Benemérita Universidad Autónoma de Puebla

Facultad de Ciencias Físico-Matemáticas
Doctorado en Ciencias (Física Aplicada)

Phase computing for automatic real-time applications

Doctoral Thesis

Author
Rigoberto Juarez-Salazar

Advisers
Dr. Carlos I. Robledo Sánchez
Dr. Cruz Meneses Fabián

July, 2014

Para mi familia.

Agradecimientos

Gracias a mis padres y hermanos por haberme dado la oportunidad de perseguir un sueño y ayudarme a alcanzarlo. Gracias a mi esposa por acompañarme en este camino de sacrificios y gloria, y a mi hijo por las lecciones de fortaleza.

Agradezco de manera especial al Consejo Nacional de Ciencia y Tecnología (CONACyT) por la beca que permitió realizar mis estudios de maestría, a Vicerrectoría de Investigación y Estudios de Posgrado (VIEP) por el apoyo para la culminación de este trabajo, al Laboratorio de Investigación de Sistemas Dinámicos Controlables y al Laboratorio de Ecuaciones Diferenciales y Modelación Matemática por la disposición de equipo y espacio para el desarrollo teórico y experimental del trabajo.

Gracias a los profesores W. Fermín Guerrero S., J. Jacobo Oliveros O., Carlos I. Robledo S., J. Fermi Guerrero Castellanos, G. Tavares Velasco, Humberto Salazar I., Oscar M. Martínez B., A. Vergara Betancourt, Uri Noe Mora Rivera, por el invaluable apoyo que me ofrecieron durante estos últimos años formándome para hacer de mí una mejor persona.

Rigoberto Juárez Salazar

Phase computing for automatic real-time applications

Abstract

PHASE COMPUTING is one of the most important techniques in many scientific areas and engineering because of a wide variety of physical quantities that can be measured with high precision by noninvasive means. The phase-based optical measurement systems attain high accuracy, fast response (even in real-time), and they are low-cost. These features have been possible through recent technological advances such as laser diodes, liquid crystal displays, high-speed cameras, and high-performance computers. However, rapid technological development also demands improving the conventional phase demodulation methods.

This thesis is focused on phase demodulation for optical measurement using interferometric and non-interferometric setups. Fringe pattern phase demodulation methods are developed. The main specific contributions of this thesis are new phase-based measurement methods described as follows.

1. Fringe pattern normalization by polynomial fitting using least squares.
2. Fourier fringe-normalized analysis.
3. Generalized phase-shifting algorithms for inhomogeneous phase shifts and spatio-temporal visibility variation.
4. Phase-unwrapping by a rounding least squares approach.
5. Phase-shifting interferometry by lateral displacement of the light source.

The first four contributions refer to *phase computing technology*, designed to be user-free, with no specialized computer resources required, and easy to implement in other applications. Contribution 5 of the previous list concerns a simple and inexpensive optical setup for interferometrical evaluations by phase-shifting where the required phase shifts are induced by coarse lateral displacement of the laser point source.

Among the five contributions of this thesis, contribution 3 is the most outstanding. This contribution refers to a generalized phase-shifting algorithm able to handle inhomogeneous nonlinear phase shifts, large fringe contrast variations, and the capacity to work with a minimum of two fringe patterns. This contribution represents a novel method that, to the best of our knowledge, is the first to solve the phase computing problem in those generalized operation conditions.

Throughout this thesis, the concepts are illustrated by using several figures and examples. The developed algorithms are implemented on the MATLAB platform and the resulting computer code is provided. The suggested algorithms are tested by executing they in both simulated and experimental scenarios. The obtained results show that the developed work could be used in automatic real-time applications.

Index Terms— Phase-shifting, Fourier fringe analysis, Phase-unwrapping, Fringe-pattern normalization, Lateral displacement, Automatic real-time applications.

Extracción de fase para aplicaciones automáticas en tiempo real

Resumen

Codificación de fase es una de las técnicas más importantes en muchas áreas científicas y de ingeniería debido a la amplia variedad de cantidades físicas que se pueden medir con alta precisión por medios no invasivos. Los sistemas de medición por codificación de fase ópticos alcanzan alta exactitud, rápida respuesta (incluso en tiempo real), y son de bajo costo. Estas características deseables han sido posibles por los recientes avances tecnológicos en diodos láser, pantallas de cristal líquido, cámaras de alta velocidad, y computadoras de alto rendimiento. Sin embargo, el rápido desarrollo tecnológico también demanda mejorar los métodos convencionales de demodulación de fase.

Esta tesis está enfocada en la extracción de fase para medición óptica usando arreglos interferométricos y no interferométricos. Se desarrollan métodos de extracción de fase a través de procesamiento de patrones de franjas. Las principales contribuciones específicas de esta tesis son nuevos métodos de medición de fase que se describen a continuación.

1. Normalización de patrones de franjas usando ajuste polinomial por mínimos cuadrados.
2. Análisis de Fourier de patrones de franjas normalizados.
3. Algoritmo de corrimiento de fase generalizado para corrimientos de fase inhomogéneos y variación espacio-temporal de la visibilidad.
4. Desenvolvimiento de fase por medio del enfoque de redondeo y mínimos cuadrados.
5. Interferometría de corrimiento de fase por desplazamiento lateral de la fuente de luz.

Las primeras cuatro contribuciones se refieren a la *tecnología de extracción de fase* diseñada para no requerir intervención del usuario, no requerir recursos computacionales especializados, y facilidad para implementarse en distintas aplicaciones. La contribución 5 de la lista anterior se refiere a un arreglo óptico simple y económico de corrimiento de fase para evaluaciones interferométricas donde los corrimientos de fase son inducidos por medio del desplazamiento lateral de una fuente láser puntual.

De las cinco contribuciones de esta tesis, la contribución número tres es la que más destaca. Esta contribución se refiere a un método de corrimiento de fase generalizado capaz de manejar corrimiento de fase no lineales e inhomogéneos, robusto a variaciones significativas de contraste de los patrones de franjas, y posibilidad de funcionamiento hasta con un mínimo de dos patrones de franjas. Esta contribución representa un método novedoso que, hasta donde sabemos, es el primero en resolver el problema de extracción de fase en condiciones de operación generalizadas.

A lo largo de esta tesis, los conceptos son ilustrados usando figuras y ejemplos. Los algoritmos diseñados son implementados en la plataforma MATLAB y los códigos usados son proporcionados. Los algoritmos sugeridos son probados ejecutándolos en escenarios tanto simulados como experimentales. Los resultados obtenidos muestran que el trabajo desarrollado puede ser usado en aplicaciones automáticas en tiempo real.

Palabras clave— Corrimiento de fase, Análisis de Fourier de patrones de franjas, Desenvolvimiento de fase, Normalización de patrones de franjas, Desplazamiento lateral, Aplicaciones automáticas en tiempo real.

Publications

THE work in this thesis was reported in the following publications.

Papers

1. Rigoberto Juarez-Salazar, Carlos Robledo-Sanchez, Fermin Guerrero-Sanchez, and A. Rangel-Huerta, *"Generalized phase-shifting algorithm for inhomogeneous phase shift and spatio-temporal fringe visibility variation"*, Opt. Express, vol. **22**, no. 4, pp. 4738–4750, 2014.

   DOI: 10.1364/OE.22.004738

2. Rigoberto Juarez-Salazar, Carlos Robledo-Sanchez, and Fermin Guerrero-Sanchez, *"Phase-unwrapping algorithm by a rounding-least-squares approach"*, Optical Engineering, vol. **53**, no. 2, pp. 024102, 2014.

   DOI: 10.1117/1.OE.53.2.024102

3. Jose F. Casco-Vasquez, Rigoberto Juarez-Salazar, Carlos Robledo-Sanchez, Gustavo Rodriguez-Zurita, Fermin Guerrero-Sanchez, Luis M. Arevalo Aguilar, and Cruz Meneses-Fabian, *"Fourier normalized-fringe analysis by zero-order spectrum suppression using a parameter estimation approach"*, Optical Engineering, vol. **52** no. 7, pp. 074109, 2013.

   DOI: 10.1117/1.OE.52.7.074109

4. Carlos Robledo-Sanchez, Rigoberto Juarez-Salazar, Cruz Meneses-Fabian, Fermin Guerrero-Sanchez, L. M. Arevalo Aguilar, Gustavo Rodriguez-Zurita, and Viridiana Ixba-Santos, *"Phase-shifting interferometry based on the lateral displacement of the light source"*, Opt. Express, vol. **21**no. 14, pp. 17228–17233, 2013. DOI: 10.1364/OE.21.017228

5. Rigoberto Juarez-Salazar, Carlos Robledo-Sanchez, Cruz Meneses-Fabian, Fermin Guerrero-Sanchez, and L. M. Arevalo Aguilar, *"Generalized phase-shifting interferometry by parameter estimation with the least squares method"*, Optics and Lasers in Engineering, vol. **51**, no. 5, pp. 626–632, 2013.

   DOI: 10.1016/j.optlaseng.2012.12.020

Proceedings

1. Rigoberto Juarez-Salazar, Carlos Robledo-Sanchez, Cruz Meneses-Fabian, Gustavo Rodriguez-Zurita, Fermin Guerrero-Sanchez, and Jorge Gonzalez-Garcia, *"Straightforward filtering to phase demodulation by a Fourier normalized-fringe approach"*, Proc. SPIE, vol. **8785**, pp. 878543–878543-5, 2013.

   DOI: 10.1117/12.2026108

2. Rigoberto Juarez-Salazar, Carlos Robledo-Sanchez, Cruz Meneses-Fabian, Gustavo Rodriguez-Zurita, Fermin Guerrero-Sanchez, and Antonio Barcelata-Pinzon, *"Automatic real-time generalized phase-shifting interferometry to process interferograms with spatio-temporal visibility"*, Proc. SPIE, vol. **8785**, pp. 878542–878542-7, 2013.

   DOI: 10.1117/12.2026103

3. Fermin Guerrero-Sanchez, Rigoberto Juarez-Salazar, Cruz Meneses-Fabian, and Carlos Robledo-Sanchez, *"Fringe pattern simulation with an adaptive pendulum"*, Proc. SPIE, vol. **8287**, pp. 828713–828713-8, 2011.

   DOI: 10.1117/12.913354

Contents

Chapter 1

Phase modulation and intensity pattern normalization

In mathematics you don't understand things.
You just get used to them.

John Von Neumann

Contents

THE phase coding is the most used method to perform high precision measurements. Phase coding could be even the only way to obtain some physical information, e.g., the imaging of transparent cells in contrast phase microscopy. In any case, the phase coding principle is preferred in many applications because this principle exhibits an excellent signal-to-noise ratio.

In this chapter the phase coding principle is described in the context of optical metrology (by interference and non-interference setups). In such applications, the phase of interest is recorded in intensity maps known as *fringe-patterns*. The *phase computing term* (the general procedure of fringe-pattern processing to extract the phase distribution of interest) will be defined.

The so-called *fringe-pattern normalization* is described and a summary of the state-of-the-art is given. Finally, a fringe-pattern normalization algorithm based on parameter estimation is presented. This algorithm is user-free and computationally efficient. Therefore, automatic real-time applications can be addressed. Several figures and examples are used in the presentation of this topic. The computer code implemented in the MATLAB platform is given.

1.1 | Introduction

Optical measurement systems are widely used because of its desirable characteristics as non-contact, high resolution, and fast evaluations [1]. In such phase-base systems, the physical variable of interest (e.g., topography, deformation, refraction index, temperature distributions, strain, optical aberrations, magnetic fields, and many others) is generally measured by sensing intensity maps in the form of *fringe-patterns*. For this reason, it is necessary to develop efficient *fringe analysis algorithms* in order to extract the encoded physical information of interest [2].

Generally, the optical measurement systems encode the interest information as a phase distribution. Since the phase of an optical radiation cannot be sensed directly, it must be recovered from one or more intensity maps (fringe-patterns) which can be modeled as [3]:

$$I_k(p) = a_k(p) + b_k(p)\cos\Phi_k(p), \tag{1.1}$$

where p (a two dimensional spatial variable for some domain) is a point on the observation plane, $I_k(p)$ $(k = 0, 1, \cdots)$ is the k-th acquired intensity distribution, $a_k(p)$ is the ***background light*** (also known as *offset light*, *bias*, and *average intensity*), $b_k(p)$ is the ***modulation light*** (also known as *fringe amplitude*), and $\Phi(p)$ is the encoded phase distribution. In some applications it is more convenient to define the fringe-patterns (1.1) as

$$I_k(p) = a_k(p)\left[1 + v_k(p)\cos\Phi_k(p)\right], \tag{1.2}$$

where

$$v_k(p) = b_k(p)/a_k(p), \tag{1.3}$$

is the so-called ***fringe visibility*** (also known as the *fringe contrast*, *fringe amplitude*, and *fringe modulation*). From now on, since all functions are depending on the spatial variable p, the dependence of p is omitted for brevity. Initially, the background, modulation, and visibility terms can be hard to understand. Useful insight about these concepts are attained by Example 1.

Fringe-patterns of the form (1.1) can be generated by interferometrical as well as non-interferometrical techniques. For example, Fig. 1.2 shows the fringe-pattern obtained from the Twyman-Green Interferometer and from a fringe projection profilometer [1]. In both optical setups, the physical information of interest (in this case, the surface's topography) is encoded as a phase distribution.

There are two important differences between the measurement systems shown in Fig. 1.2. Namely, the sensitivity and the encoding procedure of the physical information.

- Interferometers have the most high sensitivity because their measurement unit is the wavelength of the employed illumination source[1]. On the other hand, sensitivity in fringe projection techniques is moderate. However, they are useful to test long surfaces as the structure of cars. Thus, both optical setups are useful; the choice of one or another depends on the application.

- With respect to the encoding information, from Fig. 1.2 we can see that, although the physical information is the same (except scaling), the topography of the evaluating surface is encoded in different ways. The interferometer generates closed fringes while the profilometer generates open fringes. This difference is due to the fringe projection technique includes a *spatial carrier* in the resulting fringe-pattern beforehand[2]. Thus, the choice of a particular fringe analysis method depends on if a *temporal carrier* or *spatial carrier* is available. This will be discussed in the following chapters.

[1] For laser sources in the optical range, λ is in the interval $[400, 720] \times 10^{-9}$ meter [4]. Interferometers are able to sense fractions of the wavelength λ; from tenths to thousandths.

[2] For the fringe-projection techniques, the spatial carrier addition is inherent. A spatial carrier also can be generated in interferometrical setups, for example, by tilting the reference mirror M in Fig. 1.2(a).

Example 1 (Background, modulation, and fringe visibility)**.** *Let I_{max} and I_{min} be the upper and lower envelopes, respectively, of an oscillating curve. Then, the background light $a(p)$ can be easily seen as the average function:*

$$a(p) = \frac{I_{max}(p) + I_{min}(p)}{2}. \tag{1.4}$$

On the other hand, the modulation light $b(p)$ can be interpreted as the amplitude of the oscillations (around the background component); i.e.,

$$b(p) = \frac{I_{max}(p) - I_{min}(p)}{2}. \tag{1.5}$$

By using the equation (1.3), an alternative definition of fringe visibility is obtained[a]*:*

$$v(p) = \frac{b(p)}{a(p)} = \frac{I_{max}(p) - I_{min}(p)}{I_{max}(p) + I_{min}(p)}. \tag{1.6}$$

To illustrate these concepts, in Fig. 1.1 typical fringe-patterns (in one-dimensional for visualizations purposes) are shown. Notice that, although the fringe-patter in the first row of Fig. 1.1 seems to oscillate with a background and modulation near constants, the fringe visibility is poor and non-constant.

On the other hand, the fringe-pattern in the second row of Fig. 1.1, although it has a large varying background and modulation components, the visibility is constant and unitary.

[a] This is a convenient equation because it allow us to compute the visibility approximately by using only the maximum and minimum intensity levels reached in some regions of the fringe-pattern.

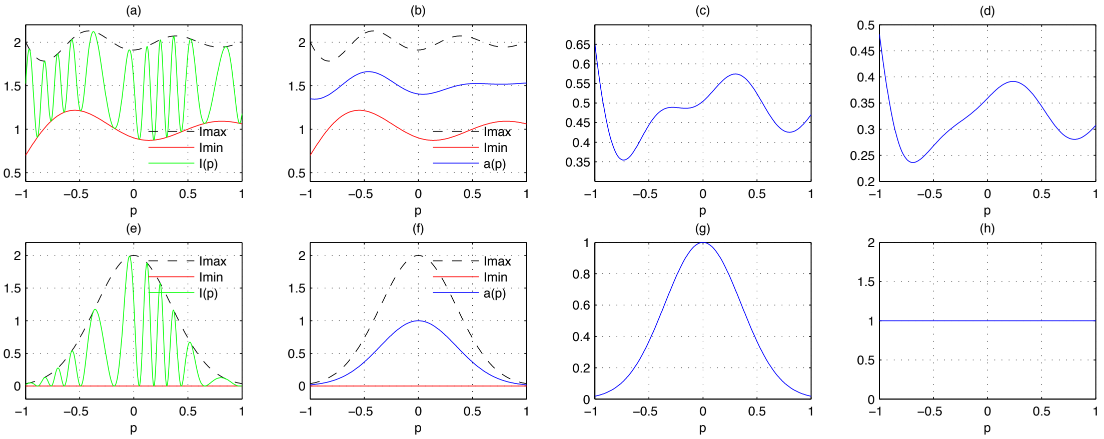


Figure 1.1: *The background, modulation, and fringe visibility concepts. (1st column) The upper envelope I_{max} and the lower envelope I_{min} of a fringe-pattern $I(p)$. (2nd column) The background light or average intensity $a(p)$. (3rd column) The modulation light or fringe amplitude $b(p)$. (4th column) The fringe visibility.*

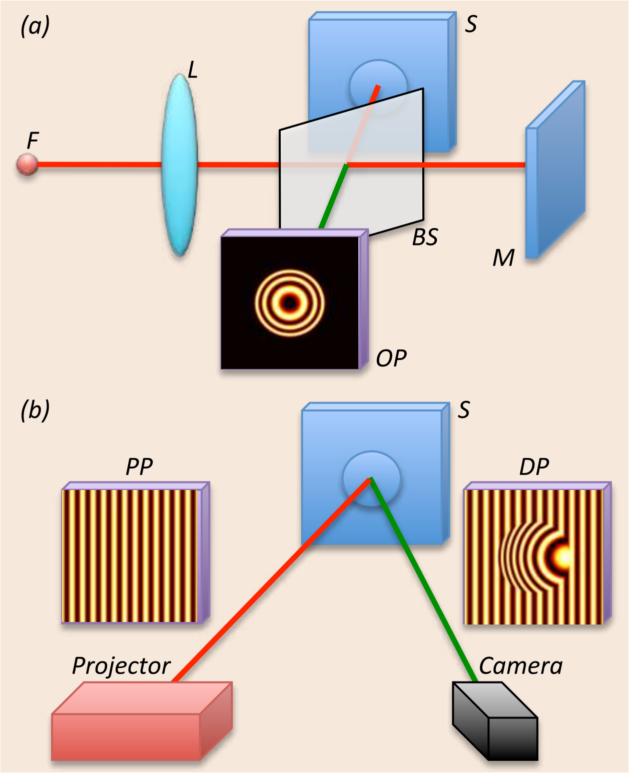


Figure 1.2: *Evaluation of the surface S (a plane with a spherical deformation at the center). (a) Twyman-Green Interferometer. F is a point laser source, L is the collimating lens, BS is a beam-splitter, M is a reference mirror, and OP is the observation plane. (b) Optical profilometer by fringe projection. PP is the projected pattern and DP is the acquired one.*

1.2 | Phase computing technology

Usually, the physical information encoded in the fringe-patterns (1.1) is associated with the phase distribution Φ_k. On the other hand, the background a_k and amplitude b_k contain information about other contributions such as environmental illumination, intensity profile of the employed source and its spatial location, object reflectance, polarization, among others [5]. This claim is illustrated in Example 2 for the case of interference of two beams.

Phase computing will be the term used to refer to the necessary data processing stages involved in the phase demodulation computer routines. The Phase demodulation is the extraction of the phase distribution Φ from one or more fringe-patterns $I(p)$ given by (1.1).

A phase computing system can be integrated of several stages such as wrapped phase extraction, phase unwrapping, fringe-pattern normalization, and noise filtering. A graphic illustration of this concept is given in Fig. 1.3. The fringe-pattern normalization is addressed in this chapter. The other fringe analysis stages are dealt in the following chapters.

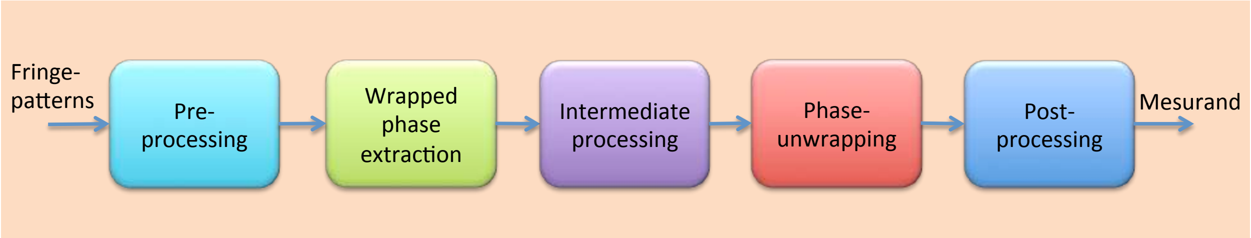


Figure 1.3: *Generic phase computing block for phase demodulation and the most common stages.*

Example 2 (Fringe-pattern generated by interference of two coherent beams)**.** *Let $E_o(p)$ and $E_r(p)$ be two coherent beams (corresponding to the object and reference beams), given, respectively, by*

$$E_o(p) = A_o(p)\exp[i\phi(p)] \text{ and}$$
$$E_r(p) = A_r(p)\exp[-i\delta(p)]$$

where i is the imaginary unit ($i^2 = -1$). For simplicity, these beams are linearly polarized into the same polarization plane. The intensity distribution from the interference of E_o and E_r is given by

$$\begin{aligned} I &= |E_o + E_r|^2 \\ &= E_o^2 + E_r^2 + 2E_oE_r\cos(\phi+\delta) \\ &= I_o + I_r + 2\sqrt{I_oI_r}\cos(\phi+\delta) \\ &= a + b\cos(\phi+\delta), \end{aligned} \tag{1.7}$$

where $|\cdot|$ denotes the module. The resulting background a and amplitude b are given as

$$a = I_o + I_r, \tag{1.8a}$$

$$b = 2\sqrt{I_oI_r}. \tag{1.8b}$$

Notice that the resulting intensity distribution (1.7) has the same mathematical description that (1.1). Clearly, the functions a and b do not provide any information about the phases ϕ or δ. In this case, the functions a and b contain only information about the intensity profiles I_o and I_r corresponding to the interfering beams.

1.3 | Fringe-pattern normalization

Many existing fringe-pattern processing algorithms for wrapped phase extraction assume that the background a_k and fringe amplitude b_k are constants so that the intensity variation correspond to the phase Φ_k only [6]. Unfortunately, only under stringent conditions, this assumption is fulfilled. Commonly, large variations of both background and modulation lights are usual in fringe-patterns [7]. For example, in holographic and speckle interferometry, the lack of illumination control (e.g., power variation of the laser source [8]) or the illumination by a divergent beam leads to a non-uniform background and modulation lights [9]. In this situation, conventional fringe-pattern processing algorithms compute an erroneous phase distribution because the variations of the background and modulation lights are misinterpreted as a phase contribution.

Consequently, the variations of the background and modulation lights are an important systematic noise source which cannot be neglected [6]. This systematic error source significantly reduces the accuracy of the final result because the assumption of cosine-shaped intensity distribution is not fulfilled [9]. According to the example 2, if the interest information is into the encoded phase $\phi+\delta$ (as is usually the case), the *fringe-pattern normalization* seems to be a desirable "filter" to suppress the less important illumination conditions before starting the phase demodulation process [5].

For a given fringe-pattern, the procedure of background light suppression and modulation light normalization is denominated ***fringe-pattern normalization*** [5–7, 10]. This process gives as result a pure sinusoidal phase modulated signal.

There exist several fringe-pattern normalization techniques, some examples are:

- Direct measurement of the background and modulations lights [11]. It is the most accuracy and simple method. However, the requirement of additional measurements increases the experimental complexity.
- Filtering in the frequency domain [12]. The most employed approaches are
  - Windowed Fourier filtering [13].
  - Orthogonal bandpass filtering [7],
  - Hilbert transform using Reisz filters [5],
  - The Schlieren filtering [7],
  - Envelope detection with spin filtering [6],
  - Adaptive filtering [14],
  - Continuous wavelet transform [12,13] and interferogram clipping [15].

  These methods have been successfully implemented in numerous applications. However, the filtering approach may distort the low-period fringes and the filter design is not easy. Thus, filtering is, in general, difficult to apply.
- The modified regularized phase tracking [16]. This approach is quite useful in fringe patterns with high frequency noise, although it requires a previous background suppression and a long processing time.
- Local histogram [17]. This is an efficient method but it depends strongly on the fringe-pattern spatial structure.
- The bidimensional empirical mode decomposition aided by the partial Hilbert transformation [10].
- The directional derivative approach [18].
- Iterative procedures [14, 16, 19]. They can be implemented for the most general situations and adaptive cases. However, the required high programming amount and processing time can be prohibitive.

In summary, the above described fringe-pattern normalization methods are either computationally exhaustive or difficult to implement. This situation can be addressed by using the parameter estimation approach.

1.4 | Statistical approach

Let a sinusoidal signal given as

$$y = a\cos\phi + \eta, \tag{1.9}$$

where ϕ is unknown and $\eta \sim \mathcal{N}(b, \sigma^2)$, or

$$y = a\cos\phi + b + \epsilon, \tag{1.10}$$

where $b = \mathcal{E}[\eta]$ and $\epsilon \sim \mathcal{N}(0, \sigma^2)$.

We have that the expected value of the signal y is

$$\mathcal{E}[y] = a\mathcal{E}[\cos\phi] + b + \underbrace{\mathcal{E}[\epsilon]}_{0}, \tag{1.11}$$

We have that the average $\mathcal{E}[\cos\phi]$ is

$$\mathcal{E}[\cos\phi] = \frac{1}{b-a}\int_a^b \cdots \tag{1.12}$$

$\cdots$. Thus, we can use

$$\mathcal{E}[\cos\phi] = 0. \tag{1.13}$$

Therefore

$$\mathcal{E}[y] = b. \tag{1.14}$$

On the other hand, we have that the variance of y leads to

$$\begin{aligned} \mathcal{V}[y] &= \mathcal{E}[(y - \mathcal{E}[y])^2] \\ &= \mathcal{E}[(a\cos\phi + \epsilon)^2] \\ &= \mathcal{E}[a^2\cos^2\phi + \epsilon^2 + 2a\epsilon\cos\phi] \\ &= \mathcal{E}[a^2(1+\cos 2\phi)/2 + \epsilon^2 + 2a\epsilon\cos\phi] \\ &= \frac{1}{2}a^2 + \frac{1}{2}a^2\underbrace{\mathcal{E}[\cos 2\phi]}_{0} + \underbrace{\mathcal{E}[\epsilon^2]}_{\sigma^2} + 2a\mathcal{E}[\epsilon\cos\phi] \end{aligned} \tag{1.15}$$

Under the assumption that the variables ϵ and $\cos\phi$ are uncorrelated, then $E[\epsilon\cos\phi] = E[\epsilon]E[\cos\phi] = 0$. Therefore,

$$\mathcal{V}[y] = \frac{1}{2}a^2 + \sigma^2. \tag{1.16}$$

Thus, the amplitude of the sinusoid can be obtained as

$$a = \sqrt{2(\mathcal{V}[y] - \sigma^2)}. \tag{1.17}$$

Finally, the sinusoidal signal y ca be normalized as

$$\bar{y} = \frac{y - \mathcal{E}[y]}{\sqrt{2(\mathcal{V}[y] - \sigma^2)}}. \tag{1.18}$$

1.5 | Fringe-pattern normalization by the parameter estimation approach

An alternative approach for fringe-pattern normalization consists on the parameter estimation. This approach uses the least-squares method to generate a fast, automatic and robust algorithm. A detailed explanation of this fringe-pattern normalization method is given as follows.

Because a least-squares approximation process is employed for fringe-pattern normalization, some assumptions derived from the requirements of the least-squares method itself must be to take into account.

Least-squares assumptions

The least-squares is a successful method to process experimental data sets corrupted with symmetrically distributed (e.g., Gaussian) noise with zero mean. Thus, many fringes (open and closed in any combination) across the recorded fringe-patterns are required in order to the sinusoidal term in (1.1) satisfies both symmetric distribution and zero mean. Then, an approximated solution for $a_k(p)$ and $b_k(p)$ can be estimated by a least-squares polynomial fitting.

1.5.1 | Intensity profiles

As it was shown by example 2, the background a and amplitude b functions can be estimated by using the fact that a and b have a close relationship with the intensity profile of the interfering beams[3]. Particularly, for a laser source, the irradiances I_o and I_r are assumed to be Gaussian functions modelled as

$$I_o = \mathcal{I}_o + R_o,$$
$$I_r = \mathcal{I}_r + R_r,$$

where $\mathcal{I}_o$ and $\mathcal{I}_r$ are P-degree polynomials (the respective truncated Taylor series) with P equal to the desired approximation order. R_o and R_r are the corresponding residual functions. Clearly, if the polynomial model is exact, the residual functions will be noise.

Since the addition of two P-degree polynomials is another P-degree polynomial, the function a can be approximated by

$$a = I_o + I_r \approx Y_P C_P X_P^T, \tag{1.19}$$

where $C_P \in \Re^{(P+1)\times(P+1)}$ is an upper cross-triangular coefficient matrix, X_P and Y_P are the respective regression matrices where its columns are the basis functions employed. Notice that the coefficient matrix C_P defines an bi-polynomial (on basis X_P and Y_P) of degree at most P.

Similarly, since the product of two P-degree polynomial is a $2P$-degree polynomial, the function b^2 can be approximated by

$$b^2 = 4I_oI_r \approx 4Y_{2P}C_{2P}X_{2P}^T, \tag{1.20}$$

where $C_{2P} \in \Re^{(2P+1)\times(2P+1)}$ is an upper cross-triangular coefficient matrix, X_{2P} and Y_{2P} are the respective regression matrices.

1.5.2 | The degree of the polynomials

When the polynomial description of the intensity profile is chosen, the profile is fully characterized by setting the polynomial basis and the degree. For example, the Taylor expansion is a polynomial representation in canonical basis of degree P where P is the order of approximation.

Accordingly, since the background light is the sum of two P-degree polynomials, say $I_o + I_r$ as shows (1.19), we conclude that the background a must be a P-degree polynomial.

[3] Although the particular case of two interfering beams is exposed, the conclusion about that the functions a and b include only illumination conditions is valid for non-interfering situations (e.g. fringe-projection). Thus, this normalization approach can be implemented to other applications. Perhaps some minor adaptations are necessary; for example, the definition of appropriate basis functions.

On the other hand, the modulation light is the square root of the product of two P-degree polynomials, say $2(I_oI_r)^{1/2}$ as shows (1.20). Thus, seems logical that the modulation b is a P-degree polynomials as well because, even though the product I_oI_r results in a $2P$-degree polynomial, the square root operation should reduce the polynomial degree by half. Unfortunately, this is true for *perfect square polynomials* only. The example 3 will illustrate this.

Example 3 (Perfect and non-perfect square polynomials)**.** *For simplicity, the 1-dimensional case is addressed. Let $p_1(x)$ and $p_2(x)$ be two 2nd-degree polynomials, say*

$$\begin{aligned} p_1(x) &= (x-1)^2, \\ p_2(x) &= (x+1)^2, \end{aligned} \tag{1.21}$$

for $x \in [-2, 2]$ as shows Fig. 1.4(a). Now, the product $p_1(x)p_2(x)$ is computed and shown in Fig. 1.4(a). Finally, the square root is applied to the resulting function. Fig. 1.4(b) shows the result of this procedure.

From Fig. 1.4(b), it is clear that the curve $[p_1(x)p_2(x)]^{1/2}$ is not a 2nd-degree polynomial (a parabola curve). For this particular case, the curve $[p_1(x)p_2(x)]^{1/2}$ is equal to

$$|x^2 - 1|.$$

This last function can not be a single polynomial for any abscissa x.

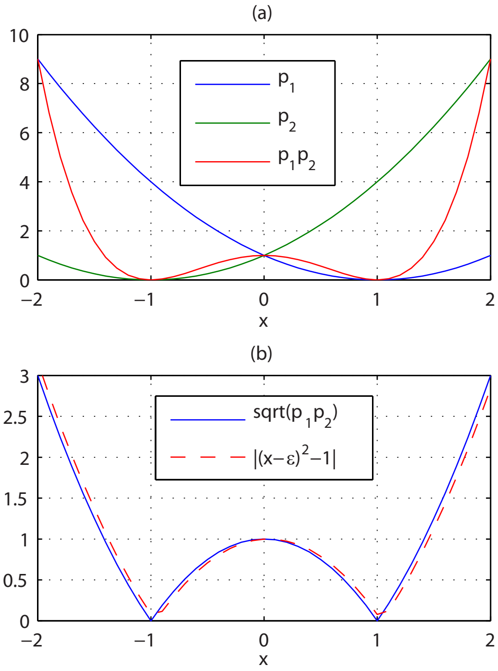


Figure 1.4: *The square root of the product of two 2nd-degree polynomials is not necessary another 2nd-degree polynomial. The small value ε in (b) was added in order to visualize the two superposed curves.*

The explicit structure (polynomial) of the functions a and b^2, described by (1.19) and (1.20), is exploited for fringe-pattern normalization. Moreover, this approach allow us to perform estimation of a and b in *single* and *multiple* mode.

Single and multiple estimation mode

Usually, the parameters a and b in (1.1) are restricted to be functions of p only. In other words, all fringe-patterns (1.1) must have the same background and modulation lights. This situation will be referred to as ***single estimation***.

The most general case is that the parameters a and b change for each fringe-pattern[a]. This situation will be referred to as ***multiple estimation***.

[a] Some examples of this situation are: phase-shifting interferometry by controlling the electrical current of the laser diode [8], amplitude modulation [20], diffraction with amplitude grating [21, 22], and lateral displacement of the illumination source [23].

1.5.3 | Multiple estimation — $a_k(p)$ and $b_k(p)$

If the parameters a_k and b_k are required to be estimated for each k-th fringe-pattern I_k (1.1) (or there is a single fringe-pattern), the least-squares method is used to fit the polynomial (1.19) to the experimental measurements (1.1) finding the coefficient matrix $C_P(k)$ as

$$\hat{C}_P(k) = Y_P^{\dagger} I_k X_P^{\ddagger}. \tag{1.22}$$

With the above computed matrix $\hat{C}_P(k)$, the background light in I_k is estimated by (1.19). Namely

$$\hat{a}_k = Y_P \hat{C}_P(k) X_P^T, \tag{1.23}$$

where $\hat{a}_k$ is the best approximation of a_k (in least-squares sense for the particular basis functions employed).

Next, for the estimation of the modulation light, we compute first the elementwise exponentiation

$$\begin{aligned}(I_k - \hat{a}_k)^2 &= b_k^2 \cos^2 \Phi_k \\ &= \frac{1}{2} b_k^2 + \frac{1}{2} b_k^2 \cos(2\Phi_k),\end{aligned} \tag{1.24}$$

where the last equation was obtained by applying the trigonometric identity $\cos^2 x = [1 + \cos(2x)]/2$.

Notice that (1.24) has the same structure as (1.1). This allow us to estimate $b_k^2/2$ in a similar form to the estimation of a_k. Accordingly, the least-squares method is used for fitting the polynomial (1.20) to $(I_k - \hat{a}_k)^2$ given by (1.24). For this, the corresponding matrix coefficients $\hat{C}_{2P}$ is found as

$$\hat{C}_{2P}(k) = Y_{2P}^{\dagger} (I_k - \hat{a}_k)^2 X_{2P}^{\ddagger}. \tag{1.25}$$

Then the quantity b_k^2 is reconstructed as

$$\hat{b}_k^2 = 2 Y_{2P} \hat{C}_{2P}(k) X_{2P}^T. \tag{1.26}$$

Finally, the modulation light $\hat{b}_k$ is obtained by computing the pointwise square root of the fitted polynomial (1.20). Again, as before with $\hat{a}_k$, $\hat{b}_k$ is the best approximation of b_k.

1.5.4 | Single estimation — $a(p)$ and $b(p)$

For specific applications, the equality of background and modulation lights for all fringe-patterns can be guaranteed by the optical setup [**?**]. For this case, it is required to compute a single pair background-modulation from all provided frames. Mathematically, instead of to find the solutions $\hat{C}_P(k)$ and $\hat{C}_{2P}(k)$ for each k with the equations

$$\begin{aligned} Y_P\hat{C}_P(0)X_P^T &= I_0, \\ &\vdots \\ Y_P\hat{C}_P(K-1)X_P^T &= I_{K-1}, \end{aligned} \tag{1.27}$$

and

$$\begin{aligned} 2Y_{2P}\hat{C}_{2P}(0)X_{2P}^T &= (I_0-\hat{a}_0)^2, \\ &\vdots \\ 2Y_{2P}\hat{C}_{2P}(K-1)X_{2P}^T &= (I_{K-1}-\hat{a}_{K-1})^2, \end{aligned} \tag{1.28}$$

respectively, it is required to find just two coefficient matrices, $\hat{C}_p$ and $\hat{C}_{2P}$, such that

$$\begin{bmatrix} Y_P \\ \vdots \\ Y_P \end{bmatrix} \hat{C}_P X^T = \begin{bmatrix} I_0 \\ \vdots \\ I_{K-1} \end{bmatrix}, \tag{1.29}$$

and

$$2\begin{bmatrix} Y_{2P} \\ \vdots \\ Y_{2P} \end{bmatrix} \hat{C}_{2P} X^T = \begin{bmatrix} (I_0-\hat{a}_0)^2 \\ \vdots \\ (I_{K-1}-\hat{a}_{K-1})^2 \end{bmatrix}, \tag{1.30}$$

where $\hat{a}$ and $\hat{b}^2$ approximate all $\hat{a}_k$ and all $\hat{b}_k^2$ ($k=\overline{0,K-1}$), respectively.

It is not difficult to prove that the matrices $\hat{C}_p$ and $\hat{C}_{2P}$ are the arithmetic mean of $\hat{C}_p(k)$ and $\hat{C}_{2P}(k)$, respectively. This claim is verified for $\hat{C}_p$ in the example 4. In summary

$$\hat{C}_P = \mathcal{Y}_P^\dagger \mathcal{J} X_P^\ddagger = \frac{1}{K}\sum_{k=0}^{K-1}\hat{C}_P(k), \tag{1.31}$$

$$\hat{C}_{2P} = \mathcal{Y}_{2P}^\dagger [\mathcal{J}-\mathcal{A}]^2 X_{2P}^\ddagger = \frac{1}{K}\sum_{k=0}^{K-1}\hat{C}_{2P}(k), \tag{1.32}$$

where $\mathcal{Y}_P = [Y_P^T \;\cdots\; Y_P^T]^T$, $\mathcal{Y}_{2P} = [Y_{2P}^T \;\cdots\; Y_{2P}^T]^T$, $\mathcal{J} = [I_0^T \;\cdots\; I_{K-1}^T]^T$, and $\mathcal{A} = [\hat{a}^T \;\cdots\; \hat{a}^T]^T$ are block matrices formed by K times the matrix Y_P, Y_{2P}, I_0, or $\hat{a}$ depending on the case.

1.5.5 | Saturation

Once the background and modulation lights are estimated, the normalization of the fringe-patterns is carried out as follows. With the recovered functions $\hat{a}$ and $\hat{b}$, we compute

$$\tilde{I}_k = [I_k - \hat{a}_k]/\hat{b}_k \tag{1.33}$$

for all p such that[4] $\hat{b}(p) \neq 0$.

[4] The points p where $\hat{b}(p) = 0$ cannot be processed. However it is not a serious problem because the computation is point-wise. Therefore, the intensity values at such singular points can be determined indirectly by an appropriate interpolation from the nonsingular neighborhood points.

Example 4. *For estimation of a single background function from the given K fringe-patterns, the matrix system*

$$\begin{bmatrix} Y_P \\ \vdots \\ Y_P \end{bmatrix} C_P X^T = \begin{bmatrix} I_0 \\ \vdots \\ I_{K-1} \end{bmatrix}, \tag{1.34}$$

is solved for C_P in the least-squares sense as

$$\hat{C}_P = \begin{bmatrix} Y_P \\ \vdots \\ Y_P \end{bmatrix}^{\dagger} \begin{bmatrix} I_0 \\ \vdots \\ I_{K-1} \end{bmatrix} X^{\ddagger}. \tag{1.35}$$

By the definition of the least-squares inverse $[\cdot]^{\dagger}$, we have that

$$\begin{aligned} \begin{bmatrix} Y_P \\ \vdots \\ Y_P \end{bmatrix}^{\dagger} &= \frac{1}{K}(Y_P^T Y_P)^{-1}[Y_P^T \ \cdots \ Y_P^T] \\ &= \frac{1}{K}[Y_P^{\dagger} \ \cdots \ Y_P^{\dagger}], \end{aligned}$$

then

$$\begin{aligned} \hat{C}_P &= \frac{1}{K}[Y_P^{\dagger} \ \cdots \ Y_P^{\dagger}] \begin{bmatrix} I_0 \\ \vdots \\ I_{K-1} \end{bmatrix} X^{\ddagger} \\ &= \frac{1}{K}[Y_P^{\dagger} I_0 + \cdots + Y_P^{\dagger} I_{K-1}] X^{\ddagger} \\ &= \frac{1}{K}[\hat{C}_P(0) + \cdots + \hat{C}_P(K-1)]. \end{aligned} \tag{1.36}$$

In words, the matrix $\hat{C}_P$ is obtained as the arithmetic mean of the matrices $\hat{C}_P(k)$ with $k = \overline{0, K-1}$.

Notice that Eq. (1.33) does not guarantee that $\tilde{I}_k \in [-1, 1]$ because of the noise in I_k and the fact that $\hat{a}$ and $\hat{b}$ are approximations. Therefore, $\tilde{I}_k = \cos \Phi_k$ will have, in generally, complex solution for Φ_k. Then, to ensure that the solution is in real values, the *saturation function* is applied as

$$\bar{I}_k = \operatorname{sat}\left(\tilde{I}_k\right) = \cos \Phi_k, \tag{1.37}$$

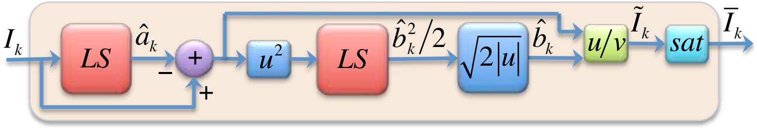
where $\bar{I}_k$ are the normalized fringe-patterns, and the ***saturation function*** is defined by

$$\operatorname{sat}(x) = \begin{cases} 1 & \text{if} \quad x > 1, \\ x & \text{if} \quad |x| \leq 1, \\ -1 & \text{if} \quad x < -1. \end{cases} \tag{1.38}$$

with $|\cdot|$ denoting the absolute value. Fig. 1.6 shows a plot of this saturation function.

This fringe-pattern normalization method is graphically represented in Fig. 1.5(a). `FPNorm` is the computational function resulting of an implementation of this algorithm in the Matlab software. The code is shown in Fig. 1.5(b).

The inputs and outputs involved by the function `FPNorm` are listed below.



(a) Block diagram. The letters u and v denote the first and second block's input, respectively.

```
function [bI,ha,hb,x,y] = FPNorm(I,deg,mode)
%Fringe-pattern normalization by parameter
%estimation method.
%
%This code is an implementation of the work
%reported in [Optics and Lasers in Engineering,
%Vol. 51(5), pp. 626 - 632 (2013). DOI:
%http://dx.doi.org/10.1016/j.optlaseng.2012.12.020
%by Rigoberto Juarez-Salazar, et. al.

    MNK = size(I);
    if numel(MNK) == 2; MNK = [MNK 1]; end

    x = (2*(0:(MNK(2)-1))/(MNK(2)-1) - 1)';
    y = (2*(0:(MNK(1)-1))/(MNK(1)-1) - 1)';

    % Bulding the regression matrices
    Xp = zeros(MNK(2),deg+1);
    Yp = zeros(MNK(1),deg+1);
    X2p = zeros(MNK(2),2*deg+1);
    Y2p = zeros(MNK(1),2*deg+1);
    for k = 0:deg
        Xp(:,k+1) = x.^k;
        Yp(:,k+1) = y.^k;
    end
    X2p(:,1:deg+1) = Xp;
    Y2p(:,1:deg+1) = Yp;
    for k = deg+1:2*deg
        X2p(:,k+1) = x.^k;
        Y2p(:,k+1) = y.^k;
    end

    % Computing the least-squares inverses
    Ypdag   = (Yp'*Yp)\Yp';
    Xpddag =  Xp/(Xp'*Xp);
    Y2pdag  = (Y2p'*Y2p)\Y2p';
    X2pddag =  X2p/(X2p'*X2p);

    Cpk  = zeros(deg+1,deg+1,MNK(3));
    C2pk = zeros(2*deg+1,2*deg+1,MNK(3));
    ha = zeros(MNK);
    for k=1:MNK(3)
        Cpk(:,:,k)  = Ypdag*I(:,:,k)*Xpddag;
        ha(:,:,k)   = Yp*Cpk(:,:,k)*Xp';

        C2pk(:,:,k) = Y2pdag*(I(:,:,k) ...
                      - ha(:,:,k)).^2*X2pddag;
    end

    if strcmp(mode,'multiple')
        hb = zeros(MNK);
        for k=1:MNK(3)
            hb(:,:,k) = ...
                sqrt(abs(2*Y2p*C2pk(:,:,k)*X2p'));
        end
        bI = sat((I - ha)./hb);
    else
        Cp  = sum(Cpk, 3)/MNK(3);
        C2p = sum(C2pk,3)/MNK(3);
        ha   = Yp*Cp*Xp';
        hb   = sqrt(abs(2*Y2p*C2p*X2p'));

        tI = zeros(MNK);
        for k=1:MNK(3)
            tI(:,:,k) = (I(:,:,k) - ha)./hb;
        end
        bI = sat(tI);
    end
end

function B = sat(A)
%Saturation function, all entries of the matrix A
%are restricted to be within the interval [-1,1].
    B = A;
    for k=1:size(A,3)
        for j=1:size(A,2)
            for i=1:size(A,1)
                if     A(i,j,k)> 1; B(i,j,k)  =  1;
                elseif A(i,j,k)<-1; B(i,j,k)  = -1;
                end
            end
        end
```

(b) Implementation by a Matlab function.

Figure 1.5: *Fringe pattern normalization method by using the parameter estimation approach.*

Inputs :

- `I` is a $M \times N \times K$ matrix where $M \times N$ is the size of each data frame and K is the number of available frames,
- `deg` is an integer denoting the polynomial degree used for estimation the background light (by definition, a `2*deg`-th polynomial will be used to estimate the modulation light),
- `mode` is an string type variable to select between single (default) (by setting `mode='single'`) and multiple (by setting `mode='multiple'`) estimation modes.

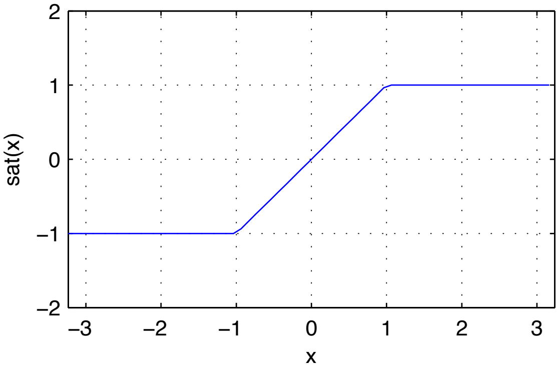


Figure 1.6: *Plot of the saturation function defined by Eq.* (1.38)*. Notice that this function is equal to the identity function $f(x) = x$ in the interval $x \in [-1, 1]$.*

Outputs :

- `bI` is the matrix of normalized fringe-patterns. This matrix has the same size than the input matrix `I`.
- `ha` is the recovered background light.
- `hb` is the recovered modulation light .
  The matrices `ha` and `hb` are either $M \times N$ or $M \times N \times K$ matrices depending on if `'single'` or `'multiple'` mode estimation was selected, respectively.
- `x` and `y` are vectors used to build the lattice where the fitted polynomials were defined.

Below, the functionality of this method is shown by computer simulation and by processing experimental data frames.

1.5.6 | Computer simulation

To show the functionality of the parameter estimation approach for fringe-pattern normalization, a computer simulation is designed.

Consider the background light, the modulation light, and the phase distribution given, respectively, by

$$a(p) = 160 - 15x - 15y - 36x^2 - 36y^2, \tag{1.39a}$$

$$\begin{aligned} b(p) =& 15 + 8x + 35x^2 - x^3 - 30x^4 \\ &+ 13y + 35y^2 - y^3 - 30y^4, \end{aligned} \tag{1.39b}$$

$$\phi(p) = 35\pi \frac{x^2 - y^2}{x^2 + y^2 + \epsilon} xy, \tag{1.39c}$$

for $x, y \in [-1, 1]$ and where ϵ is the machine epsilon (the least possible number in finite arithmetic) in order to avoid a singularity at the point $p = (0, 0)$. The respective plots of the above functions are shown in Figs. 1.7(a)-(c). The resulting fringe-pattern is displayed in Fig. 1.7(d).

Now, the simulated fringe-pattern is processed by the normalization method depicted in Fig. 1.5(a). The computed background, modulation, and normalized fringe-pattern are shown in Figs. 1.7(e)-(g), respectively. For illustration purposes, Fig. 1.7(h) shows the true normalized intensity, i.e., the cosine of (1.39c).

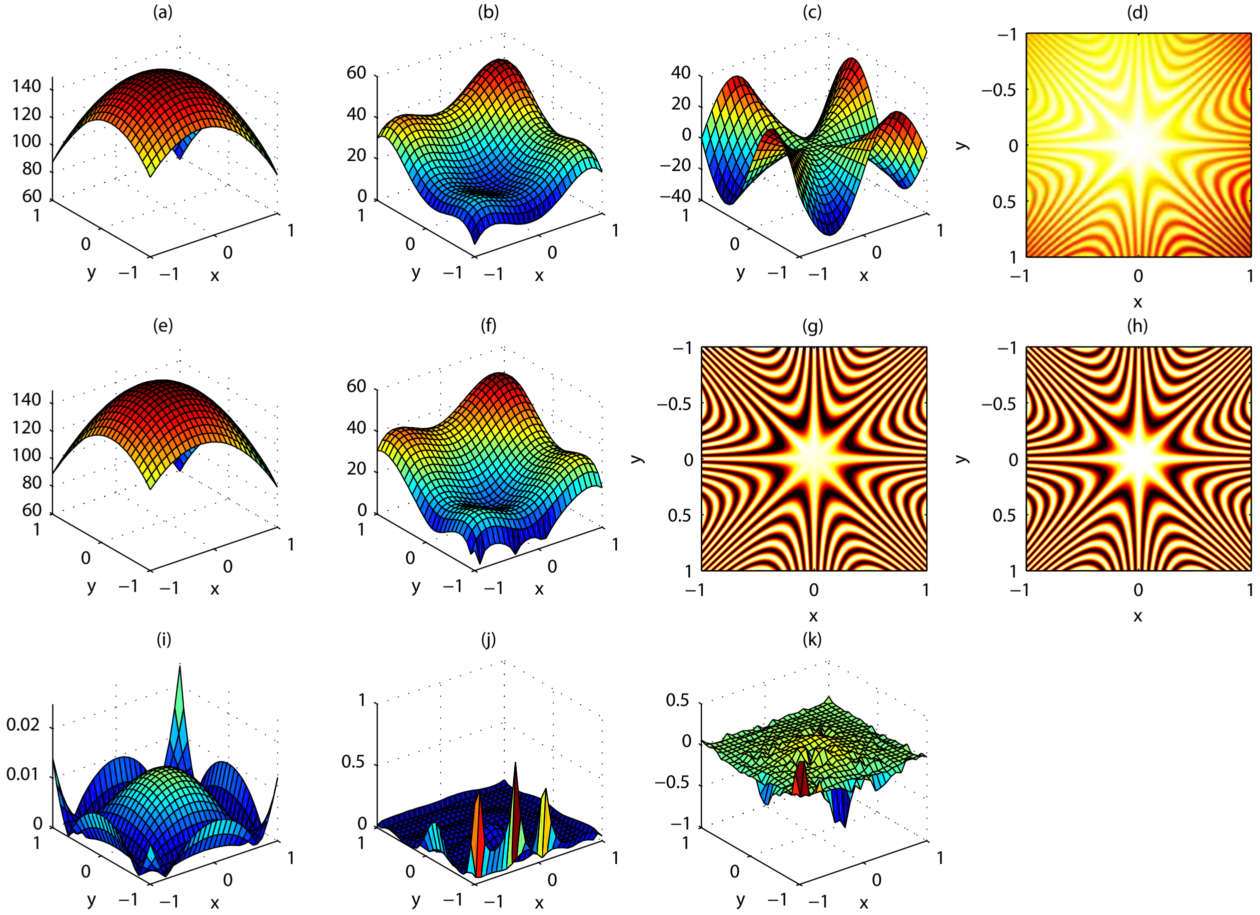


Figure 1.7: *Simulation results of fringe-pattern normalization by parameter estimation. (a)-(c) Synthetic background, modulation, and phase distribution, respectively. (d) Simulated fringe-pattern. (e)-(g) Recovered background light, modulation light, and normalized fringe-pattern, respectively. (h) True normalized intensity [the cosine of the function in (c)]. (i) and (j) show, respectively, the relative error of background and modulation estimations. (k) The difference between the normalized fringe-pattern (g) and the true normalized intensity (h).*

From Fig. 1.7, by direct comparison between the plot pairs (a)(e), (b)(f), and (g)(h), we can see that a successful data processing was performed. Moreover, in order to evaluate the accuracy levels reached by this normalization method, the plots 1.7(i) and 1.7(j) show the relative error of the background and modulation estimation. Finally, 1.7(k) shows the difference between the output $\bar{I}$ of the normalization procedure and the true normalized intensity $\cos\phi$.

1.5.7 | Processing of experimental fringe-patterns

The robustness of this normalization algorithm is tested by processing the experimental fringe-pattern shown in Fig. 1.8(a). This frame was acquired from a Twymann-Green interferometer. The frame size is of 984×1071 pixels with a pixel depth of 8-bits in scale gray.

This fringe-pattern was normalized by the described method and the computed background light,

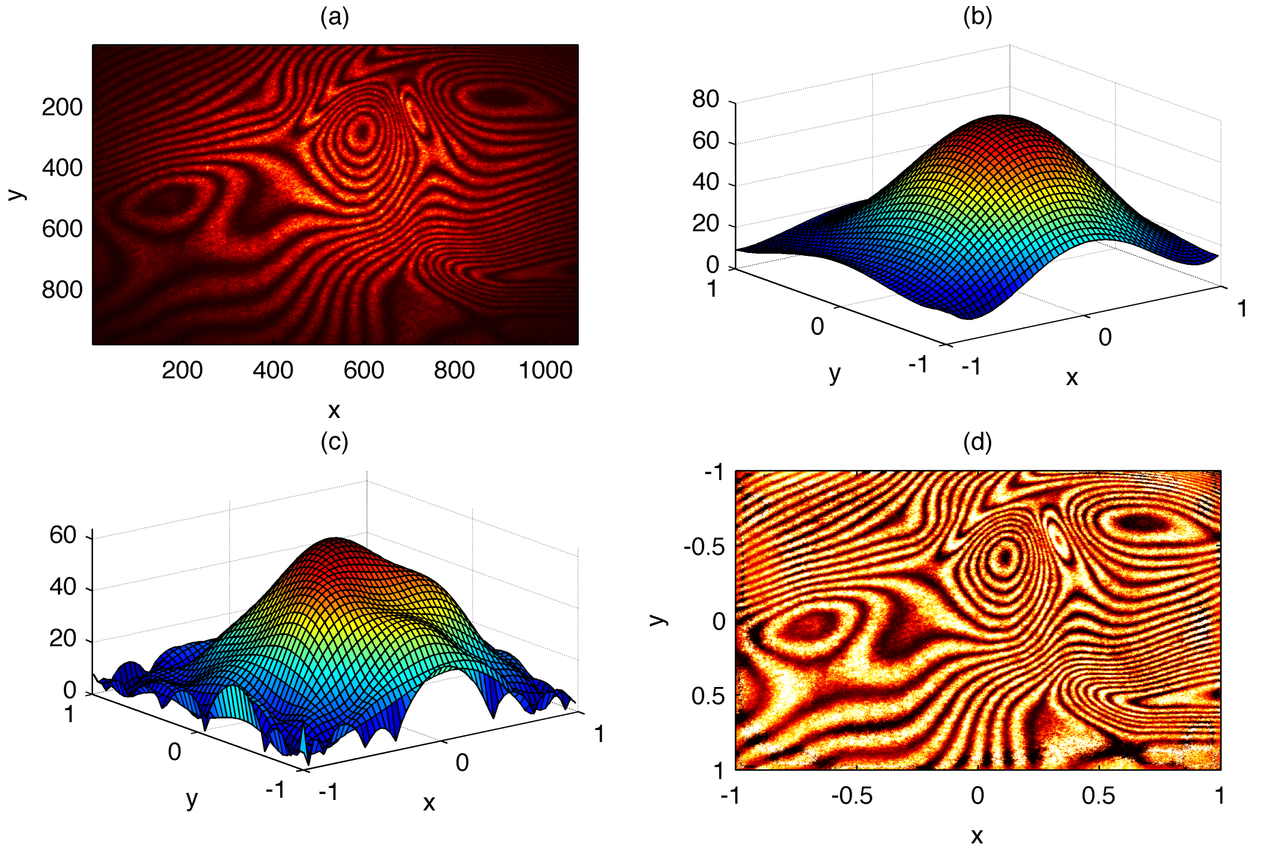


Figure 1.8: *Fringe-pattern normalization of an experimental frame. (a) Original acquired fringe-pattern. (b) and (c) Recovered background and modulation lights, respectively. (d) Resulting normalized fringe-pattern.*

modulation light, and the resulting normalized intensity are shown in Figs. 1.8(b)-(d), respectively.

1.5.8 | Application of multiple mode estimation

In the most phase-shifting techniques, it is assumed that the fringe visibility is kept for all fringe-patterns. However, for some applications this assumption is not valid. An example of this is the *electronic speckle pattern interferometry* by using a $4f$ optical processor [21].

The above referred interferometer induces the required phase shifts by translating laterally to the optical axis a Ronchi ruling on the Fourier plane. Since such a ruling is an amplitude object, it changes the illumination conditions on the output plane. Accordingly, the acquired fringe-pattern, besides to present a phase-shifting, they will exhibit a spatio-temporal fringe visibility variation. The problem of phase demodulation from phase-shifted fringe-patterns with spatio-temporal visibility can be addressed by the multiple estimation mode for fringe-pattern normalization as it is concluded in [22].

1.6 | Conclusion

A fringe-pattern normalization method by using the parameter estimation approach was presented. This method exploits the available information about the spatial structure of the intensity illumination profile. Thus, a simple algorithm to estimate the background and modulation light components can be derived. This normalization procedure is fast, robust, and computationally efficient. Moreover, it does not requires user intervention. Therefore, it could be implemented in automatic applications.

Chapter 2
Phase demodulation by spatial carrier

The old scientific method: Here are the facts. What theory can we draw from them? The new one: Here's the theory. What facts can we find to support it?

Jean-Perre Petit, in *The twin universe*

Contents

METROLOGY of extreme physical phenomena is possible by using a spatial carrier. The so-called *Fourier fringe analysis* is an important well-stocked arsenal of fringe-pattern processing that exploits the benefits of spatial carriers.

Fourier fringe analysis is the topic of this chapter. It is explained how a spatial carrier is useful to wrapped phase extraction by using the Fourier transform. The standard Fourier fringe analysis is presented and its difficulties from filtering are exposed.

Finally, the so-called *Fourier fringe-normalized analysis* is devised to overcome the filtering problems exhibited by the original unnormalized version. The simple filtering procedure, higher spatial resolution, and low computation time benefits of Fourier fringe-normalized analysis allow us to implement it on automatic real-time applications.

Throughout this chapter, the concepts are illustrated by using several figures and examples. The computer code in MATLAB software is provided.

2.1 | Introduction

The Fourier method is a powerful tool used in an important well-stocked arsenal of image processing techniques such as image compression [24] and fringe-pattern analysis to phase demodulation (Fourier fringe analysis) [3, 25].

In the previous chapter we studied the fringe-patterns modeled by Eq. (1.1), which is reproduced here for convenience:

$$I_k(p) = a_k(p) + b_k(p)\cos\Phi_k(p), \tag{2.1}$$

for $k = \overline{0, K-1}$ where K is the number of available frames. It was shown that the model (2.1) is valid regardless of the measurement method used (e.g., interference, and fringe-projection, etc.). It was explained that, in general, the information of interest is encoded into $\Phi_k(p)$ while both $a_k(p)$ and $b_k(p)$ contain less important data about the illumination setup. Accordingly, a normalization method to suppress both $a_k(p)$ and $b_k(p)$ was presented.

The problem that still remains is: to extract the phase distribution of interest from normalized fringe-patterns $\bar{I}_k(p) = \cos\Phi_k(p)$. This seems to have a trivial solution (a inverse cosine of the normalized fringe-patterns). Unfortunately, the cosine function wraps the phase $\Phi_k(p)$ in an inconvenient way (there is a sign ambiguity). Therefore, a more advanced techniques for wrapped phase extraction must be employed.

The key idea for wrapped phase extraction is to add a "reference" function into the encoded phase $\Phi_k(p)$, say

$$\Phi_k(p) = \phi(p) + \delta_k(p), \tag{2.2}$$

where $\phi(p)$ is the phase function of interest to be recovered and $\delta_k(p)$ is the reference function known as *carrier*[1]. The carrier can be *spatial* or *temporal* and, depending on it, two main approaches for wrapped phase extraction can be formulated. Namely, the frequency transformation and the phase-shifting approaches.

The ***phase-shifting approach*** for wrapped phase extraction [26] uses a ***temporal carrier*** which can be defined as

$$\delta_k = \delta(k). \tag{2.3}$$

The phase-shifting algorithms requires a set of $K > 1$ phase-shifted fringe-patters to extract (in a wrapped format) the phase distribution $\phi(p)$. This approach is analyzed in the following chapter.

On the other hand, the ***frequency transformation approach*** uses a ***spatial carrier*** defined by [27]

$$\delta(p) = 2\pi f_0 \cdot p, \tag{2.4}$$

where $f_0 = (f_x, f_y) \in \Re^2$ is known as the ***frequency carrier*** and the symbol $[\cdot]$ denotes the inner product. Notice that the spatial carrier (2.4) defines a tilted plane perpendicular to the vector $[f_x \; f_y \; -1]^T = [f_0 \; -1]^T$ or $[-f_0 \; 1]^T$. This chapter is devoted to the study of this approach.

Since this approach was devised by Mitsuo Takeda [27], it is commonly referred to as *Takeda's method*. In other references, this fringe analysis approach is also known as *Fourier transform method for fringe-pattern analysis* or simply as *Fourier transform method* [28]. When the filtering is performed by using a simple rectangular window, the method is referred to as *standard Fourier transform method*. Since the use of a tilted plane as a carrier function generates a convenient frequency transformation (a translation of the interest spectrum), this method is also known as *frequency transformation method*. In this thesis, the term *Fourier fringe analysis* will be used.

[1] The example 2 in chapter 1 shows that, for the case of interference fringes, the carrier phase $\delta_k(p)$ is the phase provided by the reference beam. Therefore, the carrier phase is controlled by the reference arm of the interferometer (e.g., translating or tilting the reference mirror). In the case of fringe-projection, the carrier phase is controlled by the projected grating.

2.2 | Fourier fringe analysis

Unlike the phase-shifting approach, the wrapped phase extraction by frequency transformation requires only a single fringe-pattern [$K = 1$ in Eq. (2.1)]. Therefore, hereafter, the variable k is not written down for brevity.

The particular structure of the carrier (2.4) can be exploited by using the *shift theorem of the Fourier transform*. This allows us to extract the function $\phi(p)$ in (2.2) by using the Fourier transform.

Theorem 1 (Fourier transform shift theorem). *If $\mathcal{F}f(p) = F(u)$ is the Fourier transform of $f(p)$, then, for the constant vector p_0,*

$$\mathcal{F}f(p - p_0) = F(u)e^{-j2\pi p_0 \cdot u}. \tag{2.5}$$

This is, shifting the origin of the coordinate system relative to a given function introduces a linear phase factor in the spatial frequency domain. The opposite is true as well due to the symmetry of the Fourier transform operation:

$$\mathcal{F}f(p)e^{-j2\pi p_0 \cdot u} = F(u + p_0). \tag{2.6}$$

The theorem 1 justifies the use of the Fourier transform to process fringe-patterns with the aim of to extract the encoded phase distribution (in wrapped format). This is explained as follows.

By using the Euler's formula, the cosine function can be written in the complex form as

$$\cos x = \frac{1}{2}e^{ix} + \frac{1}{2}e^{-ix}. \tag{2.7}$$

Accordingly, the equation (2.1) can be restated as

$$\begin{aligned} I(p) &= a(p) + b(p)\cos[\phi(p) + 2\pi f_0 \cdot p] \\ &= a(p) + \frac{1}{2}b(p)e^{i\phi(p)+i2\pi f_0 \cdot p} + \frac{1}{2}b(p)e^{-i\phi(p)-i2\pi f_0 \cdot p}, \\ &= a(p) + \frac{1}{2}b(p)e^{i\phi(p)}e^{i2\pi f_0 \cdot p} + \frac{1}{2}b(p)e^{-i\phi(p)}e^{-i2\pi f_0 \cdot p}, \\ &= a(p) + c(p)e^{i2\pi f_0 \cdot p} + c^*(p)e^{-i2\pi f_0 \cdot p}, \end{aligned} \tag{2.8}$$

where $c(p) = b(p)\exp[i\phi(p)]/2$, $i^2 = -1$, and the asterisk [*] denotes complex conjugate. Notice that the complex number $c(p)$ contains the phase distribution of interest $\phi(p)$ as its argument. Since the Fourier transform is a linear operator, it can be applied to each term in the last equality in (2.8). Moreover, the theorem 1 is applicable by the equation (2.6). Therefore, the Fourier transform of (2.8) is

$$\mathcal{I}(\mu) = A(\mu) + C(\mu - f_0) + C^*(\mu + f_0), \tag{2.9}$$

where the capital letters represent the Fourier spectrum and $\mu = (\mu_x, \mu_y)$ is the spatial frequency coordinate. From Eq. (2.9) is explicit the translation, in the Fourier space, which suffer the spectrum C and C^* by the spatial carrier f_0.

Roughly speaking, in the Fourier plane, the spectrum $\mathcal{I}(\mu)$ has three lobes: the spectrum $A(\mu)$ at the origin, and the symmetric lobes C and C^* centered at $-f_0$ and $+f_0$, respectively.

Since the complex number $c(p)$ (or its conjugate) has the information of interest, the aim is to isolate the spectrum $C(\mu - f_0)$ and translating it by f_0 toward the origin to obtain $C(\mu)$. Then, by computing its inverse Fourier transform, the complex number $c(p)$ is recovered. Finally, the extraction of the wrapped version of the phase of interest $\phi(p)$ is obtained as the argument of $c(p)$.

This procedure is depicted by the block diagram shown in Fig. 2.1. By assuming that the fast Fourier transform algorithm (FFT) is available (as it is the case for the MATLAB platform), the implementation this described method is easy as shows Fig. 2.2. The example 5 illustrates how the Fourier fringe analysis works.

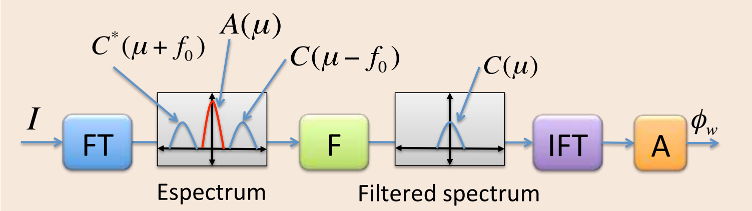


Figure 2.1: *Visual block diagram description of the Fourier fringe analysis. (FT) Fourier transform. (F) Filtering. (IFT) inverse Fourier transform. (A) Argument.*

The essential step in this procedure is the so-called *spectral filtering* which consist on the correct isolation of the spectrum $C(\mu - f_0)$ or $C^*(\mu + f_0)$ from $\mathcal{I}(\mu)$ [29]. The most evident problem is the presence of the zero order spectrum $A(\mu)$ because it can be overlapped with either of C or C^*. This problem is addressed later in this chapter by the so-called *Fourier fringe-normalized analysis* [28, 30].

Example 5. *Let the background light, modulation light, and phase distribution given, respectively, by*

$$\begin{aligned} a(p) &= 160 - 15x - 15y - 36x^2 - 36y^2, \\ b(p) &= 15 + 8x + 35x^2 - x^3 - 30x^4 \\ &\quad + 13y + 35y^2 - y^3 - 30y^4, \\ \phi(p) &= 5 \cdot \texttt{peaks(512)}, \end{aligned} \tag{2.10}$$

for the domain $x, y \in [-1, 1]$, and where `peaks(·)` *is a function defined in MATLAB software [31].*

Figs. 2.3(a)-(c) show the respective plots of (2.10). For comparison reasons, Fig. 2.3(d) shows the wrapped version of ϕ obtained as $\arctan[(\sin\phi)/(\cos\phi)]$. The resulting fringe-pattern by using the carrier frequency $f_0 = (20, 0)$ [cycles per length unit] is shown in Fig. 2.3(e) and its respective Fourier spectrum in Fig. 2.3(f). Fig. 2.3(g) shows the spectral lobe $C(\mu + f_0)$ translated toward the origin and Fig. 2.3(h) shows the argument of its inverse Fourier transform. With this, the wrapped phase was recovered as it can be seen by comparison with Fig. 2.3(d).

2.3 | Fourier fringe-normalized analysis

The essential in wrapped phase extraction by the Fourier method is the filtering procedure [29]. It consist on the separation (on the frequency domain) of the desired information from others unwanted spurious contributions such as the background component, the modulation component, and the random noise, among others [29, 32, 33]. In the Fourier transform plane, the unwanted background and modulation components are represented as a peak in the origin, i.e. the zero-order spectrum; the desired information is represented as a peak function centered at the carrier frequency (and its conjugate), i.e. the first-order spectrum; also, because the noise is generally of high frequency, it is represented by the higher-order spectrum [3].

```
function [phi ftF ft] = FourierFA(I,f,w,phi0)
%Fourier fringe analysis for fringe-pattern with
%carrier frequency f = [fx fy].
%fx  is an integer >= than 0.
%fy is any integer.
%w = [left right top-down] defines the rectangular
%filter.
%phi0 is an initial phase (a real number).

[ny nx] = size(I);

ft = fftshift(fft2(I)); %Fringe-pattern spectrum

fx  = f(1); % x-component of carrier frequency
fy  = f(2); % y-component of carrier frequency
ml  = w(1); % left cutting
mr  = w(2); % right cutting
mtd = w(3); % top-down cutting

% Carrier frequency removal
ftf0 = zeros(ny,nx);
if fy > 0
    ftf0( 1:end-fy, fx+1:floor(nx/2+fx)) ...
        = ft(fy+1:end,1:floor(nx/2));
else
    ftf0(-fy+1:end, fx+1:floor(nx/2+fx)) ...
        = ft(1:end+fy,1:floor(nx/2));
end

% Spectrum filtering
ftF = ftf0;
if ml > 0; ftF(:,1:ml) = zeros(ny,ml); end
if mr > 0
    ftF(:,floor(nx/2+fx-mr+1):floor(nx/2+fx)) ...
        = zeros(ny,mr);
end
if mtd > 0
    ftF(1:mtd,:) = zeros(mtd,nx);
    ftF(end-mtd+1:end,:) = zeros(mtd,nx);
end

%Inverse Fourier transform
%of the filtered spectrum
iftF = ifft2(fftshift(ftF));
phi  = angle(iftF*exp(1i*phi0));
end
```

Figure 2.2: *Fourier fringe analysis implemented in MATLAB software by the function* `FourierFA`.

A proper band-pass filter is usually applied to perform the filtering procedure because it will retain the first-order spectrum while both the low frequencies around the zero spectra and the high frequencies due to noise are filtered out [34]. However, the filter design is not trivial because the Fourier method is highly dependent on the distribution of the zero- and the first-order spectrums [35,36]. The so-called leakage effect is a consequence of this dependence.

The leakage effect is the overlapping of two or more orders spectrum [33]. In Fourier fringe analysis, if the carrier frequency is high and the spectrum is narrow, then the zero- and first-order are well separated from each other and the leakage effect is avoided [37]. Otherwise, the first-order spectrum cannot be effectively

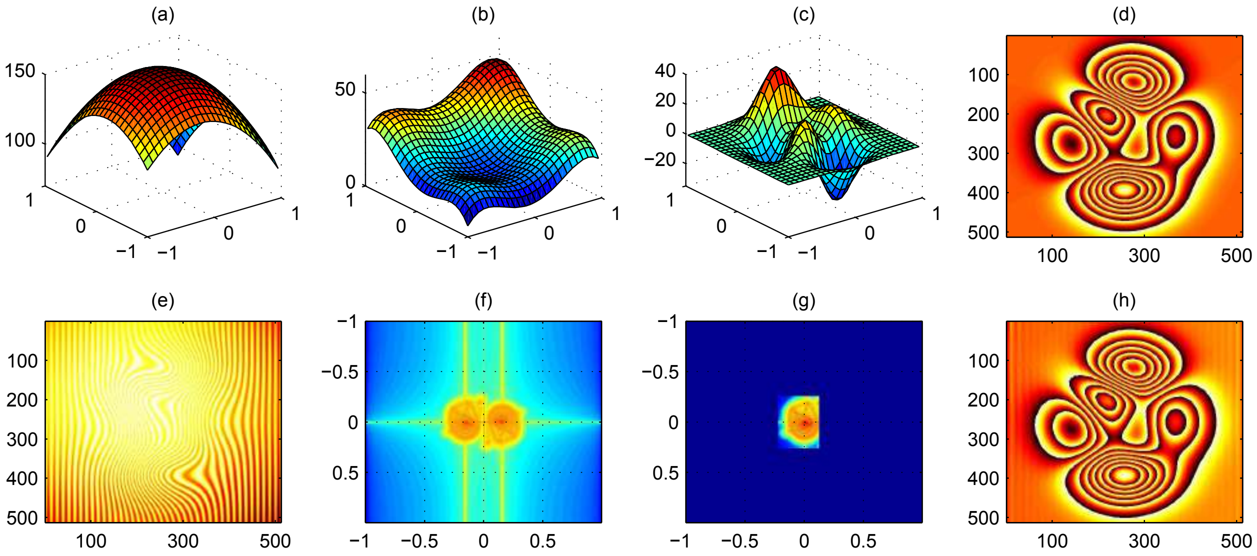


Figure 2.3: *Computer simulation of the Fourier fringe analysis. (a) Background light. (b) Modulation light. (c) Phase distribution. (d) Wrapped phase distribution. (e) Resulting fringe-pattern where the used carrier frequency is $f = [20, 0]$. (f) Modulus of the Fourier transform of (e). (g) Spectral filtering of (f). (h) Argument of the Fourier inverse transform of (g).*

isolated of the zero-order by a band-pass filter due to the order spectrum overlapping. This causes large error and can not give correct phase information [38, 39]. An illustration of this situation is given in Fig. 2.4(a).

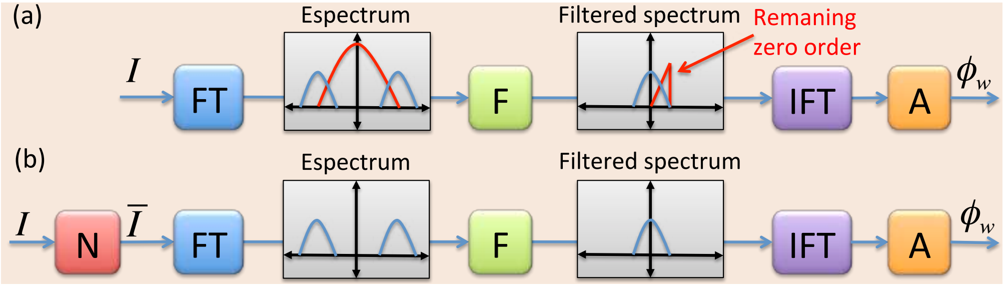


Figure 2.4: *Comparison of the (a) Fourier fringe analysis and the (b) Fourier fringe-normalized analysis schemes. Notice that, for the Fourier fringe analysis, if the carrier frequency is not sufficiently high, the orders $C(\mu - f_0)$ and $C^*(\mu + f_0)$ are overlapped with the zero order $A(\mu)$. Such an undesired situation is avoided in the Fourier fringe-normalized scheme because the zero order is no longer present.*

This problem can be overcome by selecting a higher carrier frequency, but this solution is limited by the spatial resolution of the detectors [40, 41]. Alternatively, the leakage effect can be reduced to some degree by using a narrow windows pass-band filter; however the spatial resolution is seriously reduced [42]. To cope with this situation, some advanced filters [35, 43, 44] and several filtering approaches have been implemented [13, 45–52]. However, most of them are computationally complex and/or expensive [53, 54].

Like in holographic applications, the leakage spectrum and the reduced spatial resolution problems can be solved if the zero-order spectrum is suppressed [55–58]. For a fringe-pattern, the zero-order is due to the background component, mainly. Thus, these components can be determined by either a direct measurement or an estimation procedure [59]. Finally, by subtracting it from the fringe-pattern, the zero-order is eliminated as shows Fig. 2.4(b).

Several methods for zero spectra suppression have been proposed. Some of them are the phase shifting methods [?, 32, 41, 60–64], the addition of consecutive interferograms [65], the complementary interferograms processing [42], the orthogonal/crossed fringe projection [66, 67], the empirical mode decomposition [68], the color fringe projection [69], and the optimization approach [70], among others. In general, the direct measurement of the background and modulation components requires additional measurements or more complicated systems disabling the method's ability to cope with real-time situations. In contrast, the estimation procedure of these components only demands a previous image-processing stage: the so-called fringe-pattern normalization [the red block N in Fig. 2.4(b)].

The fringe-pattern normalization approach consist of the background and modulation lights suppression. For this, the parameter estimation approach is used because, unlike other methods, it is simple, robust, user-free, and computationally efficient [71]. Thus, automatic real-time applications can be attended [30]. For more detail about the fringe-pattern normalization topic and, particularly, the parameter estimation approach, the reader is referred to the chapter 1.

Once the fringe-pattern was normalized, the filter design is not very critical because there is not zero-spectra anymore. Thus, even the simple half-plane filter can be used. This filter has the additional advantage that the spatial resolution is maximum for the Fourier method. From now on, the Fourier method applied to the analysis of normalized fringe-patterns will be referred to as *Fourier fringe-normalized analysis*.

A Fourier fringe-normalized analysis scheme can be depicted by the block diagram shown in Fig. 2.4(b). It is worth mentioning that this schemes can be built by any fringe-pattern normalization routine and any Fourier-based method. The benefits of such scheme are verified in this section. For this, the following proposals are suggested:

1. The fringe-pattern normalization method by parameter estimation. A detailed explanation of this procedure is exposed in the chapter 1.
2. The standard Fourier fringe analysis by using the simple half-plane filter. This procedure was exposed in the §2.2 of this chapter.

The above listed stages are described in the following. Next, the feasibility of the suggested scheme is verified by numerical simulation. Finally this proposal is tested by processing experimental fringe-patterns. Both numerical and experimental results show that the fringe-pattern normalization approach improves the standard Fourier fringe analysis.

From now on, in the cases where there is not risk of confusion, the dependences of the spatial variable p is not written down for brevity.

2.3.1 | Fringe-pattern normalization

The fringe-pattern normalization by parameter estimation exploits the information over the illumination profile. Specifically, we can represent both the background and modulation light by the Taylor series as

$$\begin{aligned} a &= \bar{a} + r_a, \\ b &= \bar{b} + r_b, \end{aligned} \tag{2.11}$$

where $\bar{a}$ and $\bar{b}$ are polynomials of degree equal to the desired approximation order, and r_a and r_b are its corresponding residual functions. Clearly, if the polynomial models $\bar{a}$ and $\bar{b}$ are accurate, the residual functions r_a and r_b will be just noise. So, the background and modulation light can be well approximated by fitting the polynomials $\bar{a}$ and $\bar{b}$ to the fringe-pattern data. Then, with the approximated background and modulation light, the normalization of (2.8) is carried out as

$$\bar{I}(p) = \text{sat}\left(\frac{I - \bar{a}}{\bar{b}}\right) = \cos(\phi + 2\pi f_0 \cdot p), \tag{2.12}$$

where the normalized fringe-pattern $\bar{I}$ is bounding to the interval $[-1, 1]$ by the saturation function sat$(\cdot)$. For more details on this normalization procedure the reader is referred to chapter 1.

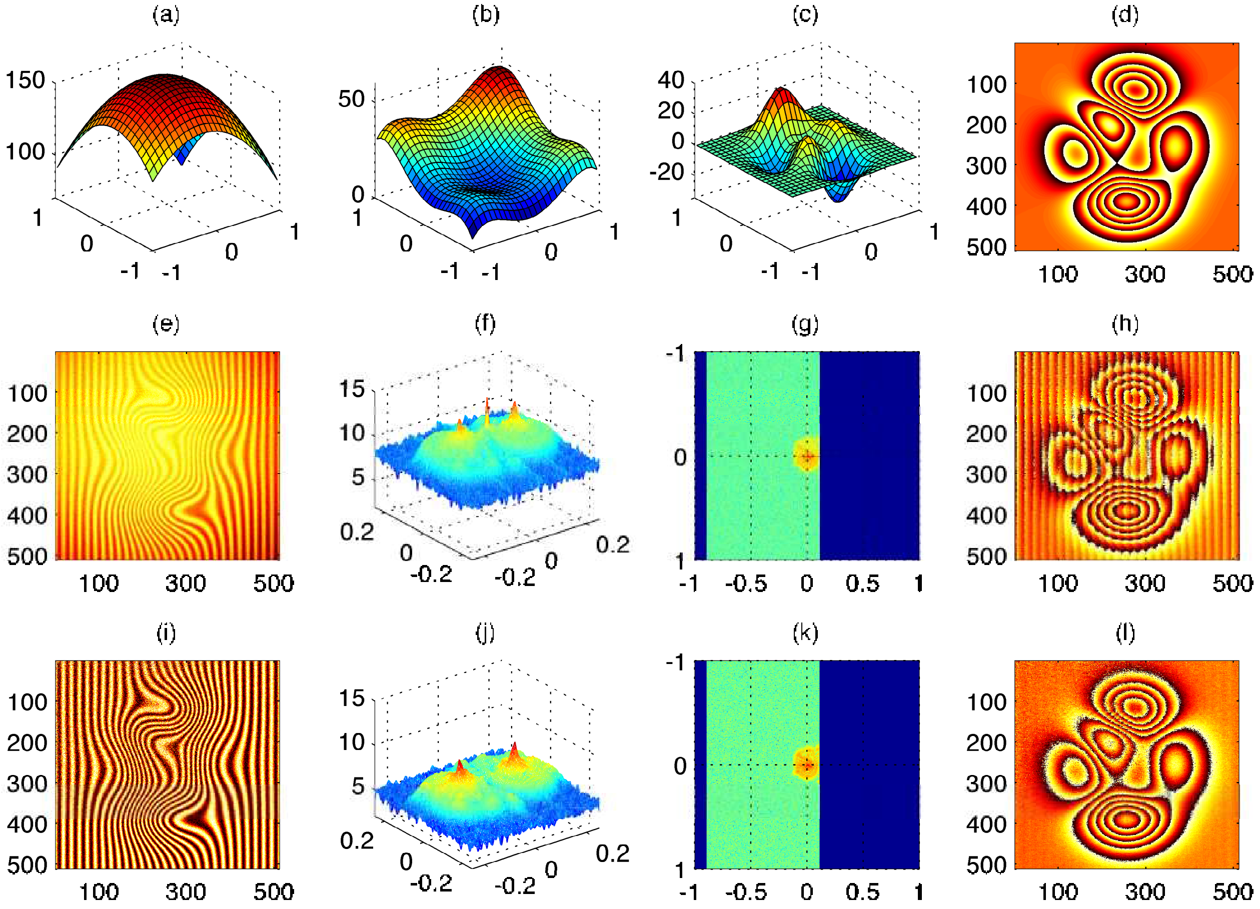


Figure 2.5: *Simulation results. (a) Background light, (b) modulation light, (c) phase distribution, (d) wrapped phase distribution. Wrapped phase extraction by using the half-plane filter in the standard Fourier transform (second row) and the Fourier fringe-normalized analysis (third row).*

2.3.2 | Standard Fourier fringe analysis

We use the half-plane filter in the standard Fourier transform method to obtain the wrapped phase distribution from a carried normalized fringe-pattern. For this, we rewrite the normalized fringe-pattern (2.12) as

$$\bar{I}(p) = c\exp(i2\pi f_0 \cdot p) + c^* \exp(-i2\pi f_0 \cdot p), \tag{2.13}$$

where $2c = \exp[i\phi]$. Applying the Fourier transform to $\bar{I}$ of (2.13) we obtain

$$\bar{\mathcal{I}}(\mu) = C(\mu - f_0) + C^*(\mu + f_0). \tag{2.14}$$

It is evident by comparing (2.9) and (2.14) that, in this fringe-normalized scheme, the filtering procedure is not very critical because the zero order [the spectrum $A(\mu)$ associated with the background light $a(p)$ in Eq. (2.8)] is not present.

The simple half-spectrum filter is applied to carried out the filtering procedure. The selected spectrum is translated by f_0 on the frequency coordinate system toward the origin to obtain $C(\mu)$. Then, the complex function $c(p)$ is obtained by applying the inverse Fourier transform to $C(\mu)$. Finally, the wrapped phase distribution is recovered as the argument of $c(p)$. This procedure is depicted by the block diagram shown in Fig. 2.4(b).

The functionality of this algorithm is illustrated by a computer simulation. Then, the scheme's feasibility is examined by processing experimental fringe-patterns.

2.3.3 | Computational simulation

Consider the background light, modulation light, and phase distribution given in (2.10). The respective plots are shown in Fig. 2.5(a)-(d). Considering the carrier frequency $f = (14, 0)$ [cycles per length unit] and an additive Gaussian noise with $\sigma(5\%)$, the resultant fringe-pattern, given by (2.8), is shown in Fig. 2.5(e).

Now, the simulated fringe-pattern shown in Fig. 2.5(e) is processed by using the half-plane filter in both the standard Fourier fringe analysis and the fringe-normalized version in order to compare their results. Thus, Figs. 2.5(f)-(h) show the module of Fourier spectrum, the filtering procedure, and the recovered wrapped phase, respectively, obtained by using the standard method. On the other hand, Figs. 2.5(i)-(l) show the normalized fringe-pattern, the module of Fourier spectrum, the filtering procedure, and the recovered wrapped phase, respectively, obtained by using the suggested Fourier fringe-normalized analysis scheme. It is evident from Figs. 2.5(f) and 2.5(j) that the zero-order spectrum suppression is satisfactory.

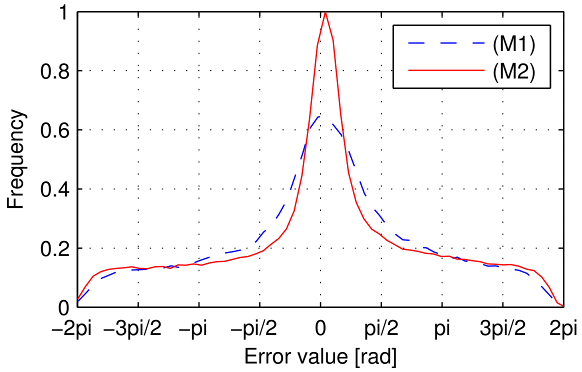


Figure 2.6: *Simulation results. Error's histogram of the standard Fourier transform method (M1), and the Fourier fringe-normalized analysis (M2).*

The direct comparison of the last column of Fig. 2.5 reveals that the normalization approach provides the best result. Moreover, Fig. 2.6 shows the error's histogram where such conclusion is verified as follow. The horizontal-axis specifies the error value in radians and the vertical-axis is the normalized frequency. The curves (M1) and (M2) in Fig. 2.6 correspond to frequency per error value both of the standard Fourier transform method and the suggested scheme, respectively. From this plot we can see that the proposal (M2) have a higher frequency for errors around to zero radians while the standard Fourier transform (M1) has a more uniform error frequency distribution. Clearly the zero-rad error of the standard Fourier transform method (M1) is less frequent because this method have a higher frequency for error around to $\pm\pi/2$ rad.

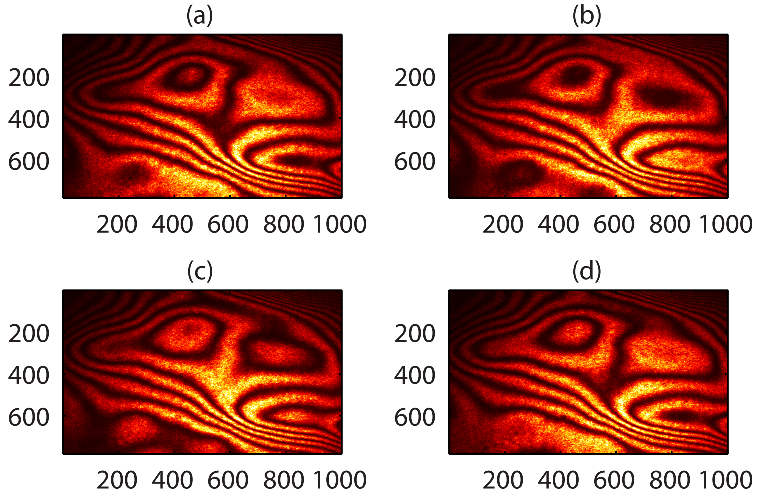


Figure 2.7: *Phase-shifted interferograms used to obtain an experimental baseline wrapped phase distribution.*

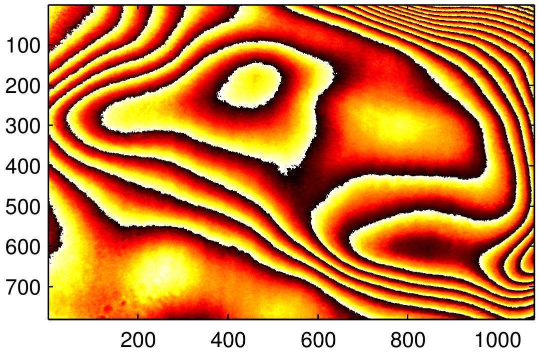


Figure 2.8: *Wrapped phase distribution obtained from the fringe-patterns shown in Fig. 2.7 by the generalized phase-shifting algorithm presented in §3.4.*

2.3.4 | Experimental validation

Twymann-Green Interferometer

The feasibility of the suggested Fourier fringe-normalized scheme was examined by an optical experiment using a Twymann-Green Interferometer illuminated with a He–Ne laser source. A deformed wavefront was generated by inserting a transparent material in the test arm. The fringe-patterns was recorded with a Gray-scale 8-bit CCD camera and a resolution of 780 × 1080 pixels.

A wrapped phase reference was obtained by using the generalized phase-shifting algorithm [71] (it will be presented in the following chapter). This algorithm was employed to process the four phase shifted fringe-patterns shown in Fig. 2.7. The obtained wrapped phase is shown in Fig. 2.8. We have used this phase data as a baseline.

Next, a carried fringe-pattern was obtained by tilting the reference mirror of the interferometer. The captured fringe-pattern is shown in Fig. 2.9(a). As in the computational simulation, the advantages of the

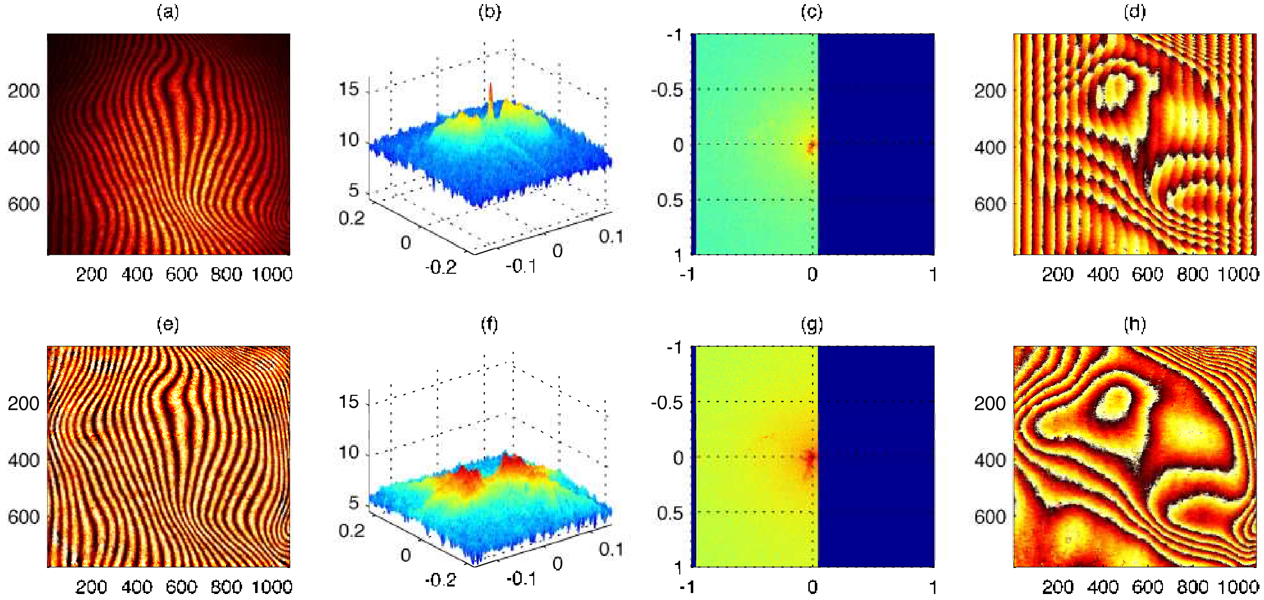


Figure 2.9: *Experimental results. Wrapped phase extraction by using the half-plane filter in the standard Fourier transform (first row) and the Fourier fringe-normalized analysis (second row). (1st column) Fringe-pattern to be processed, (2nd column) Fourier spectrum (logarithm view), (3rd column) filtering, and (4th column) recovered wrapped phase.*

normalization approach are highlighted by comparison between the results obtained using the half-plane filter in both the standard Fourier transform method, Figs. 2.9(a)-(d), and the Fourier fringe-normalized analysis, Figs. 2.9(e)-(h).

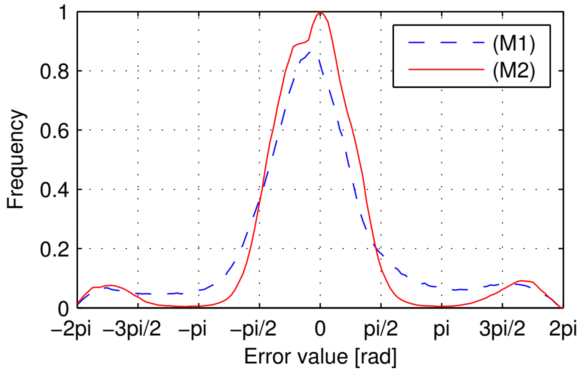


Figure 2.10: *Experimental results. Histogram of error in the standard Fourier transform method (M1), and the Fourier fringe-normalized analysis (M2).*

Similar to the simulation case, the best result is obtained from the normalization approach. This is evident by the direct comparison of the Fig. 2.8 with the last column of Fig. 2.9. The error's histogram shown in Fig. 2.10 verify this claim. For the Fourier fringe-normalized analysis, Fig. 2.10(M2), the frequency per error value around to zero-rad is higher while the error value around to $\pm\pi$ rad is near zero. In contrast, for the standard Fourier transform method, Fig. 2.10(M1), the frequency per error value around to zero-rad is lower and the frequency components around to $\pm\pi$ rad is higher with respect to the normalization approach.

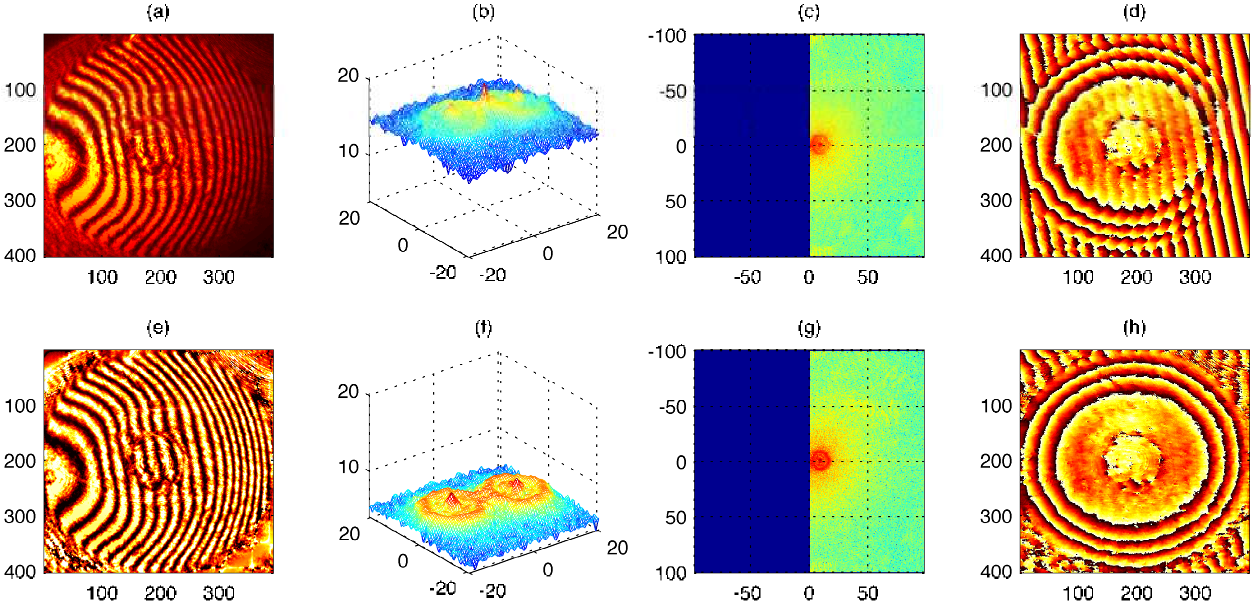


Figure 2.11: *Experimental results. Wrapped phase extraction by using the half-plane filter in the Fourier fringe analysis (1st row), and the Fourier fringe-normalized analysis (2nd row). (1st column) Fringe-pattern to be processed, (2nd column) logarithm view of the Fourier spectrum, (3rd column) filtering, and (4th column) recovered wrapped phase.*

Fizeau Interferometer

Another experiment is performed by evaluating a fringe-pattern obtained from a Fizeau interferometer by testing a concave mirror. A 404 × 392 pixel frame was recorded with a Gray-scale 8-bit CMOS camera. The captured fringe-pattern is shown in Fig. 2.11(a).

Figs. 2.11(a)-(d) show the results obtained when the Fourier fringe analysis by using the half-plane filter is applied to the original fringe-pattern. On the other hand, Figs. 2.11(e)-(h) show the respective results when the fringe-pattern normalization is performed. It can be seen that the normalization procedure is successful to zero-order suppression [compare the Fig. 2.11(b)(f) pair]. Similar to the previous cases, the best result is obtained from the Fourier fringe-normalized approach.

It is worth mentioning that the zero-order suppression justify the fact that the simple half-plane filter is sufficient to carrier out the filtering procedure from a normalized fringe-pattern as shows Fig. 2.11(g).

2.4 | Conclusion

The Fourier fringe-normalized analysis is a more advanced technique which adopt a fringe-pattern normalization stage as a data pre-processing. By the normalization approach, the spectrum leakage is avoided and the filtering procedure is less critical because the zero order spectrum is removed. Thus, the Fourier fringe-normalized analysis provides desirable benefits such as a major accuracy, maximum spatial resolution, increased robustness, faster execution speed, and a simple filtering procedure.

The Fourier fringe-normalized analysis has not been fully exploited because the most of the current fringe-pattern normalization methods are not satisfactory. To overcome this obstacle, the robust and faster fringe-pattern normalization method by parameter estimation, appropriate to real-time applications was suggested.

When the Fourier fringe-normalized analysis is used, the simple half-plane filtering is sufficient to obtain good results. It was tested by processing both synthetic and experimental fringe-patterns. With the obtained results, the benefits of fringe-pattern normalization approach for Fourier fringe analysis are verified.

Chapter 3
Phase demodulation by temporal carrier

I think that it is a relatively good approximation to truth —which is much too complicated to allow anything but approximations— that mathematical ideas originate in empirics.

John Von Neumann

Contents

The phase-shifting methods for wrapped phase extraction provide high precision, maximum spatial resolution (point-wise), low sensitivity to noise, among other good desired properties.

The use of a temporal carrier for wrapped phase extraction originates the so-called *phase-shifting techniques*. This chapter presents a brief review of the most common phase-shifting algorithms. The concepts of *standard*, *extended*, and *generalized phase-shifting* are explained. Also, the *homogeneous* and *inhomogeneous* phase shifts terms are defined.

Two generalized phase-shifting algorithms to address homogeneous as well as inhomogeneous phase shift are devised. Such algorithms can work from only two or more frames with spatio-temporal fringe visibility variation. These algorithms are user-free, computationally efficient, and robust. Thus, they can be implemented in real-time applications. Many figures and some examples are employed to illustrate several concepts and algorithms. The used computer codes, written in MATLAB software, are provided.

3.1 | Introduction

In previous chapter, the Fourier transform method to extract (in wrapped format) the encoded phase distribution was presented. One of the most important features of this approach is the requirement of a single fringe-pattern of the form

$$I(p) = a(p) + b(p)\cos[\phi(p) + \delta(p)],$$

where the so-called *spatial carrier* $\delta(p)$ is defined as

$$\delta(p) = 2\pi f \cdot p$$

with f being the *frequency carrier*. This single fringe-pattern requirement is an important property because dynamical extreme physical phenomena can be measured [25].

However, the frequency transform approach suffers of sign ambiguity (if the carrier frequency f is not high enough) and a reduced bandwidth due to the need of a high carrier frequency and the spectrum filtering. Such drawbacks are overcome by using a temporal carrier. The use of temporal carriers to wrapped phase extraction originates the so-called *phase-shifting* techniques. In this chapter, this topic is tackled.

3.2 | Phase-shifting techniques

Phase-shifting techniques are an important toolbox for extracting the wrapped phase from a set of phase-shifted fringe-patterns [3,72,73]. In general, a phase-shifting algorithm requires a set of $K > 1$ frames of the form

$$I_k(p) = a_k(p) + b_k(p)\cos[\phi(p) + \delta_k(p)], \tag{3.1}$$

for the discrete temporal variable $k = \overline{0, K-1}$, where K is the number of available frames, $I_k(p)$ is the recorded intensity distribution, $p = (x, y)$ is a two-dimensional spatial variable, $a_k(p)$ is the background light, $b_k(p)$ is the modulation light, $\phi(p)$ is the phase function of interest to be recovered, and $\delta_k(p)$ is the temporal carrier (phase shift) function[1].

The phase-shifting techniques are mainly classified by the type of phase shift function involved an the *a priori* knowledge about it.

Phase shift *a priori* knowledge

The choice of a phase-shifting algorithm depends, first, on if the phase shift is known or an unknown.

Phase shift function type

The phase shift functions can be classified in four types formed by considering spatial dependence and linearity as shows Fig. 3.1.

- Spatially non-dependent or *homogeneous*. We say that the phase shift function is *homogeneous* if it has the same value in the whole frame. In other words, a ***homogeneous phase shift*** is represented by temporal function only.

[1] In the literature, the term "phase shift" is commonly used instead of "temporal carrier". The former term will be used along this work in order to avoid the conceptual conflict if both temporal and spatial dependence of $\delta_k(p)$ is considered.

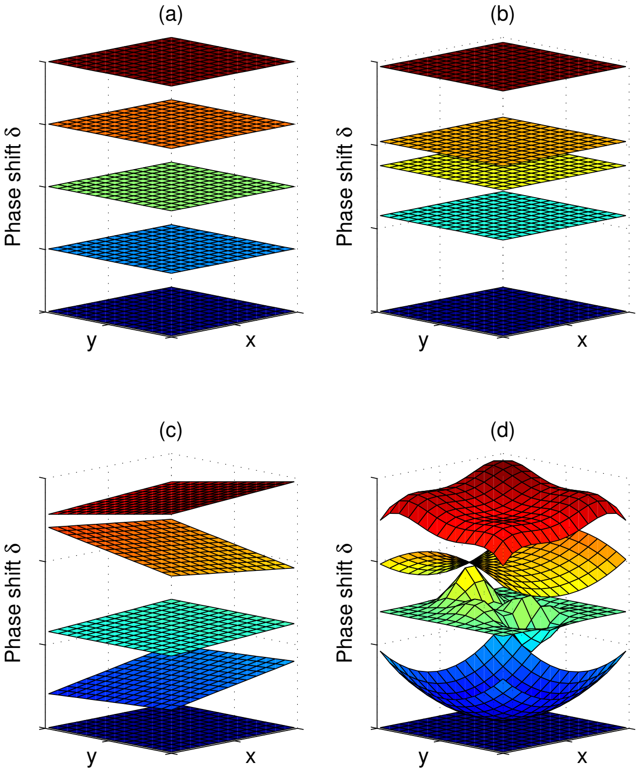


Figure 3.1: *Phase shift function types. Homogeneous phase shift: (a) linear, and (b) nonlinear on k. Inhomogeneous phase shift: (c) linear, and (d) nonlinear on p. For the inhomogeneous case, nonlinearity dependence on k is assumed in advance.*

- Spatially dependent or *inhomogeneous*. An ***inhomogeneous phase shift*** exhibits a phase variation which depend on both time and space.

The above listed groups are further classified on linear and non-linear.

Notice that for the homogeneous case, the linear and non-linear groups are defined with respect to the discrete temporal variable k. On the other hand, for the inhomogeneous case, the linearity criteria should be applied with respect to k and the spatial variable p. However, here it is assumed that the phase shift is nonlinear on k and the linearity criteria is referred to the spatial variable. This classification of phase shift functions is depicted in Fig. 3.1.

The earlier phase-shifting algorithms were proposed for homogeneous phase shift functions[2]. The homogeneous phase shift can be linear or non-linear, and it can be known *a priori* or an unknown. Thus, the phase-shifting techniques are classified in standard, extended, and generalized. This classification is summarized in Table 3.1 and illustrated in Fig. 3.2. Later in this section these algorithms will be described.

[2] Since the phase-shifting concept was devised for phase extraction by using *temporal carriers* (analogous to the frequency transform approach by using spatial carriers), phase-shifting is only applicable for the case of homogeneous phase shift. Thus, strictly speaking, the use of inhomogeneous phase shifts should be attended by a sort of hybrid between the frequency transform and the phase-shifting approaches. However, in this work such a distinction is not performed.

	Homogeneous phase shift type	
Phase-shifting	Dependence on k	Unknown
Standard	Linear	No
Extended	Non-linear	No
Generalized	Non-linear	Yes

Table 3.1: *Classification of phase-shifting techniques depending on the properties of the phase shift function.*

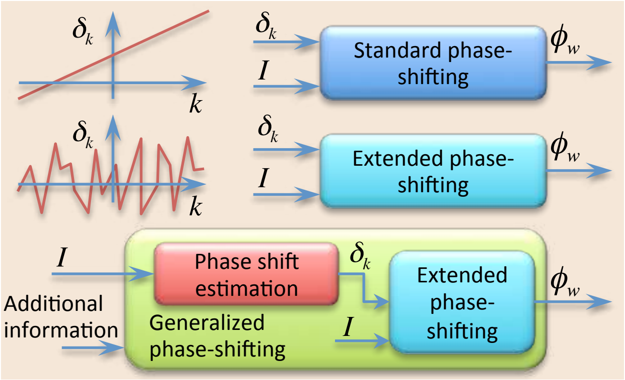


Figure 3.2: *Block diagram of (top) the standard, (middle) extended, and (bottom) generalized phase-shifting methods.*

In the following sections, the concepts of phase shift and phase steps will be frequently used. Therefore, it is convenient to present the definition of these terms.

Definition 1 (Phase step and phase shift). *From the phase-shifted fringe-patterns (3.1), the encoded phase $\phi(p) + \delta_k(p)$ is composed of two parts: the static part $\phi(p)$ and the dynamical one or* **phase shift** *$\delta_k(p)$ as shows Fig. 3.3. Thus, it is convenient to use the time derivative of the encoded phase because static and dynamic parts are separated. This leads to the concept of* **phase step** *$\alpha_k(p)$.*

The phase step is defined as the time derivative of the encoded phase (or the phase shift) as

$$\alpha_k(p) := \frac{d}{dk}[\phi(p) + \delta_k(p)] = \frac{d}{dk}\delta_k(p). \tag{3.2}$$

Hence, the phase shift $\delta_k(p)$ is computed from the phase step by

$$\delta_k(p) = \delta_0 + \int_0^k \alpha_\ell(p)\ d\ell, \tag{3.3}$$

where δ_0 is an initial reference phase or phase offset.

Because in this work the variable k is discrete, the corresponding equations of (3.2) and (3.3) are the following finite differences and cumulative sum:

$$\alpha_k(p) = \delta_k(p) - \delta_{k-1}(p), \tag{3.4}$$

$$\delta_k(p) = \delta_0 + \sum_{\ell=1}^{k} \alpha_\ell(p), \tag{3.5}$$

where $\Delta k = \Delta\ell = 1$ was assumed.

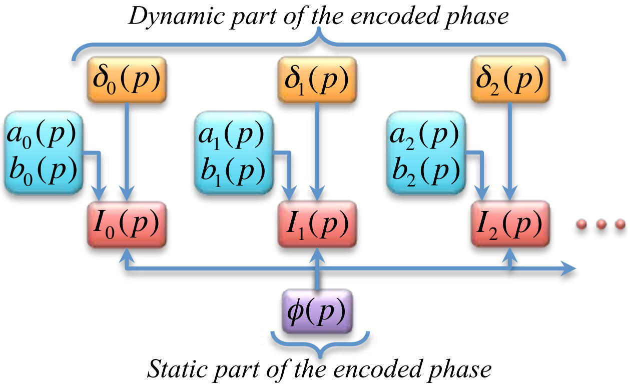


Figure 3.3: *From a set of phase-shifted fringe-patterns modeled by Eq.* (3.1)*, the interest phase distribution* $\phi(p)$ *and the phase shift* $\delta_k(p)$ *can be determined, respectively, as the dynamic and static parts of the encoded phase* $\Phi_k(p) = \phi(p) + \delta_k(p)$*.*

3.2.1 | Standard phase-shifting

The standard phase-shifting approach relies mainly on three assumptions [26]:

1. The background and modulation functions are equal in all fringe-patterns; i.e., they are not dependent on k.
2. The phase shift function is homogeneous and linear on k, Fig. 3.1(a). This is, all phase steps are equals (the phase shift function is a set of non-tilted planes).
3. The phase shift function must be known *a priori*.

Mathematically, the standard phase-shifting approach considers a simplified version of (3.1), namely

$$I_k(p) = a(p) + b(p)\cos[\phi(p) + \delta_k], \tag{3.6}$$

with the phase shift

$$\delta_k = \alpha k + \alpha_0, \tag{3.7}$$

where $\alpha \in (0, 2\pi)$ is the phase step, and α_0 is a phase offset.

The K-step algorithms [26] (also known as k-bucket) are associated with the standard approach. Computationally, this approach is the most efficient. Unfortunately, the requirement of a linear and known phase shift is a serious restriction. To satisfy this requirement, expensive shifter devices are necessary to generate with high accuracy “nominal” phase values (for example 0, $\pi/2$, π and $3\pi/2$ rad for the 4-step method). Moreover, calibration procedures are unavoidable [**?**, **?**, 74].

3.2.2 | K-step phase-shifting methods

The methods where all phase steps are equals to

$$\alpha = 2\pi/K, \tag{3.8}$$

with $K \geq 3$ and $\alpha_0 = 0$, are known as K-step phase-shifting methods[3] [26, 75]. The most common K-step methods are the cases $K = 3$ and $K = 4$.

3-step phase-shifting method

In this method, the acquired fringe patterns (3.6) are

$$\begin{aligned} I_1 &= a + b\cos\phi, \\ I_2 &= a + b\cos(\phi + 2\pi/3) \\ &= a + b\cos\phi\cos\frac{2\pi}{3} - b\sin\phi\sin\frac{2\pi}{3} \\ &= a - \frac{1}{2}b\cos\phi - \frac{\sqrt{3}}{2}b\sin\phi, \\ I_3 &= a + b\cos(\phi + 4\pi/3) \\ &= a + b\cos\phi\cos\frac{4\pi}{3} - b\sin\phi\sin\frac{4\pi}{3} \\ &= a - \frac{1}{2}b\cos\phi + \frac{\sqrt{3}}{2}b\sin\phi. \end{aligned} \tag{3.9}$$

The background light can be removed using the following subtractions:

$$\begin{aligned} \sqrt{3}(I_3 - I_2) &= 3b\sin\phi \\ 2I_1 - I_2 - I_3 &= 3b\cos\phi. \end{aligned} \tag{3.10}$$

Thus, the phase ϕ can be obtained by

$$\tan\phi = \frac{\sqrt{3}(I_3 - I_2)}{2I_1 - I_2 - I_3}. \tag{3.11}$$

On the other hand, the background light a can be obtained from Eqs. (3.9) as

$$a = \frac{1}{3}\left(I_1 + I_2 + I_3\right). \tag{3.12}$$

Similarly, the modulation light b can be obtained from Eqs. (3.10) as

$$b = \frac{1}{3}\sqrt{3(I_3 - I_2)^2 + (2I_1 - I_2 - I_3)^2} \tag{3.13}$$

4-step phase-shifting method

In this method, the acquired fringe patterns (3.6) are

$$\begin{aligned} I_1 &= a + b\cos\phi, \\ I_2 &= a + b\cos\left(\phi + \frac{1}{2}\pi\right) = a - b\sin\phi, \\ I_3 &= a + b\cos\left(\phi + \quad \pi\right) = a - b\cos\phi, \\ I_4 &= a + b\cos\left(\phi + \frac{3}{2}\pi\right) = a + b\sin\phi, \end{aligned} \tag{3.14}$$

[3] Notice that this particular phase shift is uniformly distributed in the interval $[0, 2\pi)$. This is a key property for the efficiency of this methods. This uniformity is required in other fringe-pattern processing procedures such as the intensity normalization by the max–min method [?].

where the dependence on p was removed from $I(p)$, $a(p)$, $b(p)$, and $\phi(p)$ for brevity. The background light can be removed using the following subtractions:

$$\begin{aligned} I_4 - I_2 &= 2b\sin\phi, \\ I_1 - I_3 &= 2b\cos\phi. \end{aligned} \tag{3.15}$$

Thus, the phase distribution ϕ can be obtained by

$$\tan\phi = \frac{I_4 - I_2}{I_1 - I_3}. \tag{3.16}$$

The background light a can be obtained easily from Eqs. (3.14) as

$$a = \frac{1}{4}\left(I_1 + I_2 + I_3 + I_4\right). \tag{3.17}$$

Similarly, the modulation light can be obtained from Eqs. (3.15) as

$$b = \frac{1}{2}\sqrt{(I_4 - I_2)^2 + (I_1 - I_3)^2}. \tag{3.18}$$

The *extended phase-shifting technique* is preferred because it relieves the experimental difficulties by the possibility of handle arbitrary phase shift functions as will be seen below.

3.2.3 | Extended phase-shifting

In general, a linear phase shift is not guaranteed due to many experimental difficulties [76, 77]. Moreover, some experimental setups generates non-linear phase shifts such as the quadratic phase shift induced by lateral displacement of the light source [23].

In the early 80's, were applied techniques such as the generalized data reduction and the least-squares method for wrapped phase extraction [78, 79]. Such methods relieves the constraint of linearity of the phase shift δ imposed by the standard phase-shifting. These techniques does not requires a particular phase shift. Therefore, the difficult (impossible) task of generating a particular phase shift by the optical setup is diminished. Instead, it is necessary to determine somehow the phase shift which the experimental setup has generated.

For the extended phase-shifting approach, the fringe-patterns are modelled by (3.6), but the more general phase shift

$$\delta_k = \delta(k) \text{ with } \alpha_k \in [-\pi, \pi),\ k = \overline{1, K-1}, \tag{3.19}$$

is supported. The extended phase-shifting approach exploits the fact that the set of K frames (3.6) can be rewrite as the linear equations system:

$$\begin{bmatrix} I_0(p) \\ \vdots \\ I_{N-1}(p) \end{bmatrix} = \mathbb{A} \begin{bmatrix} a(p) \\ \xi(p) \\ \zeta(p) \end{bmatrix}, \tag{3.20}$$

where $\xi(p) = b(p)\cos\phi(p)$, $\zeta(p) = b(p)\sin\phi(p)$, and the system's matrix $\mathbb{A}$ is given by

$$\mathbb{A} = \begin{bmatrix} 1 & \cos\delta_0 & -\sin\delta_0 \\ 1 & \cos\delta_1 & -\sin\delta_1 \\ \vdots & \vdots & \vdots \\ 1 & \cos\delta_{N-1} & -\sin\delta_{N-1} \end{bmatrix}. \tag{3.21}$$

It is possible to solve the equations system (3.20) for a, ξ, and ζ. For this, the criteria most widely used is the least-squares [79]. By least-squares inversion of $\mathbb{A}$, the quantities a, ξ, and ζ are obtained. Then, the phase distribution of interest $\phi(p)$ (in wrapped format denoted by $\phi_w(p)$) is computed by the relation

$$\tan\phi(p) = \zeta(p)/\xi(p). \tag{3.22}$$

The example 6 illustrates this method.

Example 6. *Let $a(p)$, $b(p)$, and $\phi(p)$ be the functions defined in (2.10). Let the phase shift be*

$$\delta_k = 0,\ \ \pi/4,\ \ -5\pi/4,$$

for $k = \overline{0,2}$, respectively, as shows Fig. 3.4(a). The resulting phase-shifted fringe-patterns are shown in Fig. 3.4(b)-(d). In this case, the system's matrix (3.21) is

$$\mathbb{A} = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 1/\sqrt{2} & -1/\sqrt{2} \\ 1 & -1/\sqrt{2} & -1/\sqrt{2} \end{bmatrix}.$$

The equation (3.20) is solved as

$$\begin{bmatrix} a(p) \\ \xi(p) \\ \zeta(p) \end{bmatrix} = A^{\dagger} \begin{bmatrix} I_0(p) \\ I_1(p) \\ I_2(p) \end{bmatrix},$$

where, since $\mathbb{A}$ is a square non-singular matrix, the least-squares inverse $\mathbb{A}^{\dagger}$ is equal to the usual inverse matrix $\mathbb{A}^{-1}$:

$$\mathbb{A}^{\dagger} = \mathbb{A}^{-1} = \begin{bmatrix} 1 & -1/\sqrt{2} & 1/\sqrt{2} \\ 0 & 1/\sqrt{2} & -1/\sqrt{2} \\ \sqrt{2} & -1-\frac{1}{\sqrt{2}} & 1-\frac{1}{\sqrt{2}} \end{bmatrix}.$$

Therefore,

$$\begin{aligned} a(p) &= I(p,0) + [I(p,2) - I(p,1)]/\sqrt{2}, \\ \xi(p) &= [I(p,1) - I(p,2)]/\sqrt{2}, \\ \zeta(p) &= \sqrt{2}I(p,0) - I(p,1) + I(p,2) \\ &\quad -[I(p,1) + I(p,2)]/\sqrt{2}. \end{aligned}$$

The obtained functions $a(p)$, $\xi(p)$, and $\zeta(p)$ are shown in Fig. 3.4(e)-(g), respectively. Finally, Figs. Fig. 3.4(h) show the wrapped phase computed by

$$\phi_w(p) = \arctan[\zeta(p)/\xi(p)].$$

Notice that this result is equal with the exact wrapped phase shown in Fig. 2.3(d).

It is worth mentioning that, although a nonlinear phase shift is supported[4], it must be known *a priori*. There are methods to determine the phase shift by using additional information. The phase-shifting techniques which includes a dedicated stage to compute the phase shift from the provided fringe-patterns are known as *generalized phase-shifting techniques*.

[4] In general, for the extended phase-shifting technique, the phase shift can be arbitrary always that $\mathbb{A}$ in (3.20) is of maximum range.

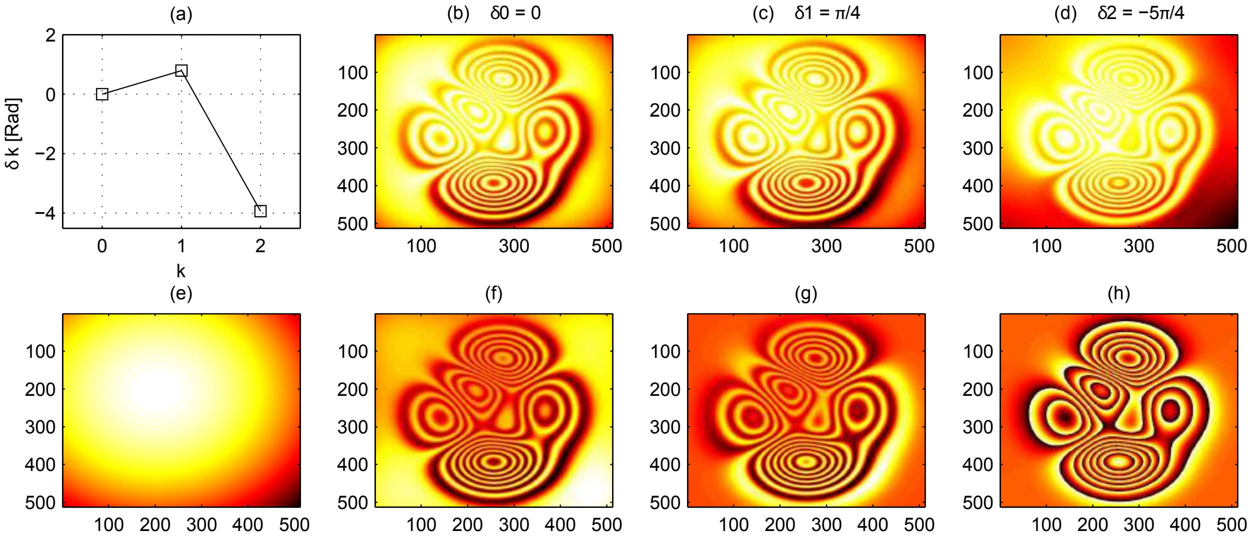


Figure 3.4: *Simulation of the extended phase-shifting method. (a) Discrete phase shift ($k = 0, 1, 2$). An interpolated curve is presented for visualization purposes. (b)-(d) Simulated fringe-patterns and its respective phase shift. (e)-(g) Computed functions a, ξ, and ζ. (h) Recovered wrapped phase.*

3.2.4 | Generalized phase-shifting

In both standard and extended phase-shifting approaches, the phase shift must be known *a priori*. For this, the phase shifter devices such as piezoelectric materials must be calibrated. However, it is difficult to determine the phase shift even from well calibrated equipment because there are many other experimental error sources [76, 77].

To overcome the above referred problems, the *generalized phase-shifting interferometry* (GPSI) was proposed [80]. The GPSI schemes include a dedicated stage for the phase shift estimation. Thus, the need of to calibrate the equipment is eliminated. Moreover, the resulting algorithms are robust against several experimental error sources. Accordingly, the high quality requirements for the phase shifter devices are so demanding; i.e., the expensive and complex phase shifters can be substituted by other more simple and cheap ones [23].

Error sources in phase-shifting

Any phase-shifting method is influenced by many error sources. They arise by instability of intensity and/or frequency of the illumination source, multiple reflected beams, irregular reflectance/transmittance of the object, miscalibration of the phase shifters, mechanical vibrations, turbulences and temperature gradients in the environment, nonlinearities[a] of the actuators (e.g., piezoelectric materials) and detectors (for instance, the gamma curve), quantization, and so on [76, 77].

[a] Some of the most common nonlinearities are the bias (linear, quadratic, cubic, etc.), hysteresis, and saturation.

The generalized phase-shifting approach does not explicitly require knowing the phase shift function. This approach includes a stage dedicated to the estimation of the phase shift. The Carrè algorithms may be the first phase-shifting techniques in this category [81, 82]. However, they are restricted to the case when the phase shift function is homogeneous and linear on k. In other words, the phase shift must be non-tilted planes separated one from another by a constant distance as shown in Fig. 3.1(a).

Example 7 (Carrè's methods)**.** *The Carrè's methods [81], before of compute the wrapped phase distribution, compute the unknown phase step $\alpha \in (0, \pi)$. This makes these methods are self-adjustable to the (linear) phase shift. This is the most important property of these methods in experimental applications.*

Another case is when the phase shift is homogeneous but is a nonlinear function of k as shown in Fig. 3.1(b). The most generalized phase-shifting algorithms fall into this group [46, 71, 80, 83–86].

In some optical setups, it is difficult to keep a homogeneous phase shift. For example, the translation of the reference mirror of an interferometer cannot be exactly perpendicular to the optical axis [87–89]. For this case, the phase shift function consists of a set of tilted planes as shown in Fig. 3.1(c). Interesting schemes to solve this problem have been developed [90–93].

Due to the development of advanced technologies (e.g. liquid crystal spatial light modulators) and versatile experimental designs [20, 23], the requirements of phase-shifting algorithms have increased. Particularly, it is necessary to process fringe-patterns with unknown inhomogeneous phase shifts that depend on both p and k in a nonlinear way as shown in Fig. 3.1(d) [94]. Additionally, other desirable properties for phase-shifting techniques are: a reduced number of necessary fringe-patterns, and the capacity to handle spatio-temporal visibility (i.e. the background and modulation lights are functions of both p and k) [95].

At this point, an apparent problem arises. If an arbitrary phase shift (changing spatially and temporally in an unknown manner) is assumed, it introduces an extra fringe-pattern. The extra fringe-pattern can cause confusion regarding what part of the encoded phase belongs to the interest phase distribution and what part belongs to the phase shift. Moreover, conceptually, the phase shift is a necessary "reference phase" useful for wrapped phase extraction. Therefore, as an arbitrary phase shift is no longer a "reference", the phase extraction in the mentioned conditions could be out of the phase-shifting framework.

In order to address this situation, the phase-shifting concept is stated as follows. In phase-shifting, the phase distribution of interest $\phi(p)$ is equal in all fringe-patterns, and the phase shift $\delta_k(p)$ varies from one to another. In other words, $\phi(p)$ and $\delta_k(p)$ are, respectively, the static and dynamic parts of the encoded phase as can be seen in Fig. 3.3.

Indeed, for the continuous case, by applying the derivative of the encoded phase with respect to k, the interest phase distribution is suppressed and the remainder is the phase step $\alpha_k(p)$ defined by (3.2). Then, the phase shift is reconstructed by (3.3). Finally, the interest phase distribution $\phi(p)$ is recovered by using the computed phase shift $\delta_k(p)$. So the functions $\phi(p)$ and $\delta_k(p)$ are clearly distinguished from one another (even when $\delta_k(p)$ is nonlinear unknown function of both p and k) and the phase computing is possible in the context of phase-shifting[5]. This explanation is valid for the discrete case.

Since early 1990, many researches have focused their effort to improve the GPSI scheme. The current GPSI advances can be classified by the approach in: Fourier transform, elliptical curve fitting, statistical, spatio-temporal, iterative, and optimization [46, 85]. A briefly description is given below.

Fourier transform. Perhaps, Lai and Yatagai [80] were the first to use an spatial carrier for to determine the temporal one in a generalized phase-shifting experiment. This method consists on the generation of Fizeau fringes by using additional optical setup. The phase shift is obtained by analyzing the Fizeau fringes by the Fourier transform method [27] (see chapter 2). With the obtained phase shift, the extended phase-shifting problem is solved.

Elliptical curve fitting. This geometrical approach consists on the evaluation of the Lissajous figures generated from the intensity values at two or more points of the given fringe-patterns [83, 96]. There are

[5] The wrapped phase extraction is possible if $\alpha_k(p) \neq 0$. This is a basic assumption in the phase-shifting approach.

several versions of this approach. An example consist on calibrate the phase steps from the resulting intensity ellipses. A benefit of this technique is the possibility of to use a few points of the fringe-patterns. However, this lead to a low robustness (i.e., high sensitivity to noise). A robust versions of this method is by using a forced fit method [97, 98]. However, such a method is limited to process interferograms with a constant visibility.

Statistical. Several statistical algorithms have been proposed. They determine the actual phase steps from measured intensities with a statistical-based algorithm [99]. This approach exploits the statistical nature of the phase distribution to obtain the phase steps in a very precise and stable way [100]. However, its applications is limited to speckle phase such as holography and phase-retrieval.

Spatio-temporal. In this category is the histogram-based algorithm where the phase difference between two adjacent frames is analyzed [101]. From the resulting histogram, the phase shift is accurately extracted.

Another alternative is the max-min algorithm [84]. This algorithms performs a comparison of each pixel in an interferogram with the pixels at the corresponding location of acquired interferograms. This removes the sensitivity to spatial intensity variations. Then, a comparison of each pixel intensity with the intensities of two reference pixels within the same interferogram es performed. This approach does not requires initial guess about the phase distribution (in contrast to the statistical approach) or additional information (such as the complementary Fizeau interferograms in the Fourier transform approach). However, the spatio-temporal algorithms are computationally exhaustive because they require a large number of data frames ($N \geq 15$) for reliable operation.

Iterative. In this category is the iterative least-squares technique to extract the phase steps [102]. The resulting algorithm is immune to nonlinearities and errors due to uncalibrated piezoelectric devices. However, this algorithm finds only local optimums. Therefore, a good initial guess must be provided.

The above referred drawbacks are overcome by the advanced iterative algorithm reported in [86]. This algorithm computes the solution by iterating between temporal and spatial domains. Fig. 3.5 shows a block diagram illustrating this algorithm.

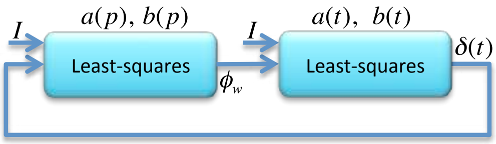


Figure 3.5: *Block diagram of the advanced iterative algorithm for generalized phase-shifting interferometry [86].*

In summary, the Fourier transform approach reported in [80] requires additional optical setup to generate Fizeau fringes. The elliptical curve fitting requires a forced fit method to cope with noisy frames. The statistical algorithms are very accurate but the computational load is high and a large number of frames is required. For the iterative approach, an initial guess is necessary and only local convergence is guaranteed. On the other hand, the noniterative optimization are robust, faster, and, by appropriate constraints, converges to the desired result.

In the following, a noniterative optimization algorithm for generalized phase-shifting is presented. It uses the least-squares method to estimate the background light, the modulation light, the phase step, and the wrapped phase distribution.

3.3 | Least-squares parameter estimation for generalized phase-shifting

In chapter 1 the principle of to use the spatial information about the parameters involved in the fringe-patterns was exploited for fringe-pattern normalization. Now, such a principle is used here for generalized phase-shifting.

The parameter estimation principle for generalized phase-shifting allows us to design efficient, fast, robust, and user-free algorithms able to real-time applications. Other important advantages are the natural structure to process fringe-patterns with spatio-temporal fringe visibility variation and both homogeneous and inhomogeneous phase shift.

In this section two generalized phase-shifting algorithms are presented. The first one is very simple because it was designed to process homogeneous phase shift. The second one is more advanced because homogeneous as well as inhomogeneous phase shifts can be addressed. Both algorithms can process frames with spatio-temporal fringe visibility because the normalization stage presented in 1.5 was used.

3.4 | Generalized phase-shifting algorithm for homogeneous phase shift

Unlike other generalized phase-shifting algorithms, the parameter estimation scheme can process fringe-patterns of the form

$$I_k(p) = a_k(p) + b_k(p)\cos[\phi(p) + \delta_k]. \tag{3.23}$$

for $k = \overline{0, K-1}$ with K being the number of given frames. Notice that the background and modulation lights, $a_k(p)$ and $b_k(p)$ respectively, are functions of both space p and time k. Also, it is worth mentioning that an homogeneous phase shift δ_k is assumed. However, more later in this chapter, this parameter estimation scheme is developed to cope with inhomogeneous phase shifts $\delta_k(p)$ (a function of both space p and time k).

From the phase-shifted fringe patterns (3.23), the following four parameters can be distinguished:

- Background light $a_k(p)$,
- Modulation light $b_k(p)$,
- Phase shift δ_k, and
- Phase distribution of interest $\phi(p)$.

The extraction of the phase distribution (in wrapped format) from a set of K given by (3.23) is, in general, a nonlinear problem. By exploiting spatial information about the parameters $a_k(p)$ and $b_k(p)$, it is possible to divide the original nonlinear problem in a series of four linear problems in cascade [71]. Specifically, the generalized phase-shifting algorithm by parameter estimation consists of three stages:

1. Fringe-pattern normalization. The background and modulation lights are estimated by the least-squares polynomial fitting method.
2. Phase shift estimation. The least-squares method is employed to recover the phase steps and then the phase shift parameter is computed.
3. Wrapped phase extraction. With the computed phase shift function, the wrapped phase distribution is obtained by using the extended phase-shifting approach.

This is the main idea of wrapped phase extraction by parameter estimation that will be described here.

In the following, the algorithm's principles are explicated. Then, the feasibility of the algorithm is illustrated by a numerical simulation. Finally, experimental fringe-patterns are processed to show the usefulness of the proposal. The phase shift estimation quality is verified by comparison with the conventional ellipse-fitting [96] and the Fourier transform [32] methods.

3.4.1 | Theoretical principles

Fringe pattern normalization

The so-called ***fringe patter normalization*** is the procedure of suppression from the given fringe-patterns both the background and modulation lights. For this, we use the parameter estimation method. It solves two linear problems in cascade to obtain the normalized version $\bar{I}_k(p)$ of the frames $I_k(p)$ given by (3.23) as

$$\bar{I}_k(p) = \cos[\phi(p) + \delta_k], \qquad k = \overline{0, K-1}. \tag{3.24}$$

For details on this fringe-patterns normalization method, the reader is referred to the chapter 1 or Ref. [71].

Phase shift estimation

The phase shift is obtained by computing the corresponding phase steps. Then, from these phase steps, the phase shift is reconstructed by the equation (3.5).

There are direct methods to compute the phase steps for the case when normalized fringe-patterns are available. For example, the average of the absolute value of the phase difference or the histogram phase difference [101]. These methods obtain the phase steps for each point in the frame and, by applying some criteria (e.g. average and statistical mode), a single phase step is deduced. However, such methods are sensitive to random noise and can return multiple solutions.

From the normalized fringe-patterns given by (3.24), to estimate the phase step α_ℓ from two consecutive normalized fringe-patterns $\bar{I}_{\ell-1}$ and $\bar{I}_\ell$:

$$\begin{aligned} \bar{I}_{\ell-1} &= \cos[\phi + \delta_{\ell-1}], \\ \bar{I}_\ell &= \cos[\phi + \delta_{\ell-1} + \alpha_\ell], \end{aligned} \tag{3.25}$$

with $\ell = \overline{1, N-1}$, we perform the addition and subtraction:

$$\begin{aligned} \bar{I}_{\ell-1} + \bar{I}_\ell &= 2\cos\left(\phi + \delta_{\ell-1} + \frac{\alpha_\ell}{2}\right)\cos\frac{\alpha_\ell}{2}, \\ \bar{I}_{\ell-1} - \bar{I}_\ell &= 2\sin\left(\phi + \delta_{\ell-1} + \frac{\alpha_\ell}{2}\right)\sin\frac{\alpha_\ell}{2}. \end{aligned}$$

Then, by using the trigonometric identity $\sin^2(\phi + \delta_{\ell-1} + \alpha_\ell/2) + \cos^2(\phi + \delta_{\ell-1} + \alpha_\ell/2) = 1$, we have

$$\begin{gathered} \left(\frac{\bar{I}_{\ell-1} - \bar{I}_\ell}{2\sin\frac{\alpha_\ell}{2}}\right)^2 + \left(\frac{\bar{I}_{\ell-1} + \bar{I}_\ell}{2\cos\frac{\alpha_\ell}{2}}\right)^2 = 1 \\ (\bar{I}_{\ell-1} - \bar{I}_\ell)^2 + (\bar{I}_{\ell-1} + \bar{I}_\ell)^2\tan^2\frac{\alpha_\ell}{2} = 4\sin^2\frac{\alpha_\ell}{2}. \end{gathered}$$

Now, by using the identities $2\sin^2 x = 1 - \cos(2x)$ and $\tan^2 = [1 - \cos(2x)]/[1 + \cos(2x)]$, we have

$$(\bar{I}_{\ell-1} - \bar{I}_\ell)^2 + (\bar{I}_{\ell-1} + \bar{I}_\ell)^2\frac{1 - \cos\alpha_\ell}{1 + \cos\alpha_\ell} = 2(1 - \cos\alpha_\ell),$$

or

$$(\bar{I}_{\ell-1} - \bar{I}_\ell)^2(1 + \cos\alpha_\ell) + (\bar{I}_{\ell-1} + \bar{I}_\ell)^2(1 - \cos\alpha_\ell) = 2(1 - \cos^2\alpha_\ell). \tag{3.26}$$

Expanding the above equation,

$$(\bar{I}_{\ell-1} - \bar{I}_\ell)^2 + (\bar{I}_{\ell-1} + \bar{I}_\ell)^2 + \left[(\bar{I}_{\ell-1} - \bar{I}_\ell)^2 - (\bar{I}_{\ell-1} + \bar{I}_\ell)^2\right]\cos\alpha_\ell = 2(1 - \cos^2\alpha_\ell),$$

and simplifying this last expression, we can reach

$$\begin{aligned} 2(\bar{I}_{\ell-1}^2 + \bar{I}_\ell^2) - 4\bar{I}_{\ell-1}\bar{I}_\ell \cos\alpha_\ell &= 2(1 - \cos^2\alpha_\ell) \\ (\bar{I}_{\ell-1}^2 + \bar{I}_\ell^2) - 2\bar{I}_{\ell-1}\bar{I}_\ell \cos\alpha_\ell &= \sin^2\alpha_\ell, \end{aligned}$$

where for the last equality was applied the identity $\cos^2\alpha_\ell = 1 - \sin^2\alpha_\ell$. By simplicity, we restated the last equation as

$$A\beta_\ell + B_\ell\gamma_\ell = C_\ell, \tag{3.27}$$

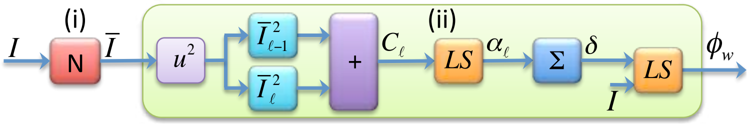
where $\beta_\ell = \sin^2\alpha_\ell$, $\gamma_\ell = \cos\alpha_\ell$, A is an all-ones matrix, $B_\ell = 2\bar{I}_{\ell-1}\bar{I}_\ell$, and $C_\ell = \bar{I}_{\ell-1}^2 + \bar{I}_\ell^2$, where $[\cdot]^2$ is an element-by-element square. Next, the system of linear equations (3.27) is solved for β_ℓ and γ_ℓ by using the least-squares method as

$$\begin{bmatrix}\beta_\ell \\ \gamma_\ell\end{bmatrix} = \mathbb{A}_\alpha^\dagger C_\ell, \tag{3.28}$$

where $\mathbb{A}_\alpha = [A \;\; B_\ell]$. The phase step is obtained by

$$\alpha_\ell = \arctan\frac{\sqrt{\beta_\ell}}{\gamma_\ell}. \tag{3.29}$$

Once the phase step function was computed, the phase shift δ_k is obtained by Eq. (3.5).



(a) Block diagram. (N) denotes the normalization stage depicted in Fig. 1.5.

```
function[Wp,alp,d,bI,a,b,x,y] = GPSbpe(I,deg,mode)
%Generalized phase-shifting algorithm for
%homogeneous phase shift by parameter estimation.
%
%This code is an implementation of the work
%reported in [Optics and Lasers in Engineering,
%Vol. 51(5), pp. 626 - 632 (2013). DOI:
%http://dx.doi.org/10.1016/j.optlaseng.2012.12.020
%by Rigoberto Juarez-Salazar, et. al.

    [bI,a,b,x,y] = FPNorm(I,deg,mode);
    [Wp,alp,d] = WPhaseExt(bI);
end
```

```
function [Wp,alp,d] = WPhaseExt(bI)
    MNK = size(bI);

    alp = zeros(MNK(3),1);
    A = ones(MNK(1:2));
    for k=2:MNK(3)
        B = 2*bI(:,:,k-1).*bI(:,:,k);
        C = bI(:,:,k-1).^2 + bI(:,:,k).^2;

        Th = [A(:) B(:)]\C(:);
        alp(k) = atan2(sqrt(abs(Th(1))),Th(2));
    end

    % Obtaining function delta
    d  = cumsum(alp);

    % Obtaining the wrapped phase
    AA = [cos(d) -sin(d)];
    Th = AA\reshape(bI(:),prod(MNK(1:2)),MNK(3))';
    Wp = atan2(Th(2,:),Th(1,:));
    Wp = reshape(Wp,MNK(1:2));
end
```

(b) Implementation in MATLAB.

Figure 3.6: *Generalized phase-shifting algorithm for homogeneous phase shift by the parameter estimation approach. Notice that the computer function* `FPNorm()`*, presented in chapter 1, is required.*

Wrapped phase extraction

Now, by using the trigonometric identity $\cos(x+y) = \cos x \cos y - \sin x \sin y$, the normalized fringe-patterns (3.24) are restated in the form

$$\begin{aligned} \bar{I}_k(p) &= \cos\phi(p)\cos\delta_k - \sin\phi(p)\sin\delta_k \\ &= \xi(p)\cos\delta_k - \zeta(p)\sin\delta_k. \end{aligned} \tag{3.30}$$

where $\xi(p) = \cos\phi(p)$ and $\zeta(p) = \sin\phi(p)$. Notice that Eq. (3.30) is linear with respect to the unknowns $\xi(p)$ and $\zeta(p)$. Thus, the system of linear equations (3.30) can be represented in matrix notation as

$$\begin{bmatrix} \bar{I}_0 \\ \vdots \\ \bar{I}_{K-1} \end{bmatrix} = \mathbb{A}_\phi^\dagger \begin{bmatrix} \xi \\ \zeta \end{bmatrix}, \tag{3.31}$$

with

$$\mathbb{A}_\phi = \begin{bmatrix} \cos\delta_0 & \sin\delta_0 \\ \vdots & \vdots \\ \cos\delta_{K-1} & \sin\delta_{K-1} \end{bmatrix}.$$

The matrix equation (3.31) is solved for ξ and ζ by using the least-squares method [78, 79] as

$$\begin{bmatrix} \xi \\ \zeta \end{bmatrix} = \mathbb{A}_\phi^\dagger \begin{bmatrix} \bar{I}_0 \\ \vdots \\ \bar{I}_{K-1} \end{bmatrix}. \tag{3.32}$$

Finally, the phase distribution ϕ is obtained (in wrapped format) by solving the equation

$$\tan\phi = -\zeta/\xi. \tag{3.33}$$

Fig. 3.6(a) summaries the above described generalized phase-shifting algorithm. Fig. 3.6(b) shows the implementation in MATLAB software.

3.4.2 | Algorithm testing

The functionality of this algorithm is tested by a computer simulation (example 8) and by processing experimental fringe-patterns (example 9).

In both the simulation and experimental cases, the parameter estimation approach obtains good results when two or more noisy fringe-patterns are processed.

The presented results were obtained by using a 2.8 GHz laptop. The computer time required to process the frames used in examples 8 and 9 (frame sizes of 500 × 500 and 600 × 800 pixels, respectively) is shown in table 3.2. In this table another experiment (40 frames of size of 480 × 440 pixels) was added with the aim to show the performance of the algorithm to process many frames.

From table 3.2 it is verified that the generalized phase-shifting algorithm by using the parameter estimation approach is faster than other similar generalized phase-shifting algorithms.

	Number of interferograms			
Size (pixels)	Two	Three	Four	Forty
500 × 500	0.36	0.41	0.46	–
600 × 800	0.65	0.76	0.84	–
480 × 440	0.31	0.36	0.39	2.21

Table 3.2: *Computation time (seconds).*

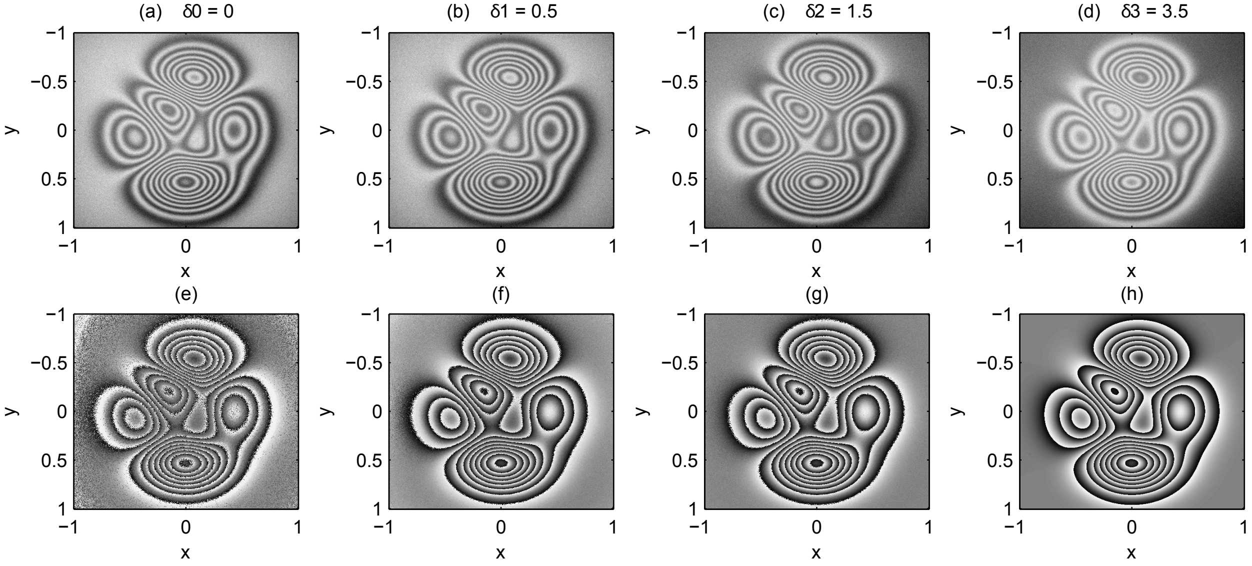


Figure 3.7: *(a)-(d) Simulated fringe-patterns I_0 to I_3 and the respective phase shift δ. Recovered wrapped phase by processing the first two (e), three (f), and four (g) fringe-patterns. (h) Exact wrapped phase.*

Example 8. *Let $a(p)$, $b(p)$, and $\phi(p)$ be the functions defined in (2.10). Let*

$$\delta(k) = 0,\ 0.5,\ 1.5,\ 3.5,$$

for $k = 0, 1, 2, 3$ respectively, be the phase shift. Figs. 3.7(a)-(d) show four noisy fringe-patterns generated by

$$I_k = a + b\cos[\phi + \delta_k] + \eta(p,k), \quad k = \overline{0,3},$$

where $\eta(p,k)$ is a random zero mean Gaussian noise with standard deviation $\sigma(5\%)$. Figs. 3.7(e)-(g) show the recovered wrapped phase by processing the first two, three, and four fringe-patterns I_0 to I_3, respectively. For comparison purposes, Fig. 3.7(h) shows the exact wrapped phase.

The quality of the estimated phase shift is analyzed by comparing the result obtained from the ellipse fitting method [96], and the Fourier transform [32]. The respective results and absolute errors are shown in Fig. 3.8. In this case, the phase shift from the Fourier method is slightly skewed. Both the ellipse fitting method and the presented parameter estimation one present an absolute error of less than 0.1 rad.

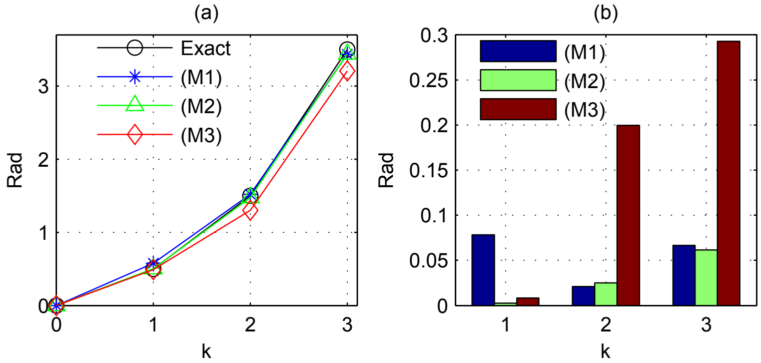


Figure 3.8: *Phase shift recovered by using the parameter estimation (M1), the ellipse fitting (M2), and the Fourier transform (M3) methods. (a) Exact and estimated phase shifts. (b) Absolute error.*

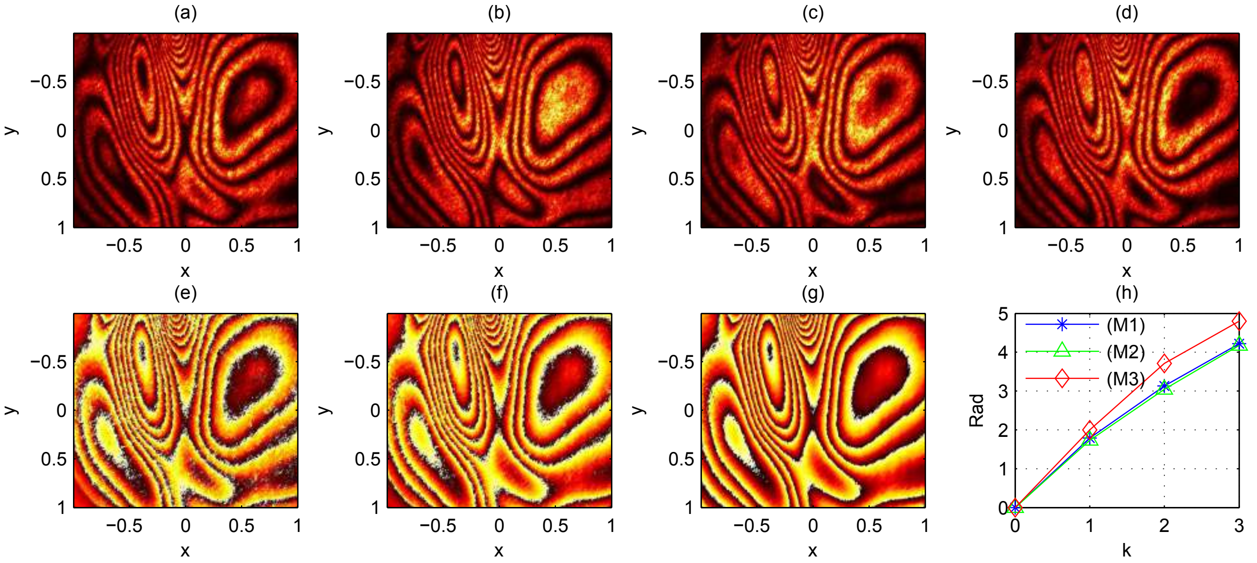


Figure 3.9: *(a)-(d) Four experimental phase-shifted fringe-patterns I_0 to I_3. Recovered wrapped phase by processing the first two (e), three (f), and four (g) fringe-patterns. (h) Estimated phase shift from the parameter estimation (M1), the ellipse fitting (M2), and the Fourier transform (M3) methods.*

Example 9 (Experimental validation). *A commercial PASCO Michelson interferometer, Fig. 3.10, illuminated with a collimated He–Ne (wavelength $\lambda = 632.8$ nm) laser source is used. A deformed wavefront was obtained by inserting a transparent material in the test arm of the interferometer. The phase shift is induced by using the precision micrometric knob to translate the movable mirror. Figs. Fig. 3.9(a)-(d) show four phase-shifted interferograms recorded with progressive phase steps but no attempt was made to make either calibrated or uniform.*

The frames were recorded by a gray scale 8-bits CMOS camera with resolution of 600×800 pixels. The spatial domain $x, y \in [-1, 1]$ was used. Figs. 3.9(e)-(f) show the recovered wrapped phase by using the parameter estimation approach from the first two, three, and four frames. Fig. 3.9(h) shows the obtained phase shift from the parameter estimation (M1), the ellipse fitting (M2), and the Fourier transform (M3) methods. In this case, the ellipse fitting and the parameter estimation methods reach an almost identical phase shift while the Fourier transform method is slightly deviated.

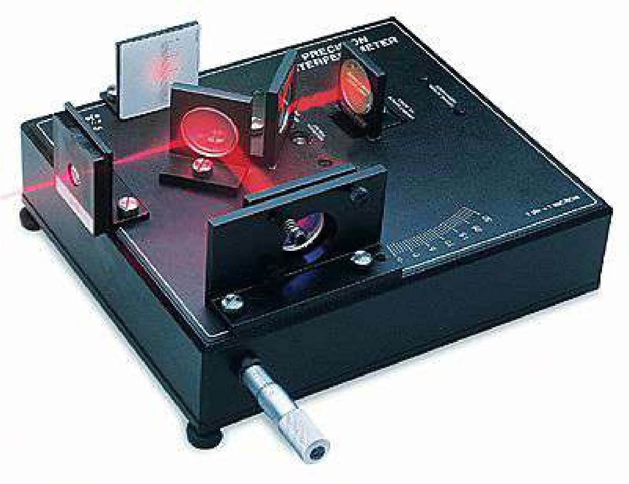


Figure 3.10: *PASCO scientific precision Michelson interferometer (Model OS-9255A).*

3.4.3 | Discussion

There are some important issues that deserve discussion.

Illumination profile. This algorithm exploit the information about the illumination profile. In general, the interfering beams have a Gaussian intensity profile, but the parameter estimation approach approximates such a profile by the respective truncated Taylor expansion (a finite-degree polynomial). In the examples a second-degree polynomial was employed by simplicity. However a higher order polynomial, even more general polynomials (e.g. splines), can be used.

Number of fringes across the frame. The least-squares fitting is successful to cope data corrupted with zero mean symmetrically distributed random noise. The presented algorithm obtains the parameters a and b by a least-squares fitting to $I_k(p)$ and $[I_k(p) - \hat{a}_k(p)]^2$ respectively. En each case, the cosinusoidal terms ($b\cos[\phi + \delta]$ and $b^2\cos[2\phi + 2\delta]/2$) are considered as noise. Therefore, many fringe (open and closed in any combination) across the frame are required with the aim of such cosinusoidal terms satisfy symmetry and zero mean assumptions.

Number of frames. The presented algorithm can works from only two or more frames. However, a major noise tolerance is experimented when a major number of frames is available as shown in Figs. 3.7(e)-(g) and 3.9(e)-(g).

Single and multiple estimation. The used fringe-pattern normalization method can compute a background-modulation pair for each frame (*multiple estimation mode*) or a single background-modulation pair for all given frames (*single estimation mode*). The conventional phase-shifting algorithms (standard, extended, and generalized) are based on single estimation mode. However, the versatility multiple/single estimation mode of the presented algorithm is convenient for more demanding experiments such as phase-shifting by electrical current laser diode variation [8], amplitude modulation [20], diffraction with amplitude grating [21, 22], and lateral displacement of the illumination source [23].

In summary, the phase-shifting algorithm by parameter estimation is fast, simple, non-iterative, and automatic. This algorithm is useful to work from only two or more frames. The unknown phase step can be between 0 to π rad. Because a normalization stage is used, the presented algorithm can process fringe-patterns with spatio-temporal visibility overcoming the restrictions imposed by conventional approaches. Due to the good properties of this algorithm such as low computational cost, high robustness, and user-free, the presented algorithm is useful for automatic real-time applications.

3.5 | Generalized phase-shifting algorithm for inhomogeneous phase shift

In some optical setups, it is difficult to keep a homogeneous phase shift. For example, the translation of the reference mirror of an interferometer cannot be exactly perpendicular to the optical axis [87–89]. For this case, the phase shift function consists of a set of tilted planes as shown in Fig. 3.1(c). Interesting schemes to solve this problem have been developed [90–93].

Due to the development of advanced technologies (e.g. liquid crystal spatial light modulators) and alternative experimental designs [20, 23], the requirements of phase-shifting algorithms have increased. Particularly, it is necessary to process phase-shifted fringe-patterns with unknown inhomogeneous phase shifts that depend on both p and k [94] as shown in Fig. 3.1(d). Additionally, other desirable properties for phase-shifting techniques are: a reduced number of necessary fringe-patterns, and the capacity to handle spatio-temporal visibility (i.e. the background and modulation lights are functions of both p and k) [95].

In this section an extension of the generalized phase-shifting algorithm [71], given in §3.4, is presented. This previous scheme can handle spatio-temporal visibility (i.e. the background and modulation lights are surfaces changing for each fringe-pattern) but it assumes homogeneous nonlinear phase shift of k, Fig. 3.1(b). Now, an advanced phase shift estimation stage is developed. Thus, the proposed algorithm can efficiently handle inhomogeneous nonlinear phase shifts of both p and k, Fig. 3.1(d). The beneficial properties of the algorithm (such as only two or more fringe-pattern requirements, robustness, user-free execution, and computational efficiency) are kept. Computer simulation and experimental results validate this algorithm.

3.5.1 | Theoretical principles

Hereafter in this paper, for simplicity, all functions of p are defined in a discretized rectangular domain. Thus, they are considered $M \times N$ matrices and the variable p is not written down for brevity. For a $M \times N$ matrix $\mathbb{A}$ with $M \geq N$ and $\operatorname{rank}(\mathbb{A}) = N$, the notation $\mathbb{A}^\dagger = (\mathbb{A}^T\mathbb{A})^{-1}\mathbb{A}^T$ denotes the least-squares inverse of $\mathbb{A}$.

The wrapped phase recovered by the proposed generalized phase-shifting algorithm is carried out through three stages: fringe-pattern normalization, phase shift estimation, and wrapped phase extraction.

Fringe-pattern normalization

The normalization procedure consists of removing the background a_k and modulation b_k lights from the fringe-patterns I_k described by Eq. (3.1). For this, we used the parameter estimation approach [71]. Specifically,

$$\tilde{a}_k = \mathbb{A}_a\mathbb{A}_a^\dagger I_k, \tag{3.34a}$$

$$\tilde{b}_k^2 = 2\mathbb{A}_b\mathbb{A}_b^\dagger(I_k - \tilde{a}_k)^2, \tag{3.34b}$$

where $\tilde{a}_k$ and $\tilde{b}_k$ are the approximations of the parameters a_k and b_k, respectively. The columns of the matrices $\mathbb{A}_a$ and $\mathbb{A}_b$ are the basis functions employed.

Depending on the applications, the basis functions can be truncated Taylor polynomials, Fourier series, piecewise polynomials (splines) of appropriate degree, B-splines, etc. For the simulation and experiments in

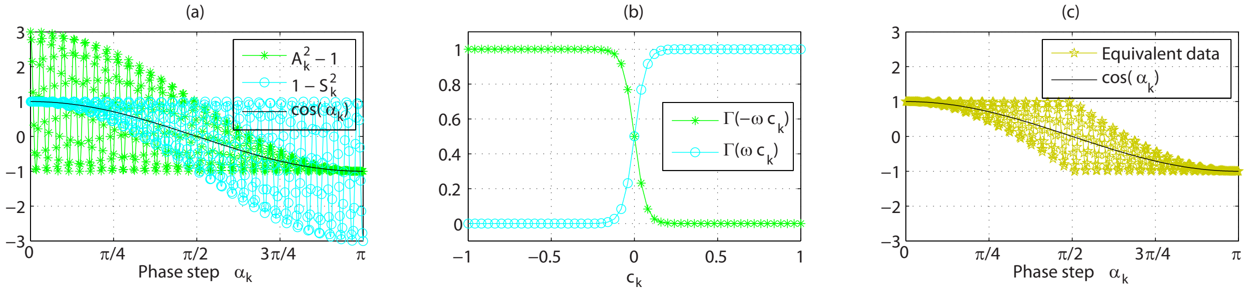


Figure 3.11: *(a) The data $A_k^2 - 1$ and $1 - S_k^2$, given by Eqs. (3.37), for the estimation of the term $\cos\alpha_k$. For $\alpha_k \in [0, \pi/2]$ it is convenient to choose $1 - S_k^2$ because the noise amplitude is lower, and vice versa for $\alpha_k \in [\pi/2, \pi]$. (b) Switch functions by hyperbolic tangent approximation of unit step functions. The constant ω sets the smoothness of the transition at $c_k = 0$. In this plot, $\omega = 15$. (c) Equivalent data to polynomial fitting by the procedure described with Eq. (3.39).*

this work, we used a 2nd-degree polynomial to estimate $\tilde{a}_k$ (a 2nd-order approximation of a Gaussian profile) and, a 4th-degree polynomial to estimate $\tilde{b}_k^2$.

Finally, the normalized fringe-patterns are computed by

$$\bar{I}_k = \operatorname{sat}\left(\frac{I_k - \tilde{a}_k}{\tilde{b}_k}\right) \approx \cos[\phi + \delta_k], \tag{3.35}$$

where the saturation function sat(·) bounds the argument to the interval $[-1, 1]$. For more details on this normalization procedure, the reader is referred to chapter 1 or Ref. [71].

Phase shift estimation

We consider the addition A_k and subtraction S_k of adjacent normalized fringe-patterns defined, respectively, by

$$A_k = \bar{I}_{k-1} + \bar{I}_k = 2\cos\frac{\alpha_k}{2}\cos\left[\phi + \frac{\delta_{k-1} + \delta_k}{2}\right], \tag{3.36a}$$

$$S_k = \bar{I}_{k-1} - \bar{I}_k = 2\sin\frac{\alpha_k}{2}\sin\left[\phi + \frac{\delta_{k-1} + \delta_k}{2}\right], \tag{3.36b}$$

for $k = 1, 2, \cdots, K-1$, where $\alpha_k = \delta_k - \delta_{k-1}$ (see Eq. (3.4)) is the *phase step* between the fringe-patterns I_k and I_{k-1}. From the above equations, by applying some trigonometric identities and a few algebraic operations, we can reach:

$$A_k^2 - 1 = \cos\alpha_k + 2\cos^2\frac{\alpha_k}{2}\eta_k, \tag{3.37a}$$

$$1 - S_k^2 = \cos\alpha_k + 2\sin^2\frac{\alpha_k}{2}\eta_k, \tag{3.37b}$$

where $\eta_k = \cos(2\phi + \delta_{k-1} + \delta_k)$. It is possible to estimate the term $\cos\alpha_k$ from any above equation by considering that the factor η_k is noise with zero mean and amplitude of $2\cos^2(\alpha_k/2)$ or $2\sin^2(\alpha_k/2)$, respectively. However, in some situations, one is more convenient than another as shown in Fig. 3.11(a). Particularly, for $\alpha_k \in [0, \pi/2]$, it is appropriate to choose $1 - S_k^2$, Eq. (3.37b), because the noise amplitude is low (and it is higher in $A_k^2 - 1$), and vice versa for $\alpha_k \in [\pi/2, \pi]$.

Since it is not possible to choose *a priori* the data $A_k^2 - 1$ or $1 - S_k^2$, we carried out a polynomial fitting for these two data. Mathematically,

$$c_{Ak} = \mathbb{A}_c\mathbb{A}_c^{\dagger}(A_k^2 - 1), \tag{3.38a}$$

$$c_{Sk} = \mathbb{A}_c \mathbb{A}_c^{\dagger}(1 - S_k^2), \tag{3.38b}$$

where c_{Ak} and c_{Sk} are the fitted polynomials which approximate the term $\cos\alpha_k$ from $A_k^2 - 1$ and $1 - S_k^2$, respectively. The columns of the matrix $\mathbb{A}_c$ are the basis functions of the polynomial used. Particularly, in this work, a 4th-degree Taylor polynomial is used.

Now, to select between the polynomials c_{Ak} and c_{Sk}, we consider the approximation $\Gamma(\cdot) = [1 + \tanh(\cdot)]/2$ of the unit step function, Fig. 3.11(b), as

$$\begin{aligned} \cos\alpha_k &= c_{Sk}\Gamma(\omega c_k) + c_{Ak}\Gamma(-\omega c_k) \\ &= c_k + \frac{1}{2}(c_{Sk} - c_{Ak})\tanh(\omega c_k), \end{aligned} \tag{3.39}$$

where $c_k = (c_{Ak} + c_{Sk})/2$, and ω is an appropriate constant (experimentally, we chose $\omega = 15$). Thus, a smooth transition between c_{Sk} and c_{Ak} at $c_k = 0$ is obtained.

It is worth mentioning that, instead of carrying out a polynomial fitting procedure to any data set as shown in Fig. 3.11(a), the procedure described by Eq. (3.39) is similar to performing the polynomial fitting to the equivalent data shown in Fig. 3.11(c).

To obtain the phase steps α_k, an inverse cosine of Eq. (3.39) is computed. Then, the phase shift function is constructed by the cumulative sum given by Eq. (3.5). It is not a loss of generality to consider that $\delta_0 = 0$. If additional information over δ_0 is available, it can be included in Eq. (3.5). For example, $\delta_0 = d(p)$ where $d(p)$ is the last phase shift of a previous evaluation.

Wrapped phase extraction

Finally, the wrapped version ϕ_w of the phase distribution of interest ϕ is computed for each point p as

$$\phi_w = \arctan(\zeta/\xi), \tag{3.40}$$

where the quantities $\zeta = \sin\phi$ and $\xi = \cos\phi$ are obtained by the least-squares method [78,79] as

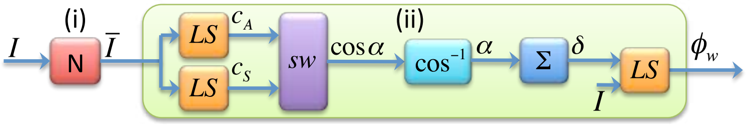
$$\begin{bmatrix} \xi \\ \zeta \end{bmatrix} = \mathbb{A}_{\phi}^{\dagger} \begin{bmatrix} I_0 \\ \vdots \\ I_{K-1} \end{bmatrix}, \tag{3.41}$$

with

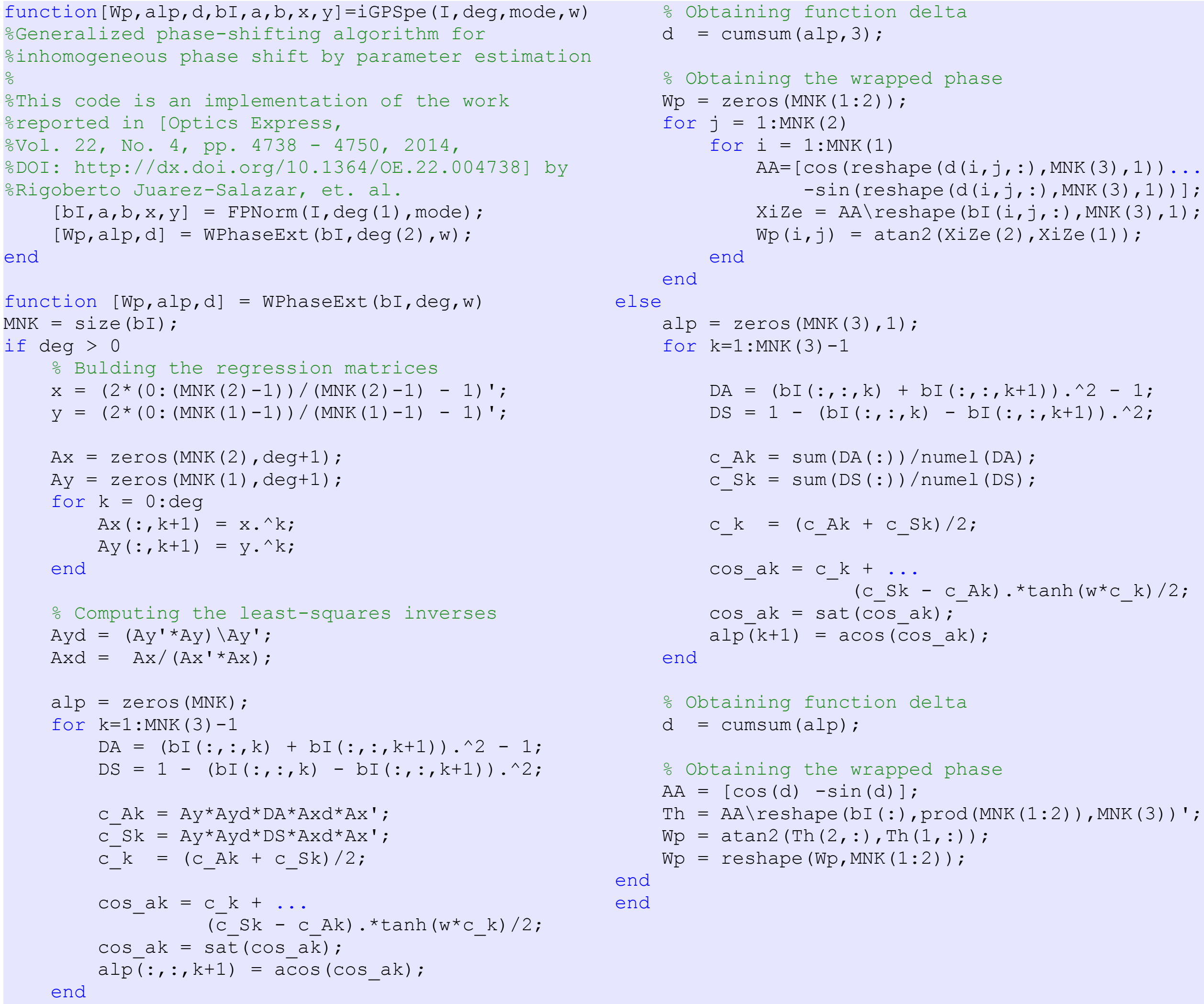
$$\mathbb{A}_{\phi} = \begin{bmatrix} \cos\delta_0 & -\sin\delta_0 \\ \cos\delta_1 & -\sin\delta_1 \\ \vdots & \vdots \\ \cos\delta_{K-1} & -\sin\delta_{K-1} \end{bmatrix}. \tag{3.42}$$

It is worth mentioning that the above matrix $\mathbb{A}_{\phi}$ is, in general, different for each point p. Therefore, this last stage is time-consuming because a matrix inversion must be performed for each point to solve Eq. (3.41). However, the runtime can be reduced by parallel computing because the wrapped phase extraction is pointwise. Particularly, the wrapped phase extraction is faster when the phase shift is homogeneous (non-tilted planes). In this case, the matrix $\mathbb{A}_{\phi}$ is the same for all points p and a single matrix inversion is necessary.

This proposed generalized phase-shifting algorithm to inhomogeneous phase shift and spatio-temporal visibility is depicted by the block diagram shown in Fig. 3.12(a). A prototype computer implementation by using the MATLAB software is given in Fig. 3.12(b).



(a) Block diagram.



```
function[Wp,alp,d,bI,a,b,x,y]=iGPSpe(I,deg,mode,w)
%Generalized phase-shifting algorithm for
%inhomogeneous phase shift by parameter estimation
%
%This code is an implementation of the work
%reported in [Optics Express,
%Vol. 22, No. 4, pp. 4738 - 4750, 2014,
%DOI: http://dx.doi.org/10.1364/OE.22.004738] by
%Rigoberto Juarez-Salazar, et. al.
    [bI,a,b,x,y] = FPNorm(I,deg(1),mode);
    [Wp,alp,d] = WPhaseExt(bI,deg(2),w);
end

function [Wp,alp,d] = WPhaseExt(bI,deg,w)
MNK = size(bI);
if deg > 0
    % Bulding the regression matrices
    x = (2*(0:(MNK(2)-1))/(MNK(2)-1) - 1)';
    y = (2*(0:(MNK(1)-1))/(MNK(1)-1) - 1)';

    Ax = zeros(MNK(2),deg+1);
    Ay = zeros(MNK(1),deg+1);
    for k = 0:deg
        Ax(:,k+1) = x.^k;
        Ay(:,k+1) = y.^k;
    end

    % Computing the least-squares inverses
    Ayd = (Ay'*Ay)\Ay';
    Axd =  Ax/(Ax'*Ax);

    alp = zeros(MNK);
    for k=1:MNK(3)-1
        DA = (bI(:,:,k) + bI(:,:,k+1)).^2 - 1;
        DS = 1 - (bI(:,:,k) - bI(:,:,k+1)).^2;

        c_Ak = Ay*Ayd*DA*Axd*Ax';
        c_Sk = Ay*Ayd*DS*Axd*Ax';
        c_k  = (c_Ak + c_Sk)/2;

        cos_ak = c_k + ...
                 (c_Sk - c_Ak).*tanh(w*c_k)/2;
        cos_ak = sat(cos_ak);
        alp(:,:,k+1) = acos(cos_ak);
    end
    % Obtaining function delta
    d  = cumsum(alp,3);

    % Obtaining the wrapped phase
    Wp = zeros(MNK(1:2));
    for j = 1:MNK(2)
        for i = 1:MNK(1)
            AA=[cos(reshape(d(i,j,:),MNK(3),1))...
                -sin(reshape(d(i,j,:),MNK(3),1))];
            XiZe = AA\reshape(bI(i,j,:),MNK(3),1);
            Wp(i,j) = atan2(XiZe(2),XiZe(1));
        end
    end
else
    alp = zeros(MNK(3),1);
    for k=1:MNK(3)-1

        DA = (bI(:,:,k) + bI(:,:,k+1)).^2 - 1;
        DS = 1 - (bI(:,:,k) - bI(:,:,k+1)).^2;

        c_Ak = sum(DA(:))/numel(DA);
        c_Sk = sum(DS(:))/numel(DS);

        c_k  = (c_Ak + c_Sk)/2;

        cos_ak = c_k + ...
                    (c_Sk - c_Ak).*tanh(w*c_k)/2;
        cos_ak = sat(cos_ak);
        alp(k+1) = acos(cos_ak);
    end

    % Obtaining function delta
    d  = cumsum(alp);

    % Obtaining the wrapped phase
    AA = [cos(d) -sin(d)];
    Th = AA\reshape(bI(:),prod(MNK(1:2)),MNK(3))';
    Wp = atan2(Th(2,:),Th(1,:));
    Wp = reshape(Wp,MNK(1:2));
end
end
```

(b) Implementation in MATLAB

Figure 3.12: *Generalized phase-shifting algorithm for inhomogeneous phase shift by the* `iGPSpe()` *function. This function call the* `FPNorm()` *routine (the fringe-pattern normalization presented in chapter 1).*

3.5.2 | Algorithm testing

The functionality of the proposal is verified by a computer simulation. Then, the feasibility and robustness are tested by processing experimental phase-shifted fringe-patterns.

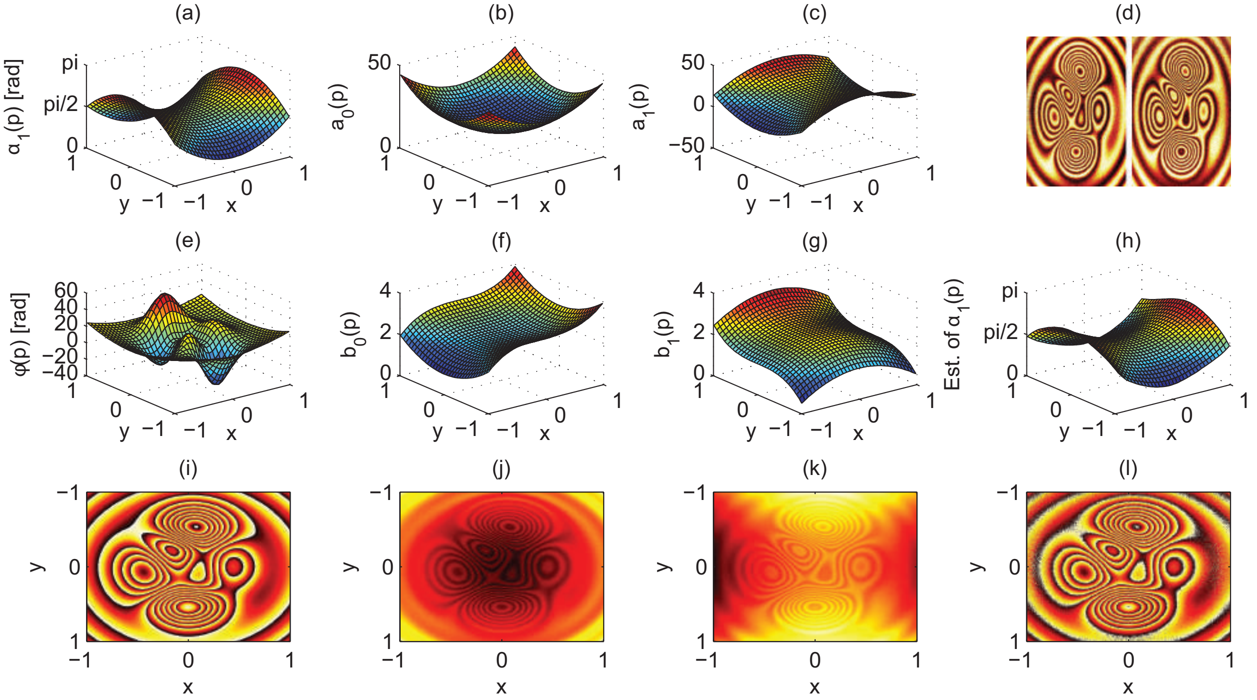


Figure 3.13: *Computer simulation. (1st column) Phase step α_1, phase distribution ϕ to be recovered and its wrapped version. (2nd column) Background a_0 and modulation b_0 lights for the first fringe-pattern I_0. Similarly, the 3rd column shows a_1, b_1 and I_1. (4th column) Normalized fringe-patterns, computed phase step, and recovered wrapped phase.*

Computer simulation

We consider the phase step α_1 and the phase distribution ϕ given, respectively, by

$$\begin{aligned} \alpha_1 &= \pi\left[\frac{1}{2}+\frac{1}{3}(x^2-y^2)\right], \\ \phi &= 6\ \texttt{peaks(500)} + 12(x^2+y^2)+1, \end{aligned} \tag{3.43}$$

for $x, y \in [-1, 1]$. The command `peaks(500)` is defined in MATLAB software and returns a linear combination of Gaussian functions. The plots of Eqs. (3.43) are shown in the first column of Fig. 3.13.

Now, two synthetic noisy fringe-patterns are simulated as

$$I_k = a_k + b_k\cos(\phi+\delta_k)+\rho_k, \tag{3.44}$$

for $k = 0, 1$, where ρ_k is a Gaussian noise with zero mean and standard deviation $\sigma(10\%)$. The respective background and modulation lights are given by

$$\begin{aligned} a_0 &= 15(x^2+y^2+1), & b_0 &= x^3+y^2+2, \\ a_1 &= 20(y^2-x^2+1)+5x, & b_1 &= y^3-x^2+2.5. \end{aligned} \tag{3.45}$$

The functions a_k, b_k, and I_k are shown in the second and third columns of Fig. 3.13 for $k = 1, 2$, respectively.

The two simulated fringe-patterns I_k, Figs. 3.13(j) and 3.13(k), are processed by the proposed algorithm. The resulting normalized fringe-patterns $\bar{I}_k$, phase step α_1, and the recovered wrapped phase are shown in the last column of Fig. 3.13.

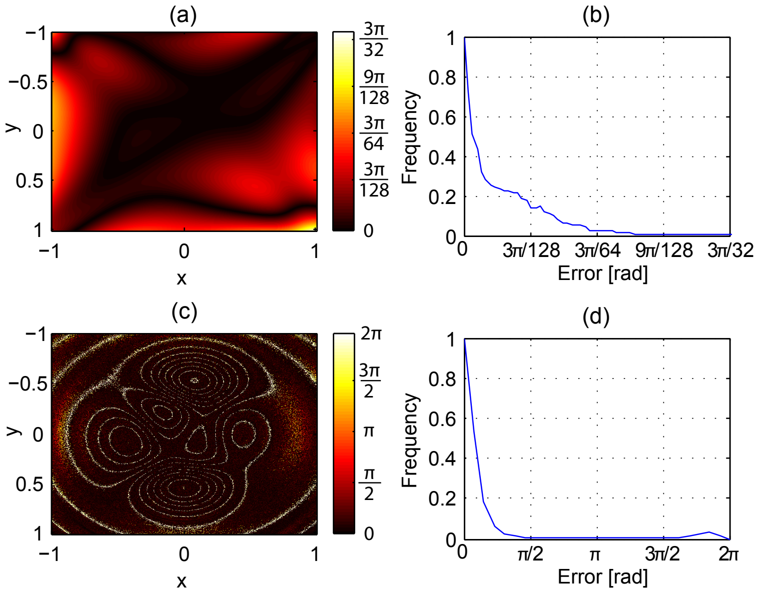


Figure 3.14: *Computer simulation. (1st column) Absolute error from the phase step and wrapped phase estimations. (2nd column) The respective error histograms.*

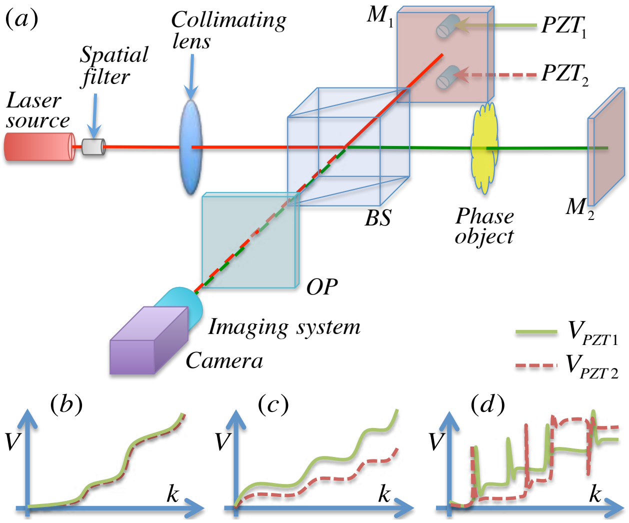


Figure 3.15: *(a) Optical setup for the phase-shifting experiments. Pictographic description of the piezo-electric control signals to generate phase shifts: (a) homogeneous, (b) inhomogeneous tilted planes, and (c) inhomogeneous surfaces.*

By direct comparison between Figs. 3.13(i) and 3.13(l), we can see that a correct solution was reached. This claim is verified by the error information shown in Fig. 3.14. The first column of Fig. 3.14 shows the absolute error from estimation of the phase step, Fig. 3.14(a), and the wrapped phase, Fig. 3.14(c). The second column of Fig. 3.14 shows the respective error histograms (normalized in frequency).

Experimental results

The feasibility of the proposed algorithm was examined by processing experimental fringe-patterns recorded from the Twyman-Green Interferometer shown in Fig. 3.15(a). A collimated laser beam was obtained by a He–Ne laser source (wavelength $\lambda = 633$ nm), a spatial filter (objective microscope and pinhole), and a collimating lens (focal length $f = 0.5$ m).

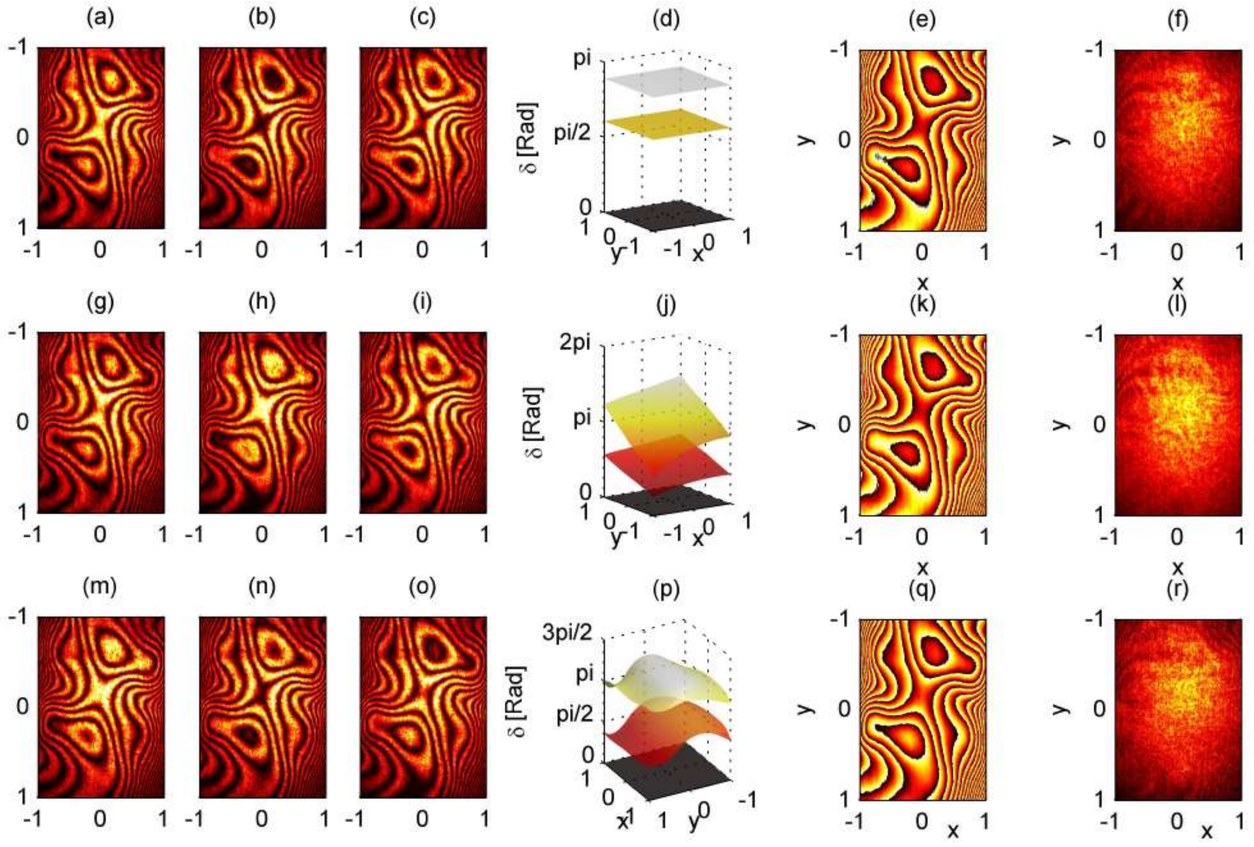


Figure 3.16: *Experimental results. Wrapped phase computed from three phase-shifted fringe-patterns when the phase shift is: (1st row) homogeneous (non-tilted planes), (2nd row) inhomogeneous tilted planes, and (3rd row) inhomogeneous surfaces. (1-3rd columns) Fringe-patterns to be processed. (4th column) Computed phase shift. (5th column) Recovered wrapped phase. (6th column) Resulting background light.*

In the first interferometer's arm, the translation of the flat mirror M_1 is driven by two piezoelectric devices, PZT_1 and PZT_2. Thus, if the control signals for these devices are equal, Fig. 3.15(b), a homogeneous phase shift is induced. Otherwise, Fig. 3.15(c), an inhomogeneous tilted planes phase shift is generated and the tilting is controlled by the difference between the control signals. Moreover, if the control signals are impulsive, Fig. 3.15(d), an inhomogeneous surface phase shift is induced due to this mechanical perturbation.

A distorted wavefront was generated by inserting a phase object in the second interferometer's arm. The fringe-patterns observed on the plane OP were recorded by using a gray-scale camera sensor (1824×1418 pixels with 8-bit pixel depth). Notice that the phase object is not previously characterized. However, it is measured by first phase-shifting with a conventional homogeneous phase shift. Thus, when inhomogeneous phase shifts are considered, the results must be consistent with the first measure.

The proposed algorithm requires only two or more fringe-patterns to work. However, we considered three fringe-patterns in order to test the phase shift estimation quality later. Accordingly, three phase-shifting experiments were carried out. The induced phase shifts are, respectively: homogeneous (i.e., non-tilted planes), Fig. 3.16(1st row), inhomogeneous tilted planes, Fig. 3.16(2nd row), and inhomogeneous surfaces, Fig. 3.16(3rd row).

There are several evaluation methods to test the performance of generalized phase-shifting algorithms [103]. In this work, the proposed algorithm is verified by using the background-phase correlation [104]. The principle of background-phase correlation is as follows. If the phase shift δ is obtained with an error $\varepsilon \ll 1$ (i.e., we have $\tilde{\delta} = \delta + \varepsilon$), the least-squares method to phase extraction solves the problem (at least three fringe-patterns are required):

$$\begin{aligned} I &= a + b\cos(\phi + \tilde{\delta}) \\ &\approx \tilde{a} + b\cos(\phi + \delta), \end{aligned}$$

where the computed background $\tilde{a} = a - \varepsilon b\sin(\phi + \delta)$ is correlated with the phase function ϕ by the term $\varepsilon b\sin(\phi + \delta)$. Thus, a simple way to test the phase shift estimation quality is by computing the correlation coefficient between the reconstructed background light and the fringe-patterns.

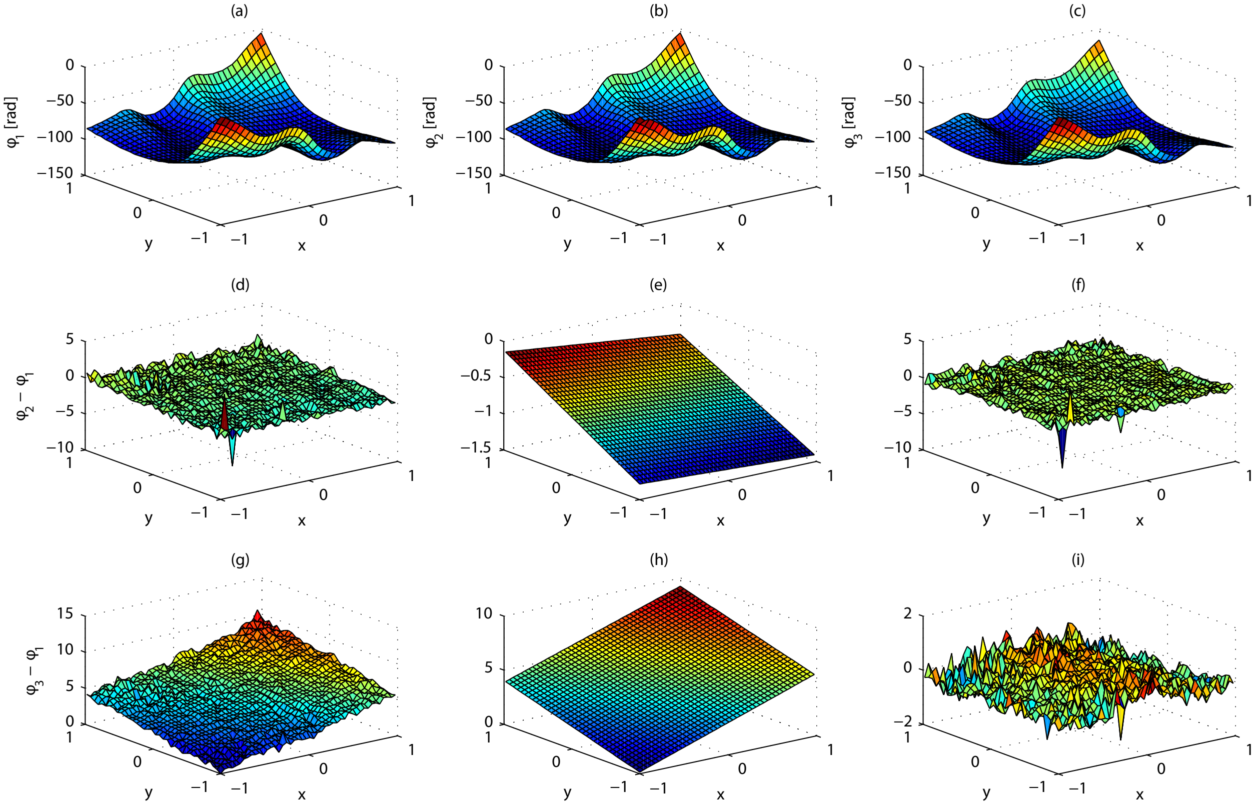


Figure 3.17: *(1st row) Unwrapped phases ϕ_1, ϕ_2, and ϕ_3, respectively, of the wrapped phase maps shown in the 5th column of Fig. 3.16. (2nd row) The difference $\phi_2 - \phi_1$, its fitted plane, and the fitting error (4.5×10^{-15} average value). (3rd row) The difference $\phi_3 - \phi_1$, its fitted plane, and the fitting error (-3.3×10^{-14} average value).*

The resulting background light for each phase-shifting experiment is shown in Fig. 3.16(6th column). In each experiment, the reconstructed background light is compared with the fringe-patterns and the three respective correlation coefficients are computed. The maximum correlation values are 0.5618 (first experiment), 0.5579 (second experiment), and 0.6271 (third experiment). The accuracy levels reached in all experiments are similar (the computed correlation coefficient is 0.5925 ± 0.0346).

3.5.3 | Discussion

Reference frame in phase-shifting

It seems that the extracted wrapped phase maps (5th column in Fig. 3.16) do not describe the same object phase because these maps are not equal. Since the measured phase object is the same in all measurements, the recovered wrapped phases must match in some sense. Indeed, as phase-shifting techniques measure only the relative phase distribution, it can be transformed by translations and/or rotations without affecting the topography of the recovered phase distribution. That is rather an effect due to the established reference frame. In particular, the "reference point" in phase-shifting is given by the phase shift $\delta_k(p)$ at $k = 0$ as is shown below.

The wrapped data of the 5th column in Fig. 3.16 are unwrapped by using the phase-unwrapping method by the rounding-least-squares approach [105] (this algorithm will be presented in the chapter 4). The respective results (which will be labeled as ϕ_1, ϕ_2, and ϕ_3) are shown in Figs. 3.17(a)-(c). The differences $\phi_2 - \phi_1$ and $\phi_3 - \phi_1$ are shown in Figs. 3.17(d) and 3.17(g), respectively. We can see that these differences are planes. This claim is verified by fitting linear functions, Figs. 3.17(e) and 3.17(h), where the fitting error, Figs. 3.17(f) and 3.17(i), present average values of 4.5×10^{-15} and -3.3×10^{-14}, respectively. Although the difference $\phi_3 - \phi_2$ is not shown, it can be proved that such difference is also a plane.[6] Accordingly, the experimental wrapped phase maps in Fig. 3.16 actually describe the same topography.

Furthermore, the differences shown in Figs. 3.17(d) and 3.17(g) are explicable. The measurements were performed in sequence (first, phase-shifting by non-tilted plane phase steps, next, tilted planes, and, lastly, surface phase shifts). In all measurements, $\delta_0(p) = 0$ was assumed. Then,

- For the first experiment, the wavefront of interest was measured and an additional phase (the non-tilted phase shift) was induced.
- For the second experiment, since $\delta_0(p) = 0$ was considered, the recovered phase distribution corresponds to the wavefront of interest plus the phase shift from the first experiment. Thus, the difference between the results from this experiment and the first one must be a non-tilted plane. Approximately, this difference is observed in Fig. 3.17(d). Again, an additional phase (a tilted plane) is induced by the phase-shifting procedure.
- For the third experiment, the wavefront of interest plus the cumulated phase shifts was measured by assuming $\delta_0(p) = 0$. Therefore, the difference between the results from this experiment and the first one is the resultant tilted plane as is observed in Fig. 3.17(g).
- In any future experiment, the wavefront of interest plus the phase added by previous phase-shifting experiments will be sensed.

Notice that, if the initial phase is known (e.g. the last induced phase shift), it can be included as $\delta_0(p)$ in Eq. (3.5). This subtracts this cumulated phase from the final wrapped phase map. In any case, the proposed phase-shifting algorithm can extract the interest wrapped phase distribution up to an initial phase shift $\delta_0(p)$ as occurs with any phase-shifting method.

[6] We consider this example: let $f_1 = \phi$ be the first phase function with ϕ being an arbitrary function. Then, for a second and third phase distributions $f_2 = \phi + g$ and $f_3 = \phi + h$ where g and h are a planes, the difference $f_3 - f_2 = g - h$ is another plane.

Spatio-temporal visibility fluctuations

The spatio-temporal visibility is addressed by the normalization stage in the algorithm's scheme. This good feature was shown by computer simulation. But, this property could not be confirmed in the experimental evaluations because the employed optical setup does not exhibit such visibility fluctuations. However, the potential to cope spatio-temporal visibility by the normalization stage was studied in experimental cases also as shown in chapter 1.

Algorithm performance

The proposed algorithm is based on the computationally efficient least-squares method. The suggested scheme shown in Fig. 3.12 provides high robustness and easy implementation. Among its processing stages, the wrapped phase extraction is the most computationally intensive because it requires a matrix inversion for each pixel of the fringe-pattern. However, this pointwise computation also allows parallel computing. By using a 2.5 GHz laptop, the computer time required to obtain the results shown in Figs. 3.13 and 3.16 is shown in Fig. 3.18.

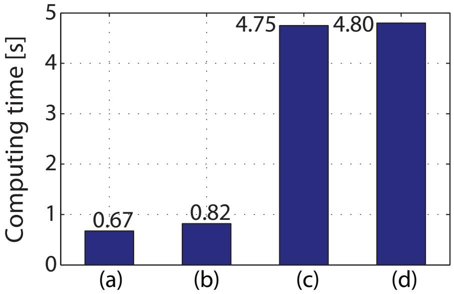


Figure 3.18: *Computing time required for processing: (a) Two* 500×500 *fringe-patterns with inhomogeneous nonlinear phase shift (Fig. 3.13). (b)-(d) Three* 1824×1418 *fringe-patterns with: homogeneous (constant), inhomogeneous linear, and inhomogeneous nonlinear phase shifts, respectively (Fig. 3.16). The hardware used was a 2.5 GHz laptop.*

From the first bar of the plot in Fig. 3.18, we can see that the phase extraction from two fringe-patterns of moderate size is carried out in less that one second. For bigger fringe-patterns, quasi-dynamic measurements are still possible as is the case of an optical workshop (e.g., interferometrical monitoring from 1824×1418 pixels fringe-patterns every 5 seconds). Notice that this algorithm is very fast when the particular homogeneous phase shift is involved (Fig. 3.18(b)) because a single matrix inversion in the wrapped phase extraction stage is sufficient.

On the other hand, the algorithm settings (mainly basis functions and degree of the polynomials) are established before the algorithm is running. Further user intervention is not required. Consequently, due to the low hardware requirements and user-free execution of the proposal, it could be used in automatic applications.

The simplicity of the suggested algorithm allows its implementation into dedicated hardware such as DSP (Digital Signal Processor). This will reduce the computing time notably. Thus, this algorithm could be used in real-time applications.

Operating requirements

The proposed algorithm is highly flexible for processing phase-shifted fringe-patterns. However, to obtain the optimal algorithm performance, there are three main issues to keep in mind.

First, the fitting procedure by using the least-squares method is efficient when the data frames are corrupted with additive symmetrically distributed zero mean noise. This method is applied to obtain the functions a_k, b_k and α_k where the additive terms $\cos(\phi + \delta_k)$ and $\cos(2\phi + \delta_{k-1} + \delta_k)$ (with their respective amplitudes) are seen as noise. Thus, many fringes (open and closed in any combination) across the recorded intensities I_k are required in order to satisfy both symmetric distribution and zero mean conditions. The property of additivity of the noise does not allow processing speckled fringe-patterns because speckle noise is multiplicative. However, the method could be applicable if pre-filtering of the fringe-patterns is accomplished.

Second, the estimation of the functions a_k, b_k, and α_k requires appropriate basis functions. Fortunately, there are a variety of options from which we can choose for each particular application. For example, the basis functions can be polynomials (such as truncated Taylor and Fourier series, Seidel, Zernike, etc.), splines (piecewise polynomials) of appropriate degree, B-splines, etc. This flexibility allows the proposed algorithm to be implemented in many applications.

And third, the phase steps α_k are computed as the argument of a cosine function, Eq. (3.38). Therefore, the phase steps are recovered without ambiguity if α_k lies within the interval $(0, \pi)$ rad. Nevertheless, the phase shift δ_k (the cumulative sum of the phase steps) may be greater than 2π as long as the matrix $\mathbb{A}_\phi$ of Eq. (3.42) satisfies $\text{rank}(\mathbb{A}_\phi) = 2$. If the phase steps exceed the interval $(0, \pi)$, they are wrapped. We believe that this issue can be overcome by an additional unwrapping procedure; however, this possibility is left as future work.

3.6 | Conclusion

The generalized phase-shifting scheme for wrapped phase extraction is the most versatile and useful in practice because it relieves the experimental requirements.

Conventional generalized algorithms require three or more frames. Moreover, such algorithms cannot address neither spatio-temporal visibility nor inhomogeneous phase shifts.

In this chapter a generalized inhomogeneous phase-shifting algorithm working with only two or more frames was presented. The algorithm is based on a cascade least-squares estimation scheme. This algorithm can handle fringe-patterns with spatio-temporal visibility and inhomogeneous surface phase shift. The algorithm features and its operating conditions have been discussed.

Computer simulations and experimental results have shown the feasibility of this algorithm. Moreover, the obtained results have shown that the wrapped phase computing is possible even when an inhomogeneous phase shift is induced (e.g., by nonlinearity of piezoelectric materials or miscalibrated phase shifters) and large visibility changes have occurred. The computational efficiency and user-free execution of this algorithm allows it to be implemented in automatic applications.

Chapter 4

Phase-unwrapping algorithm by a rounding-least-squares approach

The first principle is that you must not fool yourself —and you are the easiest person to fool.

Richard Feynman

Contents

No much actual physical information is given by a phase distribution if it is in the wrapped format. Both Fourier fringe analysis and phase-shifting provide us wrapped phase maps. Therefore, it is essential a phase-unwrapping procedure to remove the artificial discontinuities induced by the performed phase encoding of the physical information of interest.

This chapter describes the concepts of *local* and *global approaches* for phase-unwrapping and its respective properties. Then, a simple and efficient phase-unwrapping algorithm based on a rounding procedure and a global least-squares minimization is presented.

The performance of the studied phase-unwrapping algorithm is evaluated. The obtained results show that this algorithm reaches high accuracy levels by a low computing time. Moreover, since the proposed algorithm is simple, fast, and user-free, it could be used in automatic real-time applications.

4.1 | Preliminary definitions

The ***floor operator***, denoted as $\lfloor x \rfloor$, returns the first integer number equal or previous to x. It is defined as

$$\lfloor x \rfloor := \max\{m \in \mathbb{Z} | m \leq x\}. \tag{4.1}$$

A useful property of the floor operator is

$$\lfloor x + n \rfloor = \lfloor x \rfloor + n, \tag{4.2}$$

where n is any integer. The ***modulo operation***, denoted as $\mathrm{mod}(x, y)$, returns the remainder after division of x by y. This operation is defined as

$$\mathrm{mod}(x, y) := x - y\lfloor x/y \rfloor, \tag{4.3}$$

where $\mathrm{mod}(x, 0) = x$ by convention. Assuming $y > 0$, it follows that the modulo operation returns a number in the half-open interval $[0, y)$ for any real number x. A useful property of the modulo operation is

$$\mathrm{mod}(x + ky, y) = \mathrm{mod}(x, y), \tag{4.4}$$

where k is any integer number. We define the ***wrapping operator*** $\mathcal{W} : \Re \to [-\pi, \pi)$ as

$$\begin{aligned} \mathcal{W}[x] &:= \mathrm{mod}(x + \pi, 2\pi) - \pi \\ &= x - 2\pi \left\lfloor \frac{x + \pi}{2\pi} \right\rfloor . \end{aligned} \tag{4.5}$$

For any integer number k, a real number $z \in [-\pi, \pi)$real number x

The wrapping operation presents the following properties. Let x and k be any real and integer numbers, respectively, then

$$\mathcal{W}[x + 2\pi k] = \mathcal{W}[x]. \tag{4.6}$$

If y is a real number in the interval $[-\pi, \pi)$, then

$$\mathcal{W}[y] = y. \tag{4.7}$$

4.2 | Introduction

Phase-unwrapping is a key procedure in computer-aided technologies for digital analysis of fringe-patterns [106]. This is an essential procedure to sensing a wide range of static and dynamic physical information such as strain, topography, flow, temperature, magnetic field inhomogeneities, wavefront, among others. Numerous applications involves phase unwrapping, for instance optical testing, biomedical studies, geological surveys, automotive and aerospace industries, mechanical and civil engineering [2, 107].

In the mentioned applications, the physical parameter to be measured is encoded as phase in a cosinusoidal function [72]. Since the phase computing involves an inverse trigonometric operation, only the principal values (wrapped phase) can be determined [3]. Thus, before the physical information can be interpreted, the phase-unwrapping must be performed [107].

Generally speaking, the phase-unwrapping algorithms can be classified into two categories: ***temporal*** and ***spatial methods***.

The temporal methods require a set of sequentially recorded data frames. Then the phase is unwrapped along the time axis for each pixel independently of the others [108, 109]. Because the unwrapping is pointwise, the points or regions with local error do not influence the good data points; i.e., there is no spatial error propagation [110,111]. The main advantage of the temporal scheme is that it can unwrap noisy phase surfaces with complex topography and discontinuities in a very simple and efficient way [112,113]. Unfortunately, the multiple data frames requirement is a serious restriction for many fringe analysis applications. Consequently, the spatial phase-unwrapping methods are more attractive because they require a single data frame.

The spatial phase-unwrapping methods can be further classified in: *local* (path-dependent or path-following), and *global* (path-independent or minimum-norm) approaches [114]. Their operation principles are described as follows.

4.2.1 | Local phase-unwrapping approach

Let $\phi(r)$ be an unwrapped phase map where $r = (x, y)$ is a point in the plane, and let $\psi(r) = \mathcal{W}[\phi(r)]$ be the wrapped version of $\phi(r)$. From Eq. (4.5), we have that the wrapped phase map can be modeled as

$$\psi(r) = \phi(r) - 2\pi k(r), \tag{4.8}$$

where $k(r) = \lfloor(\phi+\pi)/2\pi\rfloor$ is an unknown integer-valued (piecewise constant) function. Since the phase jumps term $2\pi k(r)$ is a piecewise constant function, its gradient $2\pi\nabla k(r)$ (where ∇ is the gradient operator) is zero piecewise. This fact is helpful to obtain the gradient of the unwrapped phase by the Itoh's relation [107,115]:

$$\nabla\phi(r) = \mathcal{W}\{\nabla\psi(r)\}. \tag{4.9}$$

In general, the Eq. (4.9) is the basis of spatial methods [116].

The local phase-unwrapping approach is carried out from Eq. (4.9) by the path-integration procedure:

$$\phi(r) = \phi_0 + \int_C \mathcal{W}\{\nabla\psi(r)\}\ \mathrm{d}r, \tag{4.10}$$

where ϕ_0 is the constant of integration and C is an appropriate integration path. This approach generates fast and computationally simple unwrapping algorithms. However, choosing an integration path is not trivial [117].

For good quality (noise-free) wrapped maps, an irrotational gradient field $\mathcal{W}\{\nabla\psi(r)\}$ is obtained; i.e., the gradient field does not have residues. In this case, the integration path C is arbitrary. Unfortunately, by the unavoidable presence of many error sources in experimental situations, the gradient field $\mathcal{W}\{\nabla\psi(r)\}$ is contaminated with residues. Therefore, the choosing of an integration path C is a problem in itself [118,119]. Some relevant solutions are the residues compensation (branch-cut) [120,121], the quality maps [122–124], reliability maps [109,125], and noise filters [126,127]. However, an important drawback of these methods is the low robustness in the sense that, even for low levels of noise in the wrapped data, a different unwrapped phase can be obtained depending on the selected integration path [128,129].

4.2.2 | Global phase-unwrapping approach

Alternatively, the global approach provides robust phase-unwrapping algorithms because they do not depend on path-following techniques [107,130]. This approach formulates the unwrapping process as the minimization problem [131]:

$$\min_{\phi} \left\| \mathcal{W}[\nabla\psi(r)] - \nabla\phi(r) \right\|_p \tag{4.11}$$

where $\|\cdot\|_p$ denotes the Euclidean ℓ_p norm. Typical algorithms in this category are the cases $p = 1$ (Flynn's method) and $p = 2$ (least squares method).

For the least-squares case, the problem (4.11) is reduced to solve the Poisson's equation. Since there are computationally efficient techniques to solve this type of equation (e.g., the fast Fourier Poisson solver), the least-squares is the most common global phase-unwrapping algorithm [132,133]. Although the global methods are robust and computationally efficient, they tend to distort the phase distribution by residue-spreading and over-smoothing [134].

Residue-spreading is an effect resulting from the global nature of the approximation approach described by (4.11) [132]. In other words, global phase-unwrapping algorithms cannot isolate the residues at the local bad points or regions. On the other hand, over-smoothing is a consequence of the approximation of the input data by a function from a prefixed functions space (mainly polynomial functions and Fourier series) [135, 136]. The residue-spreading problem can be overcome by weighting strategies [137] or filtering systems [138] such as localized compensators [134], total-variation denoising [139], windowed Fourier filtering and least squares [140]. The definition of a most general functions space should solve the over-smoothing problem. Unfortunately, these solutions represent a hard computational load. The longer computing time is unacceptable for reliable and real-time applications.

Instead of working with the gradient field of the wrapped phase given by Eq. (4.9), where the phase jumps term $2\pi k(r)$ is removed, an alternative approach is to extract the gradient of the jumps function $k(r)$ and to work with it. Then, by using the computed jumps function $k(r)$, the phase-unwrapping is carried out by solving the Eq. (4.8) for $\phi(r)$. In principle, this approach would solve both the residue-spreading and over-smoothing problems inherent in (4.11) due to the following reason.

In contrast to the infinite possible values of $\nabla\psi(r)$, the possible values of $\nabla k(r)$ is a finite set D because the wrapped phase exhibits phase jumps of almost 2π in absolute value [141]. In particular,

$$D = \{-1, 0, +1\}, \tag{4.12}$$

i.e., phase jump from π to $-\pi$, no phase jump, and phase jump from $-\pi$ to π, respectively. The reduced data values of $\nabla k(r)$ makes the derived algorithms less sensitive to noise (low residue-spreading). Moreover, because the input data can be represented exactly by a simple piecewise constant functions, the over-smoothing problem is removed.

In this chapter an efficient and robust phase-unwrapping algorithm is presented. This algorithm is based on a rounding to remove the gradient of unwrapped phase and works with the remaining phase jumps. Then, a least-squares integration is applied to find the phase jumps component from its gradient. This algorithm is very simple, fast, user-free, and does not require quality maps or any other additional data or filtering stage; thus, it is appropriate for real-time applications. The feasibility of this algorithm is shown by simulated and experimental data processing. The good performance of the proposal is highlighted by comparing this algorithm with the Minimum Cost Network Flow (MCNF) [142], fast Fourier transform, quality-guided, and branch-cut methods [107].

4.3 | Theoretical algorithm description

Hereafter in this chapter, it is assumed that Ω is a uniformly sampled rectangular domain. Thus, the two-dimensional functions $\psi(r)$, $\phi(r)$ and $k(r)$ are considered as $M \times N$ matrices and the dependence of the variable r is not written down for brevity. We define the discrete gradient ∇f of any $M \times N$ matrix f as

$$\nabla f := \begin{bmatrix} \partial f/\partial x \\ \partial f/\partial y \end{bmatrix} = \begin{bmatrix} f_x \\ f_y \end{bmatrix} = \begin{bmatrix} f L_x^T \\ L_y f \end{bmatrix}, \tag{4.13}$$

with $[\cdot]^T$ denoting matrix transposition, and where L_x and L_y are the *difference operators* along the horizontal and vertical directions, respectively. The matrices L_x and L_y define the finite difference approximation. In general, higher-order methods for approximating the derivative can be applied. Nevertheless, we define the difference operators as the simple double diagonal matrices (first-order approximation):

$$L_x = \begin{bmatrix} -1 & 1 & & \\ & \ddots & \ddots & \\ & & -1 & 1 \end{bmatrix}_{(N-1)\times N}, \tag{4.14a}$$

$$L_y = \begin{bmatrix} -1 & 1 & & \\ & \ddots & \ddots & \\ & & -1 & 1 \end{bmatrix}_{(M-1)\times M}, \tag{4.14b}$$

or more briefly by

$$\begin{aligned} L_x &= \begin{bmatrix} \vec{\mathbf{0}}_{N-1} & \mathbb{I}_{N-1} \end{bmatrix} - \begin{bmatrix} \mathbb{I}_{N-1} & \vec{\mathbf{0}}_{N-1} \end{bmatrix}, \\ L_y &= \begin{bmatrix} \vec{\mathbf{0}}_{M-1} & \mathbb{I}_{M-1} \end{bmatrix} - \begin{bmatrix} \mathbb{I}_{M-1} & \vec{\mathbf{0}}_{M-1} \end{bmatrix}, \end{aligned}$$

where $\mathbf{0}_{a\times b}$ is a $a \times b$ matrix with all entries equal to zero, and $\mathbb{I}_a$ is the squared $a \times a$ identity matrix.

Usually, the difference operators L_x and L_y are defined as square matrices in order that the matrix f and its derivatives fL_x^T and $L_y f$ have the same size $M \times N$. The differentiation matrices (4.14) can be squared by using appropriate discrete differentiation formulas at the end points [143]. However, the simple rectangular definition used in (4.14) is sufficient because these matrices are used in the forms $L_x^T L_x$ and $L_y^T L_y$ (which are squared by $N \times N$ and $M \times M$, respectively) as we will see later.

We consider the gradient of the wrapped data (4.8). From this gradient field, the $1/(2\pi)$ part is taken and the result is rounded; i.e.,

$$\begin{aligned} \text{round}\left(\frac{1}{2\pi}\nabla\psi\right) &= \text{round}\left(\frac{1}{2\pi}\nabla\phi - \nabla k\right) \\ &= \text{round}\left(\frac{1}{2\pi}\nabla\phi\right) - \nabla k, \end{aligned} \tag{4.15}$$

where round$(\cdot)$ is the operator which leads each matrix entry to the nearest integer. The last equality in (4.15) is derived by the fact that ∇k is a integer-valued term (since k is an integer-valued function, its differences k_x and k_y must be integer-valued functions too). Now, if the Nyquist sample condition

$$\left|\nabla\phi\right| < \pi \qquad \forall r \in \Omega \tag{4.16}$$

is satisfied, then the quantity $|\nabla\phi/(2\pi)|$ is less than 1/2 for all r in Ω. Hence, round$[\nabla\phi/(2\pi)]$ is zero. In this context, the term ∇k is the integer component of the quantity $\nabla\psi/(2\pi)$. This convenient property allows computing the gradient field ∇k by the simple rounding:

$$\nabla k = \begin{bmatrix} k_x \\ k_y \end{bmatrix} = -\text{round}\left(\frac{1}{2\pi}\nabla\psi\right). \tag{4.17}$$

The procedure to obtain a function from its gradient (gradient field integration) is a common issue in several computer-aided contexts [143,144]. Analogous to the optimization in (4.11), the computing of k from its gradient field ∇k can be formulated as the least-squares optimization problem [145]:

$$\min_{\tilde{k}} \epsilon(\tilde{k}) = \left\|k_x - \tilde{k}L_x^T\right\|_F^2 + \left\|k_y - L_y\tilde{k}\right\|_F^2, \tag{4.18}$$

where k_x and k_y are the available data given by Eq. (4.17), $\tilde{k}$ is the approximating function and $\|\cdot\|_F$ denotes the Frobenious norm. By definition, we have that, for the real matrix A, the Frobenious norm $\|A\|_F$ satisfies

$$\|A\|_F^2 = \mathrm{tr}(A^T A),$$

where $\mathrm{tr}(\cdot)$ denotes the trace of a square matrix. The above equation allow us to restate the functional in (4.18) as

$$\epsilon = \mathrm{tr}\left[\left(k_x - \tilde{k}L_x^T\right)^T \left(k_x - \tilde{k}L_x^T\right)\right] + \mathrm{tr}\left[\left(k_y - L_y\tilde{k}\right)^T \left(k_y - L_y\tilde{k}\right)\right]. \tag{4.19}$$

Expanding this equation, we have

$$\begin{aligned}\epsilon(\tilde{k}) &= \mathrm{tr}\left[\left(k_x^T - L_x\tilde{k}^T\right)\left(k_x - \tilde{k}L_x^T\right)\right] + \mathrm{tr}\left[\left(k_y^T - \tilde{k}^T L_y^T\right)\left(k_y - L_y\tilde{k}\right)\right] \\ &= \mathrm{tr}\left[k_x^T\left(k_x - \tilde{k}L_x^T\right) - L_x\tilde{k}^T\left(k_x - \tilde{k}L_x^T\right)\right] \\ &\quad + \mathrm{tr}\left[k_y^T\left(k_y - L_y\tilde{k}\right) - \tilde{k}^T L_y^T\left(k_y - L_y\tilde{k}\right)\right] \\ &= \mathrm{tr}\left[k_x^T k_x - k_x^T\tilde{k}L_x^T - L_x\tilde{k}^T k_x + L_x\tilde{k}^T\tilde{k}L_x^T\right] \\ &\quad + \mathrm{tr}\left[k_y^T k_y - k_y^T L_y\tilde{k} - \tilde{k}^T L_y^T k_y + \tilde{k}^T L_y^T L_y\tilde{k}\right].\end{aligned} \tag{4.20}$$

At this point, it is convenient to know some properties of the trace operation.

Basic properties of the trace operation

From linear algebra, it is known that the trace is a linear mapping; i.e.,

$$\mathrm{tr}(\alpha A + \beta B) = \alpha\mathrm{tr}A + \beta\mathrm{tr}B, \tag{4.21}$$

for all square matrices A and B, and all scalars α and β. Another useful trace property is

$$\mathrm{tr}A = \mathrm{tr}A^T. \tag{4.22}$$

In words, the matrix A and its transpose A^T have the same trace. By using the properties (4.21) and (4.22), we have that

$$\begin{aligned}\mathrm{tr}[\alpha A + \beta A^T] &= \mathrm{tr}[(\alpha+\beta)A] \\ &= \mathrm{tr}[(\alpha+\beta)A^T].\end{aligned} \tag{4.23}$$

Some useful derivatives of traces are the following. If A, B, and X are three matrices of appropriate sizes, then holds that

$$\frac{\partial}{\partial X}\mathrm{tr}(AXB) = A^T B^T, \tag{4.24}$$

and

$$\frac{\partial}{\partial X}\mathrm{tr}(A^T X^T B X A) = (B^T + B)XAA^T. \tag{4.25}$$

The relations (4.24) and (4.25) are known as the first and second order derivatives of traces, respectively.

The equalities in (4.23) allow us to simplify the last equation in (4.20) as

$$\epsilon = \mathrm{tr}\left[k_x^T k_x - 2k_x^T\tilde{k}L_x^T + L_x\tilde{k}^T\tilde{k}L_x^T\right] + \mathrm{tr}\left[k_y^T k_y - 2k_y^T L_y\tilde{k} + \tilde{k}^T L_y^T L_y\tilde{k}\right]. \tag{4.26}$$

Now, we use the necessary first-order condition for optimality

$$\frac{\partial}{\partial \tilde{k}} \epsilon(\tilde{k}) = 0. \tag{4.27}$$

The linearity property of both the trace and the derivative operations allow us to write

$$-2\frac{\partial}{\partial \tilde{k}} \mathrm{tr}\left[k_x^T \tilde{k} L_x^T\right] + \frac{\partial}{\partial \tilde{k}} \mathrm{tr}\left[L_x \tilde{k}^T \tilde{k} L_x^T\right] - 2\frac{\partial}{\partial \tilde{k}} \mathrm{tr}\left[k_y^T L_y \tilde{k}\right] + \frac{\partial}{\partial \tilde{k}}\left[\tilde{k}^T L_y^T L_y \tilde{k}\right] = 0. \tag{4.28}$$

By using the Eqs. (4.24) and (4.25), the above equation become to The linearity property of both the trace and the derivative operations allow us to write

$$-2k_x L_x + 2\tilde{k} L_x^T L_x - 2L_y^T k_y + 2L_y^T L_y \tilde{k} = 0. \tag{4.29}$$

or in compact form

$$A\tilde{k} + \tilde{k}B = C, \tag{4.30}$$

where $A = L_y^T L_y$, $B = L_x^T L_x$, and $C = L_y^T k_y + k_x L_x$. In literature, equation (4.30) is known as *Lyapunov's* [146] or *Sylvester's equation* [147].

Notice that, although L_x and L_x are $(N-1) \times N$ and $(M-1) \times M$ matrices, the solution matrix $\tilde{k}$ in Eq. (4.30) has the size of $M \times N$ (the same size as the wrapped phase map ψ) as it should be.

Efficient methods to solve Eq. (4.30) have been proposed [147, 148]. Most of them are based on simplifying the original matrix equation by similarity transformations [149]. Intuitively, the solution of Eq. (4.30) for $\tilde{k}$ is obtained as follows. Let $U\Sigma V^T$ be the singular value decomposition of B. Particularly, since B is a symmetric matrix, then the orthogonal matrices U and V are equals and $\Sigma = \mathrm{diag}(\sigma_1, \cdots, \sigma_N)$ is a square diagonal matrix with σ_i the singular values of B. Accordingly, the right multiplication of Eq. (4.30) by U generates the equation

$$A\tilde{k}U + \tilde{k}BU = CU. \tag{4.31}$$

By inserting the identity matrix $I = UU^T$ in the second term of the left side of the above equation, we have $A\tilde{k}U + \tilde{k}UU^T BU = CU$ which is rewritten as

$$AK + K\Sigma = D, \tag{4.32}$$

where $K = \tilde{k}U$ and $D = CU$. It is not difficult to solve the transformed system (4.32) because K can be computed column-wise. Namely, let $K = [\kappa_1\ \kappa_2\ \cdots\ \kappa_N]$ and $D = [d_1\ d_2\ \cdots\ d_N]$ be the column partitioning of K and D, respectively. The column κ_i is computed by a standard matrix inversion as

$$\kappa_i = (A + \sigma_i I)^{-1} d_i \tag{4.33}$$

with $i = 1, 2, \cdots, N$. When all columns of K are computed, the original equation (4.30) is solved by $\tilde{k} = KU^T$. Because this procedure computes the solution $\tilde{k}$ in real values, the rounding is applied again to obtain $\bar{k}$ (integer values).

Finally, with the computed jumps function $\bar{k}$, the phase-unwrapping is carried out by solving the equation (4.8) for ϕ as

$$\phi = \psi + 2\pi\bar{k}. \tag{4.34}$$

This algorithm is summarized in Fig. 4.1 by a block diagram representation.

It is worth mentioning that the computational effort to execute this algorithm depends only on the size of the input wrapped data. In contrast, the computing time required by the path-following methods (where the MCNF is included) depends on both the size of the input data and the noise content. This is shown by the following computer simulation and experimental results.

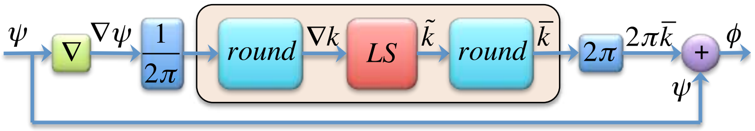


Figure 4.1: *Proposed rounding-least-squares scheme to phase-unwrapping.*

4.4 | Simulated and experimental results

In this section, the proposed algorithm is tested by processing both synthetic and experimental wrapped phase maps. In order to show the full potential of the proposed method, it is compared with the well-known Minimum Cost Network Flow (MCNF), the Fast Fourier Transform (FFT), the Quality Guided (QG) and the branch-cut methods.

4.4.1 | Computer simulation

Since unwrapping routines are sensitive to noise and the local gradient (or spatial frequency), among others [150], a computer simulation was designed to include these two harmful parameters.

A $N \times N$ (with $N = 200$) square domain is considered. The employed unwrapped phase distribution is:

$$\phi(x,\sigma) = ax^2 + \eta(x,\sigma), \tag{4.35}$$

where $a = N\pi/2$ is a constant, and $\eta(x,\sigma)$ is a zero mean Gaussian noise term along the x-axis with standard deviation of σ. The variables x and σ are defined respectively as

$$x \in [0,1], \qquad \sigma \in [0,\pi/5]\,, \tag{4.36}$$

where $\pi/5$ represent a standard deviation $\sigma(10\%)$ rad of error. These synthetic data are shown in Fig. 4.2.

Notice that the phase distribution (4.35) is such that the spatial frequency linearly increases along the x-axis (from zero at $x = 0$ until to the Nyquist frequency limit at $x = 1$) while the random noise has a constant standard deviation σ. On the other hand, the standard deviation of the random noise linearly increases along the σ axis while the spatial frequency is constant. Accordingly, the performance on phase-unwrapping is evaluated on both spatial frequency and noise level simultaneously.

The wrapped phase map shown in Fig. 4.2(c) is processed by the MCNF, the proposed rounding-least-squares, the FFT, the quality-guided, and branch-cut algorithms. The resulting unwrapped maps are shown in Figs. 4.2(d)-(h), respectively. The relative error for each unwrapped phase is shown in Figs. 4.2(i)-(m), respectively.

The last row of Fig. 4.2 reveals that the MCNF method has the best accuracy and it is seconded by our proposed algorithm. The error from the FFT method is slightly higher than our proposed method. The quality-guided method has a big error. The lower accuracy is observed in the branch-cut method because it does not work in the region with both high frequency and high noise levels.

The computer time required for each phase-unwrapping algorithm is shown in Fig. 4.3. Both our proposed algorithm and the FFT require the least amount of computer time (0.1018 ± 0.0078 s). The MCNF and the branch-cut needed much more computer time (5.68 and 6.11 s, respectively). The most computational intensive algorithm was the quality guided method by requiring the higher computer time (28.34 s).

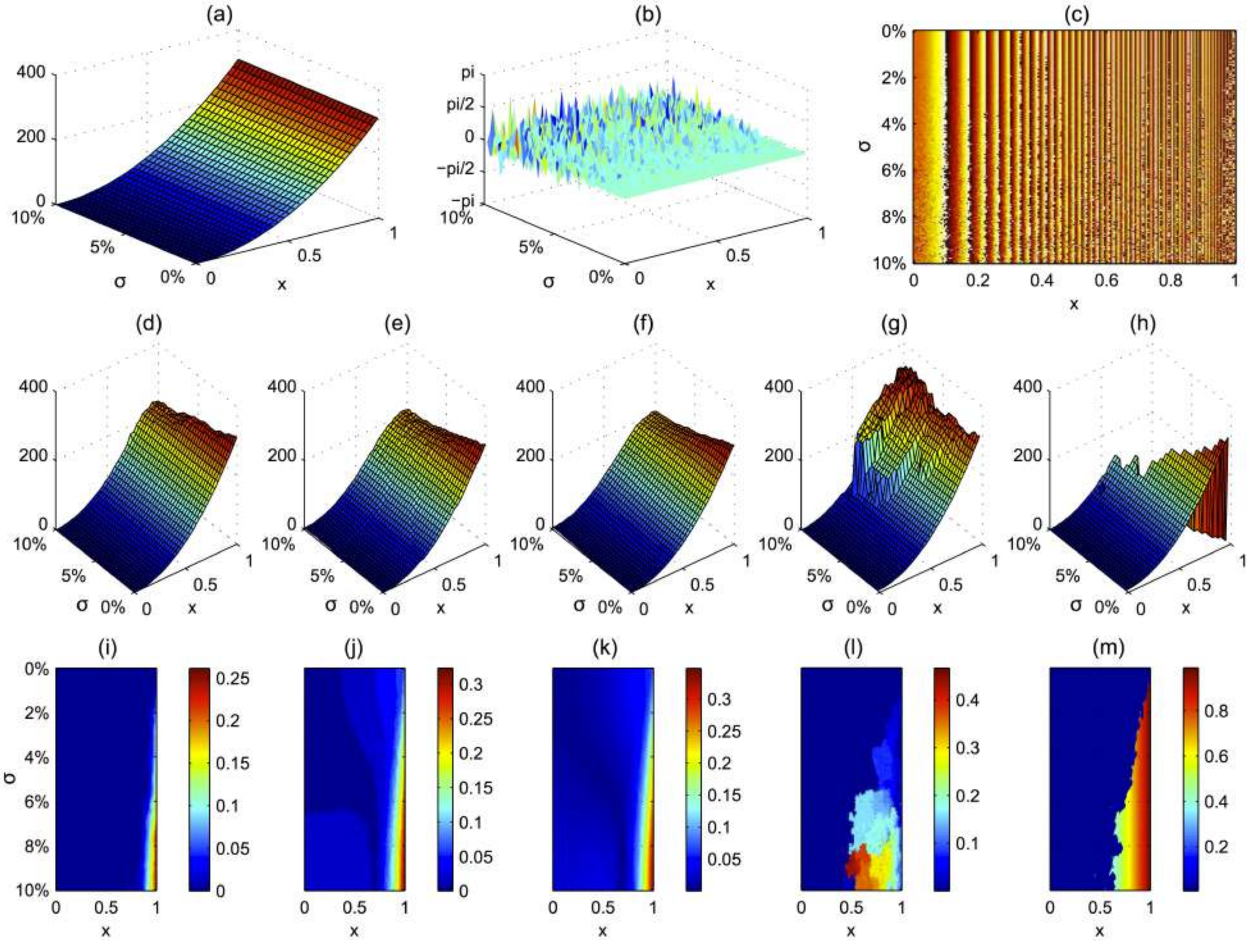


Figure 4.2: *Simulation results. (a) Synthetic noisy unwrapped phase $\phi(x, \sigma)$. (b) The noise component $\eta(x, \sigma)$. (c) Wrapped phase map. Unwrapped phase recovered by: (d) the MCNF, (e) the proposed rounding-least-squares, (f) the FFT, (g) the quality-guided, and (h) the branch-cut methods. (i)-(m) The respective relative errors.*

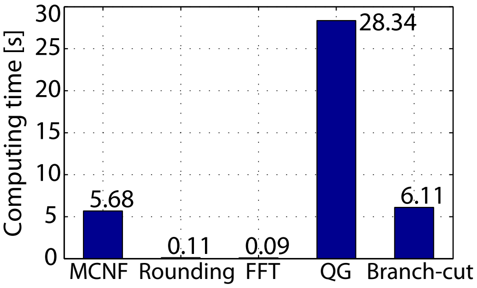


Figure 4.3: *Computing time required to process a wrapped phase map of 200×200 pixels by the MCNF, the proposed rounding-least squares, the FFT, the quality-guided, and the branch-cut methods in a 2.5 GHz laptop.*

4.4.2 | Processing of experimental wrapped phase maps

In order to test the robustness of the proposed algorithm, three experimental wrapped phase maps, shown in Fig. 4.4, were processed. These data frames were obtained from a Fizeau interferometer (by using a Fourier normalized fringe analysis method [28]), by fringe projection (by using the four-step phase-shifting method [3]), and by a Twyman-Green Interferometer (by using the generalized phase-shifting method presented in §3.4 and reported in [71]). The array sizes are 404×392, 326×421, and 205×262 pixels, respectively.

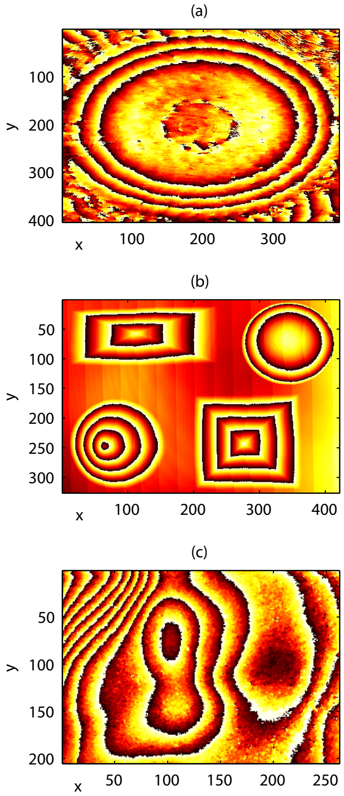


Figure 4.4: *Experimental wrapped phase maps obtained from a Fizeau Interferometer (a), fringe projection (b), and a Twyman-Green Interferometer (c).*

The computed unwrapped phase for each input frame are shown in the first three rows of Fig. 4.5, respectively. The columns in Fig. 4.5 correspond with the results obtained by the MCNF, our proposed method, the FFT method, the quality-guided, and the branch-cut, respectively.

Since the MCNF method has the higher accuracy (it was verified in the previous subsection titled *Computer simulation*), its unwrapped phase results are used as a baseline. Thus, the absolute error from the remaining methods are shown in the last three rows of Fig. 4.5. The distribution is similar to the block of unwrapped phases: the rows correspond with the input data and the columns correspond with the unwrapping method employed.

From the experimental results shown in Fig. 4.5 we can see that, the proposed rounding-least-squares method reaches a higher accuracy with respect to the FFT, quality-guided, and branch-cut methods. Moreover, for the fringe projection case, the MCNF method and our proposal reach an identical unwrapped solution (they have a difference of at the most 3×10^{-14} rad, Fig. 4.5(eg)). In Fig. 4.5, the error graphs (ec), (eh), and (em) exhibit the distortion effect inherent in the FFT method. The quality-guided method returns

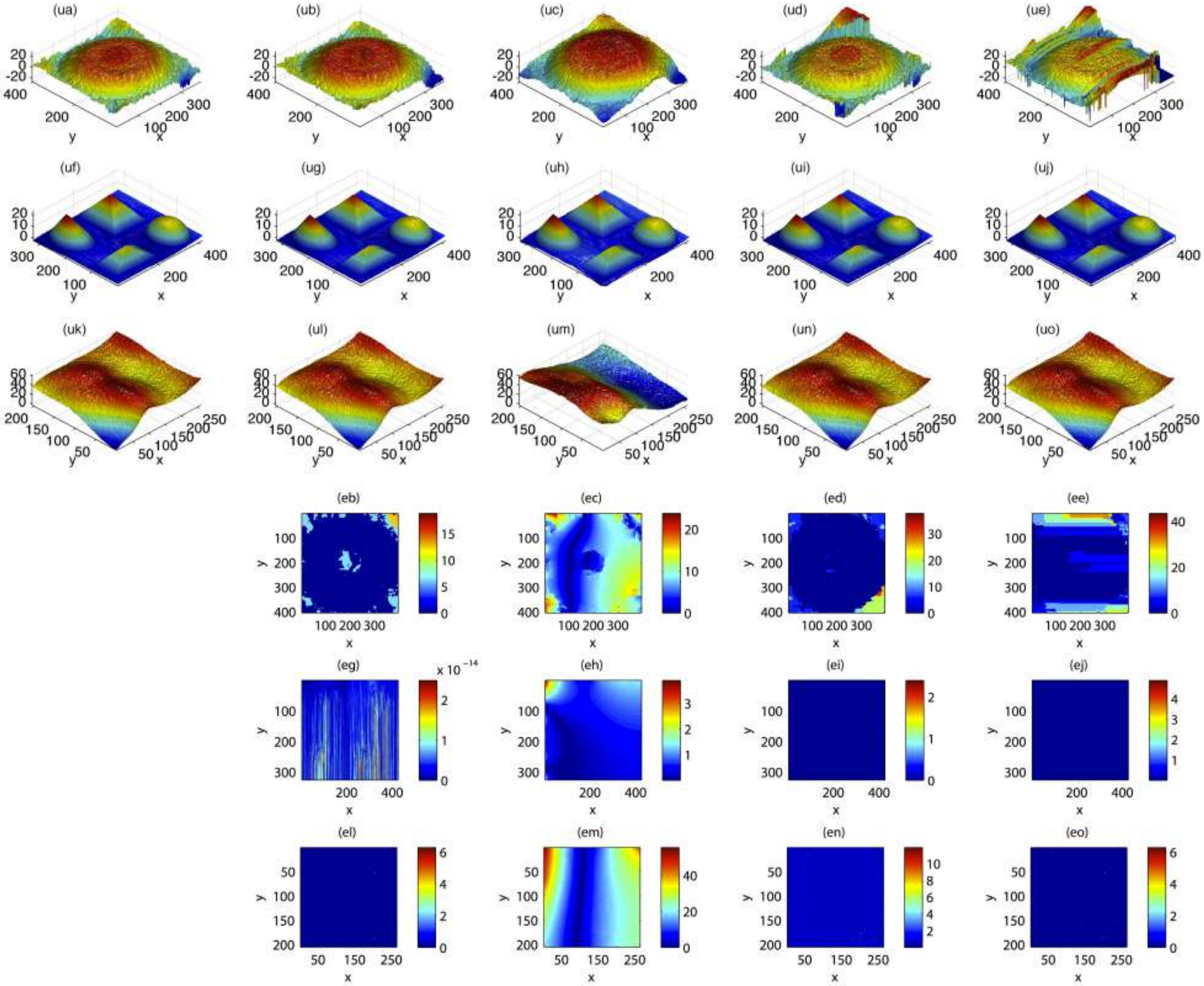


Figure 4.5: *(1st row)-(3rd row) The respective unwrapping phase computed by: (1st column) the MCNF method, (2nd column) the proposed rounding-least-squares algorithm, (3rd column) the FFT method, (4th column) the quality-guided method, and (5th column) the branch-cut method. (4th row)-(6th row) The respective absolute error where the solutions by the MCNF method are used as a baseline.*

an acceptable result if dislocation defects are not important. Finally, for the noisy wrapped map shown in Fig. 4.4(a), the unwrapped solution by the branch-cut method, Fig. 4.5(ue), presents error by divergent paths as it is verified in Fig. 4.5(ee).

Fig. 4.6 shows the computer time required in these experiments. Similar to the computer simulation in the previous subsection: our proposed algorithm and the FFT method require the least amount of computer time. More computer resources are demanded by the MCNF and the branch-cuth methods. The most computationally intensive is the quality-guided method. It is worth mentioning that, for the processed experimental data, our proposal and the FFT method require computer time of less than one second while the other tested algorithms demand from a few seconds to several minutes.

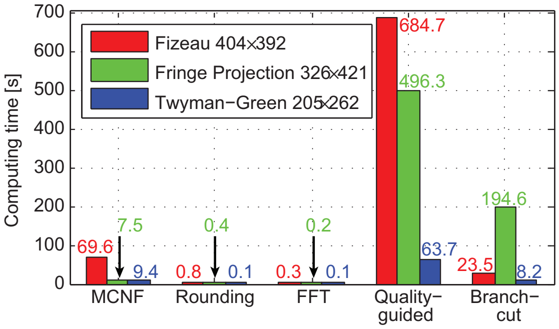


Figure 4.6: *Computing time required to obtain the results shown in Fig. 4.5. The hardware used was a 2.5 GHz laptop.*

4.5 | Discussion

Since noise is very damaging in phase-unwrapping, making it unstable and time consuming, a pre-processing stage dedicated to filter the wrapped phase map is a pertinent strategy [119, 126]. However, the filtering procedure has its own issues such as the filter tuning, the smearing of regions with dense fringes, and the trend to distort the phase distribution. In this context, the proposed algorithm is more convenient because advanced filtering procedures are not required and the phase distortion is avoided. The proposal is based on a simpler structure reaching a high robustness from a low computational effort.

Another issue in phase-unwrapping is the breaking of the fringes. Such damages may be due to, for example, holes and discontinuities of the phase distribution. Because the proposed algorithm assumes continuity of the phase distribution, the proposal cannot address the breaking of the fringes generated by true discontinuities of the phase distribution. We believe that this algorithm can overcome this situation by including a weighting method such as the regularized least-squares approach. However, this possibility is left as future work. This work is focused on the development of the alternative rounding-least-squares approach for phase unwrapping and highlight its good properties such as simplicity, accuracy, robustness and computational efficiency.

4.6 | Conclusion

A user-free and computationally efficient global phase-unwrapping algorithm based on a simple rounding-least-squares approach was proposed. Unlike the conventional phase-unwrapping methods which integrate the gradient of the unwrapped phase, the proposal operates over the gradient of the phase jumps component. Because of this, the proposed algorithm is less sensitive to residue-spreading and over-smoothing effects.

The performance of the proposed algorithm was compared, in both simulated and experimental cases, with the Minimum Cost Network Flow (MCNF), the phase-unwrapping by fast Fourier transform (FFT), the quality-guided, and the branch-cut methods. The results have shown that the higher accuracy level is obtained by the MCNF method and the lower computing time is required by the FFT method. In this context, our proposed algorithm exhibits a good performance in the sense that a high accuracy is reached (similar to the MCNF method) by demanding low computational resources (similar to the FFT method).

Due to the simplicity and robustness of this proposal, its computational performance, and that it does not require user intervention, the suggested phase-unwrapping scheme could find use in automatic real-time applications.

Chapter 5

Phase-shifting by lateral displacement of the illumination source

Real understanding of a thing comes from taking it apart oneself, not reading about it in a book or hearing about it in a classroom.

Robert B. Laughlin

Contents

The heart of any Phase-Shifting Interferometry system is the phase shifter device. In general, the phase shifter devices are expensive and/or complex to implement. The advances on fringe-processing algorithms can be exploited to relieve the high requirements on such devices. Therefore, it is possible to focus in the design of simpler optical setups.

This chapter discuss the idea of to change the spatial position of a point laser source to induce phase shifts. The principles are given based on the Twyman-Green interferometer. Thus, a simple and inexpensive phase-shifting interferometer is produced. The experimental and theoretical results show the feasibility of this unused phase-shifting technique.

5.1 | Introduction

Phase-Shifting Interferometry (PSI) has become a versatile powerful tool used widely in a variety of precise optical measurement applications [1]. In contrast with the spatial carrier based methods, PSI is especially attractive because it can ensure major sensitivity, higher accuracy, and maximum spatial resolution of the calculated phase [3].

The principle of phase-shifting methods is very simple. In general, it consists on two steps [1]:

1. A fringe-pattern is recorded for each phase step induced on the reference wavefront, and
2. The acquired fringe-patterns are processed to obtain the encoded phase distribution (in wrapped format).

It is worth mentioning that, in the first step, it is not need that the phase shift to be calibrated neither controlled because the availability of generalized phase-shifting algorithms such that the presented in the chapter 3.

Thus, it is precise to focus on simplify the experimental issues of the optical setup. Particularly, in the techniques for induce the phase shifts.

The piezo-electric device is the most employed element to induce phase shifts in interferometrical test [**?**, 26,82]. However, this device is very expensive and requires peripheral equipment such as amplifiers and control systems. Some other alternatives are the use of liquid crystal phase modulators [151], diffraction gratings [152], Bragg cells [153], polarization elements [154], the use of multiple wavelengths [155] or wavelength-shifting [8, 156], amplitude modulators [20], among others. Unfortunately, these methods require complex equipment or controlled environment conditions. A different approach is to exploit the mechanical vibrations from the operation environment [157]; however, the data processing is computationally exhaustive.

This chapter presents a simple and inexpensive phase-shifting technique. This technique is based on the phase shift induced when the illumination angle in a plane-parallel interferometer is varied [**?**]. This effect is exploited in the Twymann-Green interferometer where a point laser source is moved perpendicularly to the optical axis. It is worth mentioning that the proposed optical setup is different to induce a tilted wavefront (i.e., to generate an spatial carrier) neither a shearing interferometer.

Notation

In the following, the vectorial notation is employed. The vectors will be indicated by bold symbols, for example $\boldsymbol{d}$. The magnitude is measured by the Euclidean norm and it is indicated by no bold symbols, for example $d = \|\boldsymbol{d}\|$ with $\|\cdot\|$ denoting the Euclidean norm. Unitary vectors (including the unitary coordinate vectors) are hatted, for example $\hat{\boldsymbol{d}} = \boldsymbol{d}/\|\boldsymbol{d}\|$, $\hat{\boldsymbol{x}}$, $\hat{\boldsymbol{y}}$, $\hat{\boldsymbol{z}}$, etc. The vectorial quantities are described from its magnitude and direction, for example $\boldsymbol{d} = d\hat{\boldsymbol{d}}$. Finally, the scalar and vectorial products between two vectors, say $\boldsymbol{d}$ and $\boldsymbol{p}$, will be indicated as $\boldsymbol{d} \cdot \boldsymbol{p}$ and $\boldsymbol{d} \times \boldsymbol{p}$ respectively.

5.2 | Theoretical principles

5.2.1 | Lateral displacement on the lens focal plane of the point source

A laser point source in the configuration shown in Fig. 5.1(a) is usually employed as the illumination system in the Twyman-Green interferometer. Now, it is supposed that the point source is displaced laterally from the optical axis a quantity given by $\boldsymbol{d} = d\hat{\boldsymbol{d}}$, where $\hat{\boldsymbol{d}}$ is a unitary vector perpendicular to the optical axis as shows Fig. 5.1(b). The spherical wave emerging from the point source reach the input plane of the lens L is given by

$$A(\boldsymbol{p}, \boldsymbol{d}) = \frac{A_0}{r} \exp[ikr],$$

where $\boldsymbol{p} = (x, y)$ is a spatial variable with Cartesian coordinates x and y, $\boldsymbol{f}$ is parallel to the optical axis with f be the focal length of the lens L, $r = ||\boldsymbol{f} + \boldsymbol{p} - \boldsymbol{d}||$ is the length between the point source and the point $\boldsymbol{p}$ on the input plane of L, A_0 is an real-valued amplitude, $k = 2\pi/\lambda$ is the wave number, λ is the wavelength, and i denotes the imaginary unit ($i^2 = -1$).

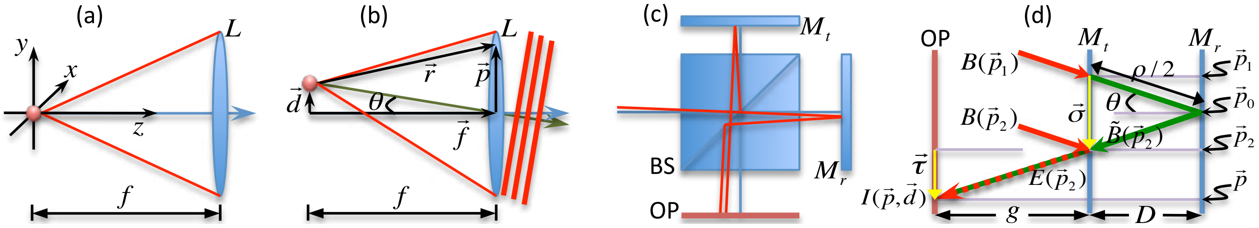


Figure 5.1: *(a) Collimation system with a centered point source. (b) Tilted plane wavefront obtained by the lateral displacement of the source. (c) Twyman-Green interferometer illuminated with a tilted plane wavefront. (d) Alternative representation of Fig. 5.1(c).*

If experimentally is satisfied that $f \gg \|\boldsymbol{p} - \boldsymbol{d}\|$ (focal length of the lens is longer than its diameter and the displacements d), then is valid second order the approximation of r (truncated Taylor series at the second order term); i.e., $r \approx f + (p^2 + d^2 - 2\boldsymbol{p} \cdot \boldsymbol{d})/2f$. Substituting this approximation in (5.2.1) the *paraxial approximation* of the complex field $A(\boldsymbol{p}, \boldsymbol{d})$ is obtained as

$$A(\boldsymbol{p}, \boldsymbol{d}) \approx B_0 \exp\left[i\frac{k}{2f}\left(p^2 + d^2 - 2\boldsymbol{d} \cdot \boldsymbol{p}\right)\right], \tag{5.1}$$

where $B_0 = (A_0/f) \exp[ikf]$ is a complex amplitude. Notice that the first order approximation $r \approx f$ for the amplitude $A_0/r \approx A_0/f$ in (5.1) was used[1].

When the wavefront $A(\boldsymbol{p}, \boldsymbol{d})$ is propagated inside the lens L and reach the output plane of L, the complex amplitude $A(\boldsymbol{p}, \boldsymbol{d})$ losses its quadratic phase term $\exp[ikp^2/(2f)]$. Thus, in the output plane of L is obtained the tilted plane wavefront

$$B(\boldsymbol{p}, \boldsymbol{d}) = B_0 \exp\left[i\frac{k}{2f}\left(d^2 - 2\boldsymbol{d} \cdot \boldsymbol{p}\right)\right]. \tag{5.2}$$

To determine the tilt of the plane wavefront (5.2), we require a vector $\boldsymbol{h}$ perpendicular to such a wavefront. This vector can be defined as the vectorial product of two linearly independent vectors $\boldsymbol{u}$, $\boldsymbol{v}$ on

[1] Because the wavelength in the visible range, the phase of a wave changes faster that its amplitude [158]. Thus, the paraxial approximation consists on a second order approximation for the phase and a first order approximation for the amplitude.

the wavefront. By simplicity, $\boldsymbol{d} = \hat{\boldsymbol{y}}$ is considered. The first vector can be defined for the case where $\boldsymbol{p} = \hat{\boldsymbol{x}}$. Then $\boldsymbol{p}$ and $\boldsymbol{d}$ are perpendiculars and will be a null phase contribution at $\boldsymbol{p}$. Thus, the vector $\boldsymbol{u}$ is defined as

$$\boldsymbol{u} = \hat{\boldsymbol{x}}.$$

Now, the case where $\boldsymbol{p} = \hat{\boldsymbol{y}}$ is considered. Then $\boldsymbol{p}$ and $\boldsymbol{d}$ are parallel and will be an phase contribution of d/f in the direction of $\hat{\boldsymbol{z}}$. Thus, the vector $\boldsymbol{v}$ is defined as

$$\boldsymbol{v} = \hat{\boldsymbol{y}} + \frac{d}{f}\hat{\boldsymbol{z}}.$$

With the vectors $\boldsymbol{u}$ and $\boldsymbol{v}$ the perpendicular vector $\boldsymbol{h}$ is constructed as

$$\boldsymbol{h} = \boldsymbol{u} \times \boldsymbol{v} = -\frac{d}{f}\hat{\boldsymbol{y}} + \hat{\boldsymbol{z}}.$$

With this, we conclude that the plane wavefront (5.2) propagates in the direction of $\boldsymbol{h}$ having a tilt of

$$\tan\theta = -d/f, \tag{5.3}$$

with respect to the optical axis. Notice that, this result agrees with the one directly obtained from geometrical arguments as it is shown in Fig. 5.1(b).

5.2.2 | Nonzero angle illumination in the Twyman-Green interferometer

Now, the Twyman-Green interferometer is illuminated with the wavefront $B(\boldsymbol{p},\boldsymbol{d})$ as shown in 5.1(c) or, equivalently, 5.1(d). It is not loss of generality to assume that the wavefront $B(\boldsymbol{p},\boldsymbol{d})$ on the output plane of the lens L is the same at the plane of the mirror M_r because the additional constant phase by the propagation from the output plane of L to the plane of M_r can be omitted.

Accordingly, at the point $\boldsymbol{p}$ on the observation plane OP, we have the superposition of the beams $B(\boldsymbol{p}_2,\boldsymbol{d})\exp[i\phi(\boldsymbol{p}_2)]$ and $\tilde{B}(\boldsymbol{p}_2,\boldsymbol{d})$ coming by reflection from the test and reference mirrors, respectively, as shows Fig. 5.1(d). Notice that the wavefront coming from the test mirror to the plane OP is the incident wavefront $B(\boldsymbol{p}_2,\boldsymbol{d})$ plus an additional phase term $\exp[i\phi(\boldsymbol{p}_2)]$ associated with the aberrations of the mirror surface.

The intensity distribution on the observation plane can be described as

$$I(\boldsymbol{p},\boldsymbol{d}) = |B(\boldsymbol{p}_2,\boldsymbol{d})\exp[i\phi(\boldsymbol{p}_2)] + \tilde{B}(\boldsymbol{p}_2,\boldsymbol{d})|^2, \tag{5.4}$$

with $|\cdot|$ denoting the module. It is worth mentioning that the beam $\tilde{B}(\boldsymbol{p}_2,\boldsymbol{d})$ is the beam $B(\boldsymbol{p}_1,\boldsymbol{d})$ (reflected from the reference mirror) plus an additional phase term[2]:

$$\rho = 2D\left[1 + \left(\frac{d}{f}\right)^2\right]^{1/2}, \tag{5.5}$$

and a translation:

$$\boldsymbol{\sigma} = \boldsymbol{p}_2 - \boldsymbol{p}_1 = \sigma\hat{\boldsymbol{d}}, \quad \text{with} \quad \sigma = -2D\frac{d}{f}; \tag{5.6}$$

this is,

$$\tilde{B}(\boldsymbol{p}_2,\boldsymbol{d}) = B(\boldsymbol{p}_2 - \boldsymbol{\sigma},\boldsymbol{d})\exp[ik\rho]. \tag{5.7}$$

[2] The additional phase term ρ corresponds to the optical path traveled from $\boldsymbol{p}_1$ on the mirror M_t, reflects on the mirror M_r and reaches again the plane of the mirror M_t but at the point $\boldsymbol{p}_2$.

From the equation (5.2), we have that

$$\begin{aligned} B(\boldsymbol{p}_2 - \boldsymbol{\sigma}, \boldsymbol{d}) &= B_0 \exp\left[i\frac{k}{2f}\left[d^2 - 2\boldsymbol{d}\cdot(\boldsymbol{p}_2 + \boldsymbol{\sigma})\right]\right] \\ &= B_0 \exp\left[i\frac{k}{2f}(d^2 - 2\boldsymbol{d}\cdot\boldsymbol{p}_2 + 2d\sigma)\right] \\ &= B(\boldsymbol{p}_2, \boldsymbol{d}) \exp\left[i\frac{k}{f}d\sigma\right]. \end{aligned}$$

Then, the equation (5.7) can be restated as

$$\tilde{B}(\boldsymbol{p}_2, \boldsymbol{d}) = B(\boldsymbol{p}_2, \boldsymbol{d}) \exp\left[i\frac{k}{f}d\sigma\right] \exp[ik\rho]. \tag{5.8}$$

It is necessary to define the translation vector:

$$\boldsymbol{\tau}(\boldsymbol{d}) = \boldsymbol{p} - \boldsymbol{p}_2 = \tau(d)\hat{\boldsymbol{d}}, \quad \text{with} \quad \tau(d) = -g\frac{d}{f}, \tag{5.9}$$

to consider the fact that the interference occurring at the point $\boldsymbol{p}_2$ on the plane of M_t is detected at the point $\boldsymbol{p}$ on the observation plane OP.

Considering the equation (5.8) and (5.9), we have that the irradiance described by (5.4) can be rewritten as

$$\begin{aligned} I(\boldsymbol{p}, \boldsymbol{d}) &= |B(\boldsymbol{p}_2, \boldsymbol{d}) \exp[i\phi(\boldsymbol{p}_2)] + \tilde{B}(\boldsymbol{p}_2, \boldsymbol{d})|^2 \\ &= |B(\boldsymbol{p}_2, \boldsymbol{d}) \exp[i\phi(\boldsymbol{p}_2)] + B(\boldsymbol{p}_2, \boldsymbol{d}) \exp\left[i\frac{k}{f}d\sigma\right] \exp[ik\rho]|^2 \\ &= |B(\boldsymbol{p}_2, \boldsymbol{d})|^2 + |B(\boldsymbol{p}_2, \boldsymbol{d})|^2 + 2|B(\boldsymbol{p}_2, \boldsymbol{d})|^2 \cos\left[\phi(\boldsymbol{p}_2) - \frac{k}{f}d\sigma - k\rho\right] \\ &= 2a + 2a\cos\left[\phi(\boldsymbol{p} - \boldsymbol{\tau}(\boldsymbol{d})) + \delta(d)\right], \end{aligned} \tag{5.10}$$

where $a = |B(\boldsymbol{p}_2, \boldsymbol{d})|^2 = A_0^2/f^2$, and $\delta(d) = -k(\sigma d/f + \rho)$ is the phase shift.

Substituting the variables ρ, Eq. (5.5), and σ, Eq. (5.6), in $\delta(d)$, we have

$$\begin{aligned} \delta(d) &= -k(\rho + \sigma d/f) \\ &= -2Dk\left\{\left[1 + \left(\frac{d}{f}\right)^2\right]^{1/2} - \left(\frac{d}{f}\right)^2\right\}. \end{aligned}$$

Because $d^2/f^2 \ll 1$, it is valid the approximation:

$$\left[1 + \left(\frac{d}{f}\right)^2\right]^{1/2} \approx 1 + \frac{1}{2}\left(\frac{d}{f}\right)^2.$$

Thus, the phase shift $\delta(d)$ can be approximated as

$$\begin{aligned} \delta(d) &\approx -2Dk\left[1 + \frac{1}{2}\left(\frac{d}{f}\right)^2 - \left(\frac{d}{f}\right)^2\right] \\ &= Dk\left(\frac{d}{f}\right)^2 - 2Dk. \end{aligned}$$

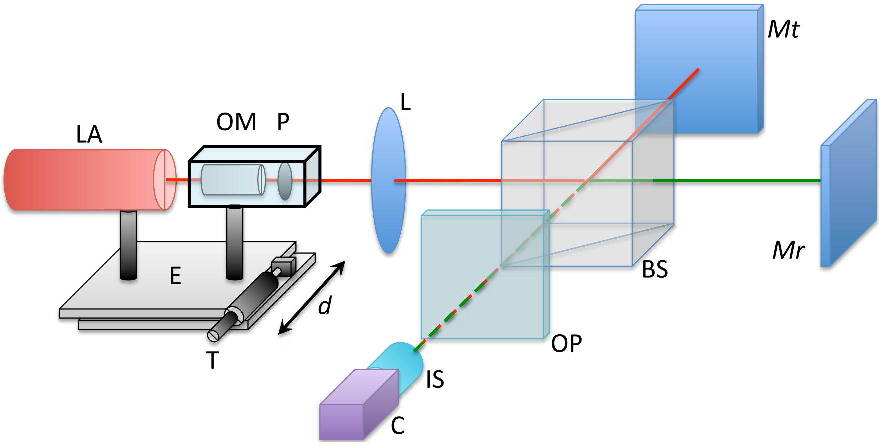


Figure 5.2: *Experimental setup for phase-shifting interferometry by lateral displacement of the laser point source in the Twyman-Green interferometer. The laser source (LA), the microscope objective (OM), and the pinhole (P) are mounted on the manual linear translation stage (E). The lateral displacements d are provided by the micrometer screw (T). The fringe-pattern on the observation screen (OP) are recorded by the camera (C) through the imaging system (SFI).*

Notice that the term $-2Dk$ is a constant phase or *offset* (with respect to the lateral displacement d of the light source) and, therefore, it can be omitted. Thus, we have that the resulting phase shift is

$$\delta(d) \approx Dk\left(\frac{d}{f}\right)^2. \tag{5.11}$$

From Eq. (5.10), we can see that, when the magnitude of $\boldsymbol{d}$ is constant, there are two effects simultaneously in the intensity distribution $I(\boldsymbol{p}, \boldsymbol{d})$:

1. A quadratic phase shift $\delta(d)$, given by (5.11).
2. A linear translation $\boldsymbol{\tau}$ of the fringe-pattern, given by (5.9).

Both effects are outlined in Fig. 5.3.

It is worth mentioning that the sensitivity of the phase shifter can be tuned by the relative distance D between the mirrors M_t and M_r. For example, for a lateral displacement of 2 mm, a collimation lens with focal length of $f = 0.5$ m, and a laser source with wavelength $\lambda = 633$ nm; the phase shift 2π rad can be obtained if the distance D is set to $\delta f^2/(kd^2) = 3.96$ cm. If more phase shift gain is required, a greater distance D is necessary and vice versa.

In the next section, the feasibility of this principle is verified by experimental measurements.

5.3 | Experimental validation

The feasibility of this proposal is experimentally verified as follows. We consider the standard Twyman-Green Interferometer arrangement depicted in Fig. 5.2. The light source used was a He–Ne laser, with wavelength $\lambda = 633$ nm. The laser beam was expanded and filtered by a microscope objective and a pinhole, respectively. The pinhole was located at the focal point of a collimating lens with focal length of $f = 0.5$ m.

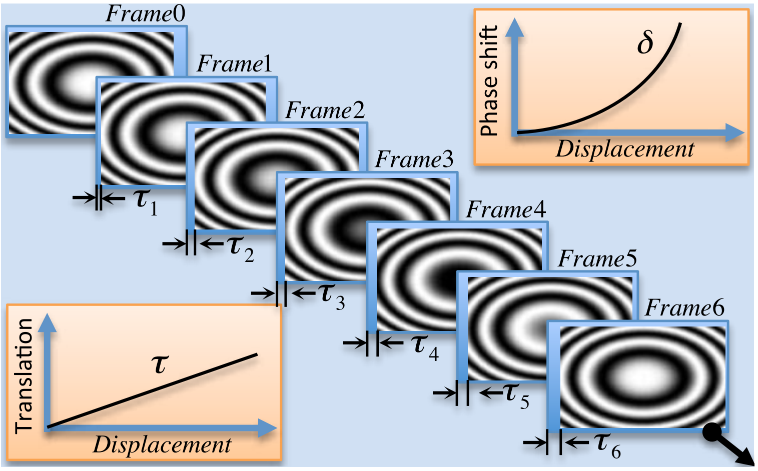


Figure 5.3: *The phase shift δ and fringe-pattern translation τ effects induced by the lateral displacement of the laser point source.*

The collimated wavefront obtained was split by a non-polarizing cube beam splitter. Two beams were produced and they were reflected by the reference M_r and test M_t mirrors. The test mirror surface was deformed in order to obtain a distorted wavefront. The fringe pattern due to the interference of these two reflected beams was observed on a screen on the observation plane OP. The fringe patterns was acquired by a gray-scale 8-bit CCD camera with a resolution of 768 × 1008 pixels.

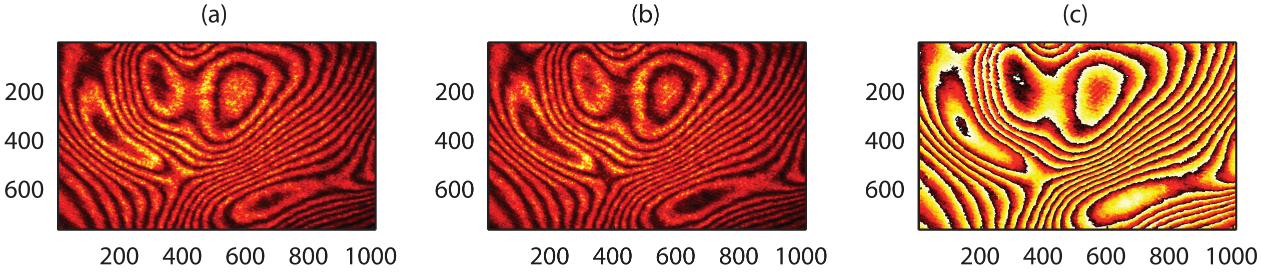


Figure 5.4: *(a) and (b) Two adjacent phase shifted interferograms acquired with the proposed PSI technique. (c) Wrapped phase distribution recovered by processing the two interferograms shown in Figs. 5.4(a) and 5.4(b) with the suggested GPSI algorithm.*

The lateral source displacement was performed by mounting the laser source, the microscope objective and the pinhole on a manual linear translation stage. The displacement resolution reached with this mechanism is of 10 μm. We considered a lateral source displacement of $d = \pm 2$ mm with steps of 100 μm. We choose the relative distance $D = 3.96$ and 7.91 cm between the mirrors in order to obtain a phase shift $\delta(2 \text{ mm}) = 2\pi$ and 4π rad, respectively. The observation plane is placed to the distance $g = 11$ cm from the reference mirror M_r. For each progressive displacement step, a phase shifted interferogram was recorded. Thus, for each value of D, 41 interferograms were acquired. Of these interferograms, we show two adjacent interferograms in Fig. 5.4 as an example.

The interferograms were processed to extract both the phase shift and the wrapped phase distribution by the GPSI algorithm described in §**??** and reported in [71]. The fringe-pattern translation was determined by the method exposed in the appendix A. The calculated phase shift and the image translation are shown in Figs. 5.5(a) and 5.5(b), respectively. The obtained experimental data values presents a mean error of

0.06 rad (with standard deviation of 0.10 rad) for the phase shift, and 1.31 μm (with standard deviation of 11.42 μm) for the image translation. These results are good considering that the displacement is induced manually.

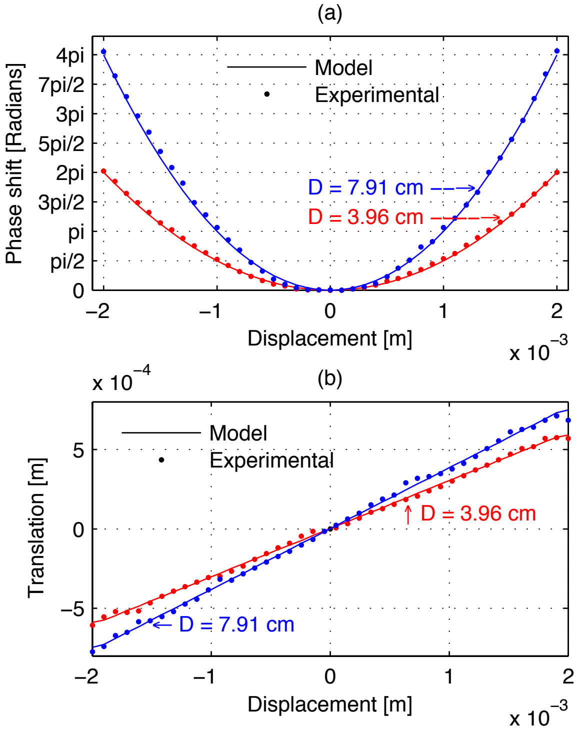


Figure 5.5: *(a) Phase shift and (b) image translation measurements versus lateral source displacement. This experiments correspond to the relative distances between the mirrors of (red) $D = 3.96$ cm, and (blue) $D = 7.91$ cm.*

It is worth mentioning that both the nonlinear phase shift and its deviation from the nominal values are not a problem because an appropriate GPSI algorithm can be implemented. Thereby, only two interferograms with an arbitrary and unknown phase shift are sufficient to wrapped phase extraction. For example, the results obtained with this algorithm when it processes the two interferograms shown in Figs. 5.4(a) and 5.4(b) are: a phase step of 0.898 rad, and the wrapped phase distribution shown in Fig. 5.4(c).

With respect to the image translation issue, it is very small (in the described experiment, $\tau = 37.82$ and 29.92 μm for $D = 7.91$ and 3.96 cm, respectively). In addition, a scaling of this translation is performed by the camera's imaging system. In our particular case, the size of the interferograms was of 3×3.9 cm and the target's size of 2.7×3.5 mm. Thus, the image translation is reduced to 3.4 and 2.7 μm for $D = 7.91$ and 3.96 cm, respectively. But, since the pixel size is 3.5 μm, such translations are not observable. Moreover, for large translations, because the translation is a linear function of the displacement, the numerical correction is very simple and consists of a translation of all the pixels of the interferogram by a certain number of pixels.

5.4 | Conclusion

A simple and inexpensive phase shifter by lateral source displacement to phase-shifting interferometry was proposed. Unlike the conventional PZT techniques, where fine nanometric translations are required, in this novel technique a coarse and miscalibrated translation stage obtained with a micrometric screw is sufficient. The phase shifter sensibility can be tuned by the optical path difference between the interferometer's mirrors.

Some phase shift problems such as the quadratic phase shift, miscalibration, and other unknown possible error sources are overcome by the use of an appropriate Generalized Phase-Shifting Interferometry (GPSI) algorithm to wrapped phase extraction. In this work, the automatic real-time GPSI algorithm by parameter estimation presented in §3.4 was used. The translation of the interferogram image can be easily corrected by a simple pixels shift. Even, the translation effect can be negligible by either setting the relative distance between the mirrors or adjusting the camera's image amplification.

A successful implementation of this technique in the Twyman-Green Interferometer was reported. The experimental results shown that the proposed scheme is a simple and inexpensive alternative to interferometrical phase evaluation. We believe that other interferometric systems such as the Fizeau interferometer (which is very important due to its industrial applications) and the *Electronic Speckle Pattern Interferometry* could incorporates this approach.

Appendix A

Image translation by lateral source displacement

Science tells us we are merely beasts, but we don't feel like that. We feel like angels trapped inside the bodies of beasts, forever craving transcendence.

Vilayanur S. Ramachandran

Contents

The lateral displacement of the laser point source provides two effects in the fringe-patterns: a phase-shifting, and an image translation. The phase-shifting is need for wrapped phase extraction. However, the image translation is an effect that must be characterized for establishment of its influence and, if necessary, compensate it into the computational routines of fringe analysis.

Theoretically, we have deduced the quantity of translation as function of the provided displacement of the illumination source. However, for validation purposes, such translation must be measured experimentally. In this appendix a method to measure the image translation induced by lateral displacement of the illumination source is presented.

A.1 | Experimental setup

When the laser point source in a Twymann-Green interferometer is laterally displaced, both a phase shift and a fringe-pattern translation are induced.

To measure the image translation, in the test arm is placed a flat mirror M_1 with the half area covered as shows Fig. A.1(a). Thus, ideally an binary interferogram *brig-dark* as shows Fig. A.1(b) is obtained. The region corresponding with the covered mirror will be dark, while the another part will be brig[1].

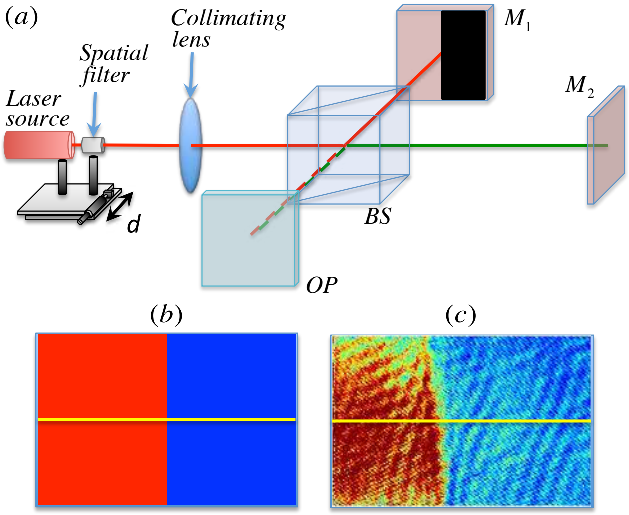


Figure A.1: *(a) Experimental setup to measure the image translation when the illumination source is displaced laterally. (b) Ideal interferogram. (c) Experimental interferogram.*

In the case of ideal interferograms, say Fig. A.1(b), the translation measurement is trivial: it is sufficient to process a single row of the image to find the point where the maximum-minimum (max-min) intensity occurs [e.g. the highlighted data by the horizontal yellow line in Fig. A.1(b)]. Unfortunately, this method is not valid when experimental interferograms as the shown in Fig. A.1(c) are processed.

A.2 | Image processing

A.2.1 | Image row average

For processing experimental interferograms, it is convenient to process the whole frame in order to reduce the intensity variations due to random noise, diffraction fringes, and other error sources. For example, Fig. A.2(c1) shows the intensity profile along the yellow line in the interferogram shown in Fig. A.1(c).

An efficient way to attenuate the noise effects is to exploit the fact that the transition max-min intensity region in the frame is a straight line (near horizontal). Thus, all rows of the frame can be fitted to a single curve by taking its average [a least-squares fitting to a zero-order polynomial (a constant)]. Fig. A.2(c2)

[1] This is not valid for a phase shift of π rad (which leads to destructive interference). However, this is a particular (and extreme) case restricted to a single point in the interval $[0, \pi]$ rad. Moreover, experimentally such a case is unlikely for the whole interferogram because the non-ideal flatness of the mirror and its inexact alignment.

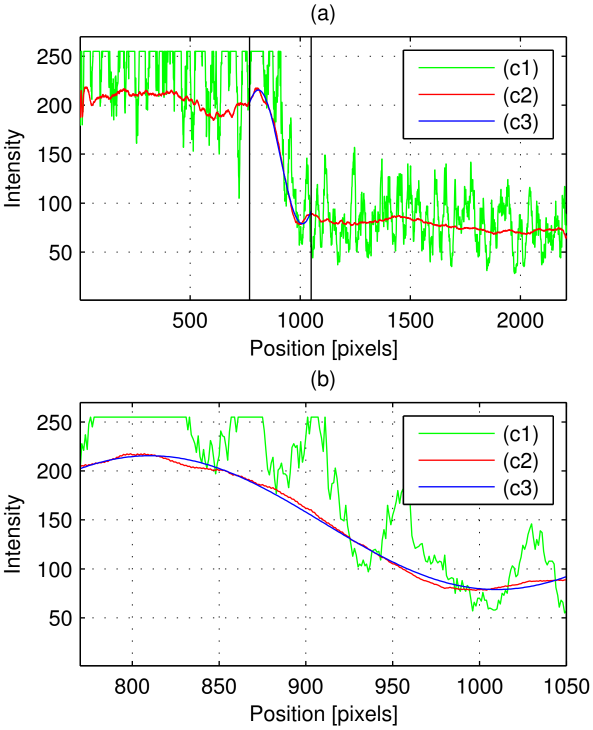


Figure A.2: *(a) Intensity profile of the interferogram shown in Fig. A.1(c). (b) Zoom of (a) in the interval indicated by the horizontal lines (interval from 770 to 1050 pixels). (c1) Intensity of the central row only from the interferogram. (c2) Intensity average from the whole frame. (c3) Fitted sinusoidal curve to the average intensity (c2) in the marked interval.*

shows the result of this procedure applied to the interferogram shown in Fig. A.1(c). It is evident that a significant noise reduction was achieved by this simple procedure.

A.2.2 | Transition point

Now, we take advantage of the previous knowledge about the point where the transition occurs. From §5.2 we known that the transition point must change as the translation $\tau(d)$ given by (5.9). For convenience purposes, we rewrite this equation here:

$$\boldsymbol{\tau}(\boldsymbol{d}) = \boldsymbol{p} - \boldsymbol{p}_2 = \tau(d)\hat{\boldsymbol{d}}, \quad \text{with} \quad \tau(d) = -g\frac{d}{f}. \tag{A.1}$$

This translation is linear with respect to the lateral displacement d of the light source. Thus, taking as reference an interferogram at $d = 0$, we can calibrate the transition point. With this information, it is sufficient to analyze the intensity curve in a neighborhood around the predicted transition point for all subsequent value of d.

From Fig. A.2(a) we can see that the transition point is between a ridge and a valley. Then, we consider a neighborhood $\tau(d) \pm v$ around the transition point $\tau(d)$ (for an appropriate value v such that the neighborhood includes a ridge and a valley). Thus the transition curve can be approximated to a sinusoidal curve as shows Fig. A.2(c3). The procedure to fit a sinusoidal curve and how to measure the actual translation from the fitted curve is shown below.

A.2.3 | Approximation of the transition curve by a sinusoidal curve

The problem of parameter estimation of a sinusoidal model is, in general, nonlinear. To solve this kind of problems, there are several numerical methods such as the Newton, Gauss-Newton, and the Levenber-Marquardt methods. However, these methods converge (if they do) to a local optimum. Moreover, they require a good initial guess for computing a better solution by an iterative proceeding. Three disadvantages can be identified:

1. An initial guess is required,
2. The optimum is local, even cannot be found (when the iterative procedure does not converge), and
3. The procedure is computationally exhaustive.

To overcome the above drawbacks, we suggest the following method. Instead of to estimate simultaneously the all parameters of a sinusoidal function (a nonlinear problem), such parameters are estimated in sequence. Thus, the original nonlinear problem is reduced to a cascade of linear problems. This method is similar to the phase extraction presented in the chapter 3. The main advantages of this approach are:

- Initial guess is not necessary,
- Global convergence,
- Efficient computational algorithms (iterative loops are not used).

Therefore, simple and fast computer routines are obtained appropriate to real-time applications. In the following, the theoretical principles are described.

A.3 | Theoretical principles

Let $\{x_i, y_i\}$ $(i = 1, 2, \cdots, N)$, with $x_1 < x_2 < \cdots < x_N$, be an experimental data set obtained by sampling a process of the form

$$y_i = \bar{a} - \bar{b}\sin(\bar{\omega}x_i), \tag{A.2}$$

where the constants $\bar{a}$, $\bar{b}$, and $\bar{\omega}$ are the parameters of the model. We proceed to approximate the given data values y_i to a sinusoidal function, defined for x in the interval[2] $[-1, 1]$, as

$$f(x) = a - b\sin(\omega x - \alpha), \tag{A.3}$$

where a, b, ω, and α are constants. We can to estimate the value of a as an average (a least-squares fitting to a constant) of the data set y_i as

$$a = \frac{1}{N}\sum_{i=1}^{N} y_i. \tag{A.4}$$

[2] The interval $[-1, 1]$ is chosen for convenience. For this interval, the position $\tau(d)$ corresponds with $x = 0$ and $\tau(d) \pm v$ corresponds with $x = \pm 1$.

By subtracting a from the experimental data y_i and computing the square we have

$$(y_i - a)^2 \approx \bar{b}^2 \sin^2(2\pi\bar{\omega}x_i) = \frac{1}{2}\bar{b}^2 + \frac{1}{2}\cos(2\bar{\omega}x_i).$$

Thus, the quantity $\bar{b}^2/2$ can be estimated on a similar form than a, namely

$$\frac{1}{2}b^2 = \frac{1}{N}\sum_{i=1}^{N}(y_i - a)^2, \tag{A.5}$$

where the desired parameter is obtained by solving the above equation for b (multiplying by 2 and computing the square root). Now, with the estimated parameters a and b, we proceed to the normalization of the experimental data by

$$z_i = \text{sat}\left[\frac{y_i - a}{b}\right] \approx \sin(\bar{\omega}x_i), \text{ para } b \neq 0,$$

where sat[·] denotes the saturation function defined by the equation (1.38):

$$\text{sat}\left(\tilde{I}_k\right) = \begin{cases} 1 & \text{if} \quad \tilde{I}_k > 1, \\ \tilde{I}_k & \text{if} \quad |\tilde{I}_k| \leq 1, \\ -1 & \text{if} \quad \tilde{I}_k < -1. \end{cases}$$

with $|\cdot|$ denoting the absolute value.

To obtain the parameters $\bar{\omega}$ of the line $2\pi\omega\bar{x}_i$ we compute the inverse sine of z as

$$r_i = \arcsin z_i \approx \bar{\omega}x_i.$$

The resulting curve r_i is a triangular function corresponding to a line wrapped in the interval $[-\pi/2, \pi/2]$ by a sinusoidal function.

A property of the triangular function r_i is that its derivative is a constant but the sign changes at the points $2\pi\bar{\omega}x_i = (2n+1)\pi/2$ with $n = 0, \pm1, \pm1, \cdots$. Assuming that the line is monotonically increasing, the sign of the function can be ignored[3]. The derivative of r_i is numerically computed as

$$r_i' \approx \left|\frac{r_{i+1} - r_i}{x_{i+1} - x_i}\right|, \quad i = 1, 2, \cdots, N-1.$$

where $|\cdot|$ denotes the absolute value. It is worth mentioning that the differentiation is an *ill-posed problem* (therefore it is unstable). However, we can "restore stability" by considering that the second derivative r_i'' is zero. In this way, we obtain the parameter ω as

$$\omega = \frac{1}{N-1}\sum_{i=1}^{N-1} r_i'. \tag{A.6}$$

Now we have estimated the parameter ω, it can be computed the parameter α by writing the equation (A.3) as

$$\begin{aligned} f(x) &= a - b\sin(\omega x)\cos\alpha + b\cos(\omega x)\sin\alpha \\ &= a - \xi\sin(\omega x) + \zeta\cos(\omega x), \end{aligned} \tag{A.7}$$

[3] If it is expected that the line is monotonically decreasing, then the absolute value of the derivative must be multiplied by -1.

with $\xi = b\cos\alpha$, $\zeta = b\sin\alpha$ and where the trigonometric identity $\sin(x+y) = \sin x\cos y - \cos x\sin y$ was applied. The equation (A.7) can be solved for a, ξ and ζ in the least-squares sense as[4]

$$\begin{bmatrix} a \\ \xi \\ \zeta \end{bmatrix} = \mathcal{A}^{\dagger} Y, \tag{A.8}$$

where

$$\mathcal{A} = \begin{bmatrix} 1 & -\sin(\omega x_1) & \cos(\omega x_1) \\ 1 & -\sin(\omega x_2) & \cos(\omega x_2) \\ \vdots & \vdots & \vdots \\ 1 & -\sin(\omega x_N) & \cos(\omega x_N) \end{bmatrix},$$

and

$$Y = \begin{bmatrix} y_1 & y_2 & \cdots & y_N \end{bmatrix}^T.$$

Finally, with the computed vector (A.8), the parameter α is retrieved by

$$\alpha = \arctan(\zeta/\xi). \tag{A.9}$$

The above described procedure is implemented in the MATLAB software as the function `fitSin()`. For illustration purposes, Fig. A.3(Left) shows the resulting computer code[5]. The use of this computer function is illustrated by the script shown in Fig. A.3(Right). This script generates the plots shown in Fig. A.2.

A.4 | Relationship between phase shift and transition point

Now, just need to show that the parameter α is linked to the difference between the predicted transition point $\tau(d)$ ($x = 0$) and the actual transition one. According to the theoretical transition point $\tau(d)$, the equation (A.2) must be centered in the considered neighborhood, i.e., at $\alpha = 0$. However, experimentally it is expected that the fitted sinusoidal function is not in phase; i.e., the transition point, instead of being at $x = 0$, is at $x = x_0$. Considering the change of coordinates $\tilde{x} = x - x_0$ and substituting in (A.2), we have

$$y_i = \bar{a} - \bar{b}\sin(\bar{\omega}\tilde{x}_i) = \bar{a} - \bar{b}\sin(\bar{\omega}x_i - \bar{\omega}x_0),$$

where it was possible to include the parameter α in (A.3) by the relation with the actual transition point x_0:

$$x_0 = \alpha/\bar{\omega} \approx \alpha/\omega. \tag{A.10}$$

With this we conclude that the difference between the theoretical transition point and the experimental one is x_0, and this quantity is computed from the estimated parameters ω and α in (A.3).

To illustrate the whole operation of this image translation measurement method, an example is presented showing the intermediate steps from the beginning until the determination of the transition point x_0.

[4] We can consider the *normalized* data values z_i (they were obtained after of two estimations) and then to solve the most simple equation $g(x) = \cos(\omega x)\sin\alpha - \sin(\omega x)\cos\alpha$. However, it is more convenient to consider the *original data values* y_i and solve the equation (A.7). This choice increases slightly the computational cost but the accuracy is improved because some error sources due to the approximation are avoided.

[5] This is a prototype computer code. It is not optimized and may not be appropriate for end-user applications.

```
function [w alp fit] = fitSin(y)
% Fitting the function f(x) = a - b*sin(wx + alp)
% by estimation of the parameters a, b, w and alp.
% Input: intensity curve data.
% Return: w, alpha and the fitted function

ny = numel(y);
x = 2*(0:ny-1)'/(ny-1) - 1;

a = sum(y)/ny;            % Estimation of a
temp = (y - a).^2;
b = sqrt(2*sum(temp)/ny);% Estimation of b

% Data normalization
z = (y - a)/b;
for k=1:ny
    if z(k) > 1;        z(k) = 1;
    elseif z(k) < -1;   z(k) = -1;
    end
end

r = asin(z);
dif = abs(r(2:end) - r(1:end-1));
dif = dif/(x(2)-x(1));
w = sum(dif)/(ny-1);      % Estimation of w

AA  = [ones(ny,1) -sin(w*x) cos(w*x)];
Th  = (AA'*AA)\AA'*y';    % Estimation of a,xi,zeta
fit = AA*Th;              % Fitted function
alp = atan2(Th(3),Th(2));% Estimation of alpha

end
```

```
tau = 910; % Value obtained by manual calibration
v   = 140; % Value obtained by manual calibration

[m n] = size(I0);       x = 1:n;

y = sum(I0,1)/m;          %Intensity curve

% Data in the invertal [tau-v,tau+v]
x_i = x(tau+(-v:v));    y_i = y(tau+(-v:v));

[w alp fit] = fitSin(y_i);%Fitting a sine functic

% Data to mark with horizontal lines
% the data inverval
x_i1 = (tau-v)*[1 1];
x_i2 = (tau+v)*[1 1];
y_i1 = [1 270];

% Plots
subplot(1,2,1);plot(x,I0(m/2,:),'-g',x,y,'-r',...
    x_i,fit,'-b',x_i1,y_i1,'-k',x_i2,y_i1,'-k')
axis([1 n 1 270]);grid on
title('(a)');xlabel('PosiciÛn en pixeles')
ylabel('Nivel de intensidad')
legend('(c1)','(c2)','(c3)')

subplot(1,2,2);plot(x,I0(m/2,:),'-g',x,y,'-r',...
    x_i,fit,'-b',x_i1,y_i1,'-k',x_i2,y_i1,'-k')
axis([tau-v tau+v 1 270]);grid on
title('(b)');xlabel('PosiciÛn en pixeles')
ylabel('Nivel de intensidad')
legend('(c1)','(c2)','(c3)')
```

Figure A.3: *(Left) Proposed method to fit a sinusoidal curve implemented in MATLAB software by the function* `fitSin()`*. (Right) Script to test the function* `fitSin()` *to generate the plots shown in Fig. A.2. In this script,* `I0` *is a matrix which contains the interferogram in gray-scale to be processed.*

Example 10 (Sinusoidal curve fitting)**.** *We consider the experimental data values y_i shown in Fig. A.2 where $i \in [\tau - v, \tau + v]$, with $\tau = 905$ and $v = 140$, defined on the interval $[-1, 1]$. Fig. A.4(a) shows the data y_i and the estimated parameter a by applying the equation (A.4). By using the equation (A.5), we obtain the quantity $b^2/2$, Fig. A.4(b), which allows us to estimate the parameter b. With the computed values a and b we compute z_i, Fig. A.4(c), which represents the normalized version of the original data set y_i. The arc sine of the normalized data are shown in Fig. A.4(d). The resulting numerical derivative r_i' and the parameter ω, estimated by applying (A.6), are shown in Fig. A.4(e). Finally, by solving the matrix equation (A.8), the function $f(x)$ (A.7) is fully determined. The superposition of the experimental data y_i and the fitted curve $f(x)$ is shown in Fig. A.4(f). The localization of the measured transition point $x = x_0$, computed by using (A.10), and the expected one $x = 0$ are highlighted in Fig. A.4(f) by the corresponding horizontal lines.*

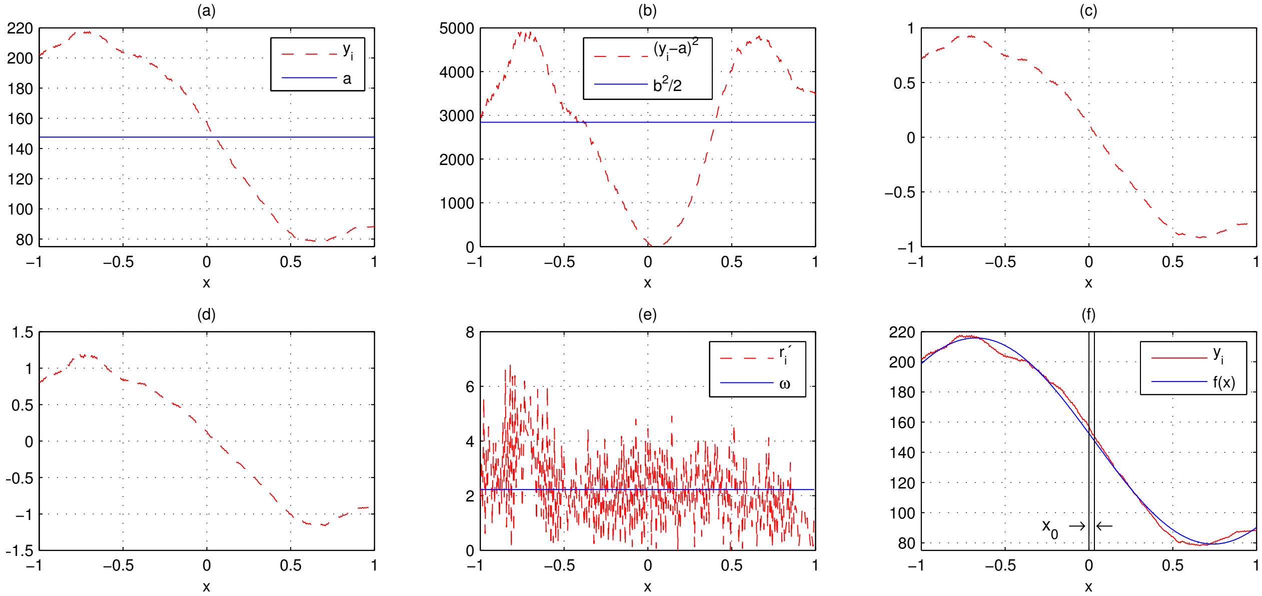


Figure A.4: *Sinusoidal curve fitting by the proposed cascade parameter estimation approach. (a) Estimation of the parameter a. (b) Estimation of $b^2/2$. (c) Normalized data. (d) Arc sine of the normalized data. (e) Estimation of the parameter ω. (f) Fitted curve to the experimental data and the points expected transition point $x = 0$ and the measured one $x = x_0$.*

Appendix B

Adquisición de datos y automatización

En matemáticas uno no entiende las cosas, se acostumbra a ellas

John Von Neumman

Contents

AUTOMATIZACIÓN se puede definir como aquella acción de equipar un sistema con *actuadores*, *sensores* y una *unidad de procesamiento* apropiados con la finalidad de que el sistema realice una tarea deseada o se mantenga en un estado de equilibrio sin necesidad de intervención del usuario. Entre las ventajas más evidentes que trae la automatización son: capacidad de respuesta rápida, altos niveles de repetibilidad y grandes volúmenes de producción.

Aquí describiremos la automatización de la extracción de fase desde interferogramas generados con el interferómetro de Michelson. En éste caso, la cámara CCD, los elementos piezoelétricos y una computadora toman el papel del sensor, actuadores y unidad de procesamiento, respectivamente. Las ventajas que se buscan son: realizar mediciones en intervalos de tiempo cortos (respuesta rápida), reducir las incertidumbres experimentales (alta repetibilidad), y obtener y procesar un conjunto grande de datos experimentales (grandes volúmenes de producción).

B.1 | Descripción del sistema

Consideremos un arreglo interferométrico cualquiera como se esboza en Fig. **??**.

B.2 | MATLAB y Simulink

MATLAB, desarrollado por MathWorks, es una potente plataforma de cómputo numérico que permite la manipulación de matrices, creación de gráficas, implementación de algoritmos, creación de interfaces, etc.

Simulink, también desarrollado por MAthWorks, es una herramienta para la modelación, simulación y análisis de sistemas dinámicos. Simulink es empleado principalmente in teoría de control y procesamiento digital de señales para simulación y diseño.

B.3 | Data Acquisition Toolbox

Éste toolbox es un software que permite configurar dispositivos de hardware externos para leer o escribir datos.

B.4 | Tarjeta de adquisición de datos (NI USB-6009 DAQ)

El dispositivo NI USB-6009 DAQ de National Instruments, es una tarjeta de adquisición de datos. Algunas de sus características son:

- 8 entradas analógicas (12-bit, 10 kS/s),
- 2 salidas analógicas (12-bit, 150 S/s)
- 12 I/O digitales,
- Conexión USB, no requiere alimentación externa.

B.5 | Adquisición de datos

Los sistemas de adquisición de datos incorporan el manejo de señales, sensores, actuadores, acondicionamiento de señales, adquisición de datos desde otros dispositivos y aplicaciones de software. En resumen, *Adquisición de datos* es el proceso de:

- Adquirir señales desde fenómenos del mundo real,
- Digitalizar las señales,
- Analizar, presentar, almacenar datos.

Podemos pensar en los DAQ como el hardware que actúa como interface entre la computadora y el mundo externo.

Un dispositivo DAQ usualmente tiene las funciones

1. Entradas y salidas analógicas,
2. Entradas y salidas digitales,
3. contadores/timers.

B.6 | Drivers y software de aplicación

Los drivers son software que son intermediarios entre el hardware y el software aplicación. Los drivers evitan la necesidad de programación de alta complejidad para comunicarse con el dardware. Para el dispositivo NI USB-6009 DAQ, es necesario instalar el driver NI-DAQmx que proporciona National Instruments. Éste se puede descargar en `www.ni.com/`.

El software de aplicación consiste básicamente en procesar las señales de entrada y generar señales de salida. Aquí las complejas rutinas de comunicación con el hardware son transparentes gracias a los drivers. El software de aplicación generalmente es implementado en lenguajes de alto nivel tal como Visual Studio, LabVIEW, MATLAB, etc.

B.7 | MAX – Measurement and Automation Explorer

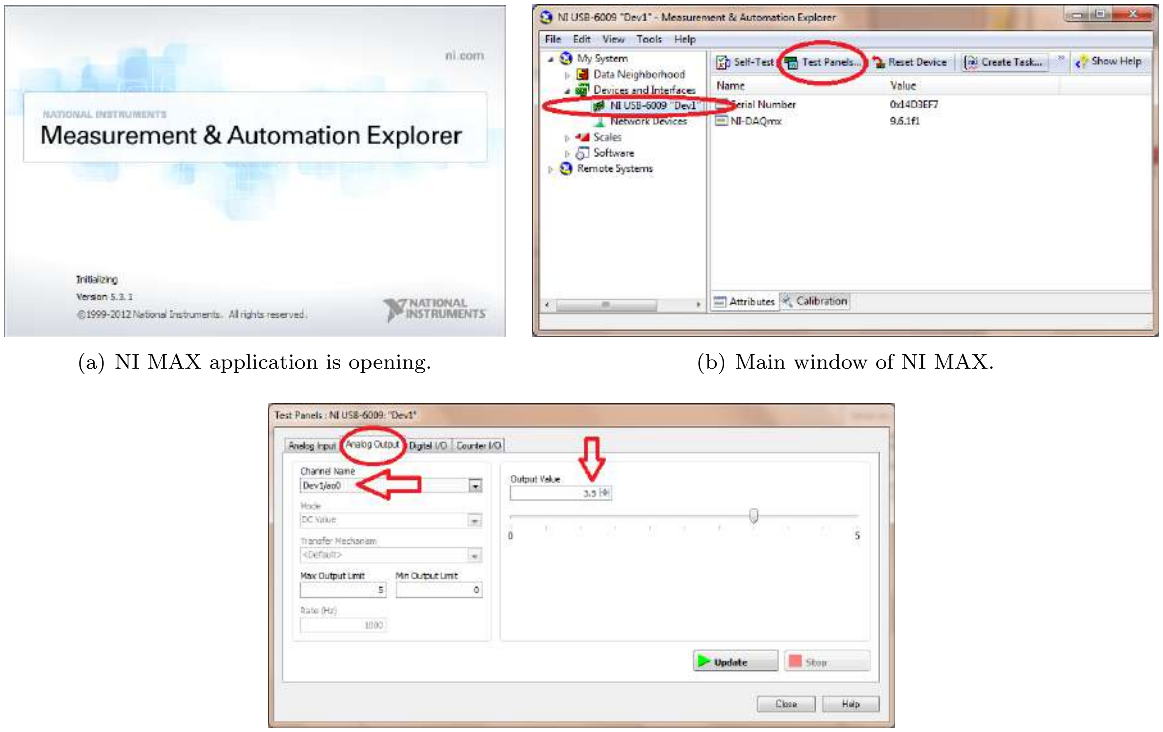
Éste es software que proporciona acceso a los dispositivos y sistemas de National Instruments. MAX permite

- Configurar hardware y software de National Instruments,
- Crear y editar canales, tareas, interfaces, escalas e instrumentos virtuales (VIs),
- Ejecutar diagnósticos del sistema,
- Ver dispositivos conectados al sistema,
- Actualizar software de National Instruments.

La aplicación NI MAX se instala cuando instalamos el driver DAQmx en configuración típica. Cuando conectamos la tarjeta, NI MAX la detecta y podemos probarla. Para ésto, procedemos como sigue:

1. Abrimos la aplicación NI MAX, Fig. B.1(a), y en la ventana principal vemos en la categoría `Devices and Interfaces` que la tarjeta DAQ se haya detectado (en el caso mostrado en la figura, la DAQ aparece como `NI USB-6009 ‘‘Dev1’’`).
2. Seleccionamos la tarjeta dando clic sobre su referencia (`NI USB-6009 ‘‘Dev1’’` en el caso mostrado) y a continuación damos clic sobre `Test Panels`. Una vez en la ventana `Panel test`, damos clic en la pestaña `Analog Output`, Fig. B.1(c).

3. La tarjeta USB-6009 tiene dos salidas analógicas, éstas son identificadas como los canales `Dev1/ao0` y `Dev1/ao1`, respectivamente. Seleccionamos el canal `Dev1/ao0`, por ejemplo, y damos un valor de voltaje entre cero y cinco (por ejemplo 3.5 como se muestra en Fig. B.1(c)) y damos clic en el botón `Undate`. Con ésto, verificamos con un multímetro que el valor de voltaje dado se registre en las terminales correspondientes de la DAQ. Con ésto, se ha probado que la DAQ funciona correctamente.



Name	Value
Serial Number	0x14D3EF7
NI-DAQmx	9.6.1f1

Channel Name: Dev1/ao0
Mode: DC Value
Transfer Mechanism: <Default>
Max Output Limit: 5
Min Output Limit: 0
Rate (Hz): 1000
Output Value: 3.5

(a) NI MAX application is opening.

(b) Main window of NI MAX.

(c) Panel Test Window.

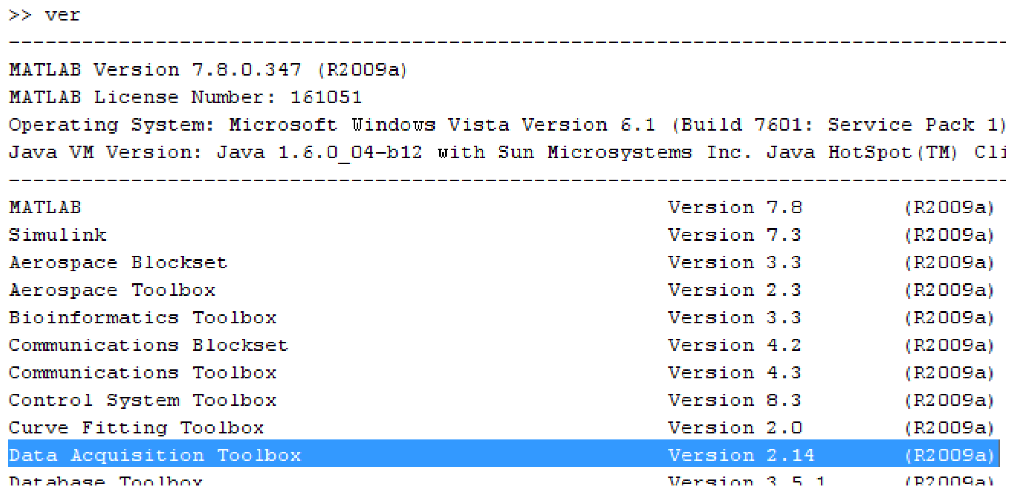
Figure B.1: *Aplicación NI MAX.*

B.8 | Acceso a la DAQ desde MATLAB

B.8.1 | Data Acquisition Toolbox

`Data Acquisition Toolbox` es un software que proporciona herramientas para leer y escribir datos a hardware externo. Usaremos este toolbox con la finalidad de escribir datos a la tarjeta USB-6009.

Para determinar is `Data Acquisition Toolbox` está instalado y su versión, escribimos el comando `ver` en la ventana de comandos de MATLAB. Si el toolbox está instalado, aparecerá listado como se muestra en Fig. B.2.



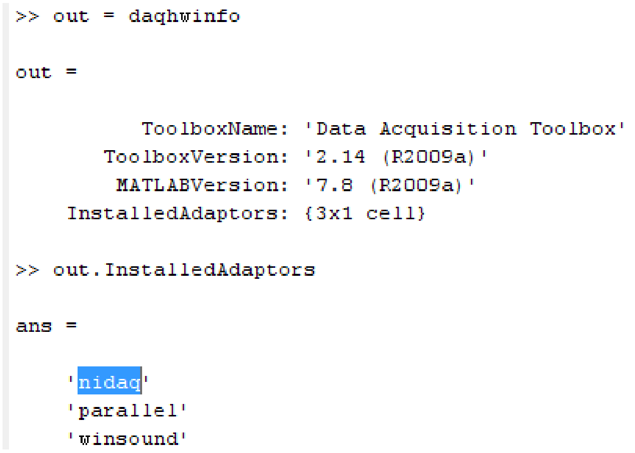
```
>> ver
--------------------------------------------------------------------------------
MATLAB Version 7.8.0.347 (R2009a)
MATLAB License Number: 161051
Operating System: Microsoft Windows Vista Version 6.1 (Build 7601: Service Pack 1)
Java VM Version: Java 1.6.0_04-b12 with Sun Microsystems Inc. Java HotSpot(TM) Cli
--------------------------------------------------------------------------------
MATLAB                                                Version 7.8         (R2009a)
Simulink                                              Version 7.3         (R2009a)
Aerospace Blockset                                    Version 3.3         (R2009a)
Aerospace Toolbox                                     Version 2.3         (R2009a)
Bioinformatics Toolbox                                Version 3.3         (R2009a)
Communications Blockset                               Version 4.2         (R2009a)
Communications Toolbox                                Version 4.3         (R2009a)
Control System Toolbox                                Version 8.3         (R2009a)
Curve Fitting Toolbox                                 Version 2.0         (R2009a)
Data Acquisition Toolbox                              Version 2.14        (R2009a)
Database Toolbox                                      Version 3.5.1       (R2009a)
```

Figure B.2: *Respuesta del comando* `ver` *de MATLAB. Se puede comprobar que* `Data Acquisition Toolbox` *está instalado.*

B.8.2 | Prueba de DAQ en MATLAB

Anteriormente se verificó el funcionamiento de la DAQ por medio de la aplicación NI MAX. Ahora probaremos la tarjeta desde MATLAB.

Comenzaremos por verificar si el driver `DAQmx` se ha instalado correctamente. Para ésto, asegúrese ejecutar MATLAB como administrador[1]. Escriba las siguientes instrucciones en la ventaja de comandos de MATLAB: `out = daqhwinfo` y `out.InstalledAdaptors`. Obtendrá una respuesta similar a la mostrada en Fig. B.3. Si el driver `DAQmx` se instaló correctamente, debemos ver `nidaq` en la lista de adaptadores.



```
>> out = daqhwinfo

out =

          ToolboxName: 'Data Acquisition Toolbox'
       ToolboxVersion: '2.14 (R2009a)'
        MATLABVersion: '7.8 (R2009a)'
    InstalledAdaptors: {3x1 cell}

>> out.InstalledAdaptors

ans =

    'nidaq'
    'parallel'
    'winsound'
```

Figure B.3: *Respuesta de los comandos* `out = daqhwinfo` *y* `out.InstalledAdaptors`*. Se comprueba con ésto que el driver* `DAQmx` *está instalado correctamente puesto que* `nidaq` *aparece en la lista de adaptadores.*

[1] Para abrir MATLAB como administrador, de clic izquierdo en el icono de MATLAB y seleccione la opción `Ejecutar como administrador`.

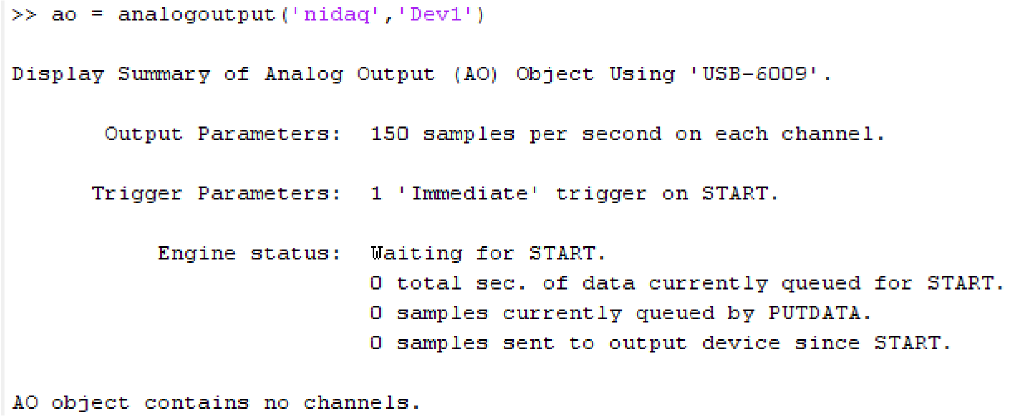
B.8.3 | Simple aplicación con la DAQ

Las aplicaciones básicas con las tarjetas de adquisición de datos consisten de:

- Iniciar sesión,
- Leer/Escribir,
- Cerrar sesión.

Iniciar sesión

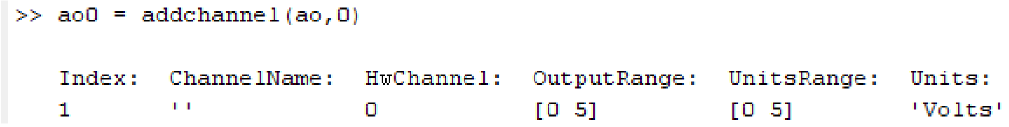
En esta estación se especifica que tipo de dispositivo se va a usar. En particular, usaremos la salida analógica de NI USB-6009. usamos el comando `analogoutput()` como se muestra en Fig. B.4



```
>> ao = analogoutput('nidaq','Dev1')

Display Summary of Analog Output (AO) Object Using 'USB-6009'.

       Output Parameters:  150 samples per second on each channel.

      Trigger Parameters:  1 'Immediate' trigger on START.

           Engine status:  Waiting for START.
                           0 total sec. of data currently queued for START.
                           0 samples currently queued by PUTDATA.
                           0 samples sent to output device since START.

AO object contains no channels.
```

Figure B.4: *Creación del objeto* `ao`*. Note que el argumento* `Dev1` *es la etiqueta que NI MAX le asigna a los dispositivos que se conectan al equipo. Por tanto esta etiqueta puede ser diferente si tiene más de un dispositivo en su sistema. Vea la documentación* `NI MAX` *(Measurement & Automation Explorer) de National Instruments.*

Ahora definiremos el(los) canal(es) que se van a usar. Como vio antes, la tarjeta USB-6009 tiene dos canales para salida analógica. Usamos el comando `addchannel()` para agregar el canal cero al objeto `ao` como se muestra en Fig. B.5



```
>> ao0 = addchannel(ao,0)

   Index:  ChannelName:  HwChannel:  OutputRange:  UnitsRange:  Units:
   1       ''            0           [0 5]         [0 5]        'Volts'
```

Figure B.5: *Agregando el canal 0 al objeto* `ao` *(salida analógica de la DAQ USB-6009).*

Leer/Escribir

En particular, solo estamos interesados en escribir datos (analógicos). Para escribir datos en el puerto analógico (en nuestro caso, en el canal cero), usamos el comando `putsample(ao,numerical_value)`, donde numerical_value es un número que debe estar en el rango UnitsRange (en nuestro caso, entre cero y cinco. Ver Fig. B.5). Por ejemplo, ejecute putsample(ao,3.5) en la ventaja de comandos de MATLAB y verifique con un multímetro que el voltaje en las terminales correspondientes de la tarjeta USB-6009 sea de 3.5 Volts.

Cerrar sesión

Cuando se ha finalizado la adquisición de datos, se usa el comando delet() para cerrar la conexión. Por ejemplo, para cerrar la conexión de salida analógica ao que usamos en los párrafos anteriores, escribimos delet(ao) en la ventana de comandos de MATLAB.

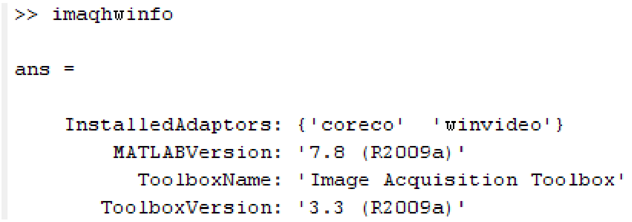
B.9 | La cámara PixeLINK

Como dispositivo de captura de imágenes, en este trabajo se empleó la cámara PL-B781U de PixeLINK. Ésta cámara es monocromática con 256 niveles de grises, tiene resolución de 2208×3000 pixeles (6.6 mega pixeles) y velocidad de captura de 5 fps (frames per second) en resolución máxima. El sensor es de tecnología CMOS con área activa de 7.73×10.5 mm (13.1 mm diagonal) y tamaño de pixel de 3.5×3.5 μm.

La captura de imágenes puede hacerse desde el software interfaz gráfica de usuario PixeLINK Capture OEM. Sin embargo, para fines de automatización, aquí describiremos la captura de imágenes desde MATLAB.

B.9.1 | Captura de imágenes desde MATLAB

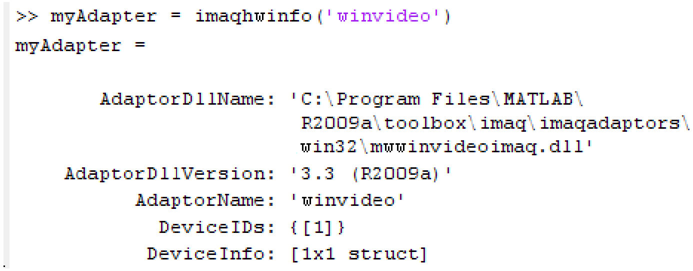
Para capturar imágenes con la cámara PL-B781U desde MATLAB, se requiere del Image Acquisition Toolbox y el adaptador de video winvideo. Podemos verificar que ambos estén instalados en el sistema escribiendo el comando imaqhwinfo en la ventana de comandos de MATLAB como se muestra en Fig. B.6.



```
>> imaqhwinfo

ans =

    InstalledAdaptors: {'coreco'  'winvideo'}
        MATLABVersion: '7.8 (R2009a)'
          ToolboxName: 'Image Acquisition Toolbox'
       ToolboxVersion: '3.3 (R2009a)'
```

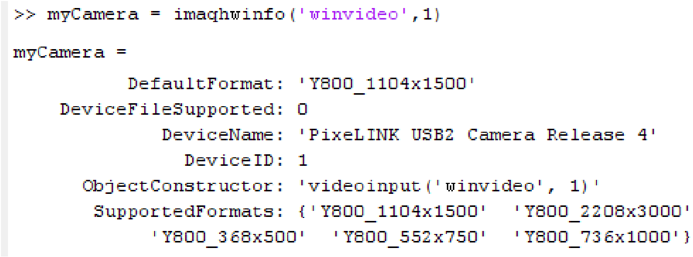
Figure B.6: *Información sobre Image Acquisition Toolbox y los adaptadores de video instalados.*

Para verificar que la cámara conectada al equipo ha sido detectada por el sistema, usamos el comando imaqhwinfo('winvideo') como se muestra en Fig. B.7. En la información que se despliega podemos ver el número de dispositivos detectados (DeviceIDs). Las características de la cámara detectada se ven agregando el ID del dispositivo, como se muestra en Fig. B.8.



```
>> myAdapter = imaqhwinfo('winvideo')
myAdapter =

       AdaptorDllName: 'C:\Program Files\MATLAB\
                        R2009a\toolbox\imaq\imaqadaptors\
                        win32\mwwinvideoimaq.dll'
    AdaptorDllVersion: '3.3 (R2009a)'
          AdaptorName: 'winvideo'
            DeviceIDs: {[1]}
           DeviceInfo: [1x1 struct]
```

Figure B.7: *Aplicación NI MAX. NI MAX application is opening.*



```
>> myCamera = imaqhwinfo('winvideo',1)
myCamera =
          DefaultFormat: 'Y800_1104x1500'
    DeviceFileSupported: 0
             DeviceName: 'PixeLINK USB2 Camera Release 4'
               DeviceID: 1
      ObjectConstructor: 'videoinput('winvideo', 1)'
       SupportedFormats: {'Y800_1104x1500'  'Y800_2208x3000'
            'Y800_368x500'  'Y800_552x750'  'Y800_736x1000'}
```

Figure B.8: *Aplicación NI MAX. Main window of NI MAX.*

Appendix C
Mathematical preliminaries

En matemáticas uno no entiende las cosas, se acostumbra a ellas

John Von Neumman

Contents

Con la finalidad de hacer la exposición del trabajo directa, algunos de los desarrollos matemáticos y herramientas que se usan a lo largo de todo el trabjo se presentan en este apéndice.

C.1 | Gradient field integration

Poisson's equation is a partial differential equation of the form

$$\nabla_p^2 \phi(p) = \rho(p) \tag{C.1}$$

which is defined on a closed domain D of the plane and where

$$\nabla_p^2 = \frac{\partial^2}{\partial p_1^2} + \frac{\partial^2}{\partial p_2^2} + \cdots + \frac{\partial^2}{\partial p_n^2}, \tag{C.2}$$

is the Laplacian operator and $p = (p_1, p_2, \cdots, p_n)$ is a n-dimensional independent variable ($n = 2$ for our applications). The solution $\phi(p)$ of Eq. (C.1) is unique up to an additive constant if the Neumann boundary conditions are given, which specify the values of the normal derivative $\partial\phi/\partial\nu$ on the boundary of D. However, under certain conditions, the boundary conditions are not necessary.

Suppose that one limits the set of possible solutions $\phi(p)$ to the periodic functions. A periodic function is a function that, for all $p \in \Re^n$, satisfies

$$\phi(p) = \phi(p + q_1) = \phi(p + q_2) = \cdots = \phi(p + q_n),$$

where q_i $(i = \overline{1, n})$ are n-dimensional vectors of the form

$$\begin{aligned} q_1 &= \begin{bmatrix} \bar{q}_1 & 0 & 0 & \cdots & 0 \end{bmatrix}, \\ q_2 &= \begin{bmatrix} 0 & \bar{q}_2 & 0 & \cdots & 0 \end{bmatrix}, \\ q_3 &= \begin{bmatrix} 0 & 0 & \bar{q}_3 & \cdots & 0 \end{bmatrix}, \\ &\vdots \\ q_n &= \begin{bmatrix} 0 & 0 & 0 & \cdots & \bar{q}_n \end{bmatrix}, \end{aligned}$$

with $\bar{q}_i$ $(i = \overline{1, n})$ being fixed constants.

C.2 | The least-squares method

For the experimental areas, one of the most important mathematical tools to data analysis is the least-squares method. Particularly, we consider a linear system which can be modelled as

$$Ax = b, \tag{C.3}$$

where $A \in \Re^{M \times N}$ is a $M \times N$ matrix which represents the model of a linear system, $x \in \Re^N$ is the coefficients vector and $b \in \Re^M$ a observations vector.

If an *appropriate* observation b is available and the operator A is given (a polynomial, for example), the problem is to obtain the coefficients vector x such that the error

$$E = b - Ax, \tag{C.4}$$

(for a particular norm) is minimum. Particularly, in this work it is considered the ℓ_2-norm for vectorial

N-dimensional euclidean space. Thus,

$$\begin{aligned} S(x) &= \|E\|^2 = \|b - Ax\|^2 = (b - Ax)^T(b - Ax) \\ &= \|b\|^2 + \|Ax\|^2 - 2b^TAx \\ &= \|b\|^2 + x^TA^TAx - 2b^TAx. \end{aligned} \tag{C.5}$$

the minimum can be found by

$$\frac{\partial}{\partial x}S(x) = 0, \tag{C.6}$$

where

$$\frac{\partial}{\partial x}S(x) = \frac{\partial}{\partial x}(x^TA^TAx) - 2\frac{\partial}{\partial x}(b^TAx). \tag{C.7}$$

For the first term of the right hand of the above equation we have that the vector of interest x is operated by the symmetric matrix A^TA. For illustration purposes, even $x \in \Re^n$, we consider the simple case $x \in \Re^3$, therefore

$$\begin{aligned} \frac{\partial}{\partial x}(x^TA^TAx) &= \frac{\partial}{\partial x}\left(\begin{bmatrix} x_1 & x_2 & x_3 \end{bmatrix} \begin{bmatrix} \alpha_1 & \alpha_4 & \alpha_5 \\ \alpha_4 & \alpha_2 & \alpha_6 \\ \alpha_5 & \alpha_6 & \alpha_3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}\right) \\ &= \frac{\partial}{\partial x}\begin{pmatrix} \alpha_1x_1^2 + \alpha_4x_1x_2 + \alpha_5x_1x_3 + \\ \alpha_4x_1x_2 + \alpha_2x_2^2 + \alpha_6x_2x_3 + \\ \alpha_5x_1x_3 + \alpha_6x_2x_3 + \alpha_3x_3^2 \end{pmatrix} \\ &= \begin{bmatrix} 2\alpha_1x_1 + 2\alpha_4x_2 + 2\alpha_5x_3 \\ 2\alpha_4x_1 + 2\alpha_2x_2 + 2\alpha_6x_3 \\ 2\alpha_5x_1 + 2\alpha_6x_2 + 2\alpha_3x_3 \end{bmatrix} \\ &= 2\begin{bmatrix} \alpha_1 & \alpha_4 & \alpha_5 \\ \alpha_4 & \alpha_2 & \alpha_6 \\ \alpha_5 & \alpha_6 & \alpha_3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \\ &= 2A^TAx. \end{aligned} \tag{C.8}$$

Similarly,

$$\begin{aligned} \frac{\partial}{\partial x}(b^TAx) &= \frac{\partial}{\partial x}\left(\begin{bmatrix} b_1 & b_2 & b_3 \end{bmatrix} \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}\right) \\ &= \frac{\partial}{\partial x}\begin{pmatrix} a_{11}b_1x_1 + a_{12}b_1x_2 + a_{13}b_1x_3 + \\ a_{21}b_2x_1 + a_{22}b_2x_2 + a_{23}b_2x_3 + \\ a_{31}b_3x_1 + a_{32}b_3x_2 + a_{33}b_3x_3 \end{pmatrix} \\ &= \begin{bmatrix} a_{11}b_1 + a_{21}b_2 + a_{31}b_3 \\ a_{12}b_1 + a_{22}b_2 + a_{32}b_3 \\ a_{13}b_1 + a_{23}b_2 + a_{33}b_3 \end{bmatrix} \\ &= \begin{bmatrix} a_{11} & a_{21} & a_{31} \\ a_{12} & a_{22} & a_{32} \\ a_{13} & a_{23} & a_{33} \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} \\ &= A^Tb. \end{aligned} \tag{C.9}$$

Therefore, the Eq. (C.7) can be rewritten as

$$\frac{\partial}{\partial x}S(x) = 2A^TAx - 2A^Tb. \tag{C.10}$$

Thus, the solution for x which satisfies the equation (C.6) leads to

$$2A^T Ax - 2A^T b = 0, \tag{C.11}$$

or,

$$x = A^{\dagger} b, \tag{C.12}$$

where

$$A^{\dagger} := (A^T A)^{-1} A^T \tag{C.13}$$

is the left ***least-squares*** (Moore-Penrose, pseudo-inverse, or generalized) inverse of A.

The right least-squares inverse of A is another useful operator. It is defined as

$$A^{\ddagger} := A^T (AA^T)^{-1}. \tag{C.14}$$

C.3 | Polynomial functions

Throughout this work, the polynomial functions are frequently used. Thus, in this section an exposition of its representation is done. Also is exposed the least-squares polynomial approximation.

For one-dimension, the n-degree polynomial $p(x)$ is defined as

$$p(x) = \sum_{u=0}^{n} c_u x^u, \tag{C.15}$$

where $c_u \in \Re$ are constant coefficients.

For two-dimension, the polynomial $p(x, y)$ of degree almost n is defined as

$$p(x, y) = \sum_{u=0}^{n} \sum_{v=0}^{u} c_{uv} x^v y^{u-v}, \tag{C.16}$$

where c_{uv} are constant coefficients.

The usefulness of polynomial functions relies on the fact that any smooth function can be approximated by the Taylor series (polynomial function). Moreover, the polynomial functions are linear functions on the coefficients. Thus, these polynomial can be efficiently fitted to experimental datasets by the least-squares method.

C.3.1 | One-dimensional polynomial fitting

A common problem though this work is the polynomial fitting to a given measured data. For this, we apply the least-squares method as follows. Let $\boldsymbol{x} \in \Re^N$ be the vector

$$\boldsymbol{x} = [x_1 \; x_2 \; \cdots \; x_N]^T \tag{C.17}$$

with entries x_j such that

$$x_1 < x_2 < \cdots < x_N,$$

and where $x_{j+1} - x_j$ for $j = \overline{1, N-1}$ is not necessarily constant, i.e., the points x_i are non-uniformly distributed in general.

If for each point x_i the respective observation $p(x_i)$ is available, then the equation (C.15) can be formulated as the equations system

$$P = XC, \tag{C.18}$$

where the regression matrix $X \in \Re^{N\times n}$, the coefficient $C \in \Re^{n}$ vector, and the observation $P \in \Re^{N}$ vector, are defined, respectively, as

$$X = \begin{bmatrix} 1_{N\times 1} & \boldsymbol{x} & \boldsymbol{x}.^{2} & \cdots & \boldsymbol{x}.^{n} \end{bmatrix}_{N\times(n+1)}, \tag{C.19}$$

$$C = \begin{bmatrix} c_0 & c_1 & \cdots & c_n \end{bmatrix}^{T}_{1\times(n+1)}, \tag{C.20}$$

$$P = \begin{bmatrix} p(x_1) & p(x_2) & \cdots & p(x_N) \end{bmatrix}^{T}_{1\times N}, \tag{C.21}$$

with $[\cdot].^{k}$ denoting the elementwise k-th exponentiation. It is worth mentioning that the columns of the regression matrix X are the basis functions of the assumed model.

For a given dataset $\{x_i, p(x_i)\}$, the unknown coefficient vector C is approximated by solving the matrix equation (C.18) as

$$\hat{C} = X^{\dagger}P. \tag{C.22}$$

Then, the fitting of the polynomial (C.15) to the data observations P is performed by

$$\hat{p}(\boldsymbol{x}) = X\hat{C}, \tag{C.23}$$

where $\hat{p}(\boldsymbol{x})$ is the best approximation (in least-squares sense) of the given data $p(x_i)$.

C.3.2 | Two-dimensional polynomial fitting

The above results are easily extended to the two-dimensional polynomial (C.16). In this case, it is necessary that the observations $p(x_j, y_i)$ be given on a rectangular grid by a matrix $P \in \Re^{M\times N}$ as

$$P = \begin{bmatrix} p(x_1,y_1) & p(x_2,y_1) & \cdots & p(x_N,y_1) \\ p(x_1,y_2) & p(x_2,y_2) & \cdots & p(x_N,y_2) \\ \vdots & \vdots & \ddots & \vdots \\ p(x_1,y_M) & p(x_2,y_M) & \cdots & p(x_N,y_M) \end{bmatrix}$$

with the sample points (x_j, y_i) defined by the vectors

$$\begin{aligned} \boldsymbol{x} &= [x_1 \; x_2 \; \cdots \; x_N]^{T}, \\ \boldsymbol{y} &= [y_1 \; y_2 \; \cdots \; y_M]^{T}, \end{aligned} \tag{C.24}$$

where the nodes $\{x_j\}$ and $\{y_i\}$ are not necessary uniformly distributed.

Accordingly, the equation (C.16) can be write as

$$P = YCX^{T}, \tag{C.25}$$

where the regression matrices X and Y are given by

$$\begin{aligned} X &= \begin{bmatrix} 1_{N\times 1} & \boldsymbol{x} & \boldsymbol{x}.^{2} & \cdots & \boldsymbol{x}.^{n} \end{bmatrix}_{N\times(n+1)}, \\ Y &= \begin{bmatrix} 1_{N\times 1} & \boldsymbol{y} & \boldsymbol{y}.^{2} & \cdots & \boldsymbol{y}.^{n} \end{bmatrix}_{M\times(n+1)}, \end{aligned} \tag{C.26}$$

and the coefficients are grouped in the matrix

$$C = \begin{bmatrix} c_{00} & c_{01} & \cdots & c_{0n} \\ c_{10} & c_{11} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ c_{n0} & 0 & \cdots & 0 \end{bmatrix}_{(n+1)\times(n+1)} .$$

This coefficient matrix is an upper cross-triangular ($c_{uv} = 0$ for $u + v > n$) because a polynomial of degree at most n is represented. However, the equation (C.25) represents a general two-dimensional polynomial of $(u + v)$th degree.

Now, for a given dataset $\{x_v, y_u, p(x_v, y_u)\}$, the polynomial's coefficient C are computed from (C.25) by

$$\hat{C} = Y^{\dagger} P X^{\ddagger}. \tag{C.27}$$

Finally, the fitting procedure of the polynomial (C.16) is carried out from (C.25) as

$$\hat{p}(\boldsymbol{x}, \boldsymbol{y}) = Y \hat{C} X^T,$$

where $\hat{p}(\boldsymbol{x}, \boldsymbol{y})$ is least-squares approximation to the data P at $(\boldsymbol{x}, \boldsymbol{y})$ for the polynomial basis (C.26).

It is worth mentioning that, in both one- and two-dimensional cases, once the polynomial's coefficient $\hat{C}$ is available, the sample vector(s) can be changed by any other(s). Thus, with the corresponding regression matrices, the reconstructed polynomial $\hat{p}$ can be obtained at any value $\boldsymbol{x}$ or $(\boldsymbol{x}, \boldsymbol{y})$.

This possibility allow us perform others data processing procedures such as interpolation, extrapolation, increasing resolution, decimation, o simply a redefinition of the samples point.

C.3.3 | Multiple fitting: one-dimensional case

Consider that there are available multiple measurements Y_k with the same regression matrix and different vector of parameters

$$\begin{aligned} Y_1 &= A\Theta_1 \\ Y_2 &= A\Theta_2 \\ &\vdots \\ Y_n &= A\Theta_n \end{aligned} \tag{C.28}$$

If the measurements Y_i corresponds to the same system, then

$$\Theta_1 = \Theta_2 = \cdots = \Theta_n = \Theta. \tag{C.29}$$

Therefore, the system of matrix equations (C.28) can be written as

$$\begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_n \end{bmatrix} = \underbrace{\begin{bmatrix} A \\ A \\ \vdots \\ A \end{bmatrix}}_{\mathcal{A}} \Theta = \mathcal{A}\Theta. \tag{C.30}$$

We have the Moore–Penrose pseudoinverse

$$\begin{aligned}\mathcal{A}^{\dagger} &= (\mathcal{A}^T\mathcal{A})^{-1}\mathcal{A}^T \\ &= \left(\begin{bmatrix} A^T & \cdots & A^T \end{bmatrix}\begin{bmatrix} A \\ \vdots \\ A \end{bmatrix}\right)^{-1}\begin{bmatrix} A^T & \cdots & A^T \end{bmatrix} \\ &= \frac{1}{n}(A^TA)^{-1}\begin{bmatrix} A^T & \cdots & A^T \end{bmatrix} \\ &= \frac{1}{n}\begin{bmatrix} A^{\dagger} & A^{\dagger} & \cdots & A^{\dagger} \end{bmatrix}.\end{aligned} \tag{C.31}$$

Therefore, the solution Eq. (C.30) for Θ is

$$\begin{aligned}\Theta &= \frac{1}{n}\begin{bmatrix} A^{\dagger} & A^{\dagger} & \cdots & A^{\dagger} \end{bmatrix}\begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_n \end{bmatrix} \\ &= \frac{1}{n}(\Theta_1 + \Theta_2 + \cdots + \Theta_n).\end{aligned} \tag{C.32}$$

That is, the vector Θ can be obtained as the average of the least-squares solutions Θ_k, $k = \overline{1, n}$. Note that this requires n regressions. Alternatively, the Eq. (C.32) can be written as

$$\begin{aligned}\Theta &= \frac{1}{n}(A^TA)^{-1}\begin{bmatrix} A^T & A^T & \cdots & A^T \end{bmatrix}\begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_n \end{bmatrix} \\ &= \frac{1}{n}(A^TA)^{-1}(A^TY_1 + A^TY_2 + \cdots + A^TY_n) \\ &= \frac{1}{n}A^{\dagger}(Y_1 + Y_2 + \cdots + Y_n).\end{aligned} \tag{C.33}$$

That is, the vector Θ is obtained as the least squares solution of the average of the provided measurements Y_k. Note that, unlike the Eq. (C.32) which requires n regressions, the equation (C.33) performs a single regression.

C.3.4 | Multiple fitting: two-dimensional case

Consider that there are n measurements I_k, $k = \overline{1, n}$, which are approximated as

$$\begin{aligned}YC_1X^T &= I_1 \\ YC_2X^T &= I_2 \\ &\vdots \\ YC_nX^T &= I_n,\end{aligned} \tag{C.34}$$

where C_k are the parameter matrices of the polynomial with basis Y and X. If we require a single matrix of parameters C which solves all the provided data; namely

$$\underbrace{\begin{bmatrix} Y \\ Y \\ \vdots \\ Y \end{bmatrix}}_{\mathcal{Y}} CX^T = \underbrace{\begin{bmatrix} I_1 \\ I_2 \\ \vdots \\ I_n \end{bmatrix}}_{\mathcal{I}} \tag{C.35}$$

or

$$\mathcal{Y}CX^T = \mathcal{I}. \tag{C.36}$$

We have that

$$\begin{aligned} \mathcal{Y}^\dagger &= \frac{1}{n}(Y^TY)^{-1}\begin{bmatrix} Y^T & Y^T & \cdots & Y^T \end{bmatrix} \\ &= \frac{1}{n}\begin{bmatrix} Y^\dagger & Y^\dagger & \cdots & Y^\dagger \end{bmatrix}. \end{aligned} \tag{C.37}$$

Therefore, the solution of Eq. (C.36) for C leads to

$$\begin{aligned} C &= \frac{1}{n}\begin{bmatrix} Y^\dagger & Y^\dagger & \cdots & Y^\dagger \end{bmatrix}\mathcal{I}X^\ddagger \\ &= \frac{1}{n}(Y^\dagger I_1 + Y^\dagger I_2 + \cdots + Y^\dagger I_n)X^\ddagger \\ &= \frac{1}{n}(Y^\dagger I_1 X^\ddagger + Y^\dagger I_2 X^\ddagger + \cdots + Y^\dagger I_n X^\ddagger) \\ &= \frac{1}{n}(C_1 + C_2 + \cdots + C_n). \end{aligned} \tag{C.38}$$

That is, the least squares solution C for all measurements is obtained as the average of the parameters C_k for each provided data I_k (n regressions are required).

Similarly,

$$\begin{aligned} C &= \frac{1}{n}(Y^\dagger I_1 + Y^\dagger I_2 + \cdots + Y^\dagger I_n)X^\ddagger \\ &= \frac{1}{n}Y^\dagger(I_1 + I_2 + \cdots + I_n)X^\ddagger. \end{aligned} \tag{C.39}$$

That is, the parameters C are obtained as the least squares solution for the average of the provided data (only one regression is performed).

C.4 | Aproximación de la distribución de irradiancia

Usando una fuente de iluminación láser, la distribución de irradiancia $I(p)$ es modelada por funciones Gaussianas [?] donde se asume, sin pérdida de generalidad, están centradas en el origen del sistema coordenado como

$$I(p) = he^{p^TAp}, \tag{C.40}$$

donde $p = [x\ y]^T$ es un vector de posición, $h \in \Re$ son las amplitudes y $A \in \Re^{2\times 2}$ una matriz definida negativa. La serie de Taylor de $I(p)$ (C.40) en el punto $p_0 = [0\ 0]^T =: 0_p$ es

$$I(p) = I(p_0) + (p - p_0)^T \nabla I(p_0) + \frac{1}{2}(p - p_0)^T \nabla^2 I(p)(p - p_0) + \cdots, \tag{C.41}$$

donde el jacobiano $\nabla I_i(0_p)$ y hessiano $\nabla^2 I_i(0_p)$ son

$$\begin{aligned}\nabla I(p_0) &= 2I(p_0)Ap_0 = 0_p,\\ \nabla^2 I(p_0) &= 2I(p_0)\left[A + 2Ap_0p_0^TA\right] = 2I(p_0)A = 2hA.\end{aligned}$$

Si se considera únicamente una pequeña región de iluminación alrededor del origen 0_p, una aproximación de segundo orden es suficiente para representar la irradiancia $I(p)$ (C.40) [?]. La aproximación de segundo orden de $I(p)$ alrededor de 0_p es

$$I(p) \approx h + hp^TAp =: \mathcal{I}(p),$$

que es un polinomio de segundo grado.

C.5 | Matrix analysis

C.5.1 | Independence, subspace, basis, and dimension

A set of vectors $\{a_1, \cdots, a_n\}$ ($a_i \in \Re^m$) is *linearly independent* if the linear combination

$$Ax = 0_m, \tag{C.42}$$

with $A = [a_1, a_2, \cdots, a_n]$ and $x \in \Re^n$, has the trivial solution $x = 0_n$. Otherwise, the set of vectors a_i is said to be *linearly dependent*.

Given a collection of vectors $a_i \in \Re^m$ ($i = \overline{1, n}$), the set of all linear combinations of these vectors is a subspace referred to as the *span* of $\{a_1, \cdots, a_n\}$:

$$\text{span}\{a_1, \cdots, a_n\} = \left\{ \sum_{j=1}^{n} \beta_j a_j : \beta_j \in \Re \right\}.$$

If $\{a_i, \cdots, a_n\}$ is independent and $b \in \text{span}\{a_i, \cdots, a_n\}$, then b is a unique linear combination of the a_j.

All bases for a subspace S have the same number of elements. This number is the *dimension* and is denoted by $\dim(S)$

C.5.2 | Range, null space, and rank

There are two important subspaces associated with an m-by-n matrix A. The *range* of A is defined by

$$\text{ran}(A) = \{y \in \Re^m : y = Ax \text{ for some } x \in \Re^n\},$$

and the *null space* of A is defined by

$$\text{null}(A) = \{x \in \Re^n : Ax = 0_m\}.$$

If $A = [a_1 \ \cdots \ a_n]$ is a column partitioning, then

$$\text{ran}(A) = \text{span}\{a_1, \cdots, a_n\}.$$

The *rank* of a matrix A is defined by

$$\text{rank}(A) = \dim(\text{ran}(A)).$$

It can be shown that $\text{rank}(A^T) = \text{rank}(A)$. We say that $A \in \Re^{m\times n}$ is *rank deficient* if $\text{rank}(A) < \min\{m, n\}$. If $A \in \Re^{m\times n}$, then

$$\dim(\text{null}(A)) + \text{rank}(A) = n.$$

C.5.3 | Orthogonality and the singular value decomposition

Orthogonality has a very prominent role to play in matrix computations. Among other things, the singular value decomposition (SVD) enables us to intelligently handle the matrix rank problem. The concept of rank, though perfectly clear in the exact arithmetic context, is tricky in the presence of round off error and fuzzy data. With the SCD we can introduce the practical notion of numerical rank.

C.5.4 | Orthogonality

A set of vectors $\{x_1, \cdots, x_p\}$ in $\Re^m$ is orthogonal if $x_i^T x_j = 0$ wherever $i \neq j$ and orthonormal if $x_i^T x_j = \delta_{ij}$. Intuitively, orthogonal vectors are maximally independent for they point in totally different directions.

A collection of subspaces $S_1, \cdots, S_p$ in $\Re^m$ is *mutually orthogonal* if $x^T y = 0$ wherever $x \in S_i$ and $y \in S_j$ for $i \neq j$. The *orthogonal complement* of a subspace $S \subseteq \Re^m$ is defined by

$$S^\perp = \{y \in \Re^m : y^T x = 0 \text{ for all } x \in S\}$$

and it is not hard to show that $\text{ran}(A)^\perp = \text{null}(A^T)$. The vectors $v_1, \cdots, v_k$ form an *orthonormal* basis for subspace $S \subseteq \Re^m$ if they are orthonormal and span S.

A matrix $Q \in \Re m \times m$ is said to be *orthogonal* if $Q^T Q = I$. If $Q = [q_1 \;\cdots\; q_m]$ is orthogonal, then the q_i form an orthonormal basis for $\Re^m$. It is always possible to extend such a basis to a full orthonormal basis $\{v_1, \vdots, v_m\}$ for $\Re^m$.

Theorem 2. *If $V_1 \in \Re^{n \times r}$ has orthonormal columns, then there exists $V_2 \in \Re^{n \times (n-r)}$ such that*

$$V = [V_1 \; V_2]$$

is orthogonal. Note that $ran(V_1)^\perp = ran(V_2)$.

C.5.5 | Norms and orthogonal transformations

The 2-norm is invariant under orthogonal transformation, for if $Q^T Q = I$, then $\|Qx\|_2^2 = x^T Q^T Q x = x^T x = \|x\|_2^2$. The matrix 2-norm and the Frobenius norm are also invariant with respect to orthogonal transformations. In particular, it is easy to show that for all orthogonal Q and Z of appropriate dimensions we have

$$\|QAZ\|_F = \|A\|_F\,,$$

and

$$\|QAZ\|_2 = \|A\|_2\,.$$

C.5.6 | Inverse matrix

The inverse of two square matrices A and B of the same size is

$$(A+B)^{-1} = A^{-1} - (\mathbb{I} + A^{-1}B)^{-1} A^{-1} B A^{-1}. \tag{C.43}$$

This equality can be verified as follows. Let

$$(A+B)^{-1} = A^{-1} + X. \tag{C.44}$$

By taking into account the identity

$$\begin{aligned}\mathbb{I} &= (A^{-1}+X)(A+B)\\ &= A^{-1}A + XA + A^{-1}B + XB,\end{aligned}$$

and rearranging terms, we have

$$X(A+B) = -A^{-1}B.$$

Solving for X,

$$\begin{aligned}X &= -A^{-1}B(A+B)^{-1}\\ &= -A^{-1}B(A^{-1}+X)\\ &= -A^{-1}BA^{-1} - A^{-1}BX.\end{aligned}$$

Rearranging terms again,

$$(\mathbb{I}+A^{-1}B)X = -A^{-1}BA^{-1},$$

and solving for X, we have

$$X = -(\mathbb{I}+A^{-1}B)^{-1}A^{-1}BA^{-1}.$$

Finally, the substitution of this last equation in (C.44) proofs the equality (C.43).

The most general result is the *Woodbury matrix identity* or *matrix inversion lemma*:

$$(A+BCD)^{-1} = A^{-1} - A^{-1}B(C^{-1}+DA^{-1}B)^{-1}DA^{-1}. \tag{C.45}$$

C.5.7 | The singular value decomposition

The theory of norm developed in the previous two sections can be used to prove the extremely useful singular value decomposition.

Theorem 3 (Singular value decomposition (SVD))**.** *If A is a real $m \times n$ matrix, then there exist two orthogonal matrices*

$$\begin{aligned}U &= [u_1 \ \cdots \ u_m] \in \Re^{m\times m},\\ V &= [v_1 \ \cdots \ v_m] \in \Re^{n\times n},\end{aligned}$$

such that

$$U^T AV = diag(\sigma_1, \cdots, \sigma_p) \in \Re^{m\times n}$$

with $p = \min\{m,n\}$ and where $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_p \geq 0$.

C.6 | Taylor expansion

The Taylor series is given as

$$f(x) = \sum_{n=0}^{\infty} \frac{f^{(n)}(a)}{n!}(x-a)^n, \tag{C.46}$$

where a is the approximation point. That is, the polynomial expansion is built from the point a (where the approximation is exact).

There is another approach in Taylor series, the so-called fordward expansion of a funtion $f(x)$ about a point x:

$$f(x+h) = \sum_{n=0}^{\infty} \frac{h^n}{n!} f^{(n)}(x). \tag{C.47}$$

This equation allow us to know the value of the function f at the point $x+h$ if the value of f and its derivatives are known at the point x.

C.6.1 | Taylor expansion for multivariate functions

If the function f has more than one independent variable, say $f(x,y)$, the Taylor expansion becomes

$$\begin{aligned} f(x,y) = f(a,b) &+ (x-a)\frac{\partial f}{\partial x} + (y-b)\frac{\partial f}{\partial y} \\ &+ \frac{1}{2!}\left[(x-a)^2\frac{\partial^2 f}{\partial x^2} + 2(x-a)(y-b)\frac{\partial^2 f}{\partial x \partial y} + (y-b)^2\frac{\partial^2 f}{\partial y^2}\right] \\ &+ \frac{1}{3!}\left[(x-a)^3\frac{\partial^3 f}{\partial x^3} + 3(x-a)^2(y-b)\frac{\partial^3 f}{\partial x^2 \partial y}\right. \\ &\left. +3(x-a)(y-b)^2\frac{\partial^3 f}{\partial x \partial y^2} + (y-b)^3\frac{\partial^3 f}{\partial y^3}\right] + \cdots, \end{aligned} \tag{C.48}$$

with all derivatives evaluated at the point (a,b). Using $\alpha_j t = x_j - x_{j0}$, we may write the Taylor expansion for m independent variables n the symbolic form

$$f(x_1,\cdots,x_m) = \sum_{n=0}^{\infty} \frac{t^n}{n!}\left(\sum_{i=1}^{m} \alpha_i \frac{\partial}{\partial x_i}\right)^n f(x_1,\cdots,x_m)\bigg|_{(x_{10},\cdots,x_{m0})}. \tag{C.49}$$

For the particular case where the point where the derivatives are evaluated is $(x_{10},\cdots,x_{m0}) = (0,\cdots,0)$, we obtain the ***Maclaurin series***.

C.6.2 | Two-dimensional case

For the particular case of functions $f:\Re^2 \to \Re$, it is convenient to define the vectorial variables

$$\begin{aligned} \xi &= [x\ y]^T, \\ \alpha &= [a\ b]^T, \end{aligned}$$

then we have the truncated Taylor series in the compact form given as

$$f(\xi) = f(\alpha) + (\xi-\alpha)^T \nabla f + \frac{1}{2!}(\xi-\alpha)^T[\nabla^2 f](\xi-\alpha) \tag{C.50}$$

where

$$\nabla f = \begin{bmatrix} \partial f/\partial x \\ \partial f/\partial y \end{bmatrix}, \tag{C.51}$$

and

$$\nabla^T f = \begin{bmatrix} \partial^2 f/\partial x^2 & \partial^2 f/\partial x \partial y \\ \partial^2 f/\partial y \partial x & \partial^2 f/\partial y^2 \end{bmatrix}. \tag{C.52}$$

C.7 | Matrix form of the square of a summation

Let a polynomial of the form

$$y = a_1x_1 + a_2x_2 + \cdots + a_nx_n. \tag{C.53}$$

The square y^2 is given as

$$\begin{aligned} y^2 =& a_1^2x_1^2 + a_2^2x_2^2 + \cdots + a_n^2x_n^2 \\ &+ 2a_1a_2x_1x_2 + 2a_1a_3x_1x_3 + \cdots + 2a_1a_nx_1x_n \\ &+ 2a_2a_3x_2x_3 + 2a_2a_4x_2x_4 + \cdots + 2a_2a_nx_2x_n \\ &+ 2a_3a_4x_3x_4 + 2a_3a_5x_3x_5 + \cdots + 2a_3a_nx_3x_n \\ &+ \cdots + 2a_{n-1}a_nx_{n-1}x_n. \end{aligned} \tag{C.54}$$

The square y^2 can be stated in a compact form using matrix notation as follows. Let $\boldsymbol{a}$ and $\boldsymbol{x}$ be the vectors

$$\boldsymbol{a} = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix}, \quad \boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}. \tag{C.55}$$

Then, the polynomial y is rewritten as

$$y = \boldsymbol{a}^T\boldsymbol{x}, \tag{C.56}$$

Therefore, the square y^2 is obtained as

$$\begin{aligned} y^2 &= (\boldsymbol{a}^T\boldsymbol{x})^2 = (\boldsymbol{a}^T\boldsymbol{x})^T(\boldsymbol{a}^T\boldsymbol{x}) \\ &= \boldsymbol{x}^T\boldsymbol{a}\boldsymbol{a}^T\boldsymbol{x}. \end{aligned} \tag{C.57}$$

C.8 | Sampling signals

Consider the rectangle function $\Pi(x)$ defined as

$$\Pi(x) = \begin{cases} 1 & |x| < 1/2, \\ 1/2 & |x| = 1/2, \\ 0 & \text{otherwise.} \end{cases} \tag{C.58}$$

Notice that the rectangle Π has unit width and unit height. The integral of the rectangle function in all its domain is equal to one; i.e.,

$$\int_{-\infty}^{\infty} \Pi(x)dx = \int_{-1/2}^{1/2} 1dx = 1. \tag{C.59}$$

The shape of the rectangle function can be modified by scaling the abscissa x and modulating the amplitude. Among all possible shapes for $\Pi(x)$ we are interested in such of them that preserves the property given in Eq. (C.59).

The width of the rectangle is equal to one. A rectangle with any other width Δx can be obtained as

$$\Pi\left(\frac{x}{\Delta x}\right). \tag{C.60}$$

We are interested in rectangles that preserves the property given in Eq. (C.59). Since

$$\int_{-\infty}^{\infty} \Pi\left(\frac{x}{\Delta x}\right) dx = \int_{-\Delta x/2}^{\Delta x/2} 1 dx = \Delta x. \tag{C.61}$$

Therefore, the rectangle pulse given as

$$r(x) = \frac{1}{\Delta x}\Pi\left(\frac{x}{\Delta x}\right) \tag{C.62}$$

preserves the property given in Eq. (C.59) as required. It is worth mentioning that the rectangle pulse $r(x)$ is an even function; that is

$$r(-x) = r(x). \tag{C.63}$$

The rectangle pulse $r(x)$ lead us to the delta function as the limit $\Delta x \to 0$; i.e.,

$$\lim_{\Delta x \to 0} \frac{1}{\Delta x}\Pi\left(\frac{x}{\Delta x}\right) = \delta(x) \tag{C.64}$$

From this, the even property and unit area of the rectangle function are inherited to the delta function.

Appendix D

Implementación en MATLAB del algoritmo GPSI propuesto

En matemáticas uno no entiende las cosas, se acostumbra a ellas

John Von Neumman

Contents

PRIMERO se implementará el algoritmo propuesto anteriormente en una función escrita en código `m` de MATLAB. Más adelante se muestra la implementación en MATLAB-Simulink para aplicaciones en tiempo real y finalmente se presenta una interfaz de usuario gráfica (GUI, por las siglas en inglés: ***Graphic User Interface***) para interactuar con el programa durante el tiempo de ejecución.

D.1 | Función script de MATLAB

Podemos dividir el algoritmo propuesto en cuatro partes elementales:

1. Obtención de la luz de fondo por ajuste (**??**) del polinomio de segundo grado (1.19) a los datos experimentales (**??**).

2. Obtención de la luz de modulación al cuadrado por ajuste (**??**) del polinomio de cuarto grado (1.20) a los datos obtenidos al sustraer la luz de fondo (obtenida en el paso anterior) a los datos experimentales (1.24). Normalizar los interferogramas.

3. Solución de (3.27) por mínimos cuadrados (3.28) para obtener los pasos de fase (3.29) y recuperar el corrimiento de fase (**??**).

4. Solución de (1.37) por mínimos cuadrados (3.32) para obtener la fase envuelta (3.33).

Los dos primeros pasos de la lista anterior terminan con la normalización de los interferogramas mientras que los últimos dos finalizan con la obtención de la fase envuelta. En Fig. D.1 se presenta, en código `m` escrito en MATLAB, la función principal `GPSI` que invoca a las funciones subordinadas `NormGPSI` y `WPhase` para normalizar los interferogramas y obtener la fase envuleta, respectivamente.

La función principal `GPSI` recibe dos argumentos: la matriz de interferogramas `I` y el número de interferogramas `K`. La matriz de interferogramas es de la forma `I=[I0;I1;...;Ik;...;IK-1]` donde cada bloque `Ik` con `k=0,1,...K-1` es un interferograma a procesar. Como es posible manipular los interferogramas (matrices bidimensionales con sistema coordenado x-y) como si éstos fuesen vectores (matrices unidimensionales con sistema coordenado p); cuando inicia el procesamiento se emplea la función `MatToVec` para convertir las matrices a vectores y una vez terminado el procesamiento se emplea la función `VecToMat` para convertir los vectores a matrices y los datos se puedan visualizar en su disposición original.

D.2 | Modelo en Matlab-Simulink

El algoritmo propuesto es de ejecución relativamente rápida puesto que es posible procesar desde dos interferogramas o más, y es no iterativo. Otra propiedad importante es que no requiere intervención de usuario para dirigir su flujo de ejecución o elegir datos de entrada. Estas propiedades permiten desarrollar programas para ejecución en tiempo real. Con fines ilustrativos en Fig. D.2 se presenta la implementación del algoritmo propuesto en diagrama de bloques sobre la plataforma de Matlab-Simulink. En Fig. D.2(a) se muestra el diagrama de bloques principal en el modelo Simulink y son claramente visible las dos partes en las que se divide el algoritmo: la normalización de los interferogramas (Subsistema ⌊Normalize⌉, Fig. D.2(c)) y la recuperación de la fase envuelta (Subsistema ⌊Phase Recover⌉, Fig. D.2(d)).

En Fig. D.2(b) se presenta el subsistema ⌊Interfs⌉ que en este caso solo genera un interferograma de referencia `I0` y un interferograma `I1` con corrimiento de fase que varía de forma triangular en el intervalo desde 0.14 a 3 rad. No es muy difícil configurar el subsistema ⌊Interfs⌉ para que adquiera los interferogramas desde una cámara de video conectada a la computadora. En Fig. D.2(e) se muestra el subsistema ⌊Scope⌉ dedicado a desplegar el video de los interferogramas procesándose y la fase envuelta obtenida. El bloque ⌊Scope1⌉ muestra la superposición del corrimiento de fase exacto y estimado por el algoritmo.

En Fig. D.3 se muestra la captura de las ventanas de video cuando el programa Simulink descrito se está ejecutando.

```
function [x y a b nI alp d Wp] = GPSI(I,K)
[x y a b nI] = NormGPSI(I,K);
[alp d Wp]   = WPhase(nI,K);
end

function [x y a b nI] = NormGPSI(I,K)
[ny nx] = size(I); ny = ny/K;
x = 2*(1:nx)/nx - 1;
y = 2*(1:ny)/ny - 1;
[X Y] = meshgrid(x,y);
Ir = MatToVec(I,K); X = X(:); Y = Y(:);
%Background light fit
AA = [ones(nx*ny,1) X Y X.*Y X.^2 Y.^2];
Th = (AA'*AA)\AA'*Ir';
Th = sum(Th,2)/K;
a = AA*Th;
%Modulation light fit
IminAo2 = (Ir - ones(K,1)*a').^2;
AA = [ones(nx*ny,1) X X.^2 X.^3 X.^4 ...
    Y X.*Y X.^2.*Y X.^3.*Y Y.^2 X.*Y.^2 ...
    X.^2.*Y.^2 Y.^3 X.*Y.^3 Y.^4];
Th = (AA'*AA)\AA'*IminAo2';
Th = sum(Th,2)/K;
b2s2 = abs(AA*Th);
b = sqrt(2*b2s2);
%Normalization of interferograms
nI = sat((Ir-ones(K,1)*a')./(ones(K,1)*b'));
%Vector to Matrix Convertion
a  = reshape(a,ny,nx);
b  = reshape(b,ny,nx);
nI = VecToMat(nI,K,ny,nx);
end

function [Inorm] = sat(I)
[ny nx] = size(I);
Inorm=I;
    for i=1:ny
        for j=1:nx
            if Inorm(i,j)>1
                Inorm(i,j) = 1;
            elseif Inorm(i,j)<-1
                Inorm(i,j) = -1;
            end
        end
    end
end
```

(a) Función principal **GPSI** y la función subordinada **NormGPSI**.

```
function [alp d Wp] = WPhase(nI,K)
[nIr ny nx] = MatToVec(nI,K);
%Solving the equation: Asin^2 alp + Bcos alp = C
A = ones(nx*ny,1);
alp = zeros(1,K-1);
for ki=1:K-1
    B = 2*nIr(ki,:).*nIr(ki+1,:);
    C = nIr(ki,:).^2 + nIr(ki+1,:).^2;
    AA = [A B'];
    Th = (AA'*AA)\AA'*C';
    alp(ki) = atan2(sqrt(abs(Th(1))),Th(2));
end
% Obtaining function delta
d = [0 alp];
for ki=2:K
    d(ki) = d(ki-1) + d(ki);
end
AA = [cos(d') -sin(d')];
Th = (AA'*AA)\AA'*nIr;
Wp = atan2(Th(2,:),Th(1,:));
Wp = reshape(Wp,ny,nx);
end

function [Ir ny nx] = MatToVec(I,m)
    [ny nx] = size(I);
    ny = ny/m;
    Ir = zeros(m,nx*ny);
    for k=1:m
        Im = I(ny*(k-1)+1:ny*k,:);
        Ir(k,:) = Im(:)';
    end
end

function [I] = VecToMat(Ir,m,ny,nx)
    I = zeros(m*ny,nx);
    for k=1:m
        Ixy = reshape(Ir(k,:),ny,nx);
        I(ny*(k-1)+1:ny*k,:) = Ixy;
    end
end
```

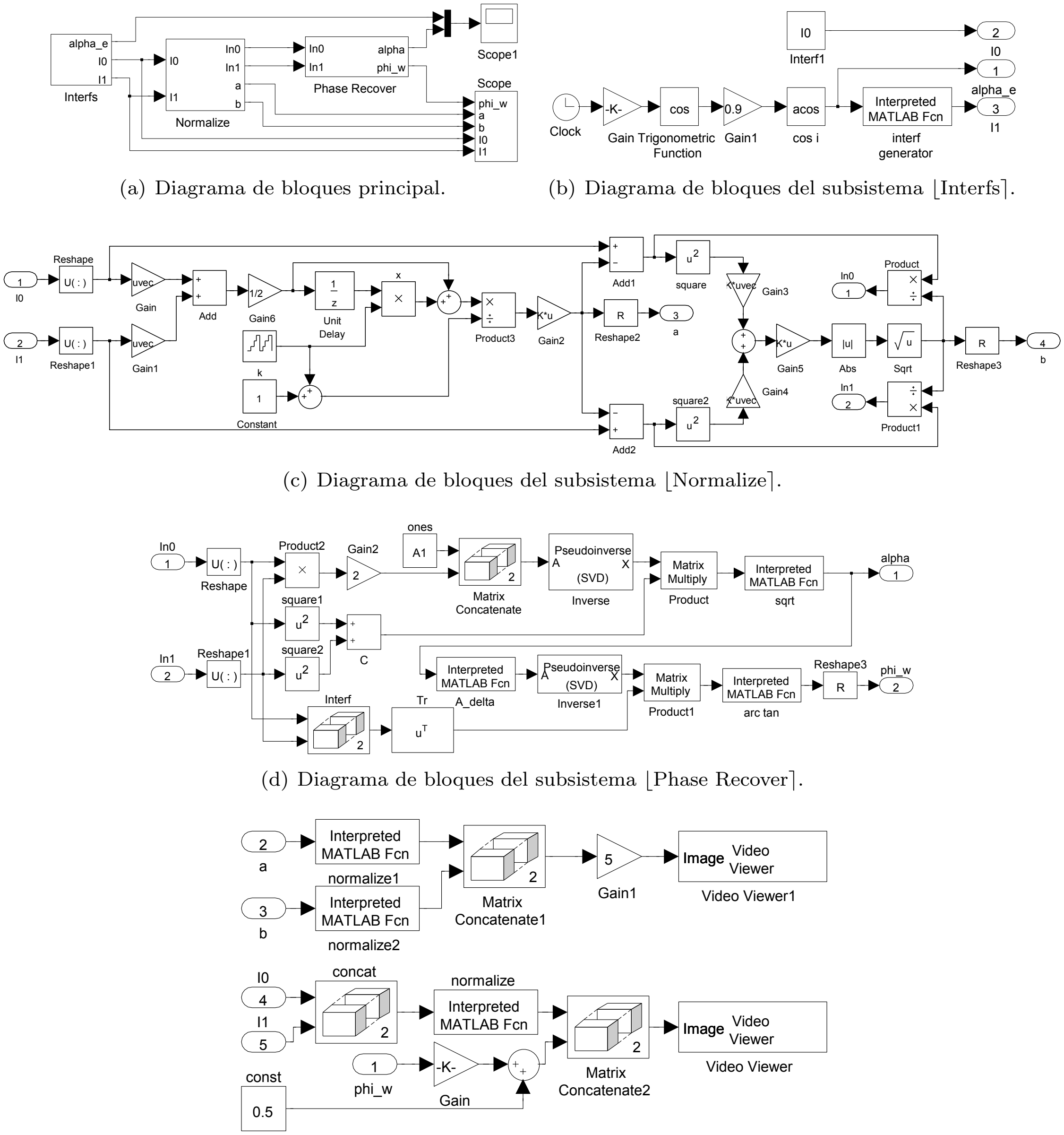
(b) Funciones subordinadas **WPhase**, **MatToVec**, **VecToMat**, y **sat**.

Figure D.1: *Implementación en MATLAB del algoritmo de interferometría de corrimiento de fase generalizado por estimación de parámetros usando el método de mínimos cuadrados.*

D.3 | Interfaz gráfica de usuario gráfica

La implementación en código **m** permite evaluar el funcionamiento del algoritmo propuesto procesando interferogramas simulados y reales. Sin embargo, para verificar el comportamiento del algoritmo en tiempo real, se presentó el modelo Simulink donde, para el caso mostrado, se toma un interferográma estático y una señal de video simulada que consiste en interferogramas que cambian el corrimiento de fase con el tiempo. Cabe señalar que no es muy difícil configurar el modelo Simulink presentado para que la señal de video a procesar se tome de una cámara digital conectada a la computadora.

Ahora, presentamos una interfaz de usuario gráfica (GUI) que agrupa las ventajas de las dos imple-



(a) Diagrama de bloques principal.

(b) Diagrama de bloques del subsistema ⌊Interfs⌉.

(c) Diagrama de bloques del subsistema ⌊Normalize⌉.

(d) Diagrama de bloques del subsistema ⌊Phase Recover⌉.

(e) Diagrama de bloques del subsistema ⌊Scope⌉.

Figure D.2: *Implementación por diagrama de bloques del algoritmo propuesto en un modelo Matlab-Simulink para aplicaciones en tiempo real.*

mentaciones anteriores. Con la GUI es posible previsualizar la señal de video obtenida de una cámara CCD y con un botón tomar un interferograma para procesarlo por el algoritmo. La GUI presenta la luz de fondo estimada, la luz de modulación y el corrimiento de fase presente en el interferográma en la señal de video. En Fig. D.4 se muestra una pantalla de la GUI ejecutándose procesando interferogramas generados en el interferómetro de Michelson (Fig. D.4(a)) y en la prueba de Ronchi (Fig. D.4(b)).

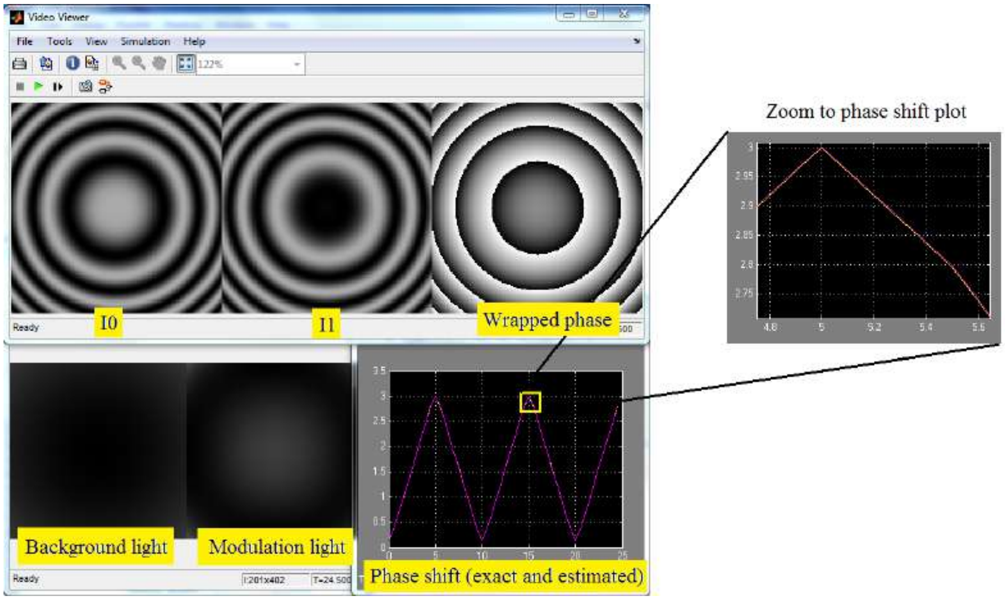


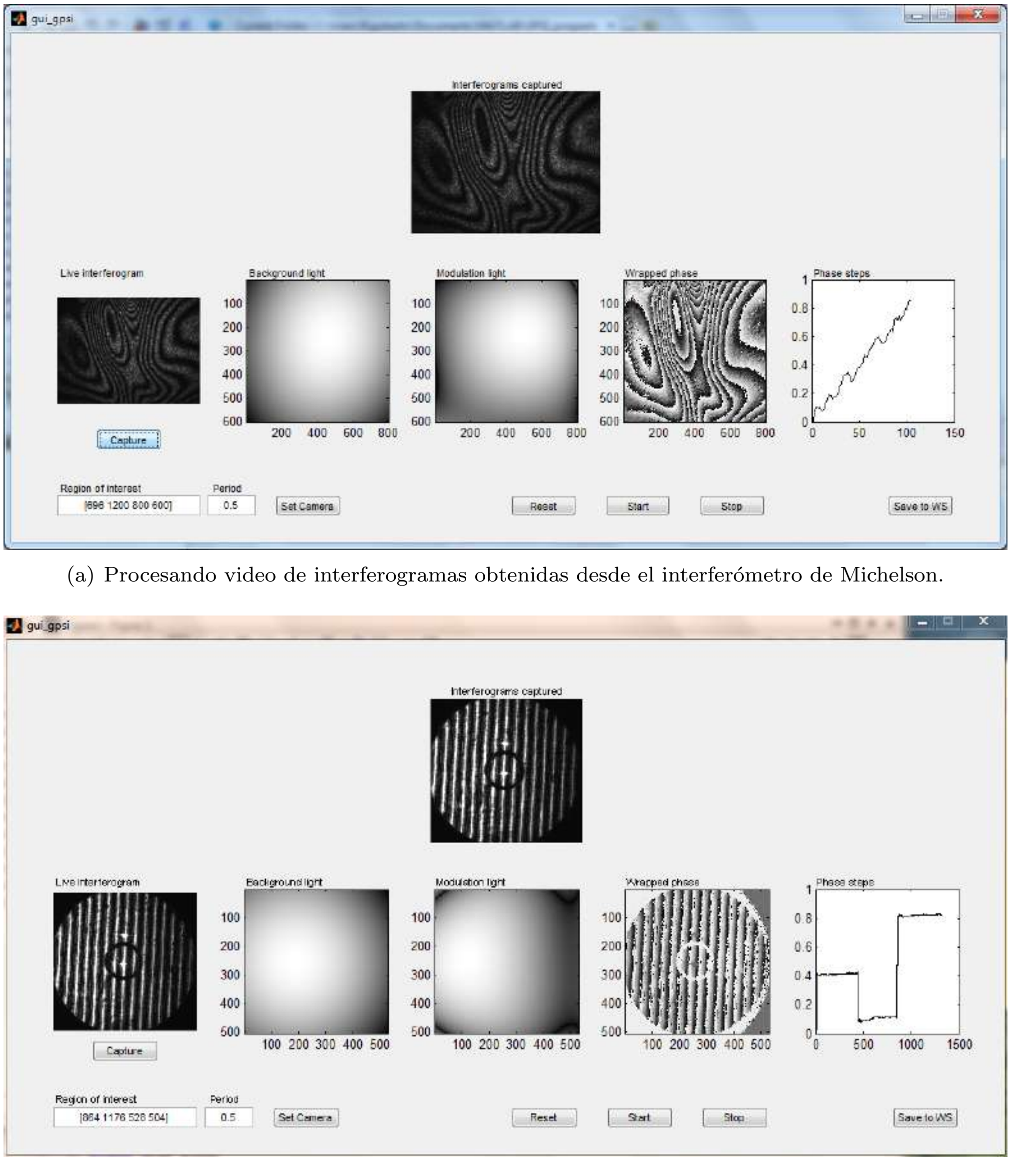
Figure D.3: *Captura de pantalla cuando el programa Simulink mostrado en Fig. D.2 se está ejecutando.*

D.4 | Automatización del interferómetro de Michelson usando piezoeléctrico y el algoritmo GPSI propuesto

En el interferómetro de Michelson, para proporcionar corrimientos de fase, es necesario desplazar el espejo de referencia. El interferómetro de Michelson comercial PASCO, está equipado con un tornillo micrométrico de precisión para desplazar el espejo de referencia y obtener corrimientos de fase. Sin embargo, al ser un dispositivo manual, los experimentos registran errores debido a la perturbación provocada por el usuario al operar sobre el tornillo. Para superar esta desventaja, se emplearon piezoelétricos instalados en el espejo de referencia. Los piezoeléctricos, a través de una tarjeta de adquisición de datos (DAQ USB 9001 de National Instruments), son operados desde la computadora estableciendo la comunicación a través de un puerto USB. Se desarrolló una GUI en Matlab y ésta genera las señales de control al piezoeléctrico, y graba el interferograma a travéz de la cámara CCD conectada.

Con el sistema desarrollado, se analizó la linealidad y precisión del piezoeléctrico. El experimento consistió en dar incrementos de voltaje a los piezoeléctricos de 5 mV. Tales incrementos de voltaje provocaría pasos de fase constantes e iguales a 0.9 rad. Se realizaron tres experimentos con 460 interferogramas cada uno (Mostrados con linea roja, verde y azul en Fig. D.5). El la primera gráfica de Fig. D.5 se muestran los pasos de fase obtenidos. En la segunda gráfica se muestra el ajuste de un polinomio de tercer grado para compensar la deriva. Sustrayendo el polinomio ajustado, se elimina el paso de fase y la deriva, quedándose solo con la incertidumbre de las mediciones como se muestra en la tercera gráfica de Fig. D.5.

De la tercera gráfica en Fig. D.5, podemos tomar como 0.02 rad. como el intervalo de incertidumbre. El cambio de fase desde una franja brillante hasta otra representa un paso de fase de 2π y equivale a una diferencia de camino óptico de la longitud de onda λ. Con los algoritmos propuestos y el sistema interferométrico considerado, la incertidumbre de 0.02 rad. señalan que se puede alcanzar precisión del orden de $\lambda/300$. La fuente láser empleada es de $\lambda = 638$ nm., por lo que con el sistema descrito se pueden distinguir perturbaciones en el frente de onda hasta de 2.13 nm.



Region of interest [896 1200 800 600]

Period 0.5

(a) Procesando video de interferogramas obtenidas desde el interferómetro de Michelson.

Region of interest [864 1176 528 504]

Period 0.5

(b) Procesando video en la prueba de Ronchi.

Figure D.4: *Interfaz de usuario en tiempo de ejecución.*

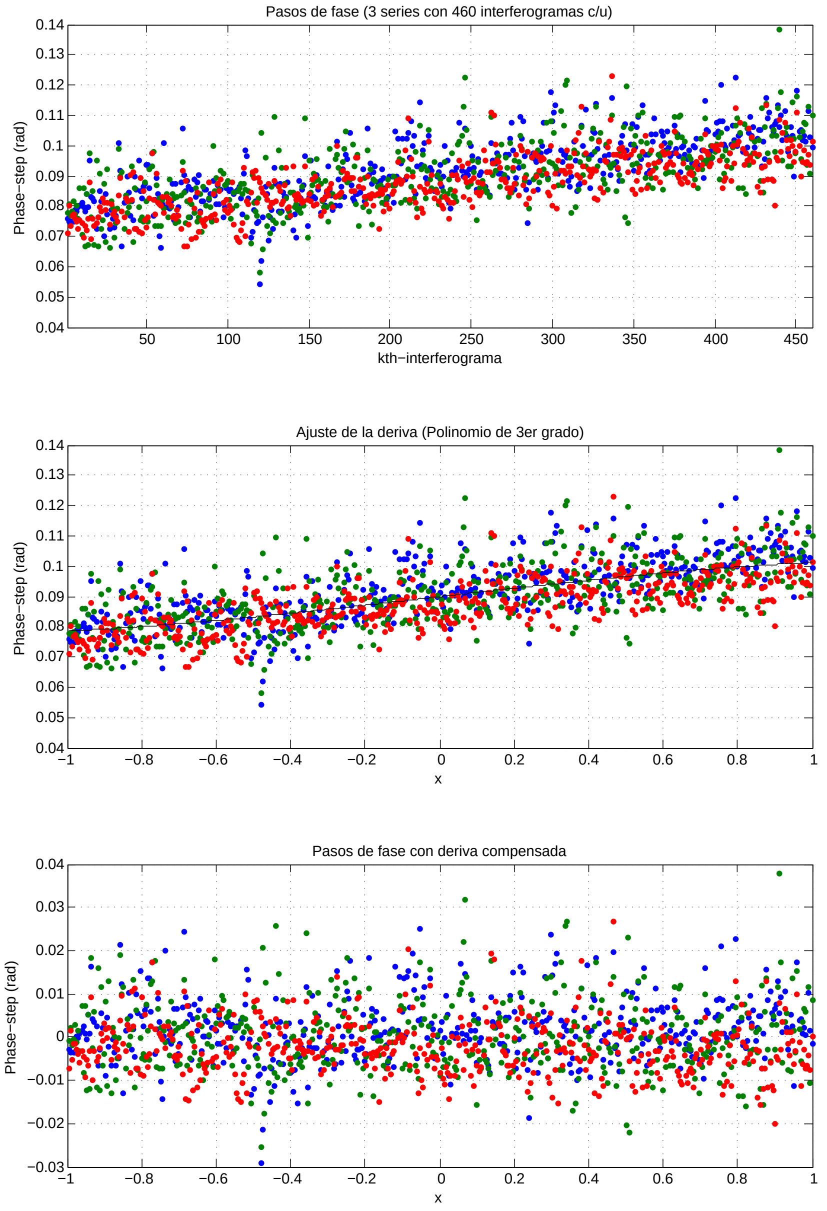


Figure D.5: *Implementación en MATLAB del algoritmo de interferometría de corrimiento de fase generalizado por estimación de parámetros usando el método de mínimos cuadrados.*

Bibliography

[1] D. Malacara, ed., *Optical shop testing.* John Wiley & Sons, Inc., third ed., 2007.

[2] G. Rajshekhar and P. Rastogi, "Fringe analysis: Premise and perspectives," *Optics and Lasers in Engineering*, vol. 50, no. 8, pp. iii – x, 2012. Fringe Analysis Methods & Applications.

[3] D. Malacara, M. Servin, and Z. Malacara, *Interferogram analysis for optical testing.* Taylor & Francis Group, second ed., 2005.

[4] D. Malacara, ed., *Óptica básica.* Fondo de Cultura Económica, México, segunda ed., 2004.

[5] J. A. Quiroga and M. Servin, "Isotropic n-dimensional fringe pattern normalization," *Optics Communications*, vol. 224, no. 4-6, pp. 221 – 227, 2003.

[6] Q. Yu, K. Andresen, W. Osten, and W. Jueptner, "Noise-free normalized fringe patterns and local pixel transforms for strain extraction," *Appl. Opt.*, vol. 35, pp. 3783–3790, Jul 1996.

[7] J. A. Quiroga, J. A. Gómez-Pedrero, and A. García-Botella, "Algorithm for fringe pattern normalization," *Optics Communications*, vol. 197, no. 1-3, pp. 43 – 51, 2001.

[8] L. Bruno and A. Poggialini, "Phase-shifting interferometry by an open-loop voltage controlled laser diode," *Optics Communications*, vol. 290, no. 0, pp. 118 – 125, 2013.

[9] Q. Yu, K. Andresen, W. Osten, and W. P. O. Jueptner, "Analysis and removal of the systematic phase error in interferograms," *Optical Engineering*, vol. 33, no. 5, pp. 1630–1637, 1994.

[10] M. B. Bernini, A. Federico, and G. H. Kaufmann, "Normalization of fringe patterns using the bidimensional empirical mode decomposition and the hilbert transform," *Appl. Opt.*, vol. 48, pp. 6862–6869, Dec 2009.

[11] Y. Fujun, D. Meiling, H. Xiaoyuan, and D. Xiaolei, "Single fringe projection profilometry based on sinusoidal intensity normalization and subpixel fitting," *Optics and Lasers in Engineering*, vol. 49, no. 3, pp. 465 – 472, 2011.

[12] A. Federico and G. H. Kaufmann, "Local denoising of digital speckle pattern interferometry fringes by multiplicative correlation and weighted smoothing splines," *Appl. Opt.*, vol. 44, pp. 2728–2735, May 2005.

[13] Q. Kemao, "Windowed fourier transform for fringe pattern analysis," *Appl. Opt.*, vol. 43, pp. 2695–2702, May 2004.

[14] J. A. Guerrero, J. L. Marroquin, M. Rivera, and J. A. Quiroga, "Adaptive monogenic filtering and normalization of espi fringe patterns," *Opt. Lett.*, vol. 30, pp. 3018–3020, Nov 2005.

[15] M. Servin, J. L. Marroquin, and F. J. Cuevas, "Demodulation of a single interferogram by use of a two-dimensional regularized phase-tracking technique," *Appl. Opt.*, vol. 36, pp. 4540–4548, Jul 1997.

[16] R. Legarda-Sáenz, W. Osten, and W. Jüptner, "Improvement of the regularized phase tracking technique for the processing of nonnormalized fringe patterns," *Appl. Opt.*, vol. 41, pp. 5519–5526, Sep 2002.

[17] S. D. Nicola, P. Ferraro, I. Gurov, R. Koviazin, and M. Volkov, "Fringe analysis for moiré interferometry by modification of the local intensity histogram and use of a two-dimensional fourier transform method," *Measurement Science and Technology*, vol. 11, no. 9, p. 1328, 2000.

[18] N. A. Ochoa and A. Silva-Moreno, "Normalization and noise-reduction algorithm for fringe patterns," *Optics Communications*, vol. 270, no. 2, pp. 161 – 168, 2007.

[19] E. Robin, V. Valle, and F. Brémand, "Phase demodulation method from a single fringe pattern based on correlation with a polynomial form," *Appl. Opt.*, vol. 44, pp. 7261–7269, Dec 2005.

[20] C. Meneses-Fabian and U. Rivera-Ortega, "Phase-shifting interferometry by wave amplitude modulation," *Opt. Lett.*, vol. 36, pp. 2417–2419, Jul 2011.

[21] A. Barcelata-Pinzon, C. Meneses-Fabian, L. Moreno-Alvarez, and R. Pastrana-Sanchez, "Common-path speckle interferometer for phase objects studies," *Optics Communications*, vol. 304, no. 0, pp. 153 – 157, 2013.

[22] R. Juarez-Salazar, C. Robledo-Sanchez, C. Meneses-Fabian, G. Rodriguez-Zurita, F. Guerrero Sanchez, and A. Barcelata-Pinzon, "Automatic real-time generalized phase-shifting interferometry to process interferograms with spatio-temporal visibility," 2013.

[23] C. Robledo-Sanchez, R. Juarez-Salazar, C. Meneses-Fabian, F. Guerrero-Sánchez, L. M. A. Aguilar, G. Rodriguez-Zurita, and V. Ixba-Santos, "Phase-shifting interferometry based on the lateral displacement of the light source," *Opt. Express*, vol. 21, pp. 17228–17233, Jul 2013.

[24] J. L. Vago, H. C. Vermeulen, and A. Verga, "Fast fourier transform based image compression algorithm optimized for speckle interferometer measurements," *Optical Engineering*, vol. 36, no. 11, pp. 3052–3063, 1997.

[25] M. Takeda, "Fourier fringe analysis and its application to metrology of extreme physical phenomena: a review [*Invited*]," *Appl. Opt.*, vol. 52, pp. 20–29, Jan 2013.

[26] J. H. Bruning, D. R. Herriott, J. E. Gallagher, D. P. Rosenfeld, A. D. White, and D. J. Brangaccio, "Digital wavefront measuring interferometer for testing optical surfaces and lenses," *Appl. Opt.*, vol. 13, pp. 2693–2703, Nov 1974.

[27] M. Takeda, H. Ina, and S. Kobayashi, "Fourier-transform method of fringe-pattern analysis for computer-based topography and interferometry," *J. Opt. Soc. Am.*, vol. 72, pp. 156–160, Jan 1982.

[28] J. F. Casco-Vasquez, R. Juarez-Salazar, C. Robledo-Sanchez, G. Rodriguez-Zurita, F. G. Sanchez, L. M. Arévalo Aguilar, and C. Meneses-Fabian, "Fourier normalized-fringe analysis by zero-order spectrum suppression using a parameter estimation approach," *Optical Engineering*, vol. 52, no. 7, pp. 074109–074109, 2013.

[29] D. J. Bone, H.-A. Bachor, and R. J. Sandeman, "Fringe-pattern analysis using a 2-d fourier transform," *Appl. Opt.*, vol. 25, pp. 1653–1660, May 1986.

[30] R. Juarez-Salazar, C. Robledo-Sanchez, C. Meneses-Fabian, G. Rodriguez-Zurita, F. Guerrero-Sanchez, and J. Gonzalez-Garcia, "Straightforward filtering to phase demodulation by a fourier normalized-fringe approach," 2013.

[31] D. J. Higham and N. J. Higham, *MATLAB Guide.* SIAM, 2005.

[32] T. Kreis, "Digital holographic interference-phase measurement using the fourier-transform method," *J. Opt. Soc. Am. A*, vol. 3, pp. 847–855, Jun 1986.

[33] J. H. Massig and J. Heppner, "Fringe-pattern analysis with high accuracy by use of the fourier-transform method: Theory and experimental tests," *Appl. Opt.*, vol. 40, pp. 2081–2088, May 2001.

[34] C. Gorecki, "Interferogram analysis using a fourier transform method for automatic 3d surface measurement," *Pure and Applied Optics: Journal of the European Optical Society Part A*, vol. 1, no. 2, p. 103, 1992.

[35] L.-C. Chen, H.-W. Ho, and X.-L. Nguyen, "Fourier transform profilometry (ftp) using an innovative band-pass filter for accurate 3-d surface reconstruction," *Optics and Lasers in Engineering*, vol. 48, no. 2, pp. 182 – 190, 2010. Fringe Projection Techniques.

[36] L. Kai and Q. Kemao, "Dynyamic 3d profiling with fringe projection using least squares method and windowed fourier filtering," *Optics and Lasers in Engineering*, vol. 51, no. 1, pp. 1 – 7, 2013.

[37] Y. Fu, C. J. Tay, C. Quan, and H. Miao, "Wavelet analysis of speckle patterns with a temporal carrier," *Appl. Opt.*, vol. 44, pp. 959–965, Feb 2005.

[38] S. Vanlanduit, J. Vanherzeele, P. Guillaume, B. Cauberghe, and P. Verboven, "Fourier fringe processing by use of an interpolated fourier-transform technique," *Appl. Opt.*, vol. 43, pp. 5206–5213, Sep 2004.

[39] Z. Zhang, Z. Jing, Z. Wang, and D. Kuang, "Comparison of fourier transform, windowed fourier transform, and wavelet transform methods for phase calculation at discontinuities in fringe projection profilometry," *Optics and Lasers in Engineering*, vol. 50, no. 8, pp. 1152 – 1160, 2012. Fringe Analysis Methods & Applications.

[40] J. García-Márquez, D. Malacara-Hernández, and M. Servín, "Analysis of interferograms with a spatial radial carrier or closed fringes and its holographic analogy," *Appl. Opt.*, vol. 37, pp. 7977–7982, Dec 1998.

[41] R. Langoju, A. Patil, and P. Rastogi, "Super-resolution fourier transform method in phase shifting interferometry," *Opt. Express*, vol. 13, pp. 7160–7173, Sep 2005.

[42] R. Vander, S. G. Lipson, and I. Leizerson, "Fourier fringe analysis with improved spatial resolution," *Appl. Opt.*, vol. 42, pp. 6830–6837, Dec 2003.

[43] J. L. Marroquin, J. E. Figueroa, and M. Servin, "Robust quadrature filters," *J. Opt. Soc. Am. A*, vol. 14, pp. 779–791, Apr 1997.

[44] J. L. Marroquin, M. Servin, and R. Rodriguez-Vera, "Adaptive quadrature filters and the recovery of phase from fringe pattern images," *J. Opt. Soc. Am. A*, vol. 14, pp. 1742–1753, Aug 1997.

[45] J. B. Liu and P. D. Ronney, "Modified fourier transform method for interferogram fringe pattern analysis," *Appl. Opt.*, vol. 36, pp. 6231–6241, Sep 1997.

[46] K. Larkin, "A self-calibrating phase-shifting algorithm based on the natural demodulation of two-dimensional fringe patterns," *Opt. Express*, vol. 9, pp. 236–253, Aug 2001.

[47] W. Chen, X. Su, Y. Cao, Q. Zhang, and L. Xiang, "Method for eliminating zero spectrum in fourier transform profilometry," *Optics and Lasers in Engineering*, vol. 43, no. 11, pp. 1267 – 1276, 2005.

[48] W. Chen, X. Su, Y. Cao, L. Xiang, and Q. Zhang, "Fourier transform profilometry based on a fringe pattern with two frequency components," *Optik - International Journal for Light and Electron Optics*, vol. 119, no. 2, pp. 57 – 62, 2008.

[49] J. Zhong and J. Weng, "Generalized fourier analysis for phase retrieval of fringe pattern," *Opt. Express*, vol. 18, pp. 26806–26820, Dec 2010.

[50] C. Quan, H. Niu, and C. Tay, "An improved windowed fourier transform for fringe demodulation," *Optics and Laser Technology*, vol. 42, no. 1, pp. 126 – 131, 2010.

[51] Y. Fu, J. Wu, and G. Jiang, "Fourier transform profilometry based on defocusing," *Optics & Laser Technology*, vol. 44, no. 4, pp. 727 – 733, 2012.

[52] M. M. Hasan, K. Teramoto, and S. Tanemura, "Windowed fourier assisted two-dimensional hilbert transform for fringes phase extraction," *Optik - International Journal for Light and Electron Optics*, no. 0, pp. –, 2013.

[53] W. Gao, Q. Kemao, H. Wang, F. Lin, and H. S. Seah, "Parallel computing for fringe pattern processing: A multicore cpu approach in matlab® environment," *Optics and Lasers in Engineering*, vol. 47, no. 11, pp. 1286 – 1292, 2009.

[54] L. Huang, Q. Kemao, B. Pan, and A. K. Asundi, "Comparison of fourier transform, windowed fourier transform, and wavelet transform methods for phase extraction from a single fringe pattern in fringe projection profilometry," *Optics and Lasers in Engineering*, vol. 48, no. 2, pp. 141 – 148, 2010. Fringe Projection Techniques.

[55] Y. Takeda, Y. Oshida, and Y. Miyamura, "Random phase shifters for fourier transformed holograms," *Appl. Opt.*, vol. 11, pp. 818–822, Apr 1972.

[56] S. Lai, B. King, and M. A. Neifeld, "Wave front reconstruction by means of phase-shifting digital in-line holography," *Optics Communications*, vol. 173, no. 1–6, pp. 155 – 160, 2000.

[57] Y. Zhang, Q. Lu, and B. Ge, "Elimination of zero-order diffraction in digital off-axis holography," *Optics Communications*, vol. 240, no. 4–6, pp. 261 – 267, 2004.

[58] G.-L. Chen, C.-Y. Lin, M.-K. Kuo, and C.-C. Chang, "Numerical suppression of zero-order image in digital holography," *Opt. Express*, vol. 15, pp. 8851–8856, Jul 2007.

[59] C. Roddier and F. Roddier, "Interferogram analysis using fourier transform techniques," *Appl. Opt.*, vol. 26, pp. 1668–1673, May 1987.

[60] K. A. Goldberg and J. Bokor, "Fourier-transform method of phase-shift determination," *Appl. Opt.*, vol. 40, pp. 2886–2894, Jun 2001.

[61] E. Hu and Y. He, "Surface profile measurement of moving objects by using an improved π phase-shifting fourier transform profilometry," *Optics and Lasers in Engineering*, vol. 47, no. 1, pp. 57 – 61, 2009.

[62] F. Bai, F. Han, Y. Xu, X. Bao, and S. Gan, "Modified fourier-transform method for phase-shift calibration," *Optics and Lasers in Engineering*, vol. 49, no. 7, pp. 932 – 936, 2011.

[63] M. Shan, B. Hao, Z. Zhong, M. Diao, and Y. Zhang, "Parallel two-step spatial carrier phase-shifting common-path interferometer with a ronchi grating outside the fourier plane," *Opt. Express*, vol. 21, pp. 2126–2132, Jan 2013.

[64] S. Liu, L. Zhong, F. Zhang, and X. Lu, "Frequency domain phase extraction algorithm for phase-shifting interferometry with random phase-shifting amount and low-sampling rate," *Optics Communications*, vol. 291, no. 0, pp. 150 – 154, 2013.

[65] D. Lovrić, Z. Vučić, J. Gladić, N. Demoli, S. Mitrović, and M. Milas, "Refined fourier-transform method of analysis of full two-dimensional digitized interferograms," *Appl. Opt.*, vol. 42, pp. 1477–1484, Mar 2003.

[66] P. J. Tavares and M. A. Vaz, “Orthogonal projection technique for resolution enhancement of the fourier transform fringe analysis method,” *Optics Communications*, vol. 266, no. 2, pp. 465 – 468, 2006.

[67] P. J. Tavares and M. A. Vaz, “Single image orthogonal fringe technique for resolution enhancement of the fourier transform fringe analysis method,” *Optics Communications*, vol. 290, no. 0, pp. 33 – 36, 2013.

[68] S. Li, X. Su, W. Chen, and L. Xiang, “Eliminating the zero spectrum in fourier transform profilometry using empirical mode decomposition,” *J. Opt. Soc. Am. A*, vol. 26, pp. 1195–1201, May 2009.

[69] F. Da and H. Huang, “A novel color fringe projection based fourier transform 3d shape measurement method,” *Optik - International Journal for Light and Electron Optics*, vol. 123, no. 24, pp. 2233 – 2237, 2012.

[70] S. lin Cui, F. Tian, and D. hua Li, “Method to eliminate the zero spectra in fourier transform profilometry based on a cost function,” *Appl. Opt.*, vol. 51, pp. 3194–3204, Jun 2012.

[71] R. Juarez-Salazar, C. Robledo-Sanchez, C. Meneses-Fabian, F. Guerrero-Sanchez, and L. A. Aguilar, “Generalized phase-shifting interferometry by parameter estimation with the least squares method,” *Optics and Lasers in Engineering*, vol. 51, no. 5, pp. 626 – 632, 2013.

[72] A. Patil and P. Rastogi, “Moving ahead with phase,” *Optics and Lasers in Engineering*, vol. 45, no. 2, pp. 253 – 257, 2007. Phase Measurement Techniques and their applications.

[73] X. Xu, G. Lu, G. Han, F. Gao, Z. Jiao, and D. Li, “Phase stitching and error correction in aperture synthesis for generalized phase-shifting interferometry,” *Appl. Opt.*, vol. 52, pp. 4864–4870, Jul 2013.

[74] X. Chen, M. Gramaglia, and J. A. Yeazell, “Phase-shift calibration algorithm for phase-shifting interferometry,” *J. Opt. Soc. Am. A*, vol. 17, pp. 2061–2066, Nov 2000.

[75] K. Creath, “Phase measurement interferometry techniques,” in *Progress in optics* (E. Wolf, ed.), vol. 26, pp. 349–393, Elsevier Science Publishers, 1988.

[76] J. van Wingerden, H. J. Frankena, and C. Smorenburg, “Linear approximation for measurement errors in phase shifting interferometry,” *Appl. Opt.*, vol. 30, pp. 2718–2729, Jul 1991.

[77] L. L. Deck, “Suppressing phase errors from vibration in phase-shifting interferometry,” *Appl. Opt.*, vol. 48, pp. 3948–3960, Jul 2009.

[78] C. J. Morgan, “Least-squares estimation in phase-measurement interferometry,” *Opt. Lett.*, vol. 7, pp. 368–370, Aug 1982.

[79] J. E. Greivenkamp, “Generalized data reduction for heterodyne interferometry,” *Optical Engineering*, vol. 23, no. 4, pp. 234350–234352, 1984.

[80] G. Lai and T. Yatagai, “Generalized phase-shifting interferometry,” *J. Opt. Soc. Am. A*, vol. 8, pp. 822–827, May 1991.

[81] P. Carre, “Installation et utilisation du comparateur photohlectrique et interfhrentiel du bureau international des poids et mesures,” *Metrologia*, vol. 2, no. 1, 1966.

[82] P. Hariharan, B. F. Oreb, and T. Eiju, “Digital phase-shifting interferometry: a simple error-compensating phase calculation algorithm,” *Appl. Opt.*, vol. 26, pp. 2504–2506, Jul 1987.

[83] C. T. Farrell and M. A. Player, “Phase-step insensitive algorithms for phase-shifting interferometry,” *Measurement Science and Technology*, vol. 5, no. 6, p. 648, 1994.

[84] X. Chen, M. Gramaglia, and J. A. Yeazell, “Phase-shifting interferometry with uncalibrated phase shifts,” *Appl. Opt.*, vol. 39, pp. 585–591, Feb 2000.

[85] A. Patil and P. Rastogi, “Approaches in generalized phase shifting interferometry,” *Optics and Lasers in Engineering*, vol. 43, no. 3-5, pp. 475–490, 2005.

[86] Z. Wang and B. Han, “Advanced iterative algorithm for phase extraction of randomly phase-shifted interferograms,” *Opt. Lett.*, vol. 29, pp. 1671–1673, Jul 2004.

[87] M. Chen, H. Guo, and C. Wei, “Algorithm immune to tilt phase-shifting error for phase-shifting interferometers,” *Appl. Opt.*, vol. 39, pp. 3894–3898, Aug 2000.

[88] A. Dobroiu, D. Apostol, V. Nascov, and V. Damian, “Tilt-compensating algorithm for phase-shift interferometry,” *Appl. Opt.*, vol. 41, pp. 2435–2439, May 2002.

[89] K. Patorski, A. Styk, L. Bruno, and P. Szwaykowski, “Tilt-shift error detection in phase-shifting interferometry,” *Opt. Express*, vol. 14, pp. 5232–5249, Jun 2006.

[90] F. Zeng, Q. Tan, H. Gu, and G. Jin, “Phase extraction from interferograms with unknown tilt phase shifts based on a regularized optical flow method,” *Opt. Express*, vol. 21, pp. 17234–17248, Jul 2013.

[91] J. Xu, Q. Xu, and L. Chai, “Tilt-shift determination and compensation in phase-shifting interferometry,” *Journal of Optics A: Pure and Applied Optics*, vol. 10, no. 7, p. 075011, 2008.

[92] J. Xu, Q. Xu, and L. Chai, “Iterative algorithm for phase extraction from interferograms with random and spatially nonuniform phase shifts,” *Appl. Opt.*, vol. 47, pp. 480–485, Jan 2008.

[93] O. Soloviev and G. Vdovin, “Phase extraction from three and more interferograms registered with different unknown wavefront tilts,” *Opt. Express*, vol. 13, pp. 3743–3753, May 2005.

[94] K. Hibino, B. F. Oreb, D. I. Farrant, and K. G. Larkin, “Phase-shifting algorithms for nonlinear and spatially nonuniform phase shifts,” *J. Opt. Soc. Am. A*, vol. 14, pp. 918–930, Apr 1997.

[95] M. Afifi, K. Nassim, and S. Rachafi, “Five-frame phase-shifting algorithm insensitive to diode laser power variation,” *Optics Communications*, vol. 197, no. 1-3, pp. 37 – 42, 2001.

[96] 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, p. 953, 1992.

[97] F. L. Bookstein, “Fitting conic sections to scattered data,” *Computer Graphics and Image Processing*, vol. 9, no. 1, pp. 56 – 71, 1979.

[98] A. Fitzgibbon, M. Pilu, and R. Fisher, “Direct least square fitting of ellipses,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 21, pp. 476–480, may 1999.

[99] L. Z. Cai, Q. Liu, and X. L. Yang, “Phase-shift extraction and wave-front reconstruction in phase-shifting interferometry with arbitrary phase steps,” *Opt. Lett.*, vol. 28, pp. 1808–1810, Oct 2003.

[100] L. Z. Cai, Q. Liu, and X. L. Yang, “Generalized phase-shifting interferometry with arbitrary unknown phase steps for diffraction objects,” *Opt. Lett.*, vol. 29, pp. 183–185, Jan 2004.

[101] J. Xu, Y. Li, H. Wang, L. Chai, and Q. Xu, “Phase-shift extraction for phase-shifting interferometry by histogram of phase difference,” *Opt. Express*, vol. 18, pp. 24368–24378, Nov 2010.

[102] G.-S. Han and S.-W. Kim, “Numerical correction of reference phases in phase-shifting interferometry by iterative least-squares fitting,” *Appl. Opt.*, vol. 33, pp. 7321–7325, Nov 1994.

[103] J. Li, Y. Wang, X. Meng, X. Yang, and Q. Wang, "An evaluation method for phase shift extraction algorithms in generalized phase-shifting interferometry," *Journal of Optics*, vol. 15, no. 10, p. 105408, 2013.

[104] H. Guo, "Blind self-calibrating algorithm for phase-shifting interferometry by use of cross-bispectrum," *Opt. Express*, vol. 19, pp. 7807–7815, Apr 2011.

[105] R. Juarez-Salazar, C. Robledo-Sanchez, and F. Guerrero-Sanchez, "Phase-unwrapping algorithm by a rounding-least-squares approach," *Optical Engineering*, vol. 53, no. 2, p. 024102, 2014.

[106] E. Zappa and G. Busca, "Comparison of eight unwrapping algorithms applied to fourier-transform profilometry," *Optics and Lasers in Engineering*, vol. 46, no. 2, pp. 106 – 116, 2008.

[107] D. C. Ghiglia and M. D. Pritt, *Two-dimensional phase unwrapping. Theory, algorithms, and software.* New York: John Wiley & Sons, Inc., 1998.

[108] J. M. Huntley and H. Saldner, "Temporal phase-unwrapping algorithm for automated interferogram analysis," *Appl. Opt.*, vol. 32, pp. 3047–3052, Jun 1993.

[109] 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.

[110] J. Burke and H. Helmers, "Matched data storage in espi by combination of spatial phase shifting with temporal phase unwrapping," *Optics & Laser Technology*, vol. 32, no. 4, pp. 235 – 240, 2000.

[111] A. Baldi, F. Bertolino, and F. Ginesu, "A temporal phase unwrapping algorithm for photoelastic stress analysis," *Optics and Lasers in Engineering*, vol. 45, no. 5, pp. 612 – 617, 2007. Optical methods in experimental mechanics.

[112] H. O. Saldner and J. M. Huntley, "Temporal phase unwrapping: application to surface profiling of discontinuous objects," *Appl. Opt.*, vol. 36, pp. 2770–2775, May 1997.

[113] J. Tian, X. Peng, and X. Zhao, "A generalized temporal phase unwrapping algorithm for three-dimensional profilometry," *Optics and Lasers in Engineering*, vol. 46, no. 4, pp. 336 – 342, 2008.

[114] G. Fornaro, G. Franceschetti, R. Lanari, E. Sansosti, and M. Tesauro, "Global and local phase-unwrapping techniques: a comparison," *J. Opt. Soc. Am. A*, vol. 14, pp. 2702–2708, Oct 1997.

[115] K. Itoh, "Analysis of the phase unwrapping algorithm," *Appl. Opt.*, vol. 21, pp. 2470–2470, Jul 1982.

[116] J. Strand and T. Taxt, "Performance evaluation of two-dimensional phase unwrapping algorithms," *Appl. Opt.*, vol. 38, pp. 4333–4344, Jul 1999.

[117] B. Osmanoglu, T. H. Dixon, S. Wdowinski, and E. Cabral-Cano, "On the importance of path for phase unwrapping in synthetic aperture radar interferometry," *Appl. Opt.*, vol. 50, pp. 3205–3220, Jul 2011.

[118] D. Zheng and F. Da, "A novel algorithm for branch cut phase unwrapping," *Optics and Lasers in Engineering*, vol. 49, no. 5, pp. 609 – 617, 2011.

[119] Q. Kemao, W. Gao, and H. Wang, "Windowed fourier-filtered and quality-guided phase-unwrapping algorithm," *Appl. Opt.*, vol. 47, pp. 5420–5428, Oct 2008.

[120] Y.-T. Zhang, M.-J. Huang, H.-R. Liang, and F.-Y. Lao, "Branch cutting algorithm for unwrapping photoelastic phase map with isotropic point," *Optics and Lasers in Engineering*, vol. 50, no. 5, pp. 619 – 631, 2012.

[121] Z. Sen, Z. Heping, and T. Jinsong, "Dendriform branch cut algorithm based on minimum spanning tree for phase unwrapping," *Procedia Engineering*, vol. 29, no. 0, pp. 1154 – 1159, 2012. 2012 International Workshop on Information and Electronics Engineering.

[122] S. Liu, W. Xiao, F. Pan, F. Wang, and L. Cong, "Complex-amplitude-based phase unwrapping method for digital holographic microscopy," *Optics and Lasers in Engineering*, vol. 50, no. 3, pp. 322 – 327, 2012.

[123] Y. Lu, W. Zhao, X. Zhang, W. Xu, and G. Xu, "Weighted-phase-gradient-based quality maps for two-dimensional quality-guided phase unwrapping," *Optics and Lasers in Engineering*, vol. 50, no. 10, pp. 1397 – 1404, 2012.

[124] L. Chen, C. Quan, C. J. Tay, and Y. Huang, "Fringe contrast-based 3d profilometry using fringe projection," *Optik - International Journal for Light and Electron Optics*, vol. 116, no. 3, pp. 123 – 128, 2005.

[125] W. Li and S. Fang, "Reliability-guided phase unwrapping algorithm following noncontinuous path based on color fringe projection," *Optik - International Journal for Light and Electron Optics*, vol. 123, no. 6, pp. 537 – 546, 2012.

[126] K. Qian, S. H. Soon, and A. Asundi, "A simple phase unwrapping approach based on filtering by windowed fourier transform," *Optics & Laser Technology*, vol. 37, no. 6, pp. 458 – 462, 2005.

[127] F. Qian, X. Wang, X. Wang, and Y. Bu, "Adaptive filter for unwrapping noisy phase image in phase-stepping interferometry," *Optics & Laser Technology*, vol. 33, no. 7, pp. 479 – 486, 2001.

[128] H. A. Zebker and Y. Lu, "Phase unwrapping algorithms for radar interferometry: residue-cut, least-squares, and synthesis algorithms," *J. Opt. Soc. Am. A*, vol. 15, pp. 586–598, Mar 1998.

[129] K. A. Stetson, "Are residues of primary importance in phase unwrapping?," in *Fringe 2005* (W. Osten, ed.), pp. 38–45, Springer Berlin Heidelberg, 2006.

[130] P. D. Ruiz, G. H. Kaufmann, and G. E. Galizzi, "Unwrapping of digital speckle-pattern interferometry phase maps by use of a minimum l^0-norm algorithm," *Appl. Opt.*, vol. 37, pp. 7632–7644, Nov 1998.

[131] D. C. Ghiglia and L. A. Romero, "Minimum lp-norm two-dimensional phase unwrapping," *J. Opt. Soc. Am. A*, vol. 13, pp. 1999–2013, Oct 1996.

[132] W. Shi, Y. Zhu, and Y. Yao, "Discussion about the dct/fft phase-unwrapping algorithm for interferometric applications," *Optik - International Journal for Light and Electron Optics*, vol. 121, no. 16, pp. 1443 – 1449, 2010.

[133] D. C. Ghiglia and L. A. Romero, "Robust two-dimensional weighted and unweighted phase unwrapping that uses fast transforms and iterative methods," *J. Opt. Soc. Am. A*, vol. 11, pp. 107–117, Jan 1994.

[134] S. Tomioka and S. Nishiyama, "Phase unwrapping for noisy phase map using localized compensator," *Appl. Opt.*, vol. 51, pp. 4984–4994, Jul 2012.

[135] J. Arines, "Least-squares modal estimation of wrapped phases: Application to phase unwrapping," *Appl. Opt.*, vol. 42, pp. 3373–3378, Jun 2003.

[136] Z.-P. Liang, "A model-based method for phase unwrapping," *Medical Imaging, IEEE Transactions on*, vol. 15, pp. 893 –897, dec 1996.

[137] Y. Lu, X. Wang, and X. Zhang, "Weighted least-squares phase unwrapping algorithm based on derivative variance correlation map," *Optik - International Journal for Light and Electron Optics*, vol. 118, no. 2, pp. 62 – 66, 2007.

[138] H. Kadono, H. Takei, and S. Toyooka, "A noise-immune method of phase unwrapping in speckle interferometry," *Optics and Lasers in Engineering*, vol. 26, no. 2-3, pp. 151 – 164, 1997. Speckle and Speckle Shearing Interferometry-I.

[139] H. Y. H. Huang, L. Tian, Z. Zhang, Y. Liu, Z. Chen, and G. Barbastathis, "Path-independent phase unwrapping using phase gradient and total-variation (tv) denoising," *Opt. Express*, vol. 20, pp. 14075–14089, Jun 2012.

[140] J. C. M. Morales, C. A. Ramos-Arreguan, A. Sotomayor-Olmedo, E. Gorrostieta-Hurtadoa, J. M. Ramos-Arreguana, J. C. Pedraza-Ortegaa, S. Tovar-Arriagaa, and J. E. Vargas-Sotoa, "A strategy for 3d object digitalization using pre-filtering and post-filtering stages," *Procedia Technology*, vol. 3, no. 0, pp. 273 – 281, 2012. The 2012 Iberoamerican Conference on Electronics Engineering and Computer Science.

[141] X. Y. He, X. Kang, C. J. Tay, C. Quan, and H. M. Shang, "Proposed algorithm for phase unwrapping," *Appl. Opt.*, vol. 41, pp. 7422–7428, Dec 2002.

[142] M. Costantini, "A novel phase unwrapping method based on network programming," *Geoscience and Remote Sensing, IEEE Transactions on*, vol. 36, pp. 813 –821, may 1998.

[143] M. Harker and P. O'Leary, "Direct regularized surface reconstruction from gradients for industrial photometric stereo," *Computers in Industry*, vol. 64, no. 9, pp. 1221 – 1228, 2013. Special Issue: 3D Imaging in Industry.

[144] R. F. Saracchini, J. Stolfi, H. C. L. ao, G. A. Atkinson, and M. L. Smith, "A robust multi-scale integration method to obtain the depth from gradient maps," *Computer Vision and Image Understanding*, vol. 116, no. 8, pp. 882 – 895, 2012.

[145] M. Harker and P. O'Leary, "Least squares surface reconstruction from measured gradient fields," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pp. 1–7, 2008.

[146] C. Lu, "Solution of the matrix equation ax + xb = c," *Electronics Letters*, vol. 7, no. 8, pp. 185–186, 1971.

[147] R. H. Bartels and G. W. Stewart, "Solution of the matrix equation ax + xb = c [f4]," *Commun. ACM*, vol. 15, pp. 820–826, Sept. 1972.

[148] G. Golub and C. Van Loan, *Matrix Computations.* Johns Hopkins Studies in the Mathematical Sciences, Johns Hopkins University Press, third ed., 1996.

[149] G. Golub, S. Nash, and C. Van Loan, "A hessenberg-schur method for the problem $ax + xb = c$," *Automatic Control, IEEE Transactions on*, vol. 24, no. 6, pp. 909–913, 1979.

[150] P. Ettl and K. Creath, "Comparison of phase-unwrapping algorithms by using gradient of first failure," *Appl. Opt.*, vol. 35, pp. 5108–5114, Sep 1996.

[151] C. Tay, C. Quan, and H. Shang, "Shape identification using phase shifting interferometry and liquid-crystal phase modulator," *Optics & Laser Technology*, vol. 30, no. 8, pp. 545 – 550, 1998.

[152] G. Rodriguez-Zurita, C. Meneses-Fabian, N.-I. Toto-Arellano, J. F. Vázquez-Castillo, and C. Robledo-Sánchez, "One-shot phase-shifting phase-grating interferometry with modulation of polarization: case of four interferograms," *Opt. Express*, vol. 16, pp. 7806–7817, May 2008.

[153] M. Atlan, M. Gross, and E. Absil, "Accurate phase-shifting digital interferometry," *Opt. Lett.*, vol. 32, pp. 1456–1458, Jun 2007.

[154] J. Min, B. Yao, P. Gao, R. Guo, J. Zheng, and T. Ye, "Parallel phase-shifting interferometry based on michelson-like architecture," *Appl. Opt.*, vol. 49, pp. 6612–6616, Dec 2010.

[155] T. Pfeifer, R. Tutsch, J. Evertz, and G. Weres, "Generalized aspects of multiple-wavelength techniques in optical metrology," *CIRP Annals - Manufacturing Technology*, vol. 44, no. 1, pp. 493 – 496, 1995.

[156] M. Vannoni, A. Sordini, and G. Molesini, “He-ne laser wavelength-shifting interferometry,” *Optics Communications*, vol. 283, no. 24, pp. 5169 – 5172, 2010.

[157] J. Vargas, J. A. Quiroga, A. Álvarez Herrero, and T. Belenguer, “Phase-shifting interferometry based on induced vibrations,” *Opt. Express*, vol. 19, pp. 584–596, Jan 2011.

[158] J. W. Goodman, *Introduction to Fourier optics*. Roberts & Company Publishers, Englewood, Colorado, third ed., 2005.

Index