We are IntechOpen,
the world's leading publisher of
Open Access books
Built by scientists, for scientists

4,800
Open access books available

122,000
International authors and editors

135M
Downloads

154
Countries delivered to

Our authors are among the

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities

WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com

Fourier Transform Profilometry in LabVIEW

Andrés G. Marrugo, Jesús Pineda, Lenny A. Romero,
Raúl Vargas and Jaime Meneses

Additional information is available at the end of the chapter

Abstract

Fourier transform profilometry (FTP) is an established non-contact method for 3D sensing in many scientific and industrial applications, such as quality control and biomedical imaging. This phase-based technique has the advantages of high resolution and noise robustness compared to intensity-based approaches. In FTP, a sinusoidal grating is projected onto the surface of an object, the shape information is encoded into a deformed fringe pattern recorded by a camera. The object shape is decoded by calculating the Fourier transform, filtering in the spatial frequency domain, and calculating the inverse Fourier transform; afterward, a conversion of the measured phase to object height is carried out. FTP has been extensively studied and extended for achieving better slope measurement, better separation of height information from noise, and robustness to discontinuities in the fringe pattern. Most of the literature on FTP disregards the software implementation aspects. In this chapter, we return to the basics of FTP and explain in detail the software implementation in LabVIEW, one of the most used data acquisition platforms in engineering. We show results on three applications for FTP in 3D metrology.

Keywords: 3D reconstruction, Fourier transform profilometry, FTP, LabVIEW

1. Introduction

Three-dimensional (3D) shape measurement techniques are widely used in many different fields such as mechanical engineering, industry monitoring, robotics, biomedicine, dressmaking, among others [1]. These techniques can be classified as passive, like in stereo vision in which two or more cameras are used to obtain the 3D reconstruction of a scene, or as active, like in fringe projection profilometry (FPP) in which a projection device is used to project a pattern onto the object to be reconstructed. When compared with other 3D measurement techniques, FPP has the advantages of high measurement accuracy and high density. There are two types of FPP

methods: phase shifting and Fourier-transform profilometry (FTP). Phase-shifting methods offer high-resolution measurement at the expense of projecting several patterns onto the object [2–4], whereas FTP is popular because only one deformed fringe pattern image is needed [5]. For this reason, FTP has been used in many dynamic applications [6] such as vibration measurement of micromechanical devices [7] and measurement of real-time deformation fields [8].

FTP was proposed by Takeda et al. [5, 9] in 1982 and has since become one of the most used methods [3, 10]. Its main advantages are full-field analysis, high precision, noise-robustness [11], among others. In FTP, a Ronchi grating, or a sinusoidal grating, or a fringe pattern from a digital projector is projected onto an object, and the depth information of the object is encoded into the deformed fringe pattern recorded by an image acquisition device as shown in **Figure 1**. The surface shape can be decoded by calculating the Fourier transform, filtering in the spatial frequency domain, and calculating the inverse Fourier transform. Compared with other fringe analysis methods, FTP can accomplish a fully automatic distinction between a depression and an elevation of the object shape. It requires no fringe order assignments or fringe center determination, and it needs no interpolation between fringes because it gives height distribution at each pixel over the entire field. Since FTP requires only one or two images of the deformed fringe pattern, it has become one of the most popular methods for real-time 3D reconstruction of dynamic scenes.

Although FTP has been extensively studied and used in many applications, to the best of our knowledge a complete reference in which the implementation details are fully described is nonexistent. In this chapter, we describe the FTP fundamentals and the implementation of an FTP system in LabVIEW one of the most used engineering development platforms for data acquisition and laboratory automation. The chapter is organized as follows. In Section 2 we describe the FTP fundamentals and a general calibration method, in Section 3 we describe how FTP is implemented in LabVIEW, and finally in Section 4 we show three applications of FTP for 3D reconstruction.
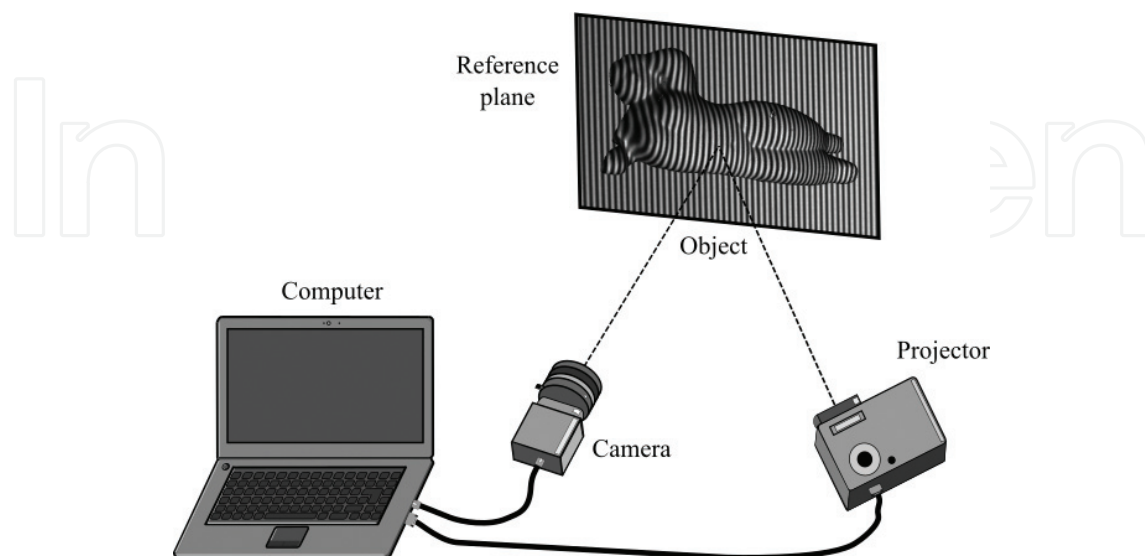
Figure 1. Fringe projection system.

2. FTP fundamentals

There are many implementations of FPP. However, all share the same underlying principle. A typical FPP setup consists of a projection device and a camera as shown in **Figure 1**. A fringe pattern is projected onto a test object, and the resulting image is acquired by the camera from a different direction. The acquired fringe pattern image is distorted according to the object shape. In terms of information theory, it is said that the object shape is encoded into a deformed fringe pattern acquired by the camera. The object shape is recovered/decoded by comparison to the original (undeformed) fringe pattern image. Therefore, the phase shift between the reference and the deformed image contains the information of the object shape.

By projecting a fringe pattern onto the reference plane, the fringe pattern (with period $p_0 = 1/f_0$) on the reference plane observed through the camera can be modeled as

$$g_0(x, y) = a_0(x, y) + b_0(x, y)\cos\big[2\pi f_0 x + \phi_0(x, y)\big]. \tag{1}$$

Likewise, when the object is placed on the reference plane, the deformed fringe pattern observed through the camera is given by

$$g(x, y) = a(x, y) + b(x, y)\cos\big[2\pi f_0 x + \phi(x, y)\big], \tag{2}$$

where $a_0(x, y)$ and $a(x, y)$ represent the non-uniform background illumination, $b_0(x, y)$ and $b(x, y)$ the contrast of the fringe pattern. f_0 is the fundamental frequency of the observed fringe pattern (also called carrier frequency). $\phi_0(x, y)$ and $\phi(x, y)$ are the original phase modulation on the reference plane R where $z(x, y) = 0$ and the phase modulations resulting from the object height distribution, respectively. $a(x, y)$, $b(x, y)$ and $\phi(x, y)$ are assumed to vary much slower than the spatial carrier frequency f_0. The principle of FTP is shown schematically in **Figure 2**. The input fringe pattern from Eqs. (1) and (2) can be rewritten using Euler's formula in the following form
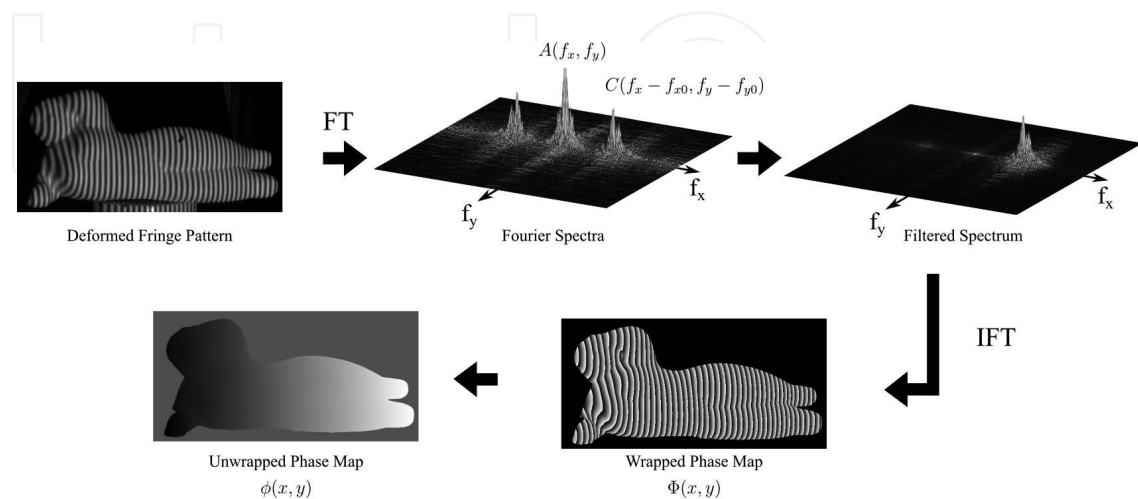
Figure 2. Principle of the filtering via Fourier transform (FT) method. IFT, inverse FT.

$$g(x,y) = a(x,y) + c(x,y)\exp\{2\pi i f_0 x\} + c^*(x,y)\exp\{-2\pi i f_0 x\}, \tag{3}$$

with

$$c(x,y) = \frac{1}{2}b(x,y)\exp\{i\phi(x,y)\}, \tag{4}$$

where $*$ denotes a complex conjugate.

Next, the phase of the fringe patterns is recovered using the Fourier Transform method. Using one-dimensional notation for simplicity, when we compute the Fourier transform of Eqs. (1) and (2) the Fourier spectrum of the fringe signals splits intro three spectrum components separated from each other, which gives

$$G(f_x,y) = A(f_x,y) + C(f_x - f_0,y) + C^*(f_x + f_0,y), \tag{5}$$

as shown in two dimensions in **Figure 2**. With an appropriate filter function, for instance, a Hanning filter, the spectra are filtered to let only the fundamental component $C(f_x - f_0,y)$. A Hanning window is given by [11],

$$H(f_x) = 0.50\left[1 + \cos\left(\beta\pi\frac{f_x - f_0}{f_c}\right)\right], \tag{6}$$

where f_c is the cutoff frequency at a 50% attenuation ratio, $\beta = 1/2$ and f_x varies from $f_0 - f_c/\beta$ to $f_0 + f_c/\beta$. The inverse Fourier Transform is applied to the filtered component, and a complex signal is obtained

$$\widehat{g}_0(x,y) = \frac{1}{2}b(x,y)\exp\{i(2\pi f_0 x + \phi_0(x,y))\}, \tag{7}$$

$$\widehat{g}(x,y) = \frac{1}{2}b(x,y)\exp\{i(2\pi f_0 x + \phi(x,y))\}. \tag{8}$$

The variable related to height distribution is the phase change $\Delta\phi(x,y)$ [9]:

$$\Delta\phi(x,y) = \Phi(x,y) - \Phi_0(x,y) = \phi(x,y) - \phi_0(x,y), \tag{9}$$

with

$$\Phi_0(x,y) = \tan^{-1}\left(\frac{\mathfrak{I}[\widehat{g}_0(x,y)]}{\mathfrak{R}[\widehat{g}_0(x,y)]}\right), \tag{10}$$

$$\Phi(x,y) = \tan^{-1}\left(\frac{\mathfrak{I}[\widehat{g}(x,y)]}{\mathfrak{R}[\widehat{g}(x,y)]}\right), \tag{11}$$

where $\mathfrak{I}[.]$ and $\mathfrak{R}[.]$ denote the imaginary and the real part, respectively. The phases obtained from Eqs. (10) and (11) are wrapped into the principal value $[-\pi, \pi]$. The wrapped phase is

unwrapped by using a suitable phase unwrapping algorithm [12] that gives the desired phase map as shown in **Figure 2**. The phase map $\Delta\phi(x, y)$ is proportional to the height of the object surface.

2.1. System calibration

The calibration of FPP systems plays an essential role in the accuracy of the 3D reconstructions. Here we describe a simple yet extensively used calibration called the reference-plane-based technique, i.e., to convert the unwrapped phase map $\Delta\phi(x, y)$ to height z.

The optical axis geometry of the FTP measurement system is depicted in **Figure 3**. The optical axis $E'_p - E_p$ of a projector lens crosses the optical axis $E'_c - E_c$ of a camera lens at a point O on a reference plane R. This reference plane is normal to the optical axis $E'_c - E_c$ and serves as a reference to measure the height of the object $z(x, y)$. d is the distance between the projector and the camera, l_0 is the distance between the camera and the reference plane. The fringe pattern image (with period p) is formed by the projector lens on plane I through point O. p is related to the carrier frequency by $f_0 = 1/p_0 = \cos\theta/p$. The height of the object surface is measured relative to R. From the point of view of the projector, point A on the object surface has the same phase value as point C on the reference plane R, $\Phi_A = \Phi_C^R$, where the superindex R denotes a point on the reference plane. On the camera sensor, point A on the object surface and point D on the reference plane are imaged on the same pixel. By subtracting the reference phase map from the object phase map, we obtain the phase difference at this specific pixel
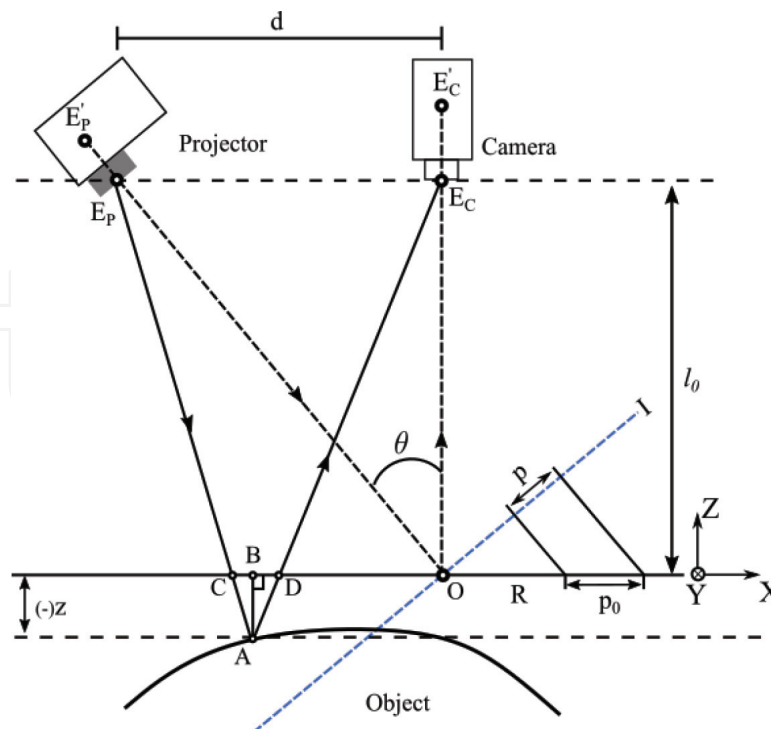
Figure 3. Fringe projection system.

$$\Delta\Phi_{AD} = \Phi_A - \Phi_D^R = \Phi_C^R - \Phi_D^R = \Phi_{CD}^R. \tag{12}$$

The triangles $\Delta E_p E_c A$ and ΔCDA are similar, and the height $\overline{AB}$ of point A on the object surface relative to the reference plane can be related to the distance between points C and D

$$\Delta z(x,y) = \overline{AB} \approx \frac{l_0}{d}\overline{CD} \propto \Delta\Phi_{CD}^R = \Phi_A - \Phi_D^R. \tag{13}$$

Combining Eqs. (12) and (13) a proportional relation between the phase map and the surface height can be obtained for any point (x,y)
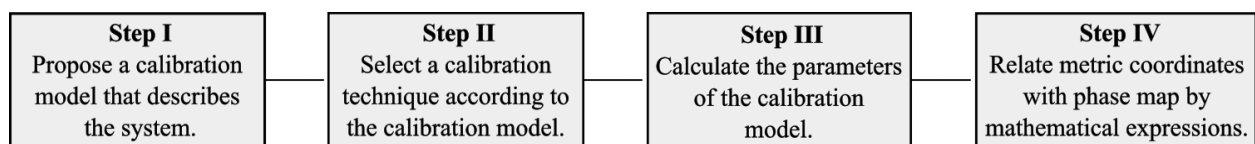
$$\Delta z(x,y) \propto \Delta\phi(x,y) = \Phi(x,y) - \Phi_0(x,y), \tag{14}$$

where $\Phi(x,y)$ is the object phase map and $\Phi_0(x,y)$ is the reference plane phase map. Assuming the reference plane has a depth of z_0, the depth value for each measured point can be represented as

$$z(x,y) = z_0 + k_0 \times [\Phi(x,y) - \Phi_0], \tag{15}$$

where k_0 is a constant determined through calibration and z_0 is usually set to 0.

We have shown how the object surface height is related to the recovered phase through FTP. The model described by Eq. (15) has many underlying assumptions and is often extended to cover more degrees of freedom. Moreover, a general calibration process in FPP can be carried out employing the methodology shown in **Figure 4**. First, we propose a model that best describes the system, while also considering metrological requirements such as speed, robustness, accuracy, flexibility and reconstruction scale. Some authors have proposed to use several calibration models based on polynomial or fractional fitting functions [13, 14], bilinear interpolation by look-up table (LUT) [15] and stereo triangulation [16–18]. These calibration models require different strategies or techniques that allow relating metric coordinates with phase values. In step II, we select or design a strategy that fits the proposed calibration model and characteristics of the elements to a given experimental setup, such as the type of projector (i.e., analog or digital projection) and camera (i.e., monochrome or color). These strategies consist in projecting and capturing fringe patterns onto 3D-objects [19] or 2D-targets [16, 20] with highly accurate known measurements. In some cases, the calibration consists in displacing the targets along the z axis using a linear translation stage [19]. The purpose is to obtain a correspondence between a metric coordinate system and the phase images captured with the camera. In step III, the correspondences are used to calculate the parameters that are part of the proposed

Step I Propose a calibration model that describes the system.	**Step II** Select a calibration technique according to the calibration model.	**Step III** Calculate the parameters of the calibration model.	**Step IV** Relate metric coordinates with phase map by mathematical expressions.

Figure 4. General calibration methodology.

model, and the best data obtained in step II. Finally in step IV, with the complete model, we can find mathematical expressions that convert phase maps to XYZ-coordinates.

3. LabVIEW implementation

In this section, we explain the details of the FTP software implementation in LabVIEW. LabVIEW stands for Laboratory Virtual Instrument Engineering Workbench and is a system-design platform and development environment for a visual programming language from National Instruments [21]. It allows integrating hardware, acquiring and analyzing data, and sharing results. Because it is a visual programming language based on function blocks, it is a highly intuitive integrated development environment (IDE) for engineers and scientists familiar with block diagrams and flowcharts. Every LabVIEW block diagram also has an associated front panel, which is the user interface of the application.

The acquisition and processing strategies described in this section require the installation of the following software components:

- **NI vision acquisition software**, which installs **NI-IMAQdx**. This software driver allows the integration of cameras with different control protocols such as USB3 Vision, GigE Vision devices, IEEE 1394 cameras compatible with IIDC, IP (Ethernet) and DirectShow compatible USB devices (e.g., cameras, webcams, microscopes, scanners). **NI vision acquisition software** also includes the driver **NI-IMAQ** for acquiring from analog cameras, digital parallel and Camera Link, as well as NI Smart Cameras. This hardware compatibility is the main advantage of using LabVIEW for vision systems. This compatibility greatly facilitates the development of applications for different types of cameras and busses.

- **NI vision development module (VDM)**. This package provides machine vision and image processing functions. It includes IMAQ Vision, a library of powerful functions for vision processing. In this library, there is a group of VIs that analyze and process images in the frequency domain. We will make use of these functions throughout the entire chapter.

NI VDM and Vision Acquisition Software are supported on the following operating systems:

• Windows 10; Windows 8.1; Windows 7 (SP1) 32-bit; Windows 7 (SP1) 64-bit; Windows Embedded Standard 7 (SP1); Windows Server 2012 R2 64-bit; Windows Server 2008 R2 (SP1) 64-bit.

3.1. Image acquisition

There are two primary ways to obtain images in LabVIEW: loading an image file or acquiring directly from a camera. The wiring diagram in **Figure 5(a)** illustrates how to perform a continuous (grab) acquisition in LabVIEW using Vision Acquisition Software. A Grab acquisition begins by initializing the camera specified by the Camera Name Control and configuring the driver for acquiring images continuously. Using IMAQ Create, we create a temporary
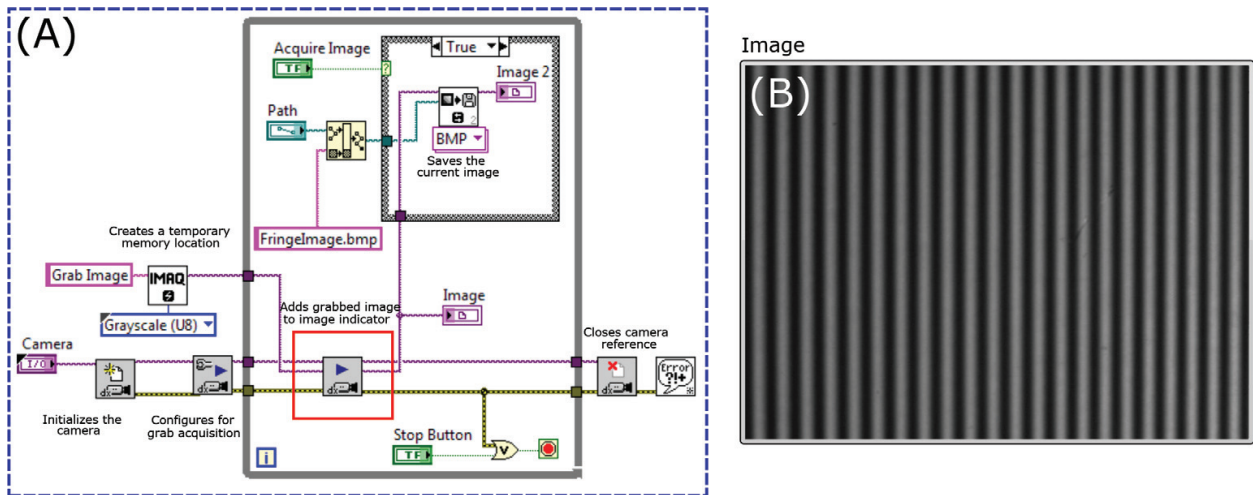
Figure 5. Grab acquisition in LabVIEW. (a) Block diagram. (b) Image indicator in front panel.

memory location for the acquired image. This function returns an IMAQ image reference to the buffer in memory where the image is stored. The reference is the input to the IMAQ Grab VI for starting the acquisition. The grabbed image is displayed on the LabVIEW front panel using an Image Indicator (see **Figure 5(b)**), which points to the location in memory referenced by the IMAQ image reference. A while loop statement allows adding each grabbed image to the image indicator as a single frame. Finally, the image acquisition is finished by calling the IMAQ close VI that releases resources associated with the camera and the interface.

The acquired image is written to a file in a specified format by using the IMAQ Write File 2 VI. The graphics file formats supported by this function are BMP (windows bitmap), JPEG, PNG (portable network graphics), and TIFF (tagged image file format). However, note that lossy compression formats, such as JPEG, introduce image artifacts and should be avoided to ensure accurate image-based measurements. The saved image can be displayed in a secondary image indicator by enabling the Snapshot option. When enabling the Snapshot Mode, the Image Display control continues to display the image as it was when the image was saved during the Case Structure execution, even when the inspection image has changed. To configure the Image Display control for working in Snapshot Mode, right-click on the control on the front panel and enable the Snapshot option.

Another way to acquire an image using a camera is presented in the **Figure 6**. This example uses the NI Vision Acquisition Express to perform the acquisition stage. The Vision Acquisition Express VI is located in the Vision Express palette in LabVIEW, and it is commonly used to quickly develop image acquisition applications due to its versatility and intuitive development environment. Double-clicking on the Vision Acquisition Express VI makes a configuration window appear which allows choosing a device from the list of available acquisition sources, selecting an acquisition type, and configuring the acquisition settings. Concerning the acquisition types, there are four main modes: single acquisition with processing, continuous acquisition with inline processing, finite acquisition with inline processing and finite acquisition with post-processing. The last two acquisition types are similar, except that for a finite acquisition with post-processing the images are only available after they are all acquired. The configuration of the
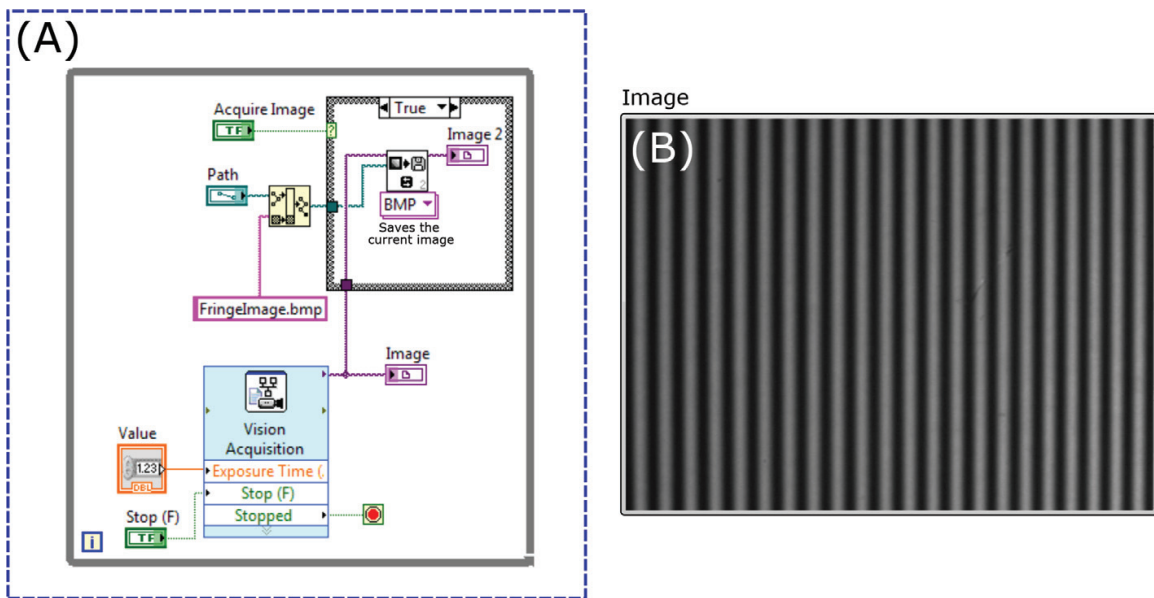
Figure 6. Continuous acquisition using IMAQ vision acquisition express. (a) Block diagram. (b) Image indicator in front panel.

acquisition settings is one of the most relevant processes during configuration and allows the simultaneous manipulation of camera attributes like Exposure Time, Trigger Mode, Gain, Gamma Factor, among others. For this example, we configured the acquisition for working in a continuous acquisition with inline processing mode, which continuously acquires images until an event stops the acquisition. Additionally, the Exposure Time attribute can be modified during the acquisition process by using a Numeric Control. As with the example in **Figure 5**, the captured image is displayed in a secondary image indicator during the Case Structure execution.

In Fringe Projection systems, the manipulation of certain camera attributes (e.g., the Exposure Time attribute) is required to capture high-quality images and to enable to work under different lighting environments with different constraints. In the example above, we introduced the possibility of manipulating camera attributes during acquisition using the Vision Acquisition Express. This manipulation of attributes is also possible by programming a simple snap, grab, or sequence operation based on low-level VIs (as in the example in **Figure 5**) using IMAQdx property nodes. The attribute manipulation requires providing the property node with the name of the attribute we want to modify and identifying the attribute representation, which can be an integer, float, Boolean, enumeration, string or command. In general, cameras share many attributes; however, they often have specific attributes depending on the manufacturer. These attributes should be known beforehand to ensure good acquisition control. At the development stage, LabVIEW does not know or display the name of the attributes or representations. Furthermore, if the documentation is not available, we suggest using the Measurement and Automation Explorer (MAX). MAX is a tool that allows the configuration of different acquisition parameters and is useful when it is required to manipulate attributes of a device with a specific interface within the LabVIEW programming environment. For example, suppose we want to modify the exposure time of our camera (Basler Aca 1600-60gm), but we do not have information about supported attributes. Here is where MAX becomes a

powerful tool for vision system developers. This attribute verification is done by selecting the desired attribute from the Camera Attributes tab in the Measurement and Automation Explorer and identifying its name (i.e., ExposureTimeAbs) and representation (i.e., floating-point format). Therefore, the section of the block diagram inside a red box in **Figure 5** can be modified in order to allow setting the ExposureTimeAbs attribute value using a Property Node as shown in **Figure 7**.

Both acquisition methods have their advantages and disadvantages concerning their implementation in vision systems. On the one hand, the use of the NI Vision Acquisition Express allows to quickly and easily develop acquisition applications, even without having a high knowledge of the tools for image acquisition offered by LabVIEW. However, this could be a disadvantage if our purpose is to have complete control over the acquisition. On the other hand, the low-level VIs provide greater control and versatility over the application development, but the implementation of vision systems based on low-level VIs can be a complicated task for novice users of NI Vision Acquisition Software and LabVIEW.

Once the acquired fringe image file has been written to disk, it is loaded for processing. The block diagram in **Figure 8** illustrates how to perform this procedure in LabVIEW. The IMAQ ReadFile VI opens and reads an image from a file stored on the computer into an image reference. The loaded pixels are converted automatically into the image type supplied by IMAQ Create VI. From now on we refer to the Fringe Image to the loaded fringe image.

3.2. Fringe pattern projection

In the previous section, we described several acquisition methods for capturing images from a camera in LabVIEW. However, in fringe projection systems there are many different fringe pattern projection technologies and choosing the correct one becomes extremely important for an accurate three-dimensional reconstruction. A fringe pattern projector can be considered as an analog device (e.g., LED pattern projector) or as a digital device (e.g., DLP, LCoS, and LCD digital display technologies). LED pattern projectors are ideal for high-resolution three-dimensional reconstruction applications. If equipped with an objective lens and a stripe pattern reticle, these projectors offer great versatility for manipulating the optics of the system and obtaining results according to the metrological requirements. The main disadvantage of this type of projection system is the impossibility of manipulating the projected fringe pattern. Therefore, its use is often restricted to techniques in which only a single fringe image is necessary to obtain the 3D information, such as in the case of FTP.
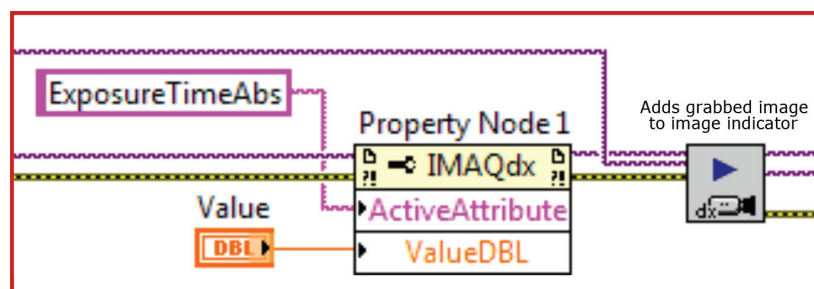
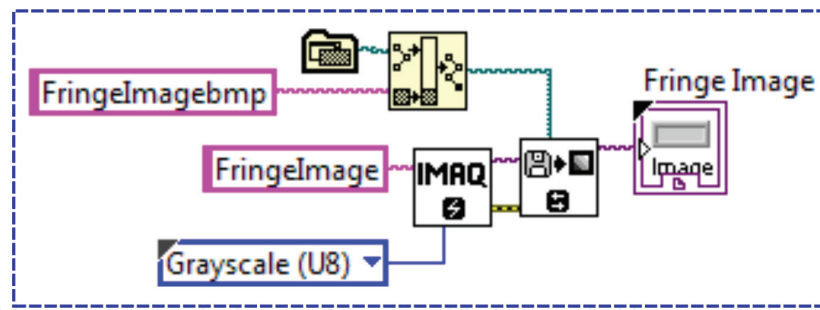**Figure 7.** Setting the ExposureTimeAbs attribute value using a property node.

Figure 8. Reading an image file in LabVIEW.

Fringe Projection systems can also take advantage of a computer to generate sinusoidal fringe patterns that are projected using a digital projector. The key to a successful 3D reconstruction system based on digital fringe projection focuses on generating high-quality fringes to meet the metrological requirements. Ideally, assuming the projector is linear in that it projects grayscale values ranging from 0 to 255 (0 black, and 255 white), the computer-generated fringe patterns can be described as follows,

$$I(i,j) = \frac{255}{2}\left[1 + cos\left(\frac{2\pi i}{p_d} + \varphi\right)\right],$$ (16)

where p_d represents the number of pixels per fringe period, φ refers to the phase shift, and (i,j) are the pixel indices. Eq. (16) is implemented using the numeric functions provided by the NI LabVIEW Base Package. An example of a pattern generator block diagram is shown in **Figure 9**. In this program the Numeric Indicators enable the modification of the fringe pitch and the phase shift according to the application requirements.

An alternative to a block diagram implementation of Eq. (16) LabVIEW provides a MathScript RT Module as a scripting language. The module allows the combination of textual and graphical approaches for algorithm development. In **Figure 10** we provide an example on how to use the MathScript RT Module for fringe generation in LabVIEW.

Once the fringe images have been generated, they are sent to a digital video projector for projection. A video projector is essentially a second monitor. Therefore the fringe image is displayed by
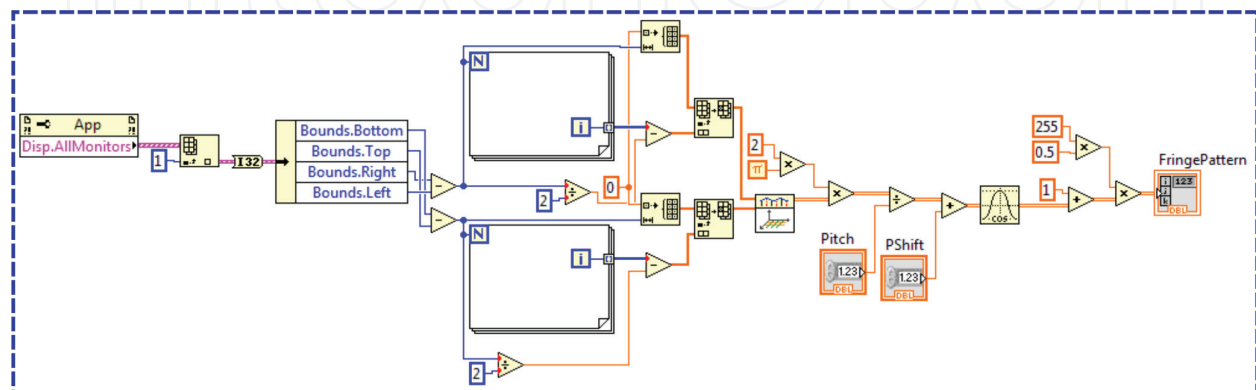

Figure 9. Block diagram for fringe pattern generation.
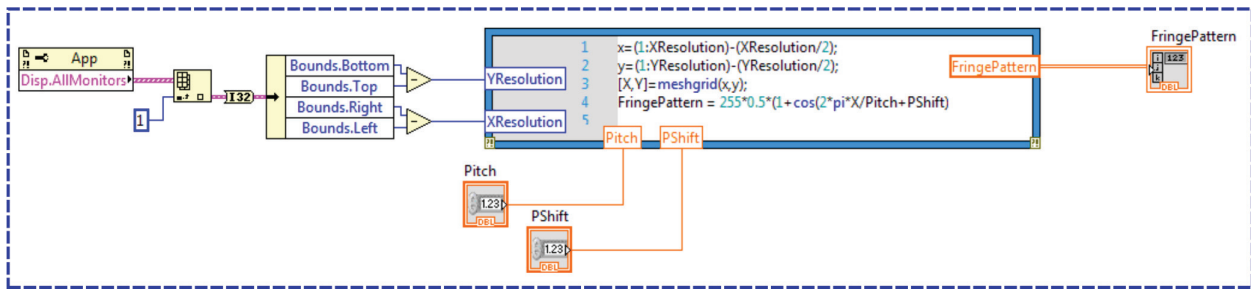
Figure 10. Fringe pattern generation example using the LabVIEW MathScript RT module.

using the External Display VIs provided by the NI Vision Development Module. Here, we use IMAQ WindDraw VI to display the image in an external image window. The image window appears automatically when the VI is executed. Having beforehand the information from all the available displays on the computer, including their resolution and bounding rectangles, we set the position of the image window to be displayed on the desired monitor. This setting is done with IMAQ WindMove VI. Additionally, using IMAQ WindSetup VI the appearance and attributes of the window can be modified to hide the title bar. Note that the default value for this attribute is TRUE which shows the title bar. The block diagram in **Figure 11** illustrates a projection stage in LabVIEW. Here, we use a Property Node for obtaining the information about all the monitors on the computer. The Disp.AllMonitors property Returns information about their bounding rectangles and bit depths.

3.3. Phase retrieval

Phase retrieval is carried out by Fourier transform profilometry. In LabVIEW, the IMAQ FFT VI computes the discrete Fourier transform of the fringe image. This function creates a complex image in which low frequencies are located at the edges, and high frequencies are grouped at the center of the image. Note that for the IMAQ FFT VI a reference to the destination image must be specified and configured as a Complex(CSG) image. Once the deformed fringe pattern is 2-D Fourier transformed, the resulting spectra are converted into a complex
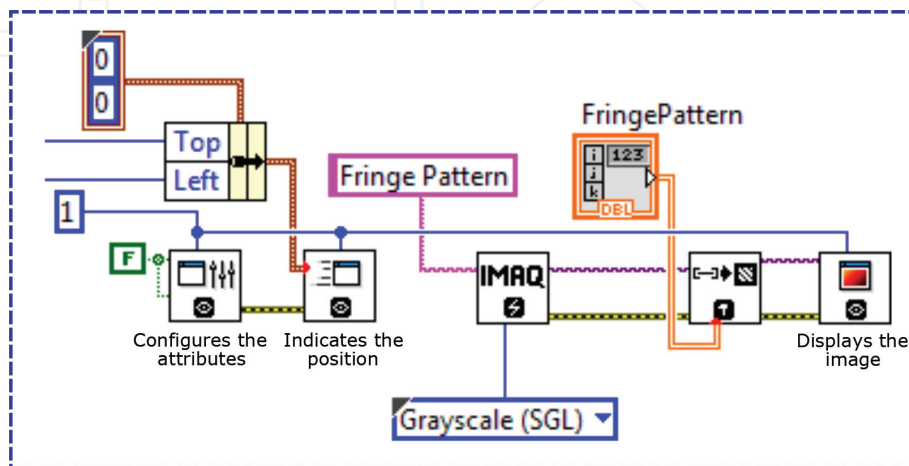

**Figure 11.** Second monitor configuration in LabVIEW.

2D array to perform the filtering procedure, thus obtaining the fundamental frequency spectrum in the frequency domain. The following step is to compute the inverse Fourier transform of the fundamental component. The Inverse FFT VI is for computing the inverse discrete Fourier transform (IDFT) of a complex 2D array. By using this function, we calculate the inverse FFT of the fundamental component which contains the 3D information. Finally, we obtain the phase by applying Eq. (11). Here, we use Complex To Re/Im Function to break the complex 2D array into its rectangular components and Inverse Tangent(2 Input) Function for performing the arctangent operation. With the example in **Figure 12(a)** we illustrate the phase retrieval process in LabVIEW. In this figure, the Fringe Image and Hanning W refer to the fringe pattern image shown in **Figure 12(b)** and the Hanning window filter array, respectively. The resultant wrapped phase map is shown in **Figure 12(c)**.

3.4. Hanning filter design

In Section 2 we showed that in FTP a filtering procedure is performed to obtain the fundamental frequency spectrum in the frequency domain. Once the Fourier transform is computed, the resultant spectrum is filtered by a 2-D Hanning window defined by Eq. (6). In LabVIEW, the IMAQ Select Rectangle VI is commonly used to specify a rectangular region of interest (ROI) in an image. We use the IMAQ Select Rectangle VI for manually selecting the region in the Fourier spectrum corresponding to the fundamental frequency component. Here, the image is displayed in an external display window and through the use of the rectangle tools, provided by the IMAQ Select Rectangle VI, we estimate the optimal size and location of the filtering
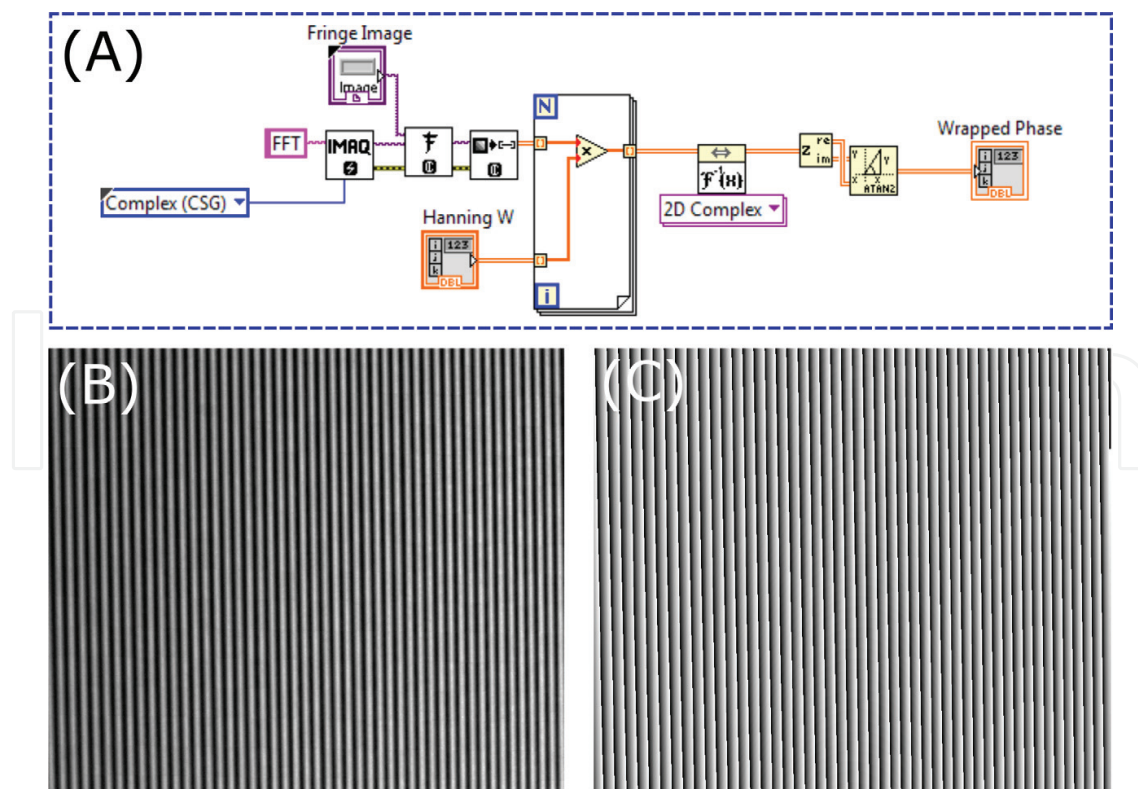

Figure 12. Phase retrieval process in LabVIEW. (a) Block diagram. (b) Fringe pattern image. (c) Wrapped phase map.
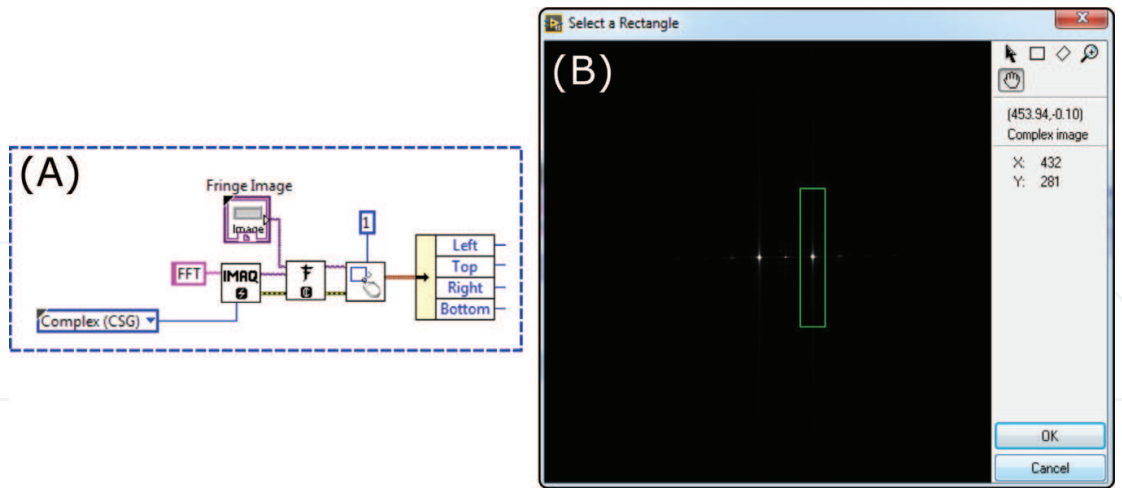
Figure 13. Manual selection of the filtering window. (a) Block diagram. (b) External display window and rectangle tools.

window that guarantees the separation between the fundamental frequency component and other unwanted contributions. The block diagram shown in **Figure 13(a)** indicates the IMAQ Select Rectangle VI to manually select the region corresponding to the first order spectrum. The Fringe Image is the fringe pattern image in **Figure 12(b)**. The IMAQ FFT VI computes the
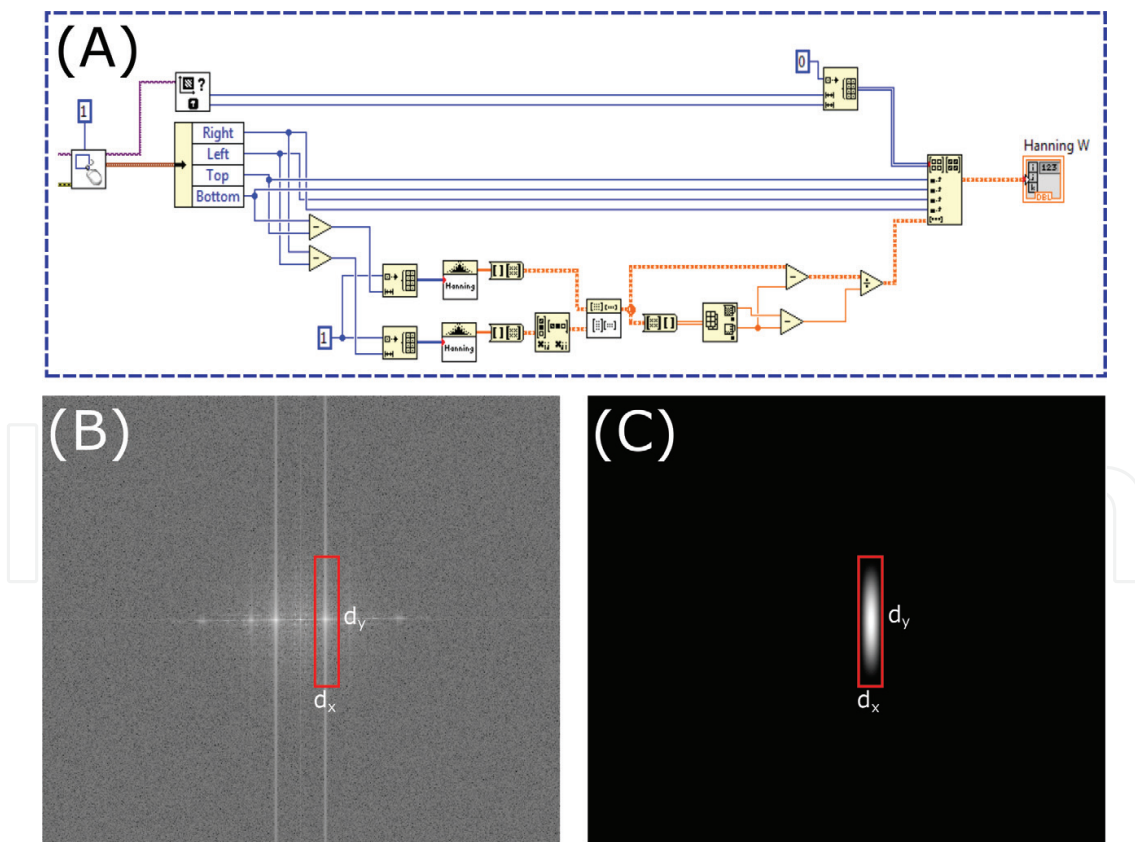

Figure 14. Hanning filter design in LabVIEW. (a) Continuation of the block diagram in **Figure 13(a)**. (b) Fourier transform magnitude spectra displayed by the external window in **Figure 13(b)**. d_x and d_y relate to the size in x and y of the filtering window, respectively. (c) 2D-hanning window.

discrete Fourier transform of the Fringe Image. The resultant complex spectrum is displayed using an external display window as shown in **Figure 13(b)**. By using the selection tools located on the right side of the window, we can manually select the rectangular area of interest.

The IMAQ Select Rectangle VI returns the coordinates (i.e., left, top, right and button) of the chosen rectangle as a cluster. Therefore, it is necessary to access each element from the cluster to extract the window information. For this reason, we add the Unbundle By Name function to the block diagram which unbundles a cluster element by name. Based on this information, we calculate the size and location of the Hanning window filter. Finally, using the Hanning Window VI two 1-D Hanning windows are created whose lengths correspond to the size of x and y of the filtering window, respectively. The two-dimensional Hanning window is obtained by the separable product of these two 1-D Hanning windows [22]. The block diagram in **Figure 14(a)** illustrates the filtering design stage in LabVIEW. d_x and d_y, in **Figure 14(b)**, relate to the size in x and y of the selected filtering window, respectively. Finally, the obtained 2D Hanning window is shown in **Figure 14(c)**.

3.5. Phase unwrapping

The phase unwrapping process is carried out comparing the wrapped phase at neighborhoods and adding, or subtracting, an integer number of 2π, thus obtaining a continuous phase. This definition is for the one-dimensional phase unwrapping process. However, for two-dimensional (2-D) phase unwrapping this is not readily applicable, and additional steps must be taken to
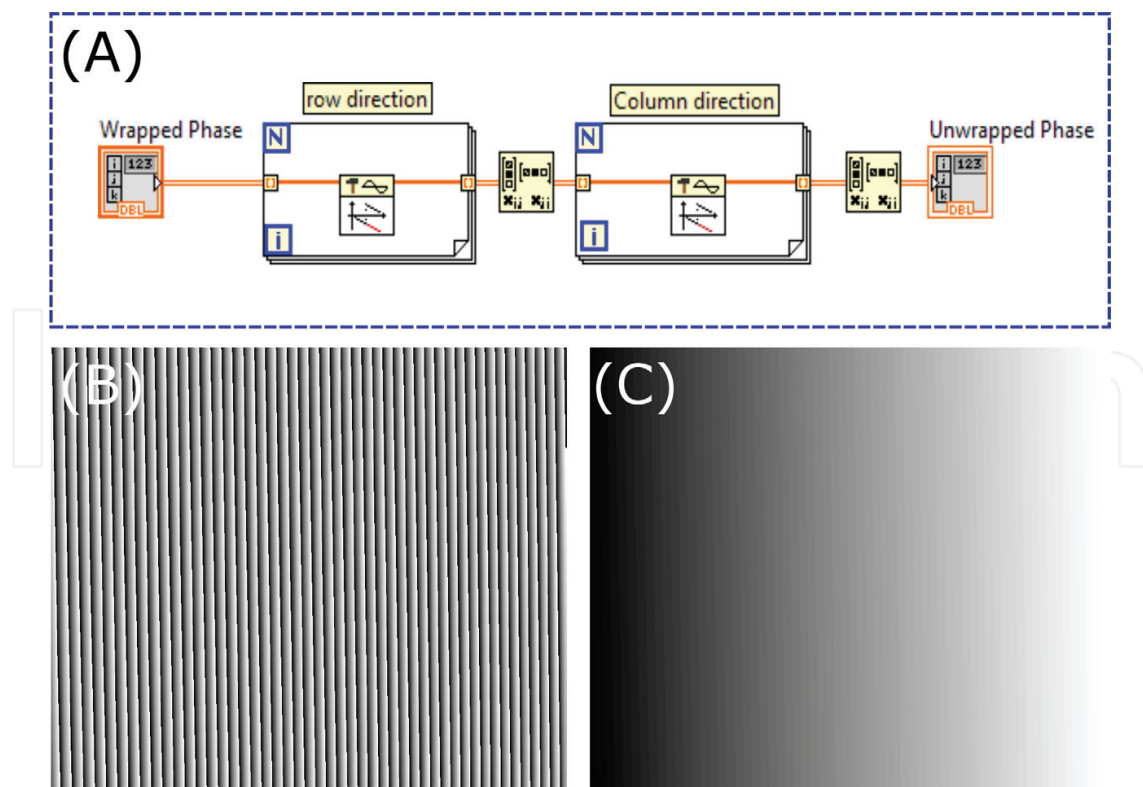

**Figure 15.** Bidimensional phase unwrapping in LabVIEW. (a) Wrapped phase map. (b) Unwrapped phase map.

obtain the unwrapped solution. The conventional approach for 2-D phase unwrapping can be accomplished by applying 1-D phase unwrapping first row-wise followed by 1-D phase unwrapping column-wise in two steps. The block diagram in **Figure 15(a)** illustrates this process. Here, the Unwrap Phase VI unwraps a 1D-phase array by eliminating discontinuities whose absolute values exceed π. Thus, a for loop is required to compute the continuous phase for each row of the 2-D wrapped phase array. For 1-D phase unwrapping column-wise, we use the Transpose Matrix Function to calculate the conjugate transpose of the resultant array before executing the for loop statement. **Figure 15(b)** and **(c)** show a wrapped phase map and its unwrapped counterpart, respectively. In addition to this approach, many 2D phase-unwrapping algorithms have been proposed, especially to address discontinuities and noise [12]. These other methods can also be implemented in LabVIEW either with block diagrams, using math scripts, with precompiled C++ code in .dll files, or via integration of external functions with other environments such as MATLAB. However, an explanation of the details of these other approaches is beyond the scope of this chapter.

4. Applications

FPP is often used as a non-contact surface analysis technique in industry inspection. In this section, we show the 3D surface reconstruction of a dented steel pipe. A dent is a permanent plastic deformation of the cross-section of the pipe. In the example shown in **Figure 16** the dent was produced penetrating the pipe with a diamond cone indenter. In **Figure 16(a)** and **(b)** we show the tested object, and the deformed fringe pattern image, respectively. The goal is to measure the depth of the dent with high accuracy and to obtain the surface shape of the pipe for subsequent deformation analysis. In **Figure 16(c)** and **(d)**, we show the wrapped, and unwrapped phases obtained by FTP, respectively. The unwrapped phase map is converted to metric coordinates using a calibration model. In **Figure 17(a)**, we show the reconstructed pipe shape with the texture map. A profile across the reconstructed pipe, thought the dent, is shown in **Figure 17(b)**. Analyzing this profile, we can measure the depth of the dent to approximately 4 mm.
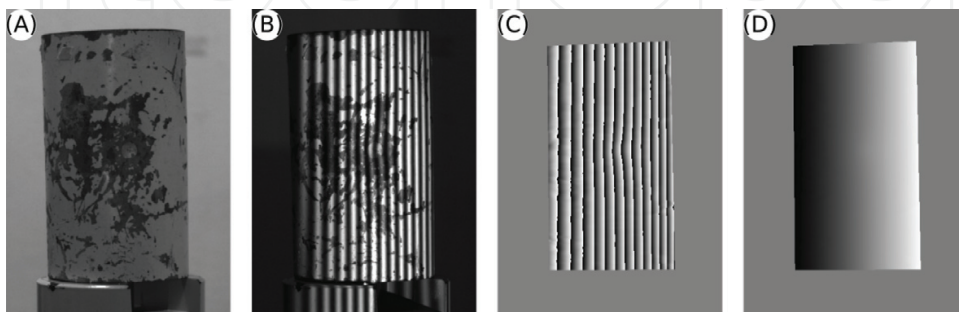

Figure 16. FTP analysis of a indented pipe. (a) Texture image. (b) Deformed fringe pattern. (c) Wrapped phase. (d) Unwrapped phase.
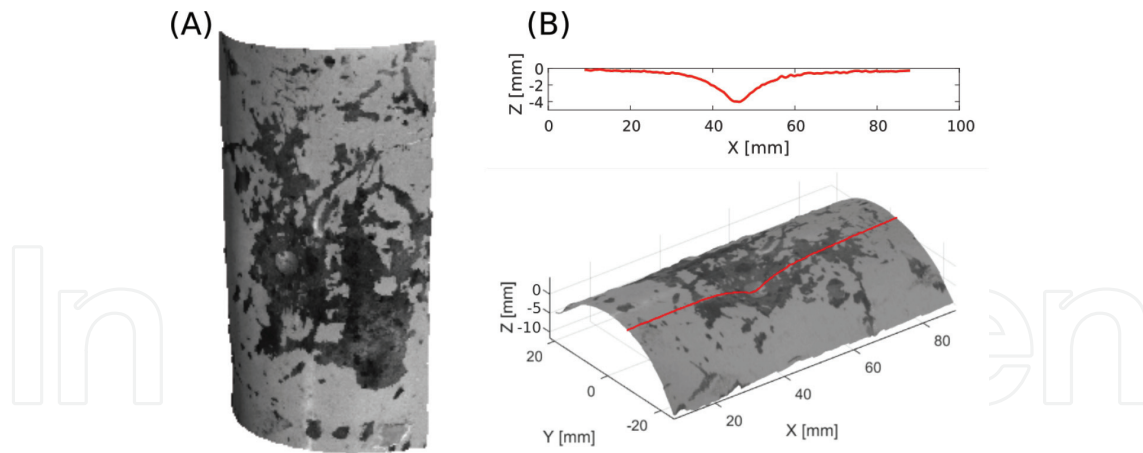
Figure 17. (a) 3D reconstructed shape. (b) Cross section of the 3D reconstruction.

Another application of FPP is in facial metrology, where several patterns are projected onto the face to obtain a 3D digital model. 3D shape measurement of faces plays an important role in several fields like in the biomedical sciences, biometrics, security, and entertainment. Human face models are widely used in medical applications for 3D facial expression recognition [24] and measurement of stretch marks [25]. Usually, the main challenge is the movement of the patient. The movement can produce errors or noise in the 3D reconstruction affecting its accuracy. Hence, 3D scanning techniques that require few images in the reconstruction process, like FTP, are commonly used. In **Figure 18** we show an experimental result of reconstructing a live human face. The captured image with the deformed fringe pattern is shown in **Figure 18(a)**. In **Figure 18(b)** and **(c)** we show the 3D geometry acquired rendered in shaded mode and with texture mapping, respectively. Note that several facial regions with hairs, like the eyebrows, are reconstructed with high detail. While other areas, under shadows, like the right side of the nose, are not correctly reconstructed.

Finally, another area where FPP has frequently been used is in cultural heritage preservation. The preservation of cultural heritage works requires accurately scanning sculptures, archeological remains, paintings, etc. In **Figure 19** we show the 3D reconstruction of a sculpture replica.
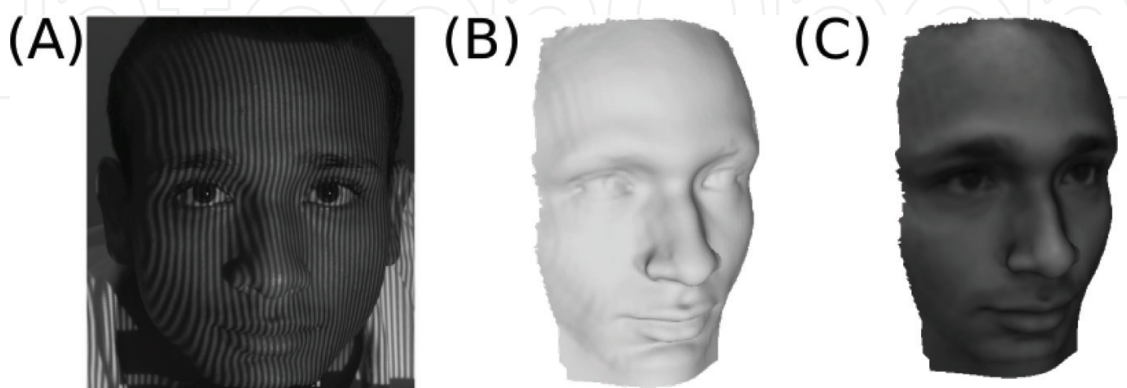
Figure 18. (a) Fringe pattern onto face. (b) 3D rendered model in shaded mode. (c) 3D rendered model with color texture mapping.
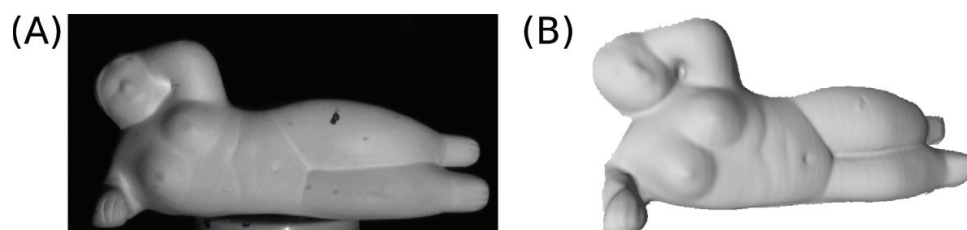
Figure 19. FTP 3D reconstruction of a sculpture replica of "Figura reclinada 92 - Gertrudis" by Fernando Botero [23]. (a) Texture image. (b) 3D reconstruction.

Acknowledgements

Author details

Andrés G. Marrugo[1]*, Jesús Pineda[1], Lenny A. Romero[2], Raúl Vargas[1] and Jaime Meneses[3]

*Address all correspondence to: agmarrugo@utb.edu.co

1 Facultad de Ingeniería, Universidad Tecnológica de Bolívar, Cartagena, Colombia

2 Facultad de Ciencias Básicas, Universidad Tecnológica de Bolívar, Cartagena, Colombia

3 Grupo de Óptica y Tratamiento de Señales, Universidad Industrial de Santander, Bucaramanga, Colombia

References

[1] Zhang S. Handbook of 3D Machine Vision: Optical Metrology and Imaging. CRC Press; 2013. pp. 1

[2] Gorthi SS, Rastogi P. Fringe projection techniques: Whither we are? Optics and Lasers in Engineering. 2010;**48**(2):133-140

[3] Zappa E, Busca G. Static and dynamic features of Fourier transform profilometry: A review. Optics and Lasers in Engineering. 2012;**50**(8):1140-1151

[4] Hariharan P, Oreb BF, Eiju T. Digital phase-shifting interferometry: A simple error-compensating phase calculation algorithm. Applied Optics. 1987;**26**(13):2504-2506

[5] Takeda M, Ina H, Kobayashi S. Fourier-transform method of fringe-pattern analysis for computer-based topography and interferometry. Journal of the Optical Society of America. 1982;**72**(1):156

[6] Su X, Zhang Q. Dynamic 3-D shape measurement method: A review. Optics and Lasers in Engineering. 2010;**48**(2):191-204

[7] Petitgrand S, Yahiaoui R, Danaie K, Bosseboeuf A, Gilles J. 3D measurement of micromechanical devices vibration mode shapes with a stroboscopic interferometric microscope. Optics and Lasers in Engineering. 2001;**36**(2):77-101

[8] Felipe-Sesé L, Siegmann P, Díaz FA, Patterson EA. Integrating fringe projection and digital image correlation for high-quality measurements of shape changes. Optical Engineering. 2014;**53**(4):044106

[9] Takeda M, Mutoh K. Fourier transform profilometry for the automatic measurement of 3-D object shapes. Applied Optics. 1983;**22**(24):3977

[10] Takeda M. Fourier fringe analysis and its application to metrology of extreme physical phenomena: A review [invited]. Applied Optics. 2013;**52**(1):20-29

[11] Lin JF, Su X. Two-dimensional Fourier transform profilometry for the automatic measurement of three-dimensional object shapes. Optical Engineering. 1995;**34**(11):3297-3302

[12] Ghiglia DC, Pritt MD. Two-Dimensional Phase Unwrapping: Theory, Algorithms, and Software. Vol. 4. New York: Wiley; 1998

[13] Huntley JM, Saldner H. Temporal phase-unwrapping algorithm for automated interferogram analysis. Applied Optics. 1993;**32**(17):3047-3052

[14] Liu H, Su W-H, Reichard K, Yin S. Calibration-based phase-shifting projected fringe profilometry for accurate absolute 3D surface profile measurement. Optics Communications. 2003;**216**(1):65-80

[15] Merner L, Wang Y, Zhang S. Accurate calibration for 3D shape measurement system using a binary defocusing technique. Optics and Lasers in Engineering. 2013;**51**(5):514-519

[16] Zhang S, Huang PS. Novel method for structured light system calibration. Optical Engineering. 2006;**45**(8):083601

[17] Li K, Bu J, Zhang D. Lens distortion elimination for improving measurement accuracy of fringe projection profilometry. Optics and Lasers in Engineering. 2016;**85**:53-64

[18] Arciniegas J, González AL, Quintero LA, Contreras CR, Meneses JE. Sistema de reconstrucción tridimensional aplicado a la exploración superficial de fallas y defectos en tuberías con refuerzo no metálico para el transporte de hidrocarburos. Revista Investigaciones Aplicadas. 2015;**9**(1):12-18

[19] Hu Q, Huang PS, Fu Q, Chiang F-P. Calibration of a three-dimensional shape measurement system. Optical Engineering. 2003;**42**(2):487-493

[20] Huang Z, Xi J, Yu Y, Guo Q. Accurate projector calibration based on a new point-to-point mapping relationship between the camera and projector images. Applied Optics. 2015; **54**(3):347-356

[21] Travis J, Kring J. LabVIEW for Everyone: Graphical Programming Made Easy and Fun. Prentice-Hall; 2007. pp. 10

[22] Easton RL Jr. Fourier Methods in Imaging. John Wiley & Sons; 2010. pp. 567

[23] IPCC. Monumento Artistico "Figura Reclinada de la Gorda Gertrudis" [Online]. 2016. Available from: http://www.ipcc.gov.co/index.php/noticias/item/260-botero

[24] Zhang S. Recent progresses on real-time 3d shape measurement using digital fringe projection techniques. Optics and Lasers in Engineering. 2010;**48**(2):149-158

[25] Gómez ALG, Fonseca JEM, Téllez JL. Proyección de franjas en metrología óptica facial. INGE CUC. 2012;**8**(1):191-206