Metrologia
Fortaleza - CE 2017

Software for imaging phase-shift interference microscope

Igor Malinovski, Ricardo dos S. França, Iakyra B. Couceiro

Instituto Nacional de Metrologia, Qualidade e Tecnologia - INMETRO

E-mail: rsfranca@inmetro.gov.br

Abstract: In recent years absolute interference microscope was created at National Metrology Institute of Brazil (INMETRO). The instrument by principle of operation is imaging phase-shifting interferometer (PSI) equipped with two stabilized lasers of different color as traceable reference wavelength sources. We report here some progress in development of the software for this instrument. The status of undergoing internal validation and verification of the software is also reported. In contrast with standard PSI method, different methodology of phase evaluation is applied. Therefore, instrument specific procedures for software validation and verification are adapted and discussed.

Keywords: Optical nanometrology, dimensional metrology, interference microscopy, software validation.

1. INTRODUCTION

The automated measurement systems with computerized processing have virtually revolutionized physical metrology in recent years. Adequate software (SW) often provides significant improvement in usability and accuracy of the instrument. However this process immediately raises important issues about validation and verification of the software as an integral part of metrology systems. We present here our ongoing work on development and validation/verification of the software for automated interference microscope (IM) that was recently upgraded with some new hardware (HW) [1].

2. GENERAL SYSTEM DESCRIPTION

Our IM is based on phase-shifting (PSI) technique combined with interference pattern digital imaging. In standard PSI methods [2] 3-5 fixed step interferograms are used for phase extraction. In our algorithm we use multiple phase-stepped interferograms (image stack) to calculate the height from waveforms that carry all necessary phase information. Each pixel of the interferometric image stack produces one waveform corresponding to one surface point (figure 1). The disadvantage of this approach is multiple interferograms processing (significant amount of data). The advantage of this method is the possibility of the phase step self-calibration. Indeed, if the whole waveform of fringe signal is available, the phase step information is simply deducted from such waveform. As more advantages or this method we should mention: less sensitivity to mechanical errors in step-shifting and higher resolution in phase determination. Specifically for this interferometric microscope we have developed dedicated custom software package that includes data acquisition, data processing and visualization. In figure 1 we demonstrate details of the phase-shifting algorithm.

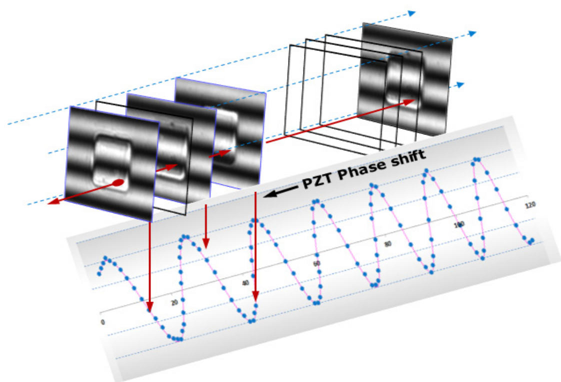

Figure 1. Image stack of interferograms (selected frames) and phase waveform signal for one pixel digitized. Each frame produces one point data on sinusoidal waveform as shown with arrows. The whole waveform results in a constant phase for one surface point.

One complete measurement with full phase-shifting scan (*measurement cycle)* typically consist about 100 points that is much more than Nyquist criterion requirement. We repeat cycles with final result averaging if noise reduction is desirable. It is clear from figure 1 that one measurement cycle results in one 4D data image stack. The image stack after processing should yield into 3D surface map of the object. The software works both with individual pixels or group of neighboring pixels (binned pixels) as option. We use pixels within XY range of interest (ROI) of CCD that is manually selected by markers on 2D screen.

3. PHYSICS OF MEASUREMENT AND MODEL

For necessary specification of the physical processes we should start with background physics of the measurement involved. From general physical principles the pseudo-continuous phase-shift results in approximately sinusoidal wave form at each pixel. Therefore, the simple sinusoidal waveform can be used as the first approximation model. Next step is to correct this model to possible phase-shifter nonlinearity that results in second order correction term of the argument of sinusoidal function. So we use the following model function:

$$F(x) = Off + A \sin (\omega x + \Omega_m x^2 + \Phi), \qquad (1)$$

where x is the value proportional to phase-shifter actuator voltage, Off is the offset of the sinusoidal fringe, A is the amplitude, ω is the frequency, Ω_m is the frequency modulation coefficient, Φ is the phase that carries height information. This model function is used to find best possible fit between measured sinusoidal waveform and model with all 5 parameters as variables. Least square criteria of the fit quality are used. We fit individually all pixels in ROI.

4. SOFTWARE SPECIFICATIONS AND STRUCTURE

From management point of view validation of the software starts with correct specifications. In our case specifying software is easy because physical part of the measurement process is clear. The general requirement is that the software must provide automated measurement with adequate calculation for obtaining meaningful result in comprehensible SI unit quantified form. The software suitable to operate automated PSI imaging instrument requires minimum 3 main functionalities as most important: (i) automation and data acquisition, (ii) data processing (phase extraction) (iii) Error control/ reporting. We will concentrate on those in next sections.

From coding point of view the implementation of above scenario it is more or less straight forward programming task. However, making the software foolproof and suitable for so called “stress tests” is a much more demanding process. We have analyzed most critical parts of the software from verification point of view. Following practice of PTB and NPL [3-4] we have defined the list of most important testable requirements with suitable test criteria for each item. We will follow the list in following sections and sub-sections.

5. AUTOMATION AND DATA ACQUISITION

The automation of the SW requires *synchronized and robust input-output data flow control*. This requirement was validated by both static analysis and white-box testing. As a first step of static program analysis the data communication code was checked in order to provide predictable code execution with any data input. This includes trapping values outside the expected range, interception of sudden jumps in values as well as flat (not changing) data input detection. The most conservative action to correct situation was adapted as a full reset of the measurement cycle in case of problem detection.

The data readout was also verified by dynamic white-box testing in which we executed the SW code together with background monitoring of the whole communication input-output calls between the SW and HW. As an auxiliary step we could trace the exact code execution with inserted check points and detailed log files. This feature was specifically implemented to ensure confidence in white-box testing. Those tests were performed in both normal operational mode and simulated problematic situations (such as low light level on CCD, saturated illumination, malfunction of phase-shifter unit, etc.)

6. DATA PROCESSING / COMPUTATION

The data received from the hardware is subject of further processing. *Correct computation* is required for each step of the processing. This requirement is subdivided in several topics due to complexity of the process.

6.1 Algorithm numerical stability

The data processing is based on numerical fit as the most critical part of the calculation methodology. Therefore, the fitting algorithm is the most important part of the processing to be verified. We should start with the curve fit criterion that is in our case the most obvious minimum least square difference between measured and fit points. Any fitting algorithm works with certain tradeoff between calculation speed and numerical stability of the output. Our fastest two-pass iteration algorithm described in [1] unfortunately was found to be insufficiently stable for some test cases (specifically cases of 2 or more local minima in multi-parameter space). Thus, the fitting procedure has been modified to avoid instability at expense of processing speed decrease. We have added one more preliminary phase of the search in which presence of several minima is verified. A resulting approximate solution is fed to same two-pass iterations as before. Final SW performance is about 0.5 ms per fit on 3.5 GHz processor.

6.2. Data filtering

There are multiple sources of noise during the measurement. Fortunately since the signal of interest lies on quite narrow frequency range we can achieve significant signal-to-noise improvement by data filtering. We apply forward-reverse Fast Fourier Transform (FFT) with intermediate Gaussian shaped filter. This filter is supposed to produce no significant phase distortions. Correctness our FFT implementation module was separately verified by black-box test comparison with published C++ code [5]. Numerical stability was tested with our test cases. The whole data filtering procedure has been verified in conjunction with fitting as described in the next section.

6.3 Dynamic black-box testing

Compliance of calculated and expected results was verified with dynamic program black-box test using test case data. Since ready test cases for our type of data are not available we have produced our own test cases by computer simulations of data sets with known function and parameters. Each test case consists of input values, expected output values, and the predicted program behavior with corresponding warnings/reporting via messages. Some of the test cases were designed to estimate type A and type B errors. Typical signal distortions were simulated in our test cases to evaluate possible deviation of the results (figure 2). We have validated our test cases with commercial off-the-shelf and open source software packages such as TableCurve 2D (systatsoftware.com) and SimFit (simfit.org.uk).

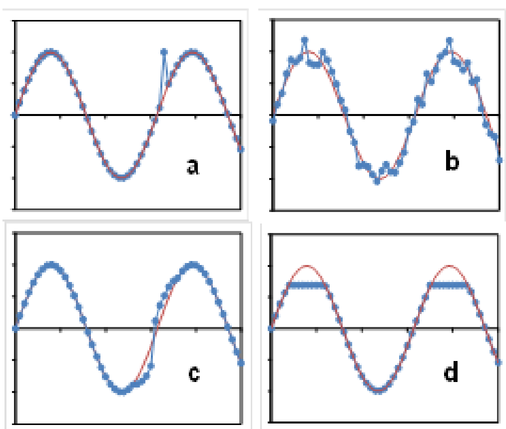

Figure 2. Numerically simulated test cases for black-box testing examples: a) "Hot pixel" peak defect, b) random (white) noise, c) phase jump (jitter), d) CCD sensor saturation. Dots are the simulated signal points and line is the model function (fit).

6.4 *Back-to-back testing*

The filtering and fitting part of the software was compared to corresponding parts of our previous SW package developed for the gauge block interferometer [6]. This instrument with its automation SW has participated at an international comparison and it can be considered as "end-to-end" tested unit [7]. No significant differences were detected during this testing.

7. CONCLUSIONS

The software package suitable for imaging phase-shift IM is under development. The SW is currently under validation and verification process. Most important observations are as follows: validated physical background is used for adequate model construction, implementation of the model is verified with both static and dynamic testing, back–to-back tests successfully passed, and insufficient numerical stability was detected and removed using more conservative minima search. We found that validation and verification performed in parallel with SW code improvements is very useful tool for SW development in its final stage.

8. REFERENCES

[1] Malinovski I, França R S, Bessa M S, Silva C R and Couceiro I B 2016 *Surf. Topogr.: Metrol. Prop.* **4** 024006

[2] Malacara D 2007 Optical Shop Testing 3rd Edition Wiley & Sons Inc. USA.

[3] Wichmann B, Parkin G and Barker R 2007 *NPL Report* DEM-ES 014.

[4] Greif N, Schrepf H, Richter D 2006 Software validation in metrology: A case study for a GUM-supporting software Measurement **39** (9) 849

[5] Press W H, Flannery B P, Teukolsky S A, and Vetterling W T 2007 Numerical Recipes - Cambridge University Press.

[6] Titov A, Malinovsky I and Massone C 2000 *Metrologia*, **37** 121

[7] Decker J E, et al. 2007 Final Report on SIM.L-K1 (SIM.4.2) Regional Comparison. Stage One: Calibration of gauge blocks by optical interferometry *Metrologia* **44** 04001