (b) 7th; and (c) 13th.

#### 6.4.2.The accuracy

Firstly, the phase is extracted from the fringe patterns by the Single-CPU-core AIA with a *double-precision*. This phase is shown in Figure 6.7(a) and is used as the benchmarking phase. The phases extracted by other implementations of the AIA will be compared with this phase for accuracy evaluation. Next, the phases are extracted by the Single-CPU-core AIA with a *single-precision*, the gAIA with a *single-precision*, and the gAIA with a *double-precision*. The difference between these phases and the benchmarking phase is shown in Figure 6.7(b)-Figure 6.7(d). The peak to valley values of the differences between the benchmarking phase and the phases extracted by the Single-CPU-core AIA with *single-precision*, the gAIA with *double-precision* and the gAIA with *single-precision* are  $2.0 \times 10^{-3}$  rad,  $4.1 \times 10^{-8}$  rad and  $1.5 \times 10^{-6}$  rad, respectively. Comparing the sequential summation with the parallel reduction based summation, a worse accuracy performance has been noticed [223], which is the reason of the Single-CPU-core AIA with a *single-precision*'s poor accuracy performance. Thus, when executing the AIA on a CPU, a *double-precision* is suggested. When executing the AIA on a GPU, both the *single-precision* and the *double-precision* could be considered. Meanwhile, for the gAIA, the implementation with a *single-precision* is 5 times faster than the implementation with a *double-precision*. Therefore, for the later comparisons, we keep the CPU operations with the *double-precision* and use the *single-precision* for GPU operations.

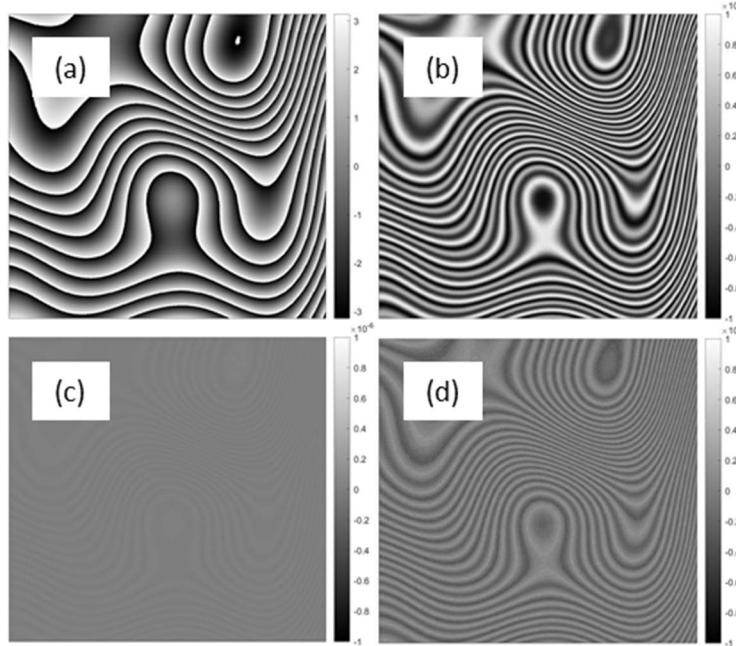


Figure 6.7. Accuracy performance of different implementations of the AIA.

- (a) Benchmark phase; (b) difference between the benchmark and the Single-CPU-core AIA with *single*-precision; (c) difference between the benchmark and the gAIA with *single*-precision; (d) difference between the benchmark and the gAIA with *double*-precision.

#### 6.4.3. The speed

From the results, we have the following remarks:

- (i) We used pixel numbers of  $N=1024 \times 1024$ ,  $2048 \times 2048$  and  $4096 \times 4096$  which are achieved by image down sampling and frame numbers of  $F=3$ , 8 and 13 which are achieved by selecting the first few fringe patterns to form nine experiment conditions. We tested all the implementations of the AIA mentioned in chapter 6.4.1. Their time costs for different experimental conditions are shown in Table 6.2.
- (ii) It is worth mentioning that all the implementations of the AIA converge with 9 iterations.
- (iii) It takes the least amount of time for the gAIA in all experimental conditions. The gAIA is  $500\times$  times faster than the Single-CPU-core AIA on average.
- (iv) Compared to the hybrid GPU-CPU AIA, the gAIA is  $10\times$  times faster while the GPU memory consumptions are similar which are both about 1.4 GB for the experiment with 13 frames of  $4096 \times 4096$  pixels fringe patterns.

(v) Real-time phase extraction could only be achieved with the gAIA, *i.e.*, extracting phase with less than 40 ms. Meanwhile, real-time phase extraction is only possible for the image sizes of  $1024 \times 1024$  and  $2048 \times 2048$ , but not  $4096 \times 4096$  by using the current GPU;

Table 6.2. Time consumption of the five implementations of the AIA (unit: ms).

Frame Number <i>M</i>	Image size <i>N</i>	Single-CPU-core	18-CPU-core	Hybrid GPU-CPU	Hybrid GPU-CPU cropping	gAIA
3	1024×1024	1671.16	198.14	750	93.75	7.42
	2048×2048	7342.99	718.53	2781.25	156.25	23.19
	4096×4096	26751.20	2693.75	11031.25	375	78.13
8	1024×1024	3353.40	339.74	1781.25	111.56	8.95
	2048×2048	13149.70	1328.36	7906.25	218.75	24.55
	4096×4096	56218.90	4941.70	24687.5	562.5	87.85
13	1024×1024	5128.63	453.25	2625	125	9.39
	2048×2048	19595.50	1711.39	10093.75	281.25	26.55
	4096×4096	80583.70	6771.37	40875	1062.5	98.42

## 6.5. Summary

In this chapter, we focus on the slow computation issue of the AIA. To solve this problem, the algorithm's structure is analyzed to show that full parallelization is possible. Next, the fully-parallelized GPU based AIA (gAIA) is proposed and realized by CUDA. Without compromising the accuracy, the gAIA is  $500\times$  times faster than the Single-CPU-core AIA. Meanwhile, comparing to the hybrid GPU-CPU AIA which crops the fringe patterns and risks convergence, the gAIA is still  $10\times$  faster. Furthermore, the AIA's phase extraction is real-time by using the gAIA. In our test, the gAIA only needs 24.5 ms to extract the phase from 13 frames of  $2048 \times 2048$  pixels' fringe patterns, which is much less than the 40ms requirement of real-time extraction.



# Chapter 7 Conclusion, and future works

## 7.1. Conclusions

Phase extraction is critical in fringe based optical metrology, for which, many phase-shifting algorithms (PSAs) were proposed. These PSAs are either with known phase shifts and based on simple triangular identities (abbreviated as kPSA) or with unknown phase shifts and based on advanced techniques (abbreviated as uPSA). Subsequently, many development and evaluation studies were conducted. However, these evaluation works are mainly for the kPSAs. In this thesis, we focused on the less-studied uPSAs and made four research contributions.

- (i) We detailly evaluated the robustness and the accuracy of different uPSAs with the help of a large number of simulations. From these simulations, we confirmed the pre-requisites of the uPSAs on different fringe parameters including the phase, the phase shifts, the background intensity and fringe amplitude; compared the accuracy of different uPSAs when there are different types of errors in the fringe patterns; and revealed the advantages and the disadvantages of different uPSAs.
- (ii) We improved the robustness, accuracy, predictability and speed of a well-recognized uPSA called advanced iterative algorithm (AIA). With a detailed understanding of the influence of different fringe parameters on the AIA, an overall phase error model was established and three individual improving practices were proposed. By combining the fringe acquisition and phase extraction, an enhanced AIA which can confirm the AIA's success with the desired accuracy is proposed. To speed up the AIA, we proposed a fully GPU-accelerated AIA (gAIA) which is based on a detailed algorithm's structure analysis and is realized by CUDA.
- (iii) We proposed a simple, accurate and robust general iterative algorithm (GIA) to deal with the most common errors in the fringe pattern. The GIA incorporates the error sources into the fringe model and estimates them together with the phase. Based on their different natures, the unknowns in the GIA are classified into three groups: (i) the background intensity and the fringe amplitude, (ii) the phase and (iii) the phase-shift related parameters. Subsequently, these three groups of unknowns are optimized alternatively by the

Levenberg-Marquart method. In addition, the background intensity and the fringe amplitude are estimated within a small window for better immunity to random noise.

(iv) We compared the accuracy of the uPSAs and kPSAs. From the compression, we noticed that it is always possible to find a suitable algorithm to tackle the presenting error source(s). However, a proper kPSA selection requires the user to have good mastering of the kPSAs. In comparison, the uPSAs with better accuracy are very strong competitors and can be powerful aids to the kPSAs. Based on this principle, a hybrid kPSA-GIA which uses a kPSA to initialize the GIA and use the GIA to extract the phase is proposed. The hybrid kPSA-GIA both improves the accuracy of the kPSAs and reliefs the GIA's restriction on fringe density.

## 7.2.Future work

### 7.2.1.Improve the computation efficiency of the GIA

As demonstrated in chapters 4 and 5, the GIA is the uPSA with an outstanding accuracy performance. However, the GIA also has an unsatisfactory computation efficiency, *i.e.*, the GIA needs about 10 minutes to extract the phase from eight frames of fringe patterns with  $1024 \times 1024$  pixels when a MATLAB implementation is used. In our tests, for one iteration, the background intensity and fringe amplitude estimation takes 14.23s, the phase estimation takes 1.26s, and the phase shifts estimation takes 72.82s. The slow computation of the GIA will affect the adaptability in industrial applications. One direct solution is to use GPU based parallel computing to accelerate the GIA. The bottleneck of the GIA is the phase shifts estimation which could be accelerated by the reduction pattern discussed in chapter 6. Besides, phase calculation can be accelerated by the pointwise pattern. Meanwhile, since the background intensity and fringe amplitude estimation is within a window, a tiling pattern should be used in this step. By using these common parallel patterns, the GIA could also be fully parallelized on GPU which will definitely improve the speed of the GIA.

Another possible solution is to use other hardware such as FPGA. As demonstrated in [224-226], in various applications, the FPGA can out-perform the CPUs and GPUs. Thus, it is rational to use the FPGAs to accelerate the GIA.

### 7.2.2.Further improvement of the GIA's accuracy

In the fringe pattern model used in the GIA, the four most common error sources are considered. However, there are some other error sources which are less common but also might be presented in the fringe patterns such as the speckle noise [227], the background intensity variation [228], etc.

The speckle noise is a noise which occurs in all coherent imaging systems. The speckle noise has its similarity to the random noise, but they are not the same. Two fringe patterns, one with the speckle noise and one with the random noise are shown in Figure 7.1. Clearly, there are differences between the two fringe patterns, thus, different models are needed for the speckle noise and random noise, which can be considered in future works.

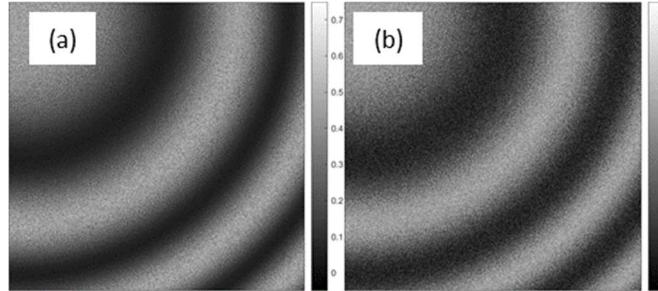


Figure 7.1.Fringe pattern with noise.

(a) With the speckle noise; (b) With the additive white random noise.

In the fringe pattern models of most PSAs, including both the kPSAs and the uPSAs, the background intensity and fringe amplitude are assumed to be constant across frames. However, this assumption might be violated when an unstable source is used. In this case, the fringe pattern model needs refinement.

### 7.2.3.Comparison between the kPSA, the uPSA and deep learning based phase-shifting algorithms

In this thesis, the kPSAs and the uPSAs are compared to reveal the advantages and disadvantages of both categories of PSAs. Very recently, there is another phase extraction method attracting increasing attention which is the deep learning based phase-shifting algorithm (dlPSA) [229, 230]. These dlPSAs have demonstrated good performance on accuracy and speed. However, there are only limited comparison works between the dlPSAs and the kPSAs. Especially, these dlPSAs are never compared with the uPSAs. The comparison among the dlPSAs, the uPSAs and the kPSAs will definitely be useful and interesting.

#### 7.2.4.Optimization based uPSA with global optimization techniques

So far, all the optimization based uPSAs use local optimization techniques. Thus, initializations are unavoidable. For example, the AIA uses the linear least-squares optimization and needs to initialize the phase shifts; and the GIA uses the Levenberg–Marquardt optimization and needs to initialize the phase and the phase shifts. In the GIA, the phase and the phase shifts are initialized by other PSA. This initialization step is avoidable if we use a global optimization technique such as the genetic algorithm [231], the stochastic tunneling algorithm [232] or the simulated annealing algorithm[233].

# Author's Publications

- Y. Chen and Q. Kemao, "Advanced iterative algorithm for phase extraction: performance evaluation and enhancement," *Optics express*, vol. 27, no. 26, pp. 37634-37651, 2019.
- Y. Chen, T. Wang, and Q. Kemao, "Parallel advanced iterative algorithm for phase extraction with unknown phase-shifts," *Optics and Lasers in Engineering*, vol. 138, p. 106408, 2021.
- Y. Chen and Q. Kemao, "General iterative algorithm for phase extraction from fringe patterns with random phase-shifts, intensity harmonics and non-uniform phase-shift distribution," *Optics Express*, vol. 29, no. 19, pp. 30905-30926, 2021.
- Y. Chen and Q. Kemao, "Phase-shifting algorithms with known and unknown phase shifts: comparison and hybrid,". *Optics Express*, vol. 30, no. 5, pp. 8275-8302, 2022.



# Reference

- [1] W. Osten, "Optical metrology: the long and unstoppable way to become an outstanding measuring tool," in *Speckle 2018: VII International Conference on Speckle Metrology*, 2018, vol. 10834: International Society for Optics and Photonics, p. 1083402.
- [2] A. G. Marrugo, F. Gao, and S. Zhang, "State-of-the-art active optical techniques for three-dimensional surface metrology: a review," *JOSA A*, vol. 37, no. 9, pp. B60-B77, 2020.
- [3] P. J. De Groot, "A review of selected topics in interferometric optical metrology," *Reports on Progress in Physics*, vol. 82, no. 5, p. 056101, 2019.
- [4] R. Hocken, N. Chakraborty, and C. Brown, "Optical metrology of surfaces," *CIRP annals*, vol. 54, no. 2, pp. 169-183, 2005.
- [5] M. Takeda, "Fourier fringe analysis and its application to metrology of extreme physical phenomena: a review," *Applied Optics*, vol. 52, no. 1, pp. 20-29, 2013.
- [6] F. B. Leloup, G. Obein, M. R. Pointer, and P. Hanselaer, "Toward the soft metrology of surface gloss: A review," *Color Research & Application*, vol. 39, no. 6, pp. 559-570, 2014.
- [7] L. Yang, X. Xie, L. Zhu, S. Wu, and Y. Wang, "Review of electronic speckle pattern interferometry (ESPI) for three dimensional displacement measurement," *Chinese Journal of Mechanical Engineering*, vol. 27, no. 1, pp. 1-13, 2014.
- [8] S. Catalucci, A. Thompson, J. Eastwood, S. Piano, D. T. Branson, and R. Leach, "Optical metrology for digital manufacturing: A review."
- [9] R. Gooch, "Optical metrology in manufacturing automation," *Sensor Review*, 1998.
- [10] J.-A. Beraldin, B. Carrier, D. MacKinnon, and L. Cournoyer, "Characterization of Triangulation-Based 3D Imaging Systems Using Certified Artifacts," *NCSLI Measure*, vol. 7, no. 4, pp. 50-60, 2012/12/01 2012, doi: 10.1080/19315775.2012.11721620.
- [11] R. Attota *et al.*, "TSOM method for semiconductor metrology," in *Metrology, Inspection, and Process Control for Microlithography XXV*, 2011, vol. 7971: International Society for Optics and Photonics, p. 79710T.
- [12] A. J. Den Boef, "Optical metrology of semiconductor wafers in lithography," in *International Conference on Optics in Precision Engineering and Nanotechnology (icOPEN2013)*, 2013, vol. 8769: International Society for Optics and Photonics, p. 876907.
- [13] A. J. den Boef, "Optical wafer metrology sensors for process-robust CD and overlay control in semiconductor device manufacturing," *Surface Topography: Metrology and Properties*, vol. 4, no. 2, p. 023001, 2016.
- [14] D. Dixit *et al.*, "Advanced applications of scatterometry based optical metrology," in *Metrology, Inspection, and Process Control for Microlithography XXXI*, 2017, vol. 10145: International Society for Optics and Photonics, p. 101451H.
- [15] N. G. Orji *et al.*, "Metrology for the next generation of semiconductor devices," *Nature electronics*, vol. 1, no. 10, pp. 532-547, 2018.
- [16] S. Sederberg, F. Kong, F. Hufnagel, C. Zhang, E. Karimi, and P. B. Corkum, "Vectorized optoelectronic control and metrology in a semiconductor," *Nature Photonics*, vol. 14, no. 11, pp. 680-685, 2020.
- [17] Y. Shimizu, L.-C. Chen, D. W. Kim, X. Chen, X. Li, and H. Matsukuma, "An insight into optical metrology in manufacturing," *Measurement Science and Technology*, vol. 32, no. 4, p. 042003, 2021.

- [18] H. Schwenke, U. Neuschaefer-Rube, T. Pfeifer, and H. Kunzmann, "Optical methods for dimensional metrology in production engineering," *CIRP Annals*, vol. 51, no. 2, pp. 685-699, 2002.
- [19] M. Mathew, A. Ellenberg, S. Esola, M. McCarthy, I. Bartoli, and A. Kortsos, "Multiscale deformation measurements using multispectral optical metrology," *Structural control and health Monitoring*, vol. 25, no. 6, p. e2166, 2018.
- [20] M. M. Barak, A. Sharir, and R. Shahar, "Optical metrology methods for mechanical testing of whole bones," *The Veterinary Journal*, vol. 180, no. 1, pp. 7-14, 2009.
- [21] J. C. Wyant and K. Creath, "Basic wavefront aberration theory for optical metrology," *Applied optics and optical engineering*, vol. 11, no. part 2, pp. 28-39, 1992.
- [22] K. Creath, J. Schmit, and J. Wyant, "Optical metrology of diffuse surfaces," *Optical shop testing*, pp. 756-807, 2007.
- [23] R. Huang, P. Su, T. Horne, G. Brusa, and J. H. Burge, "Optical metrology of a large deformable aspherical mirror using software configurable optical test system," *Optical Engineering*, vol. 53, no. 8, p. 085106, 2014.
- [24] J. C. Wyant, "Interferometric Optical Metrology," *OSA Century of Optics*, 2015.
- [25] X. Zhang *et al.*, "Correlation approach for quality assurance of additive manufactured parts based on optical metrology," *Journal of Manufacturing Processes*, vol. 53, pp. 310-317, 2020.
- [26] A. M. Urbanowicz, P. Ebersbach, D. Likhachev, D. Mezerette, and C. Hartig, "In-line control of material properties of SiOC: H based low-k dielectrics utilizing optical metrology."
- [27] R. Logan, "Optical metrology of thin films," Massachusetts Institute of Technology, 1997.
- [28] R. Shahar and S. Weiner, "Insights into whole bone and tooth function using optical metrology," *Journal of Materials Science*, vol. 42, no. 21, pp. 8919-8933, 2007.
- [29] C. Sivior, S. Grantham, D. Williamson, W. Proud, and J. Field, "Novel measurements of material properties at high rates of strain using speckle metrology," *The Imaging Science Journal*, vol. 57, no. 6, pp. 326-332, 2009.
- [30] J. Tyson, "Advanced material properties measurements with optical metrology," 2011.
- [31] A. M. Urbanowicz, P. Ebersbach, D. Likhachev, D. Mezerette, and C. Hartig, "In-line control of material properties of SiOC: H based low-k dielectrics utilizing optical metrology AM: Advanced metrology," in *2017 28th Annual SEMI Advanced Semiconductor Manufacturing Conference (ASMC)*, 2017: IEEE, pp. 24-28.
- [32] L. Blunt, P. J. Bills, and L. Fleming, "Metrology for Bio Systems," 2015.
- [33] P. K. Upputuri, M. Pramanik, K. M. Nandigana, and M. P. Kothiyal, "Multi-colour microscopic interferometry for optical metrology and imaging applications," *Optics and Lasers in Engineering*, vol. 84, pp. 10-25, 2016.
- [34] E. P. Koumoulos *et al.*, "Metrology and nano-mechanical tests for nano-manufacturing and nano-bio interface: Challenges & future perspectives," *Materials & Design*, vol. 137, pp. 446-462, 2018.
- [35] H. Lee *et al.*, "In-cell overlay metrology by using optical metrology tool," in *Metrology, Inspection, and Process Control for Microlithography XXXII*, 2018, vol. 10585: International Society for Optics and Photonics, p. 105851D.
- [36] P. K. Upputuri and M. Pramanik, "Phase shifting white light interferometry using colour CCD for optical metrology and bio-imaging applications," in *Quantitative Phase Imaging IV*, 2018, vol. 10503: International Society for Optics and Photonics, p. 105032E.
- [37] A. Chatterjee, V. Bhatia, and S. Prakash, "Biometric spoof detection using optical metrology and signal processing," Discipline of Electrical Engineering, IIT Indore, 2021.
- [38] L. Heflinger, R. Wuerker, and R. E. Brooks, "Holographic interferometry," *Journal of Applied Physics*, vol. 37, no. 2, pp. 642-649, 1966.

- [39] C. Polhemus, "Two-wavelength interferometry," *Applied Optics*, vol. 12, no. 9, pp. 2071-2074, 1973.
- [40] M. Françon, "Optical interferometry," in *Neutron interferometry*, 1979.
- [41] K. Creath, "Phase-shifting speckle interferometry," *Applied Optics*, vol. 24, no. 18, pp. 3053-3058, 1985.
- [42] K. Creath, "V phase-measurement interferometry techniques," in *Progress in optics*, vol. 26: Elsevier, 1988, pp. 349-393.
- [43] M. S. Chapman *et al.*, "Optics and interferometry with Na 2 molecules," *Physical review letters*, vol. 74, no. 24, p. 4783, 1995.
- [44] R. Bamler and P. Hartl, "Synthetic aperture radar interferometry," *Inverse problems*, vol. 14, no. 4, p. R1, 1998.
- [45] P. A. Rosen *et al.*, "Synthetic aperture radar interferometry," *Proceedings of the IEEE*, vol. 88, no. 3, pp. 333-382, 2000.
- [46] A. Quirrenbach, "Optical interferometry," *Annual Review of Astronomy and Astrophysics*, vol. 39, no. 1, pp. 353-401, 2001.
- [47] J. C. Wyant, "White light interferometry," in *Holography: A Tribute to Yuri Denisyuk and Emmett Leith*, 2002, vol. 4737: International Society for Optics and Photonics, pp. 98-107.
- [48] J. D. Monnier, "Optical interferometry in astronomy," *Reports on Progress in Physics*, vol. 66, no. 5, p. 789, 2003.
- [49] K. Harada, A. Tonomura, Y. Togawa, T. Akashi, and T. Matsuda, "Double-biprism electron interferometry," *Applied Physics Letters*, vol. 84, no. 17, pp. 3229-3231, 2004.
- [50] N. Brock *et al.*, "Dynamic interferometry," in *Novel Optical Systems Design and Optimization VIII*, 2005, vol. 5875: International Society for Optics and Photonics, p. 58750F.
- [51] R. Snieder, "The theory of coda wave interferometry," *Pure and Applied geophysics*, vol. 163, no. 2, pp. 455-473, 2006.
- [52] Y. I. Ostrovsky, M. M. Butusov, and G. Ostrovskaya, *Interferometry by holography*. Springer, 2013.
- [53] M. Crosetto, O. Monserrat, M. Cuevas-González, N. Devanthéry, and B. Crippa, "Persistent scatterer interferometry: A review," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 115, pp. 78-89, 2016.
- [54] S. Yang and G. Zhang, "A review of interferometry for geometric measurement," *Measurement Science and Technology*, vol. 29, no. 10, p. 102001, 2018.
- [55] W. Klempert, "Review of particle identification by time of flight techniques," *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 433, no. 1-2, pp. 542-553, 1999.
- [56] N. Mirsaleh - Kohan, W. D. Robertson, and R. N. Compton, "Electron ionization time - of - flight mass spectrometry: Historical review and current applications," *Mass spectrometry reviews*, vol. 27, no. 3, pp. 237-285, 2008.
- [57] S. Angeletti and M. Ciccozzi, "Matrix-assisted laser desorption ionization time-of-flight mass spectrometry in clinical microbiology: An updating review," *Infection, Genetics and Evolution*, vol. 76, p. 104063, 2019.
- [58] J. A. Jalkio, R. C. Kim, and S. K. Case, "Three dimensional inspection using multistripe structured light," *Optical Engineering*, vol. 24, no. 6, p. 246966, 1985.
- [59] J. Mundy and G. Porter, "A three-dimensional sensor based on structured light," in *Three-dimensional machine vision*: Springer, 1987, pp. 3-61.
- [60] B. Girod and S. Scherock, "Depth from defocus of structured light," in *Optics, Illumination, and Image Sensing for Machine Vision IV*, 1990, vol. 1194: International Society for Optics and Photonics, pp. 209-215.

- [61] B. Wagner, P. Stüber, T. Wissel, R. Bruder, A. Schweikard, and F. Ernst, "Accuracy analysis for triangulation and tracking based on time - multiplexed structured light," *Medical physics*, vol. 41, no. 8Part1, p. 082701, 2014.
- [62] L. Cai, Q. Liu, and X. Yang, "Phase-shift extraction and wave-front reconstruction in phase-shifting interferometry with arbitrary phase steps," *Optics letters*, vol. 28, no. 19, pp. 1808-1810, 2003.
- [63] X. Xu *et al.*, "Simple direct extraction of unknown phase shift and wavefront reconstruction in generalized phase-shifting interferometry: algorithm and experiments," *Optics letters*, vol. 33, no. 8, pp. 776-778, 2008.
- [64] X. Xu, L. Cai, Y. Wang, and R. Yan, "Direct phase shift extraction and wavefront reconstruction in two-step generalized phase-shifting interferometry," *Journal of Optics*, vol. 12, no. 1, p. 015301, 2009.
- [65] K. Okada, H. Sakuta, T. Ose, and J. Tsujiuchi, "Separate measurements of surface shapes and refractive index inhomogeneity of an optical element using tunable-source phase shifting interferometry," *Applied optics*, vol. 29, no. 22, pp. 3280-3285, 1990.
- [66] W. F. Decraemer, J. J. Dirckx, and W. R. J. Funnell, "Shape and derived geometrical parameters of the adult, human tympanic membrane measured with a phase-shift moire interferometer," *Hearing research*, vol. 51, no. 1, pp. 107-121, 1991.
- [67] Y. Kim, K. Hibino, N. Sugita, and M. Mitsuishi, "Surface profile measurement of a highly reflective silicon wafer by phase-shifting interferometry," *Applied Optics*, vol. 54, no. 13, pp. 4207-4213, 2015.
- [68] X. Dai *et al.*, "Shape measurement with modified phase-shift lateral shearing interferometry illumination and radial basis function," *Applied optics*, vol. 56, no. 21, pp. 5954-5960, 2017.
- [69] P. K. Rastogi and P. Jacquot, "Measurement of difference deformation using speckle interferometry," *Optics letters*, vol. 12, no. 8, pp. 596-598, 1987.
- [70] C. Joenathan, B. Franze, P. Haible, and H. Tiziani, "Speckle interferometry with temporal phase evaluation for measuring large-object deformation," *Applied optics*, vol. 37, no. 13, pp. 2608-2614, 1998.
- [71] J. Degrieck, W. Van Paepengem, and P. Boone, "Application of digital phase-shift shadow moiré to micro deformation measurements of curved surfaces," *Optics and Lasers in Engineering*, vol. 36, no. 1, pp. 29-40, 2001.
- [72] D. Sandwell, R. Mellors, X. Tong, M. Wei, and P. Wessel, "Open radar interferometry software for mapping surface deformation," ed: Wiley Online Library, 2011.
- [73] Y.-C. Chu, W.-Y. Chang, K.-H. Chen, J.-H. Chen, B.-C. Tsai, and K. Y. Hsu, "Full-field refractive index measurement with simultaneous phase-shift interferometry," *Optik*, vol. 125, no. 13, pp. 3307-3310, 2014.
- [74] K. Meiners-Hagen and A. Abou-Zeid, "Refractive index determination in length measurement by two-colour interferometry," *Measurement Science and Technology*, vol. 19, no. 8, p. 084004, 2008.
- [75] Z. Tian *et al.*, "Refractive index sensing with Mach-Zehnder interferometer based on concatenating two single-mode fiber tapers," *IEEE Photonics Technology Letters*, vol. 20, no. 8, pp. 626-628, 2008.
- [76] Z. Tian and S. S.-H. Yam, "In-line single-mode optical fiber interferometric refractive index sensors," *Journal of Lightwave Technology*, vol. 27, no. 13, pp. 2296-2306, 2009.
- [77] K. Meiners-Hagen, T. Meyer, J. Mildner, and F. Pollinger, "SI-traceable absolute distance measurement over more than 800 meters with sub-nanometer interferometry by two-color inline refractivity compensation," *Applied Physics Letters*, vol. 111, no. 19, p. 191104, 2017.

- [78] B. Abbott *et al.*, "GWTC-1: a gravitational-wave transient catalog of compact binary mergers observed by LIGO and Virgo during the first and second observing runs," *Physical Review X*, vol. 9, no. 3, p. 031040, 2019.
- [79] P. L. Teoh, B. Shirinzadeh, C. W. Foong, and G. Alici, "The measurement uncertainties in the laser interferometry-based sensing and tracking technique," *Measurement*, vol. 32, no. 2, pp. 135-150, 2002.
- [80] G. Giuliani and S. Donati, "Laser interferometry," *Unlocking dynamical diversity: optical feedback effects on semiconductor lasers*, pp. 217-255, 2005.
- [81] G. M. Harry, P. Fritschel, D. A. Shaddock, W. Folkner, and E. S. Phinney, "Laser interferometry for the big bang observer," *Classical and Quantum Gravity*, vol. 23, no. 15, p. 4887, 2006.
- [82] G. Marra *et al.*, "Ultrastable laser interferometry for earthquake detection with terrestrial and submarine cables," *Science*, vol. 361, no. 6401, pp. 486-490, 2018.
- [83] O. Y. Kwon, D. M. Shough, and R. A. Williams, "Stroboscopic phase-shifting interferometry," *Optics letters*, vol. 12, no. 11, pp. 855-857, 1987.
- [84] G. Lai and T. Yatagai, "Generalized phase-shifting interferometry," *JOSA A*, vol. 8, no. 5, pp. 822-827, 1991.
- [85] H. Medecki, E. Tejnil, K. Goldberg, and J. Bokor, "Phase-shifting point diffraction interferometer," *Optics letters*, vol. 21, no. 19, pp. 1526-1528, 1996.
- [86] P. de Groot, "Phase shifting interferometry," in *Optical measurement of surface topography*: Springer, 2011, pp. 167-186.
- [87] U. Schnars, C. Falldorf, J. Watson, and W. Jüptner, "Digital holography," in *Digital Holography and Wavefront Sensing*: Springer, 2015, pp. 39-68.
- [88] T. S. Huang, "Digital holography," *Proceedings of the IEEE*, vol. 59, no. 9, pp. 1335-1346, 1971.
- [89] L. P. IAroslavskii and N. S. Merzliakov, "Methods of digital holography," *Moscow Izdatel Nauka*, 1977.
- [90] T. M. Kreis, M. Adams, and W. P. Jüptner, "Methods of digital holography: a comparison," in *Optical Inspection and Micromeasurements II*, 1997, vol. 3098: International Society for Optics and Photonics, pp. 224-233.
- [91] Y. Hung, "Shearography: a new optical method for strain measurement and nondestructive testing," *Optical engineering*, vol. 21, no. 3, p. 213391, 1982.
- [92] Y. Hung, "Shearography: a novel and practical approach for nondestructive inspection," *Journal of Nondestructive Evaluation*, vol. 8, no. 2, pp. 55-67, 1989.
- [93] Y. Hung and H. Ho, "Shearography: An optical measurement technique and applications," *Materials science and engineering: R: Reports*, vol. 49, no. 3, pp. 61-87, 2005.
- [94] D. Francis, R. Tatam, and R. Groves, "Shearography technology and applications: a review," *Measurement science and technology*, vol. 21, no. 10, p. 102001, 2010.
- [95] S. Diddams and J.-C. Diels, "Dispersion measurements with white-light interferometry," *JOSA B*, vol. 13, no. 6, pp. 1120-1129, 1996.
- [96] K. G. Larkin, "Efficient nonlinear algorithm for envelope detection in white light interferometry," *JOSA A*, vol. 13, no. 4, pp. 832-843, 1996.
- [97] D. X. Hammer, A. J. Welch, G. D. Noojin, R. J. Thomas, D. J. Stolarski, and B. A. Rockwell, "Spectrally resolved white-light interferometry for measurement of ocular dispersion," *JOSA A*, vol. 16, no. 9, pp. 2092-2102, 1999.
- [98] A. Sinha, A. H. Vijay, and U. Sinha, "On the superposition principle in interference experiments," *Scientific reports*, vol. 5, no. 1, pp. 1-9, 2015.
- [99] C. Conte, G. de Alteriis, R. Schiano Lo Moriello, D. Accardo, and G. Rufino, "Drone Trajectory Segmentation for Real-Time and Adaptive Time-Of-Flight Prediction," *Drones*, vol. 5, no. 3, p. 62, 2021.

- [100] S. May, D. Dröschel, 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: IEEE, pp. 1673-1678.
- [101] A. Hata and D. Wolf, "Road marking detection using LIDAR reflective intensity data and its application to vehicle localization," in *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, 2014: IEEE, pp. 584-589.
- [102] Z. Wang, Q. Zhou, and Y. Shuang, "Three-dimensional reconstruction with single-shot structured light dot pattern and analytic solutions," *Measurement*, vol. 151, p. 107114, 2020.
- [103] Z. Wang and Y. Yang, "Single-shot three-dimensional reconstruction based on structured light line pattern," *Optics and Lasers in Engineering*, vol. 106, pp. 10-16, 2018.
- [104] S. S. Gorthi and P. Rastogi, "Fringe projection techniques: whither we are?," *Optics and lasers in engineering*, vol. 48, no. ARTICLE, pp. 133-140, 2010.
- [105] H. Guo and P. S. Huang, "Face recognition based on fringe pattern analysis," *Optical Engineering*, vol. 49, no. 3, p. 037201, 2010.
- [106] F. Lilley, M. J. Lalor, and D. R. Burton, "Robust fringe analysis system for human body shape measurement," *Optical Engineering*, vol. 39, no. 1, pp. 187-195, 2000.
- [107] T. Y.-F. Chen, Y.-L. Lo, Z.-H. Lin, and J.-Y. Lin, "Simultaneous extraction of profile and surface roughness of 3D SLM components using fringe projection method," *Rapid Prototyping Journal*, 2021.
- [108] S. R. Arellano, *Studies of Volcanic Plumes with Remote Spectroscopic Sensing Techniques-Doas and Ftir Measurements on Volcanoes of the Network for Observation of Volcanic and Atmospheric Change*. Chalmers Tekniska Hogskola (Sweden), 2014.
- [109] G. Häusler and S. Ettl, "Limitations of optical 3D sensors," in *Optical measurement of surface topography*: Springer, 2011, pp. 23-48.
- [110] G. Reid, "Automatic fringe pattern analysis: a review," *Optics and lasers in engineering*, vol. 7, no. 1, pp. 37-68, 1986.
- [111] G. Reid, "Image processing techniques for fringe pattern analysis," in *Optical Testing and Metrology II*, 1989, vol. 954: International Society for Optics and Photonics, pp. 468-477.
- [112] P. Varman and C. Wykes, "Smoothing of speckle and moiré fringes by computer processing," *Optics and Lasers in Engineering*, vol. 3, no. 2, pp. 87-100, 1982.
- [113] M. Schlüter, "Analysis of holographic interferograms with a tv picture system," *Optics & Laser Technology*, vol. 12, no. 2, pp. 93-95, 1980.
- [114] J. H. Yi, S. H. Kim, Y. K. Kwak, and Y. W. Lee, "Peak movement detection method of an equally-spaced fringe for precise position measurement," *Optical Engineering*, vol. 41, no. 2, pp. 428-434, 2002.
- [115] J. B. Schemm and C. M. Vest, "Fringe pattern recognition and interpolation using nonlinear regression analysis," *Applied optics*, vol. 22, no. 18, pp. 2850-2853, 1983.
- [116] G. A. Mastin and D. C. Ghiglia, "Digital extraction of interference fringe contours," *Applied optics*, vol. 24, no. 12, pp. 1727-1728, 1985.
- [117] K. J. Gasvik, K. G. Robbersmyr, and T. Vadseth, "Fringe location by means of a zero crossing algorithm," in *Fringe Pattern Analysis*, 1989, vol. 1163: International Society for Optics and Photonics, pp. 64-70.
- [118] T. Yatagai, S. Nakadate, M. Idesawa, and H. Saito, "Automatic fringe analysis using digital image processing techniques," *Optical Engineering*, vol. 21, no. 3, p. 213432, 1982.
- [119] Y. Hung and J. Hovanesian, "Fast detection of residual stresses in an industrial environment by thermoplastic-based shearography," in *1990 SEM Spring Conference on Experimental Mechanics*, 1990, pp. 769-775.

- [120] J. Trolinger, "Automated data reduction in holographic interferometry," *Optical Engineering*, vol. 24, no. 5, p. 245840, 1985.
- [121] W. Funnell, "Image processing applied to the interactive analysis of interferometric fringes," *Applied Optics*, vol. 20, no. 18, pp. 3245-3250, 1981.
- [122] V. Parthiban and R. S. Sirohi, "Use of gray scale coding in labeling closed loop fringe patterns," in *Fringe Pattern Analysis*, 1989, vol. 1163: International Society for Optics and Photonics, pp. 77-82.
- [123] J. Wang and D. E. Silva, "Wave-front interpretation with Zernike polynomials," *Applied optics*, vol. 19, no. 9, pp. 1510-1518, 1980.
- [124] D. Malacara-Hernandez, M. Carpio-Valadez, and J. J. Sanchez-Mondragon, "Wavefront fitting with discrete orthogonal polynomials in a unit radius circle," *Optical engineering*, vol. 29, no. 6, pp. 672-675, 1990.
- [125] U. Mieth and W. Osten, "Three methods for the interpolation of phase values between fringe pattern skeletons," in *Interferometry'89*, 1989, vol. 1121: International Society for Optics and Photonics, pp. 151-154.
- [126] J. H. Bruning, D. R. Herriott, J. Gallagher, D. Rosenfeld, A. White, and D. Brangaccio, "Digital wavefront measuring interferometer for testing optical surfaces and lenses," *Applied optics*, vol. 13, no. 11, pp. 2693-2703, 1974.
- [127] D. J. Bone, H.-A. Bachor, and R. J. Sandeman, "Fringe-pattern analysis using a 2-D Fourier transform," *Applied Optics*, vol. 25, no. 10, pp. 1653-1660, 1986.
- [128] T. R. Judge and P. Bryanston-Cross, "A review of phase unwrapping techniques in fringe analysis," *Optics and Lasers in Engineering*, vol. 21, no. 4, pp. 199-239, 1994.
- [129] X. Su and W. Chen, "Reliability-guided phase unwrapping algorithm: a review," *Optics and Lasers in Engineering*, vol. 42, no. 3, pp. 245-261, 2004.
- [130] N. H. Ching, D. Rosenfeld, and M. Braun, "Two-dimensional phase unwrapping using a minimum spanning tree algorithm," *IEEE Transactions on Image Processing*, vol. 1, no. 3, pp. 355-365, 1992.
- [131] T. J. Flynn, "Consistent 2-D phase unwrapping guided by a quality map," in *IGARSS'96. 1996 International Geoscience and Remote Sensing Symposium*, 1996, vol. 4: IEEE, pp. 2057-2059.
- [132] M. A. Herráez, D. R. Burton, M. J. Lalor, and M. A. Gdeisat, "Fast two-dimensional phase-unwrapping algorithm based on sorting by reliability following a noncontinuous path," *Applied optics*, vol. 41, no. 35, pp. 7437-7444, 2002.
- [133] Y. Li, J. Zhu, and W. Shen, "Phase unwrapping algorithms, respectively, based on path-following and discrete cosine transform," *Optik*, vol. 119, no. 11, pp. 545-547, 2008.
- [134] R. M. Goldstein, H. A. Zebker, and C. L. Werner, "Satellite radar interferometry: Two-dimensional phase unwrapping," *Radio science*, vol. 23, no. 4, pp. 713-720, 1988.
- [135] J. Huntley, "Noise-immune phase unwrapping algorithm," *Applied optics*, vol. 28, no. 16, pp. 3268-3270, 1989.
- [136] J. Huntley and J. Buckland, "Characterization of sources of  $2\pi$  phase discontinuity in speckle interferograms," *JOSA A*, vol. 12, no. 9, pp. 1990-1996, 1995.
- [137] M. A. Herraez, D. R. Burton, M. J. Lalor, and D. B. Clegg, "Robust unwrapper for two-dimensional images," in *Vision Systems: Sensors, Sensor Systems, and Components*, 1996, vol. 2784: International Society for Optics and Photonics, pp. 106-110.
- [138] D. Malacara, *Optical shop testing*. John Wiley & Sons, 2007, pp. 547-655
- [139] R. Schödel, A. Nicolaus, and G. Bönsch, "Phase-stepping interferometry: methods for reducing errors caused by camera nonlinearities," *Applied optics*, vol. 41, no. 1, pp. 55-63, 2002.
- [140] P. Hariharan, "Digital phase-stepping interferometry: effects of multiply reflected beams," *Applied optics*, vol. 26, no. 13, pp. 2506-2507, 1987.

- [141] L. Deck, *Multiple-surface phase-shifting interferometry* (International Symposium on Optical Science and Technology). SPIE, 2001.
- [142] P. J. De Groot, "Vibration in phase-shifting interferometry," *JOSA A*, vol. 12, no. 2, pp. 354-365, 1995.
- [143] N. Bobroff, "Residual errors in laser interferometry from air turbulence and nonlinearity," *Applied optics*, vol. 26, no. 13, pp. 2676-2682, 1987.
- [144] Y. Reibel, M. Jung, M. Bouhifd, B. Cunin, and C. Draman, "CCD or CMOS camera noise characterisation," *The European Physical Journal-Applied Physics*, vol. 21, no. 1, pp. 75-80, 2003.
- [145] J. A. N. Buytaert and J. J. J. Dirckx, "Study of the performance of 84 phase-shifting algorithms for interferometry," *Journal of Optics*, vol. 40, no. 3, p. 114, 2011/09/21 2011, doi: 10.1007/s12596-011-0044-y.
- [146] P. Rastogi and E. Hack, *Phase estimation in optical interferometry*. CRC, 2014, pp. 235-270.
- [147] D. Malacara, M. Servin, and Z. Malacara, *Interferogram analysis for optical testing*. CRC, 2018, pp. 359-391.
- [148] K. Creath, "Phase-measurement interferometry techniques," in *Progress in optics*, vol. 26: Elsevier, 1988, pp. 349-393.
- [149] J. Schwider, R. Burow, K.-E. Elssner, J. Grzanna, R. Spolaczyk, and K. Merkel, "Digital wave-front measuring interferometry: some systematic error sources," *Applied optics*, vol. 22, no. 21, pp. 3421-3432, 1983.
- [150] C. Morgan, "Least-squares estimation in phase-measurement interferometry," *Optics Letters*, vol. 7, no. 8, pp. 368-370, 1982.
- [151] J. E. Greivenkamp, "Generalized data reduction for heterodyne interferometry," *Optical Engineering*, vol. 23, no. 4, p. 234350, 1984.
- [152] J. Schmit and K. Creath, "Extended averaging technique for derivation of error-compensating algorithms in phase-shifting interferometry," *Applied optics*, vol. 34, no. 19, pp. 3610-3619, 1995.
- [153] J. Schmit and K. Creath, "Window function influence on phase error in phase-shifting algorithms," *Applied optics*, vol. 35, no. 28, pp. 5642-5649, 1996.
- [154] Y. Surrel, "Phase stepping: a new self-calibrating algorithm," *Applied optics*, vol. 32, no. 19, pp. 3598-3600, 1993.
- [155] K. Larkin and B. Oreb, "Design and assessment of symmetrical phase-shifting algorithms," *JOSA A*, vol. 9, no. 10, pp. 1740-1748, 1992.
- [156] Y. Surrel, "Design of algorithms for phase measurements by the use of phase stepping," *Applied optics*, vol. 35, no. 1, pp. 51-60, 1996.
- [157] K. Hibino, B. F. Oreb, D. I. Farrant, and K. G. Larkin, "Phase-shifting algorithms for nonlinear and spatially nonuniform phase shifts," *JOSA A*, vol. 14, no. 4, pp. 918-930, 1997.
- [158] 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, 1966/01 1966, doi: 10.1088/0026-1394/2/1/005.
- [159] G. Stoilov and T. Dragostinov, "Phase-stepping interferometry: Five-frame algorithm with an arbitrary step," *Optics and Lasers in Engineering*, vol. 28, no. 1, pp. 61-69, 1997/09/01/ 1997, doi: [https://doi.org/10.1016/S0143-8166\(96\)00048-6](https://doi.org/10.1016/S0143-8166(96)00048-6).
- [160] C. T. Farrell and M. A. Player, "Phase step measurement and variable step algorithms in phase-shifting interferometry," *Measurement Science and Technology*, vol. 3, no. 10, pp. 953-958, 1992/10/01 1992, doi: 10.1088/0957-0233/3/10/003.
- [161] C. Farrell and M. Player, "Phase-step insensitive algorithms for phase-shifting interferometry," *Measurement science and technology*, vol. 5, no. 6, p. 648, 1994.

- [162] L. Z. Cai, Q. Liu, and X. L. Yang, "Phase-shift extraction and wave-front reconstruction in phase-shifting interferometry with arbitrary phase steps," *Optics Letters*, vol. 28, no. 19, pp. 1808-1810, 2003/10/01 2003, doi: 10.1364/OL.28.001808.
- [163] Y. Xu, Y. Wang, Y. Ji, H. Han, and W. Jin, "Three-frame generalized phase-shifting interferometry by a Euclidean matrix norm algorithm," *Optics and Lasers in Engineering*, vol. 84, pp. 89-95, 2016.
- [164] J. Vargas, J. A. Quiroga, and T. Belenguer, "Phase-shifting interferometry based on principal component analysis," *Optics letters*, vol. 36, no. 8, pp. 1326-1328, 2011.
- [165] W. Niu, L. Zhong, P. Sun, and X. Lu, "Phase shifts extraction algorithm based on Gram-Schmidt orthonormalization of two vectors," *Optical and Quantum Electronics*, vol. 47, no. 8, pp. 2803-2810, 2015/08/01 2015, doi: 10.1007/s11082-015-0167-5.
- [166] K. Yatabe, K. Ishikawa, and Y. Oikawa, "Simple, flexible, and accurate phase retrieval method for generalized phase-shifting interferometry," *J. Opt. Soc. Am. A*, vol. 34, no. 1, pp. 87-96, 2017/01/01 2017, doi: 10.1364/JOSAA.34.000087.
- [167] M. A. Escobar, J. C. Estrada, and J. Vargas, "Phase-shifting VU factorization for interferometry," *Optics and Lasers in Engineering*, vol. 124, p. 105797, 2020/01/01/ 2020, doi: <https://doi.org/10.1016/j.optlaseng.2019.105797>.
- [168] K. Okada, A. Sato, and J. Tsujiuchi, "Simultaneous calculation of phase distribution and scanning phase shift in phase shifting interferometry," *Optics communications*, vol. 84, no. 3-4, pp. 118-124, 1991.
- [169] Z. Wang and B. Han, "Advanced iterative algorithm for phase extraction of randomly phase-shifted interferograms," *Optics letters*, vol. 29, no. 14, pp. 1671-1673, 2004.
- [170] J. Xu, Q. Xu, and L. Chai, "An iterative algorithm for interferograms with random phase shifts and high-order harmonics," *Journal of Optics A: Pure and Applied Optics*, vol. 10, no. 9, p. 095004, 2008.
- [171] T. Hoang, Z. Wang, M. Vo, J. Ma, L. Luu, and B. Pan, "Phase extraction from optical interferograms in presence of intensity nonlinearity and arbitrary phase shifts," *Applied Physics Letters*, vol. 99, no. 3, p. 031104, 2011.
- [172] M. Duan, Y. Zong, R. Zhu, and J. Li, "Phase-tilt iteration: Accurate and robust phase extraction from random tilt-shift interferograms," *Optics and Lasers in Engineering*, vol. 142, p. 106595, 2021.
- [173] Y.-C. Chen, P.-C. Lin, C.-M. Lee, and C.-W. Liang, "Iterative phase-shifting algorithm immune to random phase shifts and tilts," *Applied optics*, vol. 52, no. 14, pp. 3381-3386, 2013.
- [174] L. L. Deck, "Model-based phase shifting interferometry," *Applied optics*, vol. 53, no. 21, pp. 4628-4636, 2014.
- [175] K. Freischlad and C. L. Koliopoulos, "Fourier description of digital phase-measuring interferometry," *J. Opt. Soc. Am. A*, vol. 7, no. 4, pp. 542-551, 1990/04/01 1990, doi: 10.1364/JOSAA.7.000542.
- [176] M. Servin, J. C. Estrada, and J. A. Quiroga, "Spectral analysis of phase shifting algorithms," *Optics Express*, vol. 17, no. 19, pp. 16423-16428, 2009/09/14 2009, doi: 10.1364/OE.17.016423.
- [177] M. Servin, J. C. Estrada, and J. A. Quiroga, "The general theory of phase shifting algorithms," *Optics Express*, vol. 17, no. 24, pp. 21867-21881, 2009/11/23 2009, doi: 10.1364/OE.17.021867.
- [178] P. De Groot, "Derivation of algorithms for phase-shifting interferometry using the concept of a data-sampling window," *Applied optics*, vol. 34, no. 22, pp. 4723-4730, 1995.
- [179] Y. Surrel, "Additive noise effect in digital phase detection," *Applied optics*, vol. 36, no. 1, pp. 271-276, 1997.

- [180] K. Hibino, B. F. Oreb, D. I. Farrant, and K. G. Larkin, "Phase shifting for nonsinusoidal waveforms with phase-shift errors," *J. Opt. Soc. Am. A*, vol. 12, no. 4, pp. 761-768, 1995/04/01 1995, doi: 10.1364/JOSAA.12.000761.
- [181] K. Hibino, "Susceptibility of systematic error-compensating algorithms to random noise in phase-shifting interferometry," *Applied Optics*, vol. 36, no. 10, pp. 2084-2093, 1997/04/01 1997, doi: 10.1364/AO.36.002084.
- [182] K. Hibino and M. Yamauchi, "Phase-Measuring Algorithms to Suppress Spatially Nonuniform Phase Modulation in a Two-Beam Interferometer," *Optical Review*, vol. 7, no. 6, pp. 543-549, 2000/11/01 2000, doi: 10.1007/s10043-000-0543-6.
- [183] W. Juptner, T. M. Kreis, and H. Kreitiow, "Automatic evaluation of holographic interferograms by reference beam phase shifting," in *Industrial Applications of Laser Technology*, 1983, vol. 398: International Society for Optics and Photonics, pp. 22-29.
- [184] T. M. Kreis, "Computer aided evaluation of fringe patterns," *Optics and Lasers in Engineering*, vol. 19, no. 4, pp. 221-240, 1993/01/01/ 1993, doi: [https://doi.org/10.1016/0143-8166\(93\)90063-Q](https://doi.org/10.1016/0143-8166(93)90063-Q).
- [185] J. A. Quiroga, D. Crespo, and J. Gomez-Pedrero, "XtremeFringe: state-of-the-art software for automatic processing of fringe patterns," in *Optical Measurement Systems for Industrial Inspection V*, 2007, vol. 6616: International Society for Optics and Photonics, p. 66163Y.
- [186] S. Zhang and S.-T. Yau, "High-resolution, real-time 3D absolute coordinate measurement based on a phase-shifting method," *Optics Express*, vol. 14, no. 7, pp. 2644-2649, 2006/04/03 2006, doi: 10.1364/OE.14.002644.
- [187] Q. Kemao, S. Fangjun, and W. Xiaoping, "Determination of the best phase step of the Carré algorithm in phase shifting interferometry," *Measurement Science and Technology*, vol. 11, no. 8, p. 1220, 2000.
- [188] J. Vargas, J. A. Quiroga, and T. Belenguer, "Analysis of the principal component algorithm in phase-shifting interferometry," *Optics Letters*, vol. 36, no. 12, pp. 2215-2217, 2011/06/15 2011, doi: 10.1364/OL.36.002215.
- [189] X. Zhu, Y. Wu, and F. Liu, "GPU-accelerated phase extraction algorithm for interferograms: a real-time application," in *Optical Metrology and Inspection for Industrial Applications IV*, 2016, vol. 10023: International Society for Optics and Photonics, p. 1002327.
- [190] H. Canabal, J. A. Quiroga, and E. Bernabeu, "Improved phase-shifting method for automatic processing of moiré deflectograms," *Applied Optics*, vol. 37, no. 26, pp. 6227-6233, 1998/09/10 1998, doi: 10.1364/AO.37.006227.
- [191] P. de Groot, "Measurement of transparent plates with wavelength-tuned phase-shifting interferometry," *Applied Optics*, vol. 39, no. 16, pp. 2658-2663, 2000/06/01 2000, doi: 10.1364/AO.39.002658.
- [192] Q. Kemao, "Windowed fringe pattern analysis," 2013: Society of Photo-Optical Instrumentation Engineers.
- [193] L. L. Deck, "Suppressing phase errors from vibration in phase-shifting interferometry," *Applied optics*, vol. 48, no. 20, pp. 3948-3960, 2009.
- [194] A. Pyzara, B. Bylina, and J. Bylina, "The influence of a matrix condition number on iterative methods' convergence," in *2011 Federated Conference on Computer Science and Information Systems (FedCSIS)*, 2011: IEEE, pp. 459-464.
- [195] D. Zill and J. Dewar, *Algebra and trigonometry*. Jones & Bartlett Publishers, 2011.
- [196] B. N. Parlett, *The symmetric eigenvalue problem*. SIAM, 1998.
- [197] C. P. Brophy, "Effect of intensity error correlation on the computed phase of phase-shifting interferometry," *JOSA A*, vol. 7, no. 4, pp. 537-541, 1990.

- [198] R. Jozwicki, M. Kujawinska, and L. A. Salbut, "New contra old wavefront measurement concepts for interferometric optical testing," *Optical Engineering*, vol. 31, no. 3, pp. 422-433, 1992.
- [199] Q. Kemao, "Two-dimensional windowed Fourier transform for fringe pattern analysis: principles, applications and implementations," *Optics and Lasers in Engineering*, vol. 45, no. 2, pp. 304-317, 2007.
- [200] Q. Kemao, H. Wang, W. Gao, L. Feng, and S. H. Soon, "Phase extraction from arbitrary phase-shifted fringe patterns with noise suppression," *Optics and Lasers in Engineering*, vol. 48, no. 6, pp. 684-689, 2010.
- [201] J. Immerkaer, "Fast noise variance estimation," *Computer vision and image understanding*, vol. 64, no. 2, pp. 300-302, 1996.
- [202] L. L. Deck and J. A. Soobitsky, "Phase-shifting via wavelength tuning in very large aperture interferometers," in *Optical manufacturing and testing III*, 1999, vol. 3782: International Society for Optics and Photonics, pp. 432-442.
- [203] K. Liu, Y. Wang, D. L. Lau, Q. Hao, and L. G. Hassebrook, "Gamma model and its analysis for phase measuring profilometry," *JOSA A*, vol. 27, no. 3, pp. 553-562, 2010.
- [204] S. Zhang and S.-T. Yau, "Generic nonsinusoidal phase error correction for three-dimensional shape measurement using a digital video projector," *Applied optics*, vol. 46, no. 1, pp. 36-43, 2007.
- [205] K. Yatabe, K. Ishikawa, and Y. Oikawa, "Simple, flexible, and accurate phase retrieval method for generalized phase-shifting interferometry," *JOSA A*, vol. 34, no. 1, pp. 87-96, 2017.
- [206] C. Ai and J. C. Wyant, "Effect of piezoelectric transducer nonlinearity on phase shift interferometry," *Applied optics*, vol. 26, no. 6, pp. 1112-1116, 1987.
- [207] Y. Chen and Q. Kemao, "General iterative algorithm for phase-extraction from fringe patterns with random phase-shifts, intensity harmonics and non-uniform phase-shift distribution," *Optics Express*, vol. 29, no. 19, pp. 30905-30926, 2021.
- [208] J. Trujillo-Sevilla, J. Ramos-Rodríguez, and J. Gaudestad, "Wave front phase imaging of wafer geometry using high pass filtering to reveal nanotopography," in *Photonic Instrumentation Engineering VII*, 2020, vol. 11287: International Society for Optics and Photonics, p. 112870W.
- [209] S. Tang, R. E. Bills, and K. Freischlad, "High accuracy measurements of objects with multiple reflective surfaces with wavelength shifting interferometry," in *3rd International Symposium on Advanced Optical Manufacturing and Testing Technologies: Optical Test and Measurement Technology and Equipment*, 2007, vol. 6723: International Society for Optics and Photonics, p. 67230F.
- [210] T. Wang and Q. Kemao, "Gpu acceleration for optical measurement," 2017: SPIE.
- [211] T. Wang, Z. Jiang, Q. Kemao, F. Lin, and S. H. Soon, "GPU accelerated digital volume correlation," *Experimental Mechanics*, vol. 56, no. 2, pp. 297-309, 2016.
- [212] T. Wang, Q. Kemao, H. S. Seah, and F. Lin, "A flexible heterogeneous real-time digital image correlation system," *Optics and Lasers in Engineering*, vol. 110, pp. 7-17, 2018.
- [213] L. Zhang *et al.*, "High accuracy digital image correlation powered by GPU-based parallel computing," *Optics and Lasers in Engineering*, vol. 69, pp. 7-12, 2015.
- [214] T. Wang, L. Kai, and Q. Kemao, "Real-time reference-based dynamic phase retrieval algorithm for optical measurement," *Applied optics*, vol. 56, no. 27, pp. 7726-7733, 2017.
- [215] P. J. de Groot, "101-frame algorithm for phase-shifting interferometry," in *Optical inspection and micromeasurements II*, 1997, vol. 3098: International Society for Optics and Photonics, pp. 283-292.
- [216] D. Guide, "Cuda c programming guide," *NVIDIA*, July, vol. 29, p. 31, 2013.
- [217] M. Harris, "Optimizing parallel reduction in CUDA," *Nvidia developer technology*, vol. 2, no. 4, p. 70, 2007.

- [218] J. Nickolls, I. Buck, M. Garland, and K. Skadron, "Scalable parallel programming with cuda: Is cuda the parallel programming model that application developers have been waiting for?," *Queue*, vol. 6, no. 2, pp. 40-53, 2008.
- [219] M. Garland *et al.*, "Parallel computing experiences with CUDA," *IEEE micro*, vol. 28, no. 4, pp. 13-27, 2008.
- [220] B. N. Datta, *Numerical linear algebra and applications*. Siam, 2010.
- [221] L. Dagum and R. Menon, "OpenMP: an industry standard API for shared-memory programming," *IEEE computational science and engineering*, vol. 5, no. 1, pp. 46-55, 1998.
- [222] E. Wang *et al.*, "Intel math kernel library," in *High-Performance Computing on the Intel® Xeon Phi™*: Springer, 2014, pp. 167-188.
- [223] N. Whitehead and A. Fit-Florea, "Precision & performance: Floating point and IEEE 754 compliance for NVIDIA GPUs," *rn (A+B)*, vol. 21, no. 1, pp. 18749-19424, 2011.
- [224] S. Kestur, J. D. Davis, and O. Williams, "Blas comparison on fpga, cpu and gpu," in *2010 IEEE computer society annual symposium on VLSI*, 2010: IEEE, pp. 288-293.
- [225] D. H. Jones, A. Powell, C.-S. Bouganis, and P. Y. Cheung, "GPU versus FPGA for high productivity computing," in *2010 International Conference on Field Programmable Logic and Applications*, 2010: IEEE, pp. 119-124.
- [226] S. Asano, T. Maruyama, and Y. Yamaguchi, "Performance comparison of FPGA, GPU and CPU in image processing," in *2009 international conference on field programmable logic and applications*, 2009: IEEE, pp. 126-131.
- [227] M. Servin, J. Estrada, J. Quiroga, J. Mosiño, and M. Cywiak, "Noise in phase shifting interferometry," *Optics express*, vol. 17, no. 11, pp. 8789-8794, 2009.
- [228] Q. Liu, Y. Wang, J. He, and F. Ji, "Phase shift extraction and wavefront retrieval from interferograms with background and contrast fluctuations," *Journal of Optics*, vol. 17, no. 2, p. 025704, 2015.
- [229] S. Feng, C. Zuo, L. Zhang, W. Yin, and Q. Chen, "Generalized framework for non-sinusoidal fringe analysis using deep learning," *Photonics Research*, vol. 9, no. 6, pp. 1084-1098, 2021.
- [230] L. Zhang, Q. Chen, C. Zuo, and S. Feng, "High-speed high dynamic range 3D shape measurement based on deep learning," *Optics and Lasers in Engineering*, vol. 134, p. 106245, 2020.
- [231] C. R. Houck, J. Joines, and M. G. Kay, "A genetic algorithm for function optimization: a Matlab implementation," *Ncsu-ie tr*, vol. 95, no. 09, pp. 1-10, 1995.
- [232] E. Oblow, "SPT: a stochastic tunneling algorithm for global optimization," *Journal of Global Optimization*, vol. 20, no. 2, pp. 191-208, 2001.
- [233] P. J. Van Laarhoven and E. H. Aarts, "Simulated annealing," in *Simulated annealing: Theory and applications*: Springer, 1987, pp. 7-15.