

*3DFringe*: Digitalización de objetos  
tridimensionales por proyección de franjas

Manual de Usuario

Juan Zheng W,  
Alejandra J. Giron Vazquez,  
Rigoberto Juarez-Salazar\*

\*<http://rjuarezs.com>

# Índice general

|                                                      |           |
|------------------------------------------------------|-----------|
| <b>1. Introducción</b>                               | <b>3</b>  |
| 1.1. Descripción . . . . .                           | 3         |
| 1.2. Rutinas principales . . . . .                   | 4         |
| 1.3. Manuales . . . . .                              | 5         |
| 1.4. Datos de prueba . . . . .                       | 6         |
| <b>2. Manual de instalación</b>                      | <b>10</b> |
| 2.1. Preámbulo . . . . .                             | 10        |
| 2.2. Requerimientos . . . . .                        | 11        |
| 2.2.1. Software . . . . .                            | 11        |
| 2.2.2. Hardware . . . . .                            | 11        |
| 2.2.3. Calibración . . . . .                         | 11        |
| 2.3. Instalación de módulos Python . . . . .         | 12        |
| 2.4. Configuración . . . . .                         | 15        |
| 2.4.1. Proyector como pantalla principal . . . . .   | 15        |
| 2.4.2. Pantallas en modo extendido . . . . .         | 16        |
| 2.4.3. Archivo <code>start.bat</code> . . . . .      | 17        |
| <b>3. Manual de usuario</b>                          | <b>20</b> |
| 3.1. Preámbulo . . . . .                             | 20        |
| 3.2. General . . . . .                               | 21        |
| 3.3. Sección <code>Gratings</code> . . . . .         | 23        |
| 3.4. Sección <code>Phase shifting</code> . . . . .   | 27        |
| 3.5. Sección <code>Phase Unwrapping</code> . . . . . | 31        |
| 3.6. Sección <code>Triangulation</code> . . . . .    | 35        |

---

|                                            |           |
|--------------------------------------------|-----------|
| 3.7. Ejemplo con datos de prueba . . . . . | 40        |
| <b>4. Manual técnico</b>                   | <b>45</b> |
| 4.1. Preámbulo . . . . .                   | 45        |
| 4.2. Descripción de campos . . . . .       | 46        |
| 4.2.1. Sección Gratings . . . . .          | 46        |
| 4.2.2. Sección Phase Shifting . . . . .    | 48        |
| 4.2.3. Sección Phase Unwrapping . . . . .  | 50        |
| 4.2.4. Sección Triangulation . . . . .     | 52        |
| <b>A. Diagramas de flujo del sistema</b>   | <b>55</b> |
| A.1. Preámbulo . . . . .                   | 55        |
| A.2. Diagramas de flujo . . . . .          | 56        |
| <b>B. Funciones y código fuente</b>        | <b>62</b> |
| B.1. Preámbulo . . . . .                   | 62        |
| B.2. Funciones y código fuente . . . . .   | 63        |
| <b>Bibliografía</b>                        | <b>69</b> |

# Capítulo 1

## Introducción

### Contenido

|                                    |   |
|------------------------------------|---|
| 1.1. Descripción . . . . .         | 3 |
| 1.2. Rutinas principales . . . . . | 4 |
| 1.3. Manuales . . . . .            | 5 |
| 1.4. Datos de prueba . . . . .     | 6 |

### 1.1. Descripción

El software *3DFringe* permite realizar digitalización de objetos tridimensionales usando el método de proyección de franjas. Este método requiere un mínimo de componentes para funcionar [1]. Específicamente, solo se emplea una computadora, una cámara y un proyector de diapositivas como se muestra en la figura 1.1. Asimismo, el principio de funcionamiento de este método es simple, como se describe a continuación [2].

- Un patrón de franjas rectas uniformes, conocido como *rejilla*, es generado por computadora. La rejilla generada es enviada al proyector para iluminar el objeto a digitalizar.
- La forma tridimensional del objeto provoca que las franjas con las que ha sido iluminado el objeto se curven. La cámara es usada para capturar este nuevo patrón de franjas.

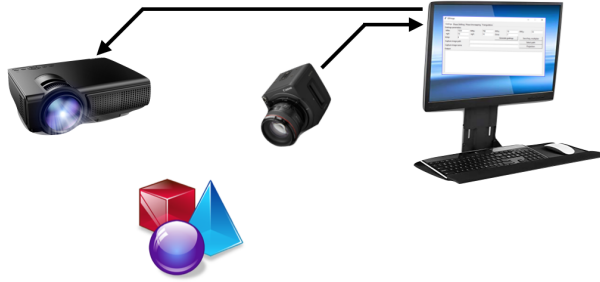


Figura 1.1: Configuración típica de un sistema de proyección de franjas para digitalización de objetos tridimensionales.

- El proceso de proyección de rejillas se repite hasta obtener el número de patrones de franjas necesarios para procesamiento posterior.
- Los patrones de franjas capturados son procesados en la computadora para obtener la forma tridimensional del objeto analizado. Específicamente, el procesamiento de datos se realiza en tres etapas: (i) extracción de fase envuelta, (ii) desenvolvimiento de fase, y (iii) triangulación.

El software *3DFringe* es una herramienta computacional unificada que permite al usuario realizar digitalización de objetos tridimensionales usando el método de proyección de franjas. A continuación se describen las rutinas principales de este software.

## 1.2. Rutinas principales

Para la implementación del método de proyección de franjas, el software *3DFringe* proporciona al usuario el control de cuatro rutinas principales.

1. Generación de rejillas, proyección y captura automática. El software proporciona controles para: (a) el número de rejillas a usar, (b) la frecuencia máxima de la rejilla, (c) el número de corrimientos de fase, y (d) la resolución. Después de la generación de rejillas, el software *3DFringe* despliega las rejillas en el proyector y captura los patrones de franjas correspondientes de forma automática.

2. Extracción de fase envuelta usando corrimiento de fase. El software procesa los patrones de franjas capturados por la rutina anterior. Como resultado, entrega un mapa de fase envuelto por cada frecuencia empleada durante la proyección de rejillas.
3. Desenvolvimiento de fase por multifrecuencia. El software procesa los mapas de fase envueltos calculados por la rutina anterior. Como resultado, el software entrega un mapa de fase desenvuelto por cada frecuencia empleada durante la proyección de rejillas. Estos mapas de fase presentan un incremento en la resolución en la medida en que se incrementa la frecuencia de la rejilla.
4. Triangulación. El software calcula las coordenadas  $(x, y, z)$  de cada punto observado por la cámara en la superficie del objeto analizado. Para esto, el software usa los mapas de fase desenvueltos calculados por la rutina anterior y los parámetros del sistema (parámetros intrínsecos y extrínsecos de la cámara y el proyector).

La implementación de estas rutinas está basada en sólidos fundamentos teóricos desarrollados en la literatura científica reciente [3, 4, 5]. En la figura 1.2 se muestra la ventana principal del software *3DFringe* donde se observan los controladores para la generación de rejillas, proyección y captura automática de patrones de franjas.

### 1.3. Manuales

La documentación del software *3DFringe* está organizada en tres manuales que se describen a continuación.

1. **Manual de instalación.** Ofrece al usuario una guía para los requerimientos computacionales, instalación de módulos complementarios y configuración del sistema.
2. **Manual de usuario.** Proporciona al usuario una descripción detallada sobre la interfaz gráfica del software *3DFringe*. Asimismo, este manual sirve como referencia para valores típicos necesarios para la correcta ejecución del software.

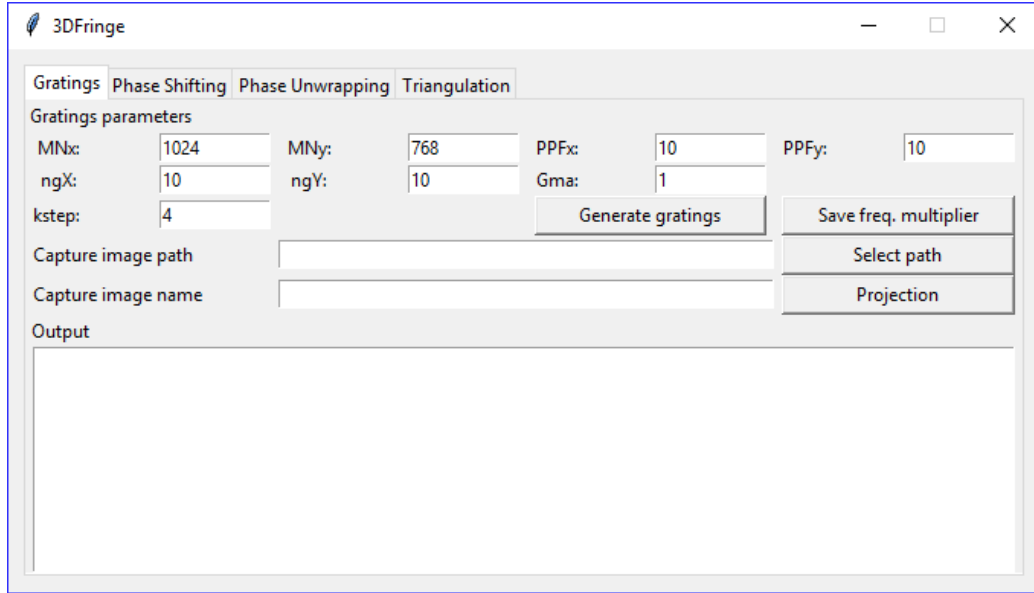


Figura 1.2: Ventana principal del software *3DFringe*.

3. **Manual técnico.** Este manual presenta información detallada sobre los campos de cada sección del software, valores de configuración, restricciones, y mensajes de error. En este manual se incluyen dos apéndices. El apéndice *A* de este manual presenta los diagramas de flujo correspondientes a las funciones principales del software *3DFringe*. El apéndice *B* de este manual describe las funciones principales del software y su correspondiente código fuente.

Estos manuales servirán al usuario como guía para la instalación, uso de la interfaz gráfica de usuario, y mantenimiento del software *3DFringe*, respectivamente.

## 1.4. Datos de prueba

El software *3DFringe* está acompañado de datos experimentales. Estos datos permitirán realizar pruebas rápidas de funcionamiento del software. Los datos fueron obtenidos durante la digitalización de tres objetos en diferentes configuraciones experimentales. A continuación se describen los datos experimentales de prueba proporcionados.

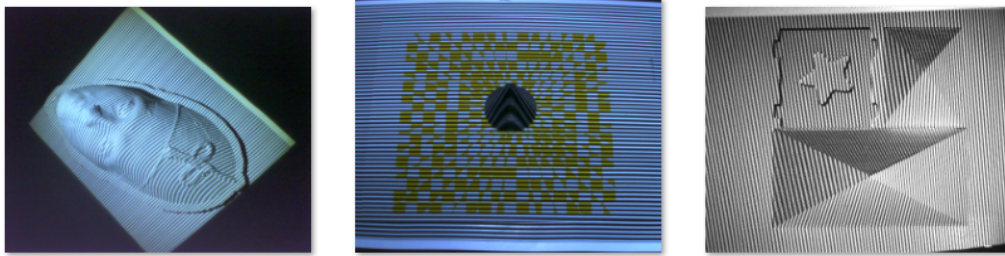


Figura 1.3: (Izq.) Uno de treinta y dos patrones de franjas obtenidos durante la digitalización de un busto de Nefertiti. (Medio) Uno de cuarenta patrones de franjas obtenidos durante la digitalización de un cono. (Der.) Uno de setenta y dos patrones de franjas obtenidos durante la digitalización de dos pirámides y una placa de madera.

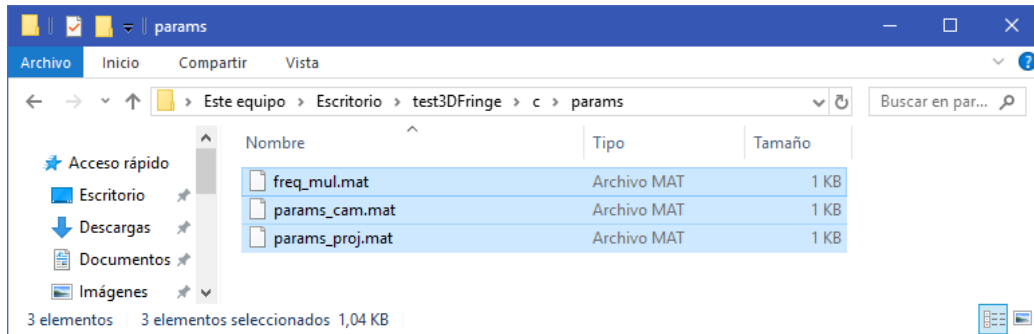


Figura 1.4: Archivos que contienen multiplicadores de frecuencia y parámetros de calibración de la cámara y proyector usados para la proyección de rejillas y captura de patrones de franjas experimentales.

### 1. Nefertiti:

- Treinta y dos patrones de franjas para digitalización de un busto de Nefertiti. Se emplearon cuatro rejillas con cuatro corrimientos de fase cada una (16 imágenes). La proyección se realizó tanto en el eje- $x$  como en el eje- $y$  del proyector ( $2 \times 16$  imágenes).
- Archivo \*.mat con los multiplicadores de frecuencia correspondientes a las rejillas usadas.
- Archivo \*.mat con los parámetros de la cámara.
- Archivo \*.mat con los parámetros del proyector.

## 2. Cono:

- Cuarenta patrones de franjas para digitalización de un cono. Se emplearon cinco rejillas con cuatro corrimientos de fase cada una (20 imágenes). La proyección se realizó tanto en el eje- $x$  como en el eje- $y$  del proyector ( $2 \times 20$  imágenes).
- Archivo \*.mat con los multiplicadores de frecuencia correspondientes a las rejillas usadas.
- Archivo \*.mat con los parámetros de la cámara.
- Archivo \*.mat con los parámetros del proyector.

## 3. Pirámides:

- Setenta y dos patrones de franjas para digitalización de dos pirámides y una placa de madera. Se emplearon seis rejillas con seis corrimientos de fase cada una (36 imágenes). La proyección se realizó tanto en el eje- $x$  como en el eje- $y$  del proyector ( $2 \times 36$  imágenes).
- Archivo \*.mat con los multiplicadores de frecuencia correspondientes a las rejillas usadas.
- Archivo \*.mat con los parámetros de la cámara.
- Archivo \*.mat con los parámetros del proyector.

La figura 1.3 muestra un patrón de franjas por cada conjunto de datos de prueba proporcionados. Por cada conjunto de datos, se incluyen los archivos \*.mat necesarios como se muestra en la figura 1.4. Estos archivos contienen los multiplicadores de frecuencia y parámetros de calibración requeridos por el software *3DFringe*. NOTA: estos archivos son diferentes en cada conjunto de datos de prueba. La figura 1.5 muestra el resultado calculado por el software *3DFringe* al procesar los datos de prueba proporcionados. En estos manuales se usarán los datos **Nefertiti** para describir la interfaz gráfica de usuario del software *3DFringe*.

Esperamos que el software *3DFringe* sea de utilidad al usuario, aporte nuevo conocimiento y permita desarrollar habilidades en el uso del método de proyección de franjas para digitalización de objetos tridimensionales.

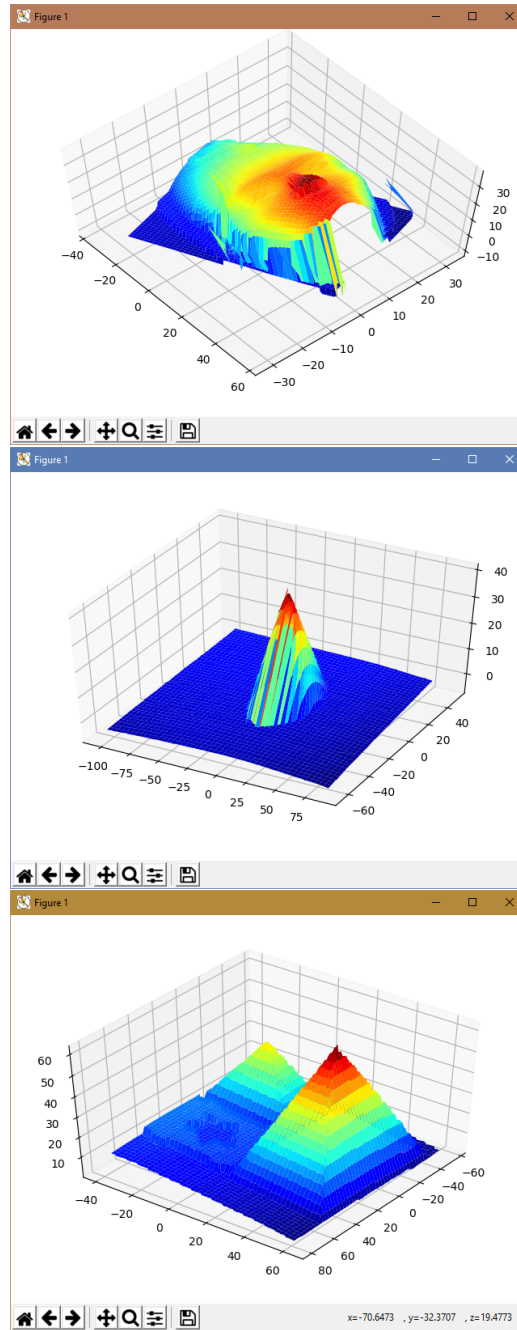


Figura 1.5: Reconstrucción digital de objetos tridimensionales usando el software *3DFringe*. (Arriba) Datos **Nefertiti**. (Medio) Datos **Cono**. (Abajo) Datos **Pirámides**.

# Capítulo 2

## Manual de instalación

### Contenido

|                                                     |           |
|-----------------------------------------------------|-----------|
| <b>2.1. Preámbulo . . . . .</b>                     | <b>10</b> |
| <b>2.2. Requerimientos . . . . .</b>                | <b>11</b> |
| 2.2.1. Software . . . . .                           | 11        |
| 2.2.2. Hardware . . . . .                           | 11        |
| 2.2.3. Calibración . . . . .                        | 11        |
| <b>2.3. Instalación de módulos Python . . . . .</b> | <b>12</b> |
| <b>2.4. Configuración . . . . .</b>                 | <b>15</b> |
| 2.4.1. Proyector como pantalla principal . . . . .  | 15        |
| 2.4.2. Pantallas en modo extendido . . . . .        | 16        |
| 2.4.3. Archivo <code>start.bat</code> . . . . .     | 17        |

### 2.1. Preámbulo

El software *3DFringe* se desarrolló usando el lenguaje de programación Python 3.7 y se ejecuta bajo el sistema operativo Windows. En este manual se describe el proceso de instalación de módulos Python complementarios y la configuración requerida para el correcto funcionamiento del software *3DFringe*.

## 2.2. Requerimientos

### 2.2.1. Software

- Sistema operativo: Windows 10 o superior.
- Lenguaje de programación: Python 3.7.

Se requiere que la computadora donde se ejecutará el software *3DFringe* tenga instalado el lenguaje de programación Python 3.7. En caso contrario, el usuario debe referirse al manual de instalación de Python disponible en

<https://wiki.python.org/moin/BeginnersGuide>

y seguir todos los pasos necesarios hasta la instalación exitosa de Python.

### 2.2.2. Hardware

- Dispositivo de salida: Proyector de diapositivas.
- Dispositivo de entrada: Cámara web.

Se requiere que los drivers de la cámara estén instalados en la computadora y la cámara funcione con normalidad. De lo contrario, el usuario debe acudir a la documentación proporcionada por el fabricante de la cámara para realizar el proceso de instalación correspondiente.

### 2.2.3. Calibración

Para reconstruir digitalmente un objeto tridimensional en unidades métricas (milímetros, por ejemplo), es necesario conocer los parámetros de calibración tanto de la cámara como del proyector. Estos parámetros dependen de los dispositivos así como de su posición y orientación en el espacio. El software *3DFringe* requiere estos parámetros de calibración para realizar triangulación. Se asume que el usuario tiene a disposición los parámetros de calibración del sistema cámara-proyector empleado. De lo contrario, el usuario deberá realizar la calibración correspondiente. Se recomienda calibrar la cámara y el proyector simultáneamente [6].

## 2.3. Instalación de módulos Python

Además de Python 3.7, el funcionamiento del software *3DFringe* requiere la instalación de los siguientes módulos complementarios.

- numpy,
- scipy,
- matplotlib,
- ipython,
- jupyter,
- pandas,
- sympy,
- nose,
- imageio,
- pillow,
- opencv-python.

La instalación de los módulos complementarios se puede realizar directamente desde el *Command Prompt* de Windows. Para abrir la ventana *Command Prompt* tecleé la combinación **Windows + R** para abrir la ventana *Run* que se muestra en la figura 2.1. En la caja de texto *Open* escribir `cmd` y dar clic en el botón *OK*. Como resultado, se abrirá la ventana *Command Prompt* como se muestra en la figura 2.2

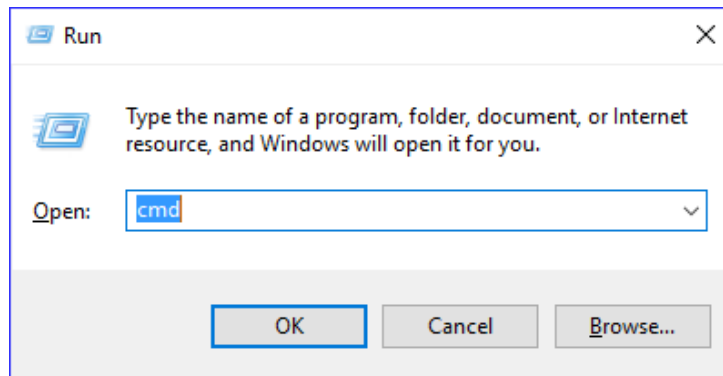


Figura 2.1: Ventana *Run* donde se escribe `cmd` para abrir el *Command Prompt* de Windows.

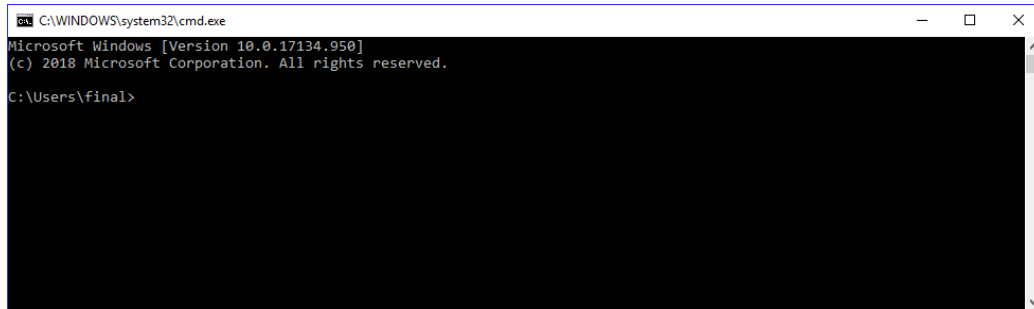


Figura 2.2: Ventana *Command Prompt* donde se solicita la instalación de módulos Python a través de texto simple.

Desde la ventana *Command Prompt* se solicita la instalación de los módulos Python requeridos a través de la siguiente instrucción, ver la figura 2.3:

```
python -m pip install --user numpy scipy matplotlib ipython  
jupyter pandas sympy nose imageio pillow opencv-python
```

Después de escribir el comando anterior, teclear ENTER y esperar a que el instalador finalice, ver la figura 2.4. Si la instalación de módulos finaliza con éxito, el *Command Prompt* mostrará el mensaje

Successfully installed.

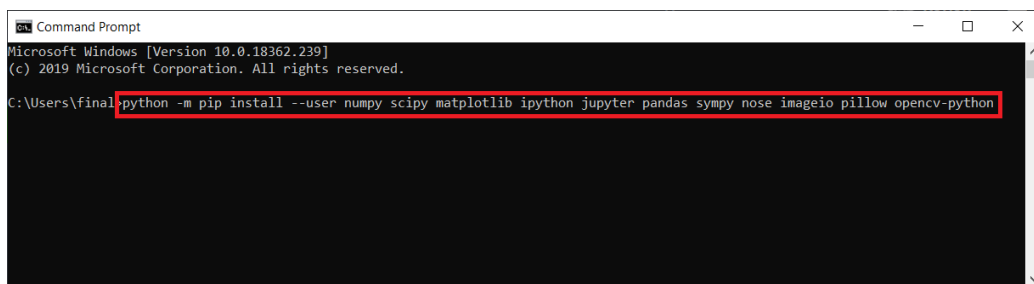


Figura 2.3: Línea de comando solicitando la instalación de los módulos Python requeridos.

```

C:\Windows\System32\cmd.exe
Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-script-location.
Running setup.py install for prometheus-client ... done
Running setup.py install for pandocfilters ... done
WARNING: The script jupyter-nbconvert.exe is installed in 'C:\Users\final\AppData\Roaming\Python\Python37\Scripts' which is not on PATH.
Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-script-location.
WARNING: The scripts jupyter-bundlerextension.exe, jupyter-nbextension.exe, jupyter-notebook.exe and jupyter-serverextension.exe are installed in 'C:\Users\final\AppData\Roaming\Python\Python37\Scripts' which is not on PATH.
Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-script-location.
WARNING: The script jupyter-console.exe is installed in 'C:\Users\final\AppData\Roaming\Python\Python37\Scripts' which is not on PATH.
Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-script-location.
Running setup.py install for mpmath ... done
WARNING: The script isympy.exe is installed in 'C:\Users\final\AppData\Roaming\Python\Python37\Scripts' which is not on PATH.
Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-script-location.
WARNING: The scripts nosetests-3.4.exe and nosetests.exe are installed in 'C:\Users\final\AppData\Roaming\Python\Python37\Scripts' which is not on PATH.
Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-script-location.
WARNING: The scripts imageio_download_bin.exe and imageio_remove_bin.exe are installed in 'C:\Users\final\AppData\Roaming\Python\Python37\Scripts' which is not on PATH.
Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-script-location.
Successfully installed MarkupSafe-1.1.1 Send2Trash-1.5.0 attrs-19.1.0 backcall-0.1.0 bleach-3.1.0 colorama-0.4.1 cycl...
0.10.0 decorator-4.4.0 defusedxml-0.6.0 entrypoints-0.3 imageio-2.5.0 ipykernel-5.1.2 ipython-7.7.0 ipython-genutils-0.2.0 ipywidgets-7.5.1 jedi-0.15.1 Jinja2-2.10.1 jsonschema-3.0.2 jupyter-1.0.0 jupyter-client-5.3.1 jupyter-console-6.0.0 jupyter-core-4.5.0 kiwisolver-1.1.0 matplotlib-3.1.1 mistune-0.8.4 mpmath-1.1.0 nbconvert-5.6.0 nbformat-4.4.0 nose-1.3.7 notebook-6.0.1 numpy-1.17.1 pandas-0.25.1 pandocfilters-1.4.2 parso-0.5.1 pickleshare-0.7.5 pillow-6.1.0 prometheus-client-0.7.1 prompt-toolkit-2.0.9 pygments-2.4.2 pyparsing-2.4.2 pyrsistent-0.15.4 python-dateutil-2.8.0 pytz-2019.2 pywinpty-0.5.5 pyzmq-18.1.0 qtconsole-4.5.4 scipy-1.3.1 six-1.12.0 sympy-1.4 terminado-0.8.2 testpath-0.4.2 tornado-6.0.3 traitlets-4.3.2 wcwidth-0.1.7 webencodings-0.5.1 widgetsnbextension-3.5.1

```

Figura 2.4: Instalación en curso de módulos Python complementarios.

Con esto, la computadora tendrá instalado todo lo necesario para ejecutar el software *3DFringe*. Sin embargo, antes de usar el software es necesario realizar algunos ajustes de configuración que se describen a continuación.

## 2.4. Configuración

### 2.4.1. Proyector como pantalla principal

El software *3DFringe* permite desplegar rejillas y capturar patrones de franjas de manera automática. Sin embargo, para que esta tarea se realice correctamente, es necesario configurar el sistema para establecer al proyector de diapositivas como *monitor principal* de la computadora. Esta configuración se realiza a través del siguiente procedimiento.

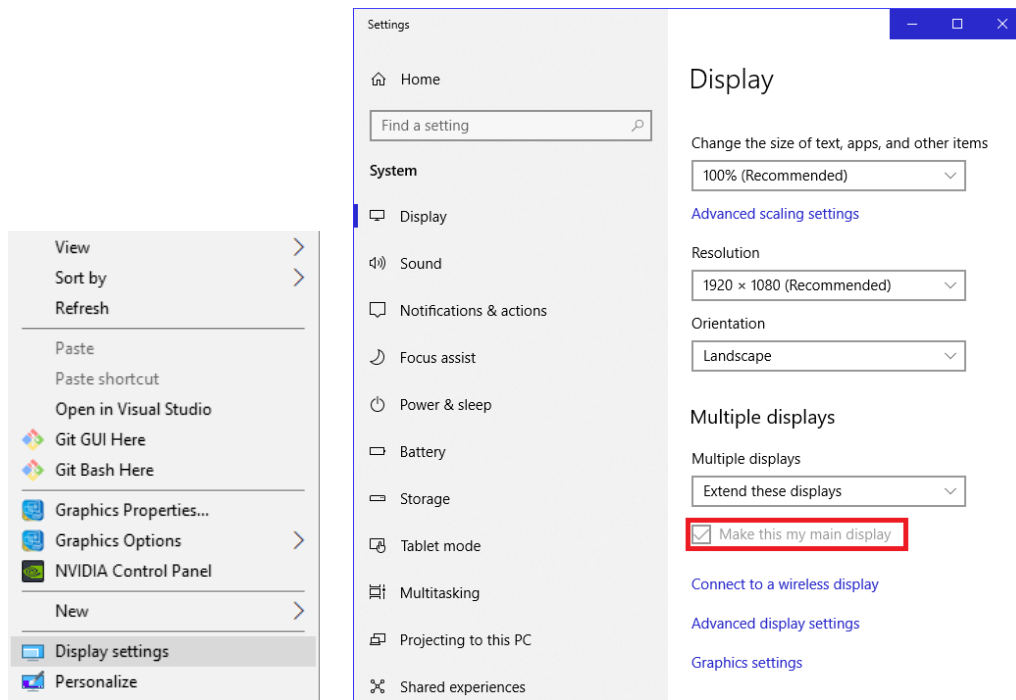


Figura 2.5: (Izq.) Menú contextual del escritorio. (Der.) Marcar un dispositivo como monitor principal de la computadora.

- Dar clic derecho en el escritorio para abrir el menú contextual que se muestra en la figura 2.5(Izq.). Seleccionar la opción *Display settings*.
- Seleccionar el monitor correspondiente al proyector de diapositivas (usualmente *Display 2*). Marcar este dispositivo como principal dando clic en la casilla *Make this my main display* como se muestra en la figura 2.5(Der.).

### 2.4.2. Pantallas en modo extendido

Además del procedimiento para establecer al proyector como pantalla principal, se deben configurar las pantallas del sistema en modo extendido. El siguiente procedimiento permite asegurar que el sistema reconoce al proyector como pantalla principal en modo extendido.

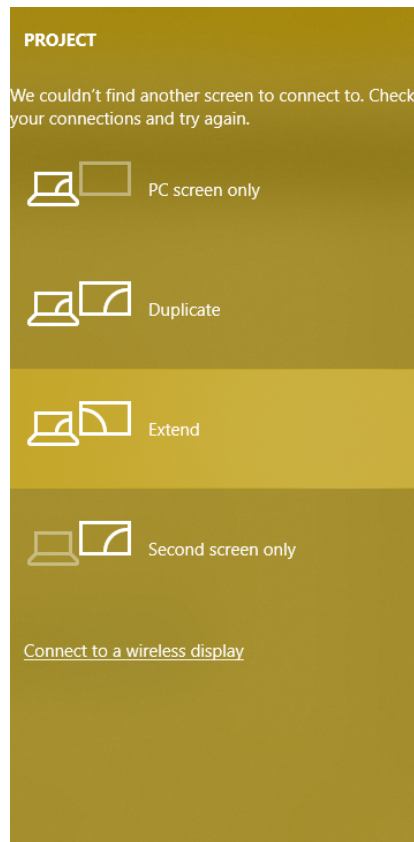


Figura 2.6: Ventana de cambio de modo de pantalla secundaria.

- Pulsar la combinación de teclas **Windows + P** para que aparezca la ventana de cambio de modo de pantalla secundaria como se muestra en la figura 2.6.
- Seleccionar la opción *Extend*.

### 2.4.3. Archivo start.bat

Python y los módulos complementarios requeridos estarán instalados en la computadora en algún directorio del disco local. Por ejemplo,

`C:\Users\Invitado\Anaconda3\python.exe`

como se muestra en la figura 2.7.

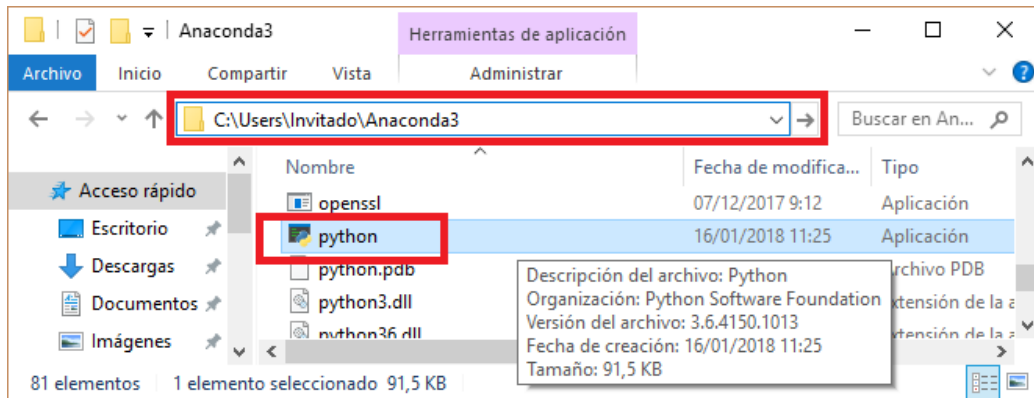


Figura 2.7: Directorio donde Python está instalado.

Por otro lado, el software *3DFringe* estará disponible en una unidad de almacenamiento externo. Por ejemplo,

`E:\software3Dfringe\`

En la unidad de almacenamiento externo se encuentran los siguientes archivos.

- fringe\_projection.py
- main.py
- functions.py
- start.bat

El programa principal del software *3DFringe* es `main.py`. Este archivo debe ser ejecutado desde Python. Esta tarea se puede realizar directamente usando el archivo `start.bat`, que abre automáticamente una instancia de Python y ejecuta en ella el archivo `main.py`. Sin embargo, para que esta opción funcione correctamente, se debe editar el archivo `start.bat` para indicar el directorio donde se encuentra instalado Python. Para ello, se debe seguir el siguiente procedimiento.

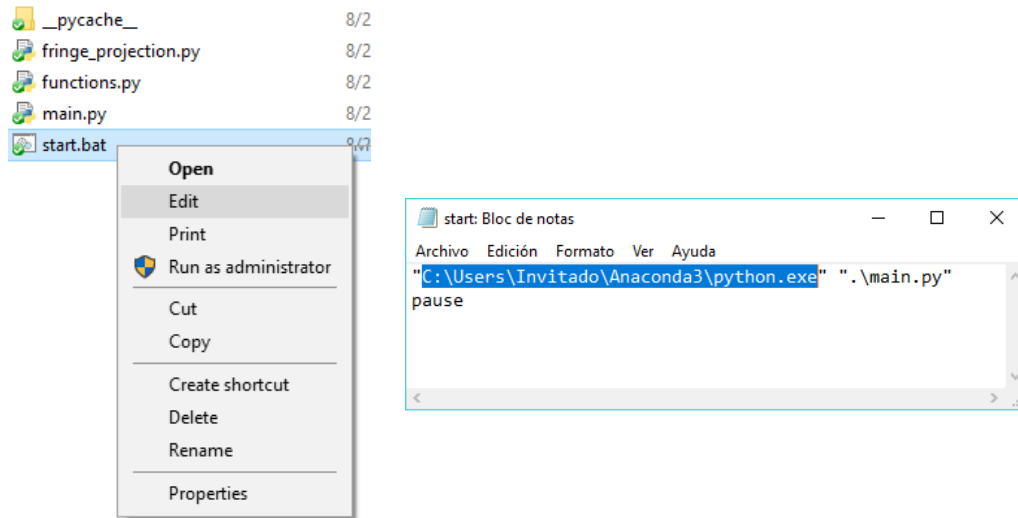


Figura 2.8: (Izq.) Menú contextual del archivo `start.bat`. (Der.) Bloc de notas donde se debe actualizar el directorio donde Python está instalado.

- Dar clic derecho sobre el archivo `start.bat` para abrir su menú contextual y seleccionar la opción *Edit* como se muestra en la figura 2.8(Izq.).
- Se abrirá una ventana del Bloc de notas donde se indica la dirección de Python. Esta dirección debe ser actualizada por la dirección donde la computadora tiene instalado Python. Por ejemplo, `C:\Users\Invitado\Anaconda3\python.exe` como se muestra en la figura 2.8(Der.).
- Guardar el archivo y cerrar la ventana del Bloc de notas.

Después de realizar esta configuración, el software *3DFringe* se podrá ejecutar fácilmente dando doble clic al archivo `start.bat`. Como resultado, se debe abrir la interfaz de usuario del software *3DFringe* que se muestra en la figura 2.9.

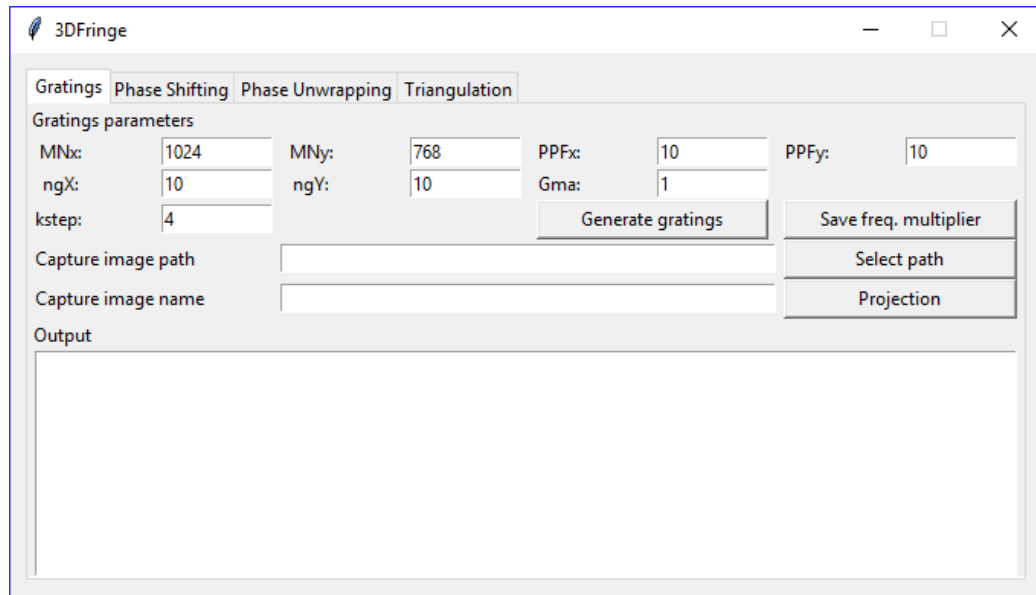


Figura 2.9: Interfaz de usuario del software *3DFringe*.

Al finalizar correctamente estos pasos de instalación y configuración, el software *3DFringe* estará listo para realizar digitalización de objetos tridimensionales usando el método de proyección de franjas. En el manual de usuario se encuentra información detallada explicando cómo usar el software *3DFringe*.

# Capítulo 3

## Manual de usuario

### Contenido

|                                            |    |
|--------------------------------------------|----|
| 3.1. Preámbulo . . . . .                   | 20 |
| 3.2. General . . . . .                     | 21 |
| 3.3. Sección Gratings . . . . .            | 23 |
| 3.4. Sección Phase shifting . . . . .      | 27 |
| 3.5. Sección Phase Unwrapping . . . . .    | 31 |
| 3.6. Sección Triangulation . . . . .       | 35 |
| 3.7. Ejemplo con datos de prueba . . . . . | 40 |

### 3.1. Preámbulo

Este manual proporciona información detallada sobre la interfaz de usuario del software *3DFringe*. En este manual se describe la estructura del software, controles, y valores típicos de configuración. Al final se presentan ejemplos de funcionamiento del software usando datos experimentales.

## 3.2. General

La figura 3.1 muestra la ventana principal del software *3DFringe*. La numeración en la figura señala los elementos que se describen a continuación.

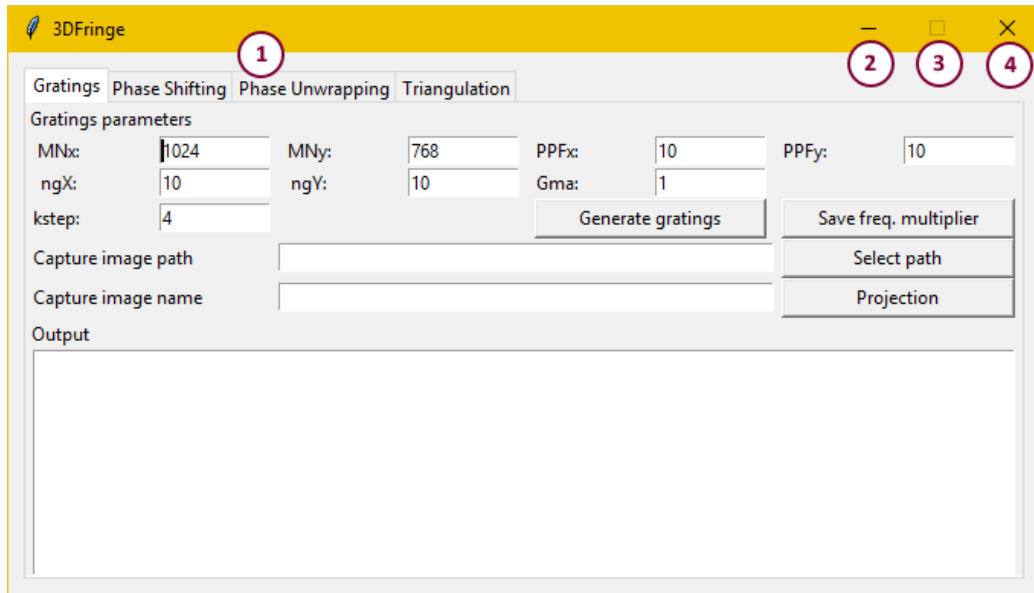


Figura 3.1: Ventana principal del software *3DFringe*.

1. **Barra de secciones:** En esta barra se encuentran las cuatro secciones del sistema:
  - a) Gratings
  - b) Phase Shifting
  - c) Phase Unwrapping
  - d) Triangulation

Cada una de estas secciones permite realizar un proceso para la digitalización tridimensional de objetos. Para cambiar entre secciones solo se debe dar clic sobre el módulo deseado.

2. **Botón minimizar:** Este botón permite minimizar la ventana del software *3DFringe*. Esto ocultará la ventana principal del software en el escritorio, pero estará visible para su posterior restauración en la barra de tareas del usuario.
3. **Botón maximizar:** Esta función esta bloqueada por el sistema. El usuario no puede modificar el tamaño de la ventana del software *3DFringe*.
4. **Botón cerrar:** Botón para cerrar la interfáz de usuario del software *3DFringe*. Todos los procesos iniciados por el software serán detenidos y se liberarán los recursos computacionales reservados por el software *3DFringe*.

### 3.3. Sección Gratings

La figura 3.2 muestra los controles de usuario de la sección **Gratings**. En esta sección, el usuario puede configurar la generación de rejillas (*gratings* en inglés). Además, esta sección tiene la funcionalidad de proyectar las rejillas generadas y capturar automáticamente los patrones de franjas resultantes.

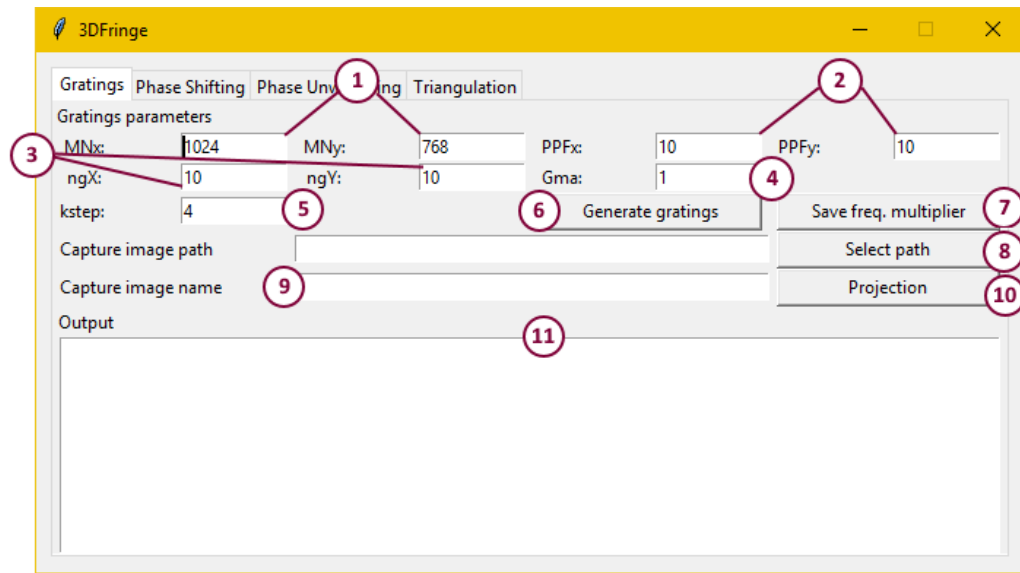


Figura 3.2: Elementos que componen la sección **Gratings**.

Los parámetros configurables en esta sección se describen a continuación.

1. **Resolución:** Número de píxeles a lo ancho (MNx) y alto (MNy) con la que el usuario desea generar las rejillas. Es importante que esta resolución sea igual a la resolución del proyector. Por defecto, el sistema usa como resolución  $1024 \times 768$  píxeles.
2. **Píxeles por franja:** Número de píxeles por franja horizontal (PPFx) y por franja vertical (PPFy). Usualmente 10 píxeles en ambas direcciones.
3. **Número de rejillas:** Número de rejillas horizontales (ngX) y verticales (ngY) a generar. Se recomienda el uso de cinco rejillas en ambas direcciones.

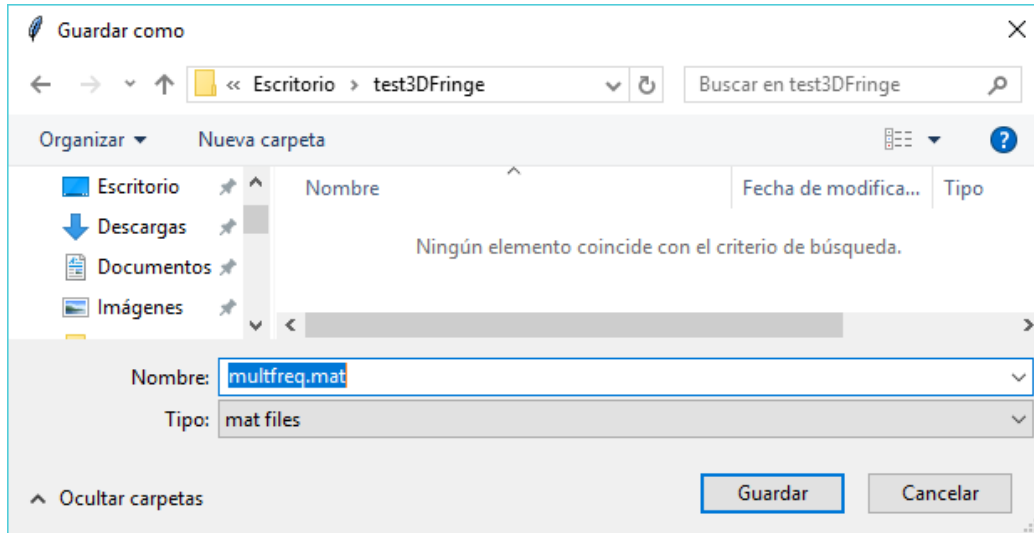


Figura 3.3: Guardar los multiplicadores de frecuencia de las rejillas generadas.

4. **Gamma:** Valor gamma con el que las rejillas serán pre-distorsionadas. Esta pre-distorsión es útil para atenuar el efecto de distorsión gamma introducida por el proyector. Usar  $\text{gamma} = 1$  si no se conoce el valor gamma del proyector.
5. **kstep:** Número (entero) de pasos de fase que serán introducidos a las rejillas. El mínimo número de pasos soportado es  $\text{kstep} = 3$ , pero el resultado es vulnerable a ruido aleatorio en las imágenes. Entre mayor es el número de pasos, la robutez al ruido incrementa, pero el número de imágenes a capturar también será mayor. Se recomienda usar  $\text{kstep} = 4$  para mediciones rápidas y  $\text{kstep} = 6$  para mediciones de mayor precisión.
6. **Generate gratings:** Botón para realizar la generación de rejillas usando los parámetros introducidos por el usuario.
7. **Save freq. multiplier:** Botón para guardar, en un archivo `*.mat`, los multiplicadores de frecuencia asociados a las rejillas generadas. Ver la figura 3.3.

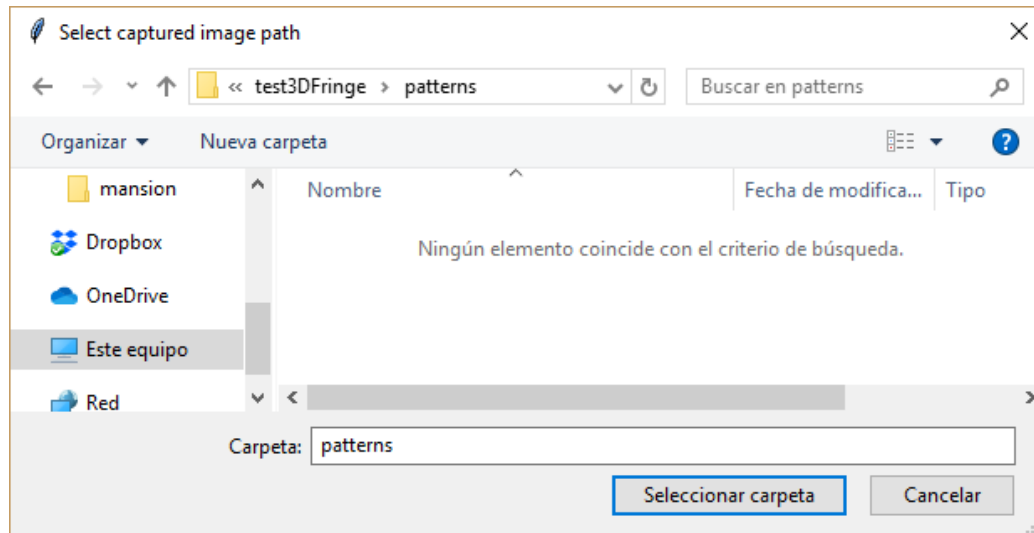


Figura 3.4: Seleccionar carpeta en donde se guardarán los patrones de franjas capturados por la cámara.

8. **Capture image path:** Botón para seleccionar el directorio donde el usuario desea guardar los patrones de franjas capturados por la cámara. Ver la figura 3.4.
9. **Capture image name:** Nombre con el que el sistema guardará las imágenes capturadas por la cámara. Ver la figura 3.5. Si no se especifica un nombre, el sistema asignará el nombre **image**.

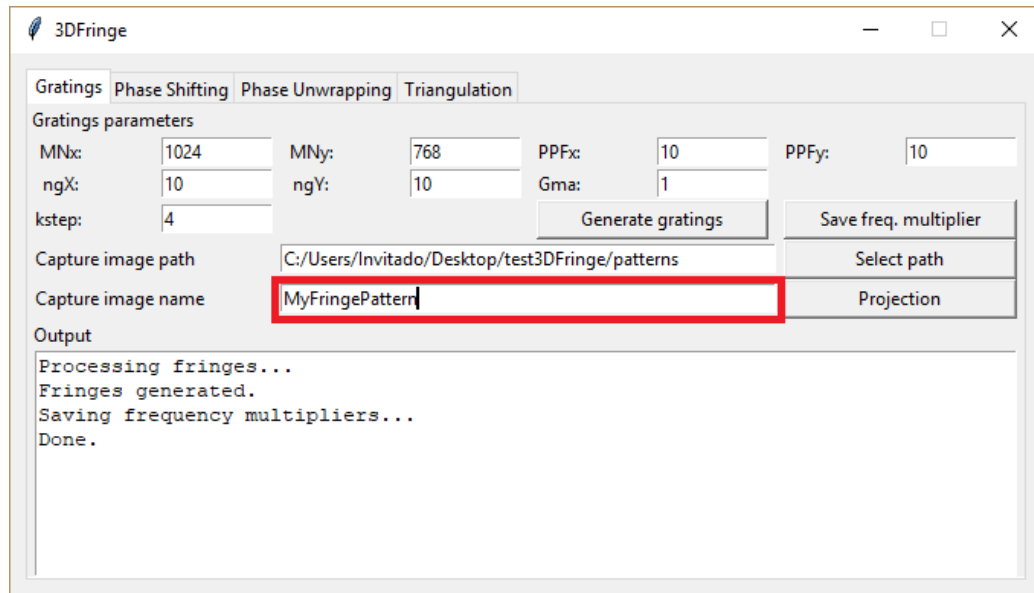


Figura 3.5: Asignar nombre de archivo para los patrones de franjas que serán capturados.

10. **Projection:** Botón para iniciar el proceso de proyección de rejillas y captura automática de patrones de franjas.
11. **Output:** Esta es una salida del sistema, donde se muestra al usuario el estatus del proceso actual, mensajes de éxito, mensajes de error, entre otros. Ésta es una caja de texto que el usuario solo puede visualizar; no permite agregar o eliminar texto.

### 3.4. Sección Phase shifting

La figura 3.6 muestra los controles de usuario para la extracción de fase envuelta usando el método de corrimiento de fase (*phase-shifting* en inglés). Esta sección está diseñada para procesar los patrones de franjas capturados en la sección **Gratings**. Sin embargo, cualquier conjunto de patrones de franjas con corrimientos de fase regulares puede ser procesado.

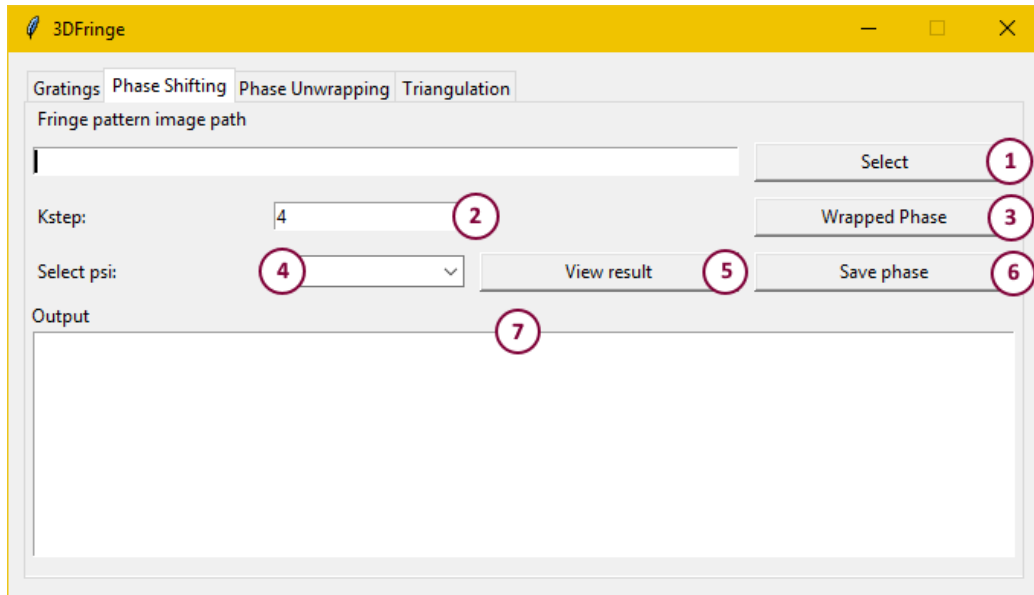


Figura 3.6: Elementos de control de la sección **Phase shifting**.

Los parámetros configurables en esta sección se describen a continuación.

1. **Select** (Fringe pattern image path): Al dar clic sobre este botón, se abrirá una ventana emergente como la que se muestra en la figura 3.7. Esta ventana permite seleccionar la carpeta donde están almacenados los patrones de franjas capturados en la sección **Gratings**. Alternativamente, se puede seleccionar alguna otra carpeta que contenga patrones de franjas que se deseen procesar.

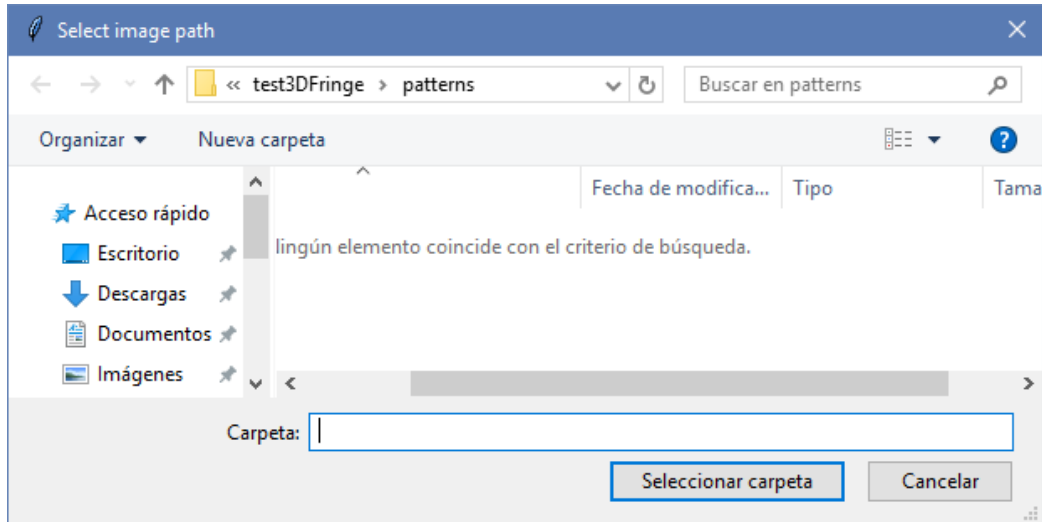


Figura 3.7: Ventana para seleccionar la carpeta donde están almacenados los patrones de franjas a ser procesados en la sección **Phase Shifting**.

2. **ksteps**: Número (entero) indicando los pasos de fase realizados durante la proyección de franjas.
3. **Wrapped Phase**: Botón para ordenar la extracción de fase envuelta usando el método de corrimiento de fase.
4. **Select psi**: Menú desplegable que se auto llena cuando el sistema finaliza el proceso de extracción de fase envuelta. El usuario puede elegir cualquier valor de **psi** que se desea pre-visualizar.
5. **View result**: Botón para ordenar la pre-visualización de la fase envuelta **psi**. Emergerá una ventana con el mapa de fase solicitado como se muestra en la figura 3.8.

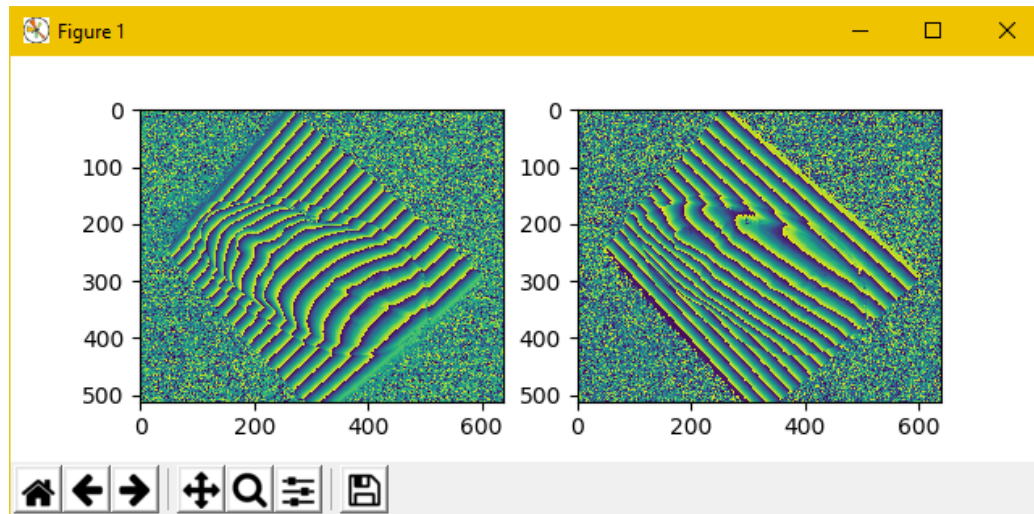


Figura 3.8: Ventana emergente con la pre-visualización del mapa de fase envuelto  $\psi$  seleccionado.

6. **Save phase:** Al dar clic sobre este botón, aparecerá una ventana como la que se muestra en la figura 3.9. Esta ventana le permitirá seleccionar la dirección donde el software *3DFringe* guardará, en un archivo *\*.mat*, la fase envuelta calculada.
7. **Output:** Esta es una salida del sistema donde se muestra al usuario el estatus del proceso actual, mensajes de éxito, mensajes de error, entre otros. Ésta es una caja de texto que el usuario solo puede visualizar; no permite agregar o eliminar texto.

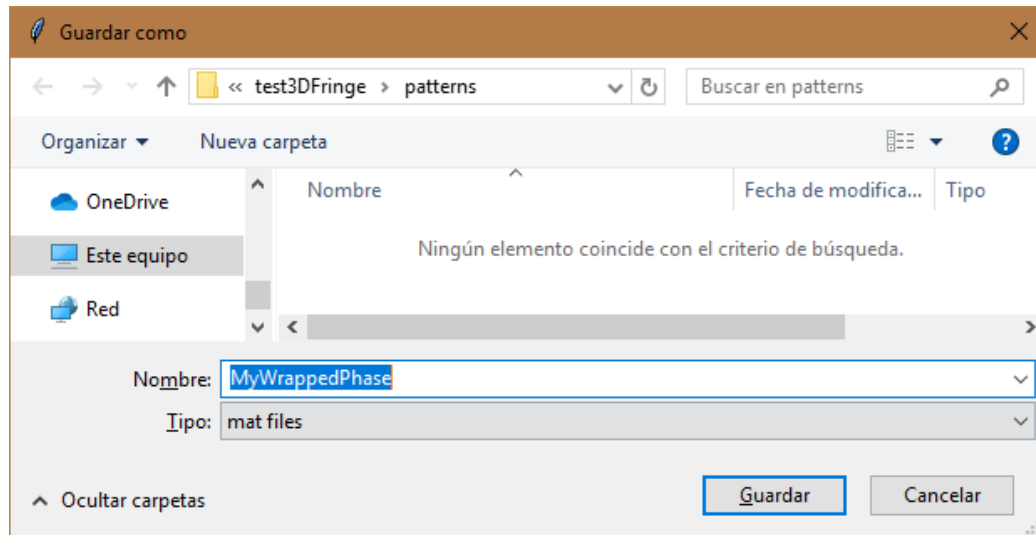


Figura 3.9: Ventana para indicar la dirección donde se guardará la fase envuelta calculada.

### 3.5. Sección Phase Unwrapping

La figura 3.10 muestra los controles de usuario para realizar desenvolvimiento de fase (*phase unwrapping* en inglés) usando el método multifrecuencia. Esta sección permite al usuario configurar y ejecutar el proceso de desenvolvimiento de fase, pre-visualizar el resultado obtenido y guardar los datos calculados.

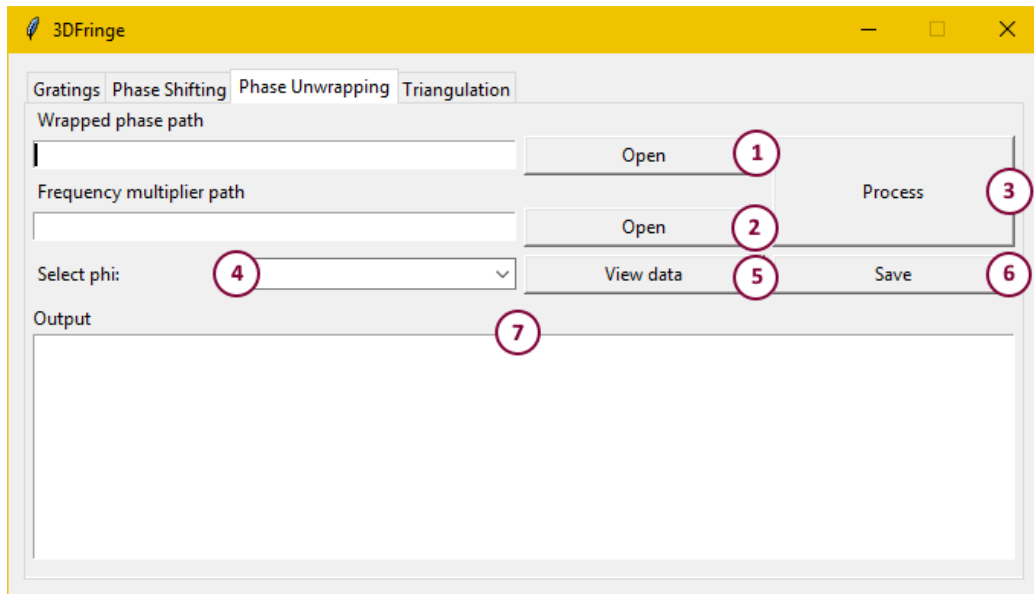


Figura 3.10: Elementos de control de la sección **Phase Unwrapping**. Los elementos están numerados siguiendo el orden en que se deben usar.

Los controles de usuario disponibles en esta sección se describen a continuación.

1. **Open** (Wrapped phase path): Botón para indicar al sistema, a través de una ventana como se muestra en la figura 3.11, el archivo `*.mat` que contiene la fase envuelta que se desea procesar. Ver sección **Phase Shifting**.
2. **Open** (Frequency multiplier path): Botón para indicar al sistema, a través de una ventana como se muestra en la figura 3.12, el archivo `*.mat` donde se encuentran los multiplicadores de frecuencia asociados a las rejillas usadas durante la proyección. Ver sección **Gratings**.

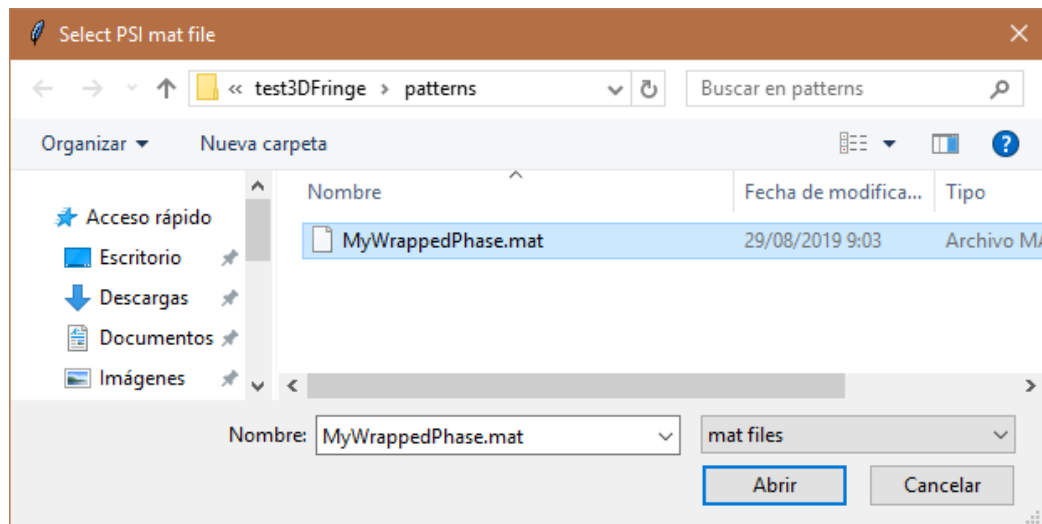


Figura 3.11: Ventana para seleccionar el archivo \*.mat que contiene la fase envuelta a procesar.

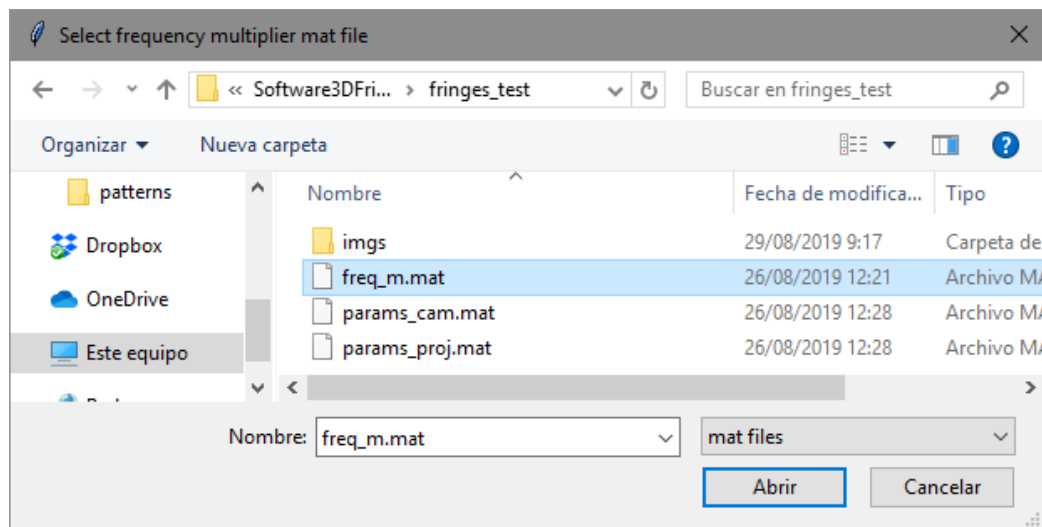


Figura 3.12: Ventana para seleccionar el archivo \*.mat que contiene los multiplicadores de frecuencia correspondientes a las rejillas proyectadas.

3. **Process:** Botón para iniciar el proceso de desenvolvimiento de fase. En el control de salida **Output** del sistema, se mostrará el mensaje  
`Unwrapping phase...`  
Si el proceso finaliza con éxito, el control de salida desplegará el mensaje  
`Phase unwrapped.`
4. **Select phi:** Menú desplegable que se auto llena cuando el sistema finaliza el proceso de desenvolvimiento de fase. El usuario puede elegir cualquier valor de **phi** que se desea pre-visualizar.
5. **View data:** Botón para ordenar la pre-visualización de la fase envuelta **phi** seleccionada. Emergerá una ventana con el mapa de fase desenvuelto solicitado como se muestra en la figura 3.13.
6. **Save:** Botón para guardar la fase desenvuelta calculada por el sistema. Al dar clic sobre este botón, aparecerá una ventana como la que se muestra en la figura 3.14. Esta ventana le permitirá seleccionar la dirección donde se guardará, en un archivo `*.mat`, la fase envuelta calculada.
7. **Output:** Esta es una salida del sistema, donde se muestra al usuario el estatus del proceso actual, mensajes de éxito, mensajes de error, entre otros. Ésta es una caja de texto que el usuario solo puede visualizar; no permite agregar o eliminar texto.

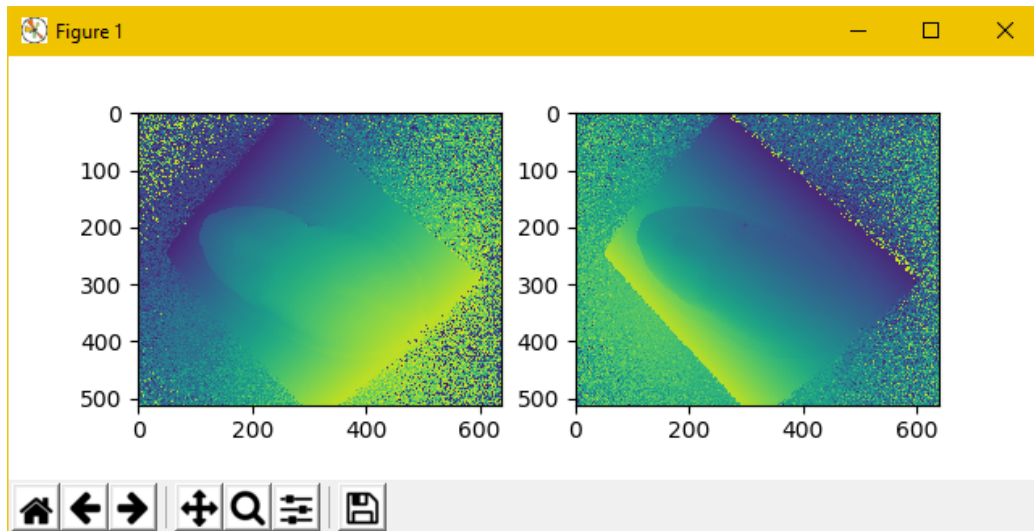


Figura 3.13: Ventana emergente con la pre-visualización de la función de fase desenvuelta seleccionada.

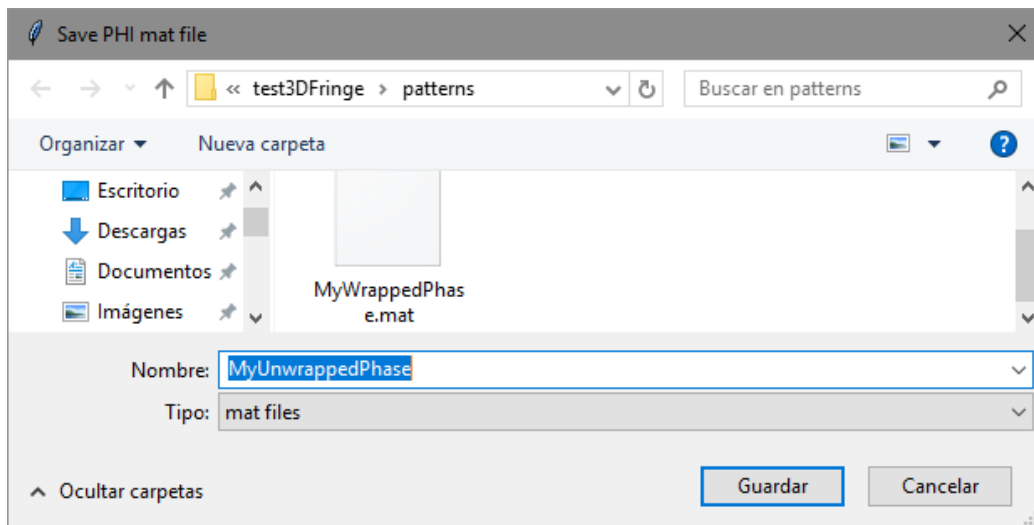


Figura 3.14: Ventana para indicar la dirección donde se guardará la fase desenvuelta calculada.

### 3.6. Sección Triangulation

Esta sección permite al usuario calcular las coordenadas  $(x, y, z)$  de cada punto observado por la cámara en la superficie del objeto analizado. Las coordenadas calculadas proporcionarán la información suficiente para reconstruir digitalmente la superficie del objeto analizado. Esta rutina computacional requiere como datos de entrada tanto los mapas de fase extraídos de los patrones de franjas, así como los parámetros de la cámara y el proyector<sup>1</sup>.

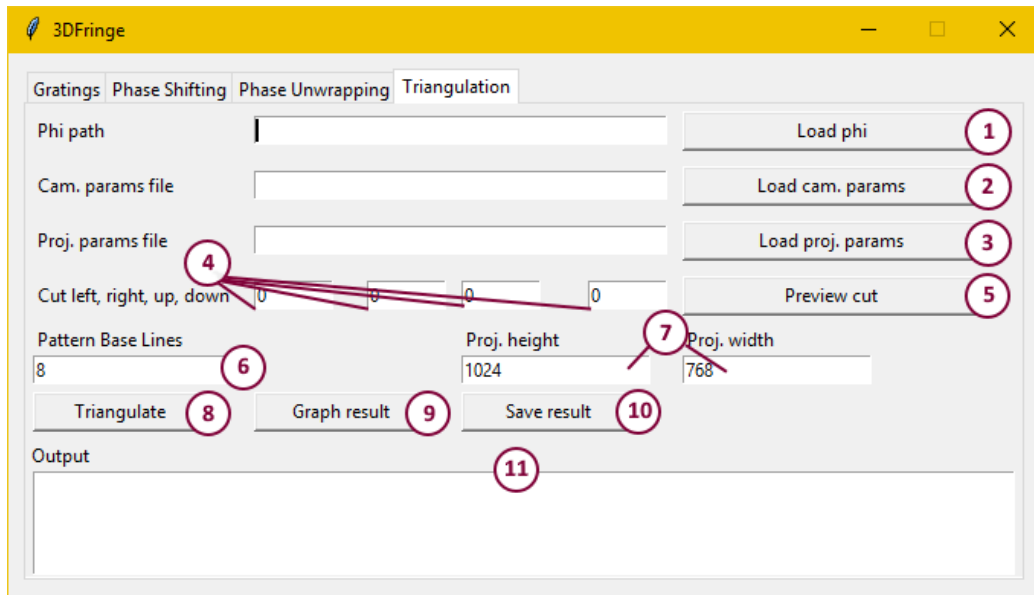


Figura 3.15: La sección **Triangulation** con los elementos que la componen.

Los controles de usuario disponibles en esta sección se describen a continuación.

1. **Load phi:** Botón para cargar en el sistema el archivo `*.mat` que contiene la fase desenvuelta que se desea procesar. Ver sección **Phase Unwrapping**. Se abrirá una ventana como la que se muestra en la figura 3.16(arriba).

<sup>1</sup>NOTA. Los parámetros de la cámara y el proyector deben estar disponibles por un proceso previo de calibración del sistema. Ver la sección *Requerimientos* del manual de instalación.

2. **Load cam. params:** Botón para cargar el archivo `*.mat` que contiene los parámetros de la cámara. Ver *Requerimientos* del manual de instalación. Se abrirá una ventana como la que se muestra en la figura 3.16(medio).
3. **Load proj. params:** Botón para cargar el archivo `*.mat` que contiene los parámetros del proyector. Ver *Requerimientos* del manual de instalación. Se abrirá una ventana como la que se muestra en la figura 3.16(abajo).
4. **Cut left, right, up, down:** Cajas de texto donde el usuario especifica el número de píxeles que se cortan al mapa de fase con la finalidad de evitar procesar zonas de ruido. Los cortes se realizan en el orden: izquierda, derecha, arriba y abajo. Ver figura 3.17.
5. **Preview cut:** Botón para ordenar el despliegue del mapa de fase recortado para fines de pre-visualización. Ver figura 3.17(Ventanas a la derecha).
6. **Pattern Base Lines:** En esta caja de texto se indica el número de líneas base que contiene el patrón de calibración usado para la estimación de los parámetros de la cámara y el proyector.
7. **Resolución del proyector:** Cajas de texto donde se especifica la resolución del proyector a lo ancho (Proj. width) y a lo alto (Proj. height).
8. **Triangulation:** Botón que ordena la ejecución del método de triangulación para calcular las coordenadas  $(x, y, z)$  de cada punto observado en la superficie del objeto.
9. **Graph result:** Botón para ordenar el despliegue de una gráfica 3D con los datos calculados como se muestra en la figura 3.18(Arriba).
10. **Save result:** Botón para guardar los datos  $(x, y, z)$  calculados por el sistema. Al dar clic sobre este botón, aparecerá una ventana como la que se muestra en la figura 3.18(Abajo). Esta ventana permite seleccionar la dirección donde se guardará, en un archivo `*.mat`, los datos calculados.

11. **Output:** Esta es una salida del sistema, donde se muestra al usuario el estatus del proceso actual, mensajes de éxito, mensajes de error, entre otros. Ésta es una caja de texto que el usuario solo puede visualizar; no permite agregar o eliminar texto.

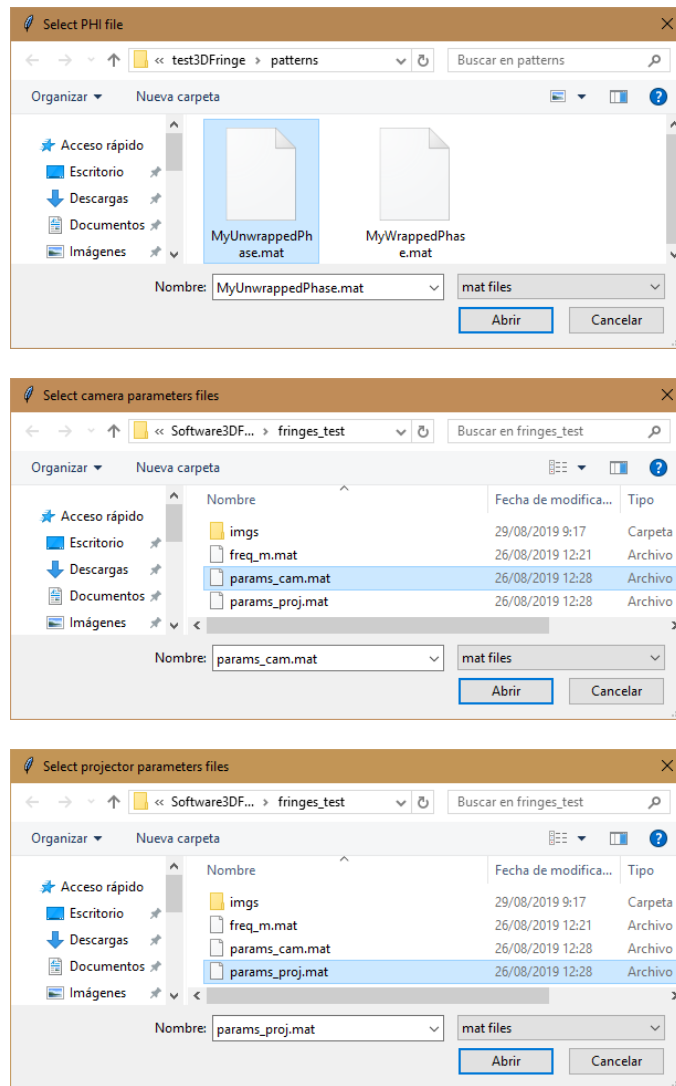


Figura 3.16: Ventana para abrir archivo que contiene: (arriba) fase, (medio) parámetros de la cámara, y (abajo) parámetros del proyector.

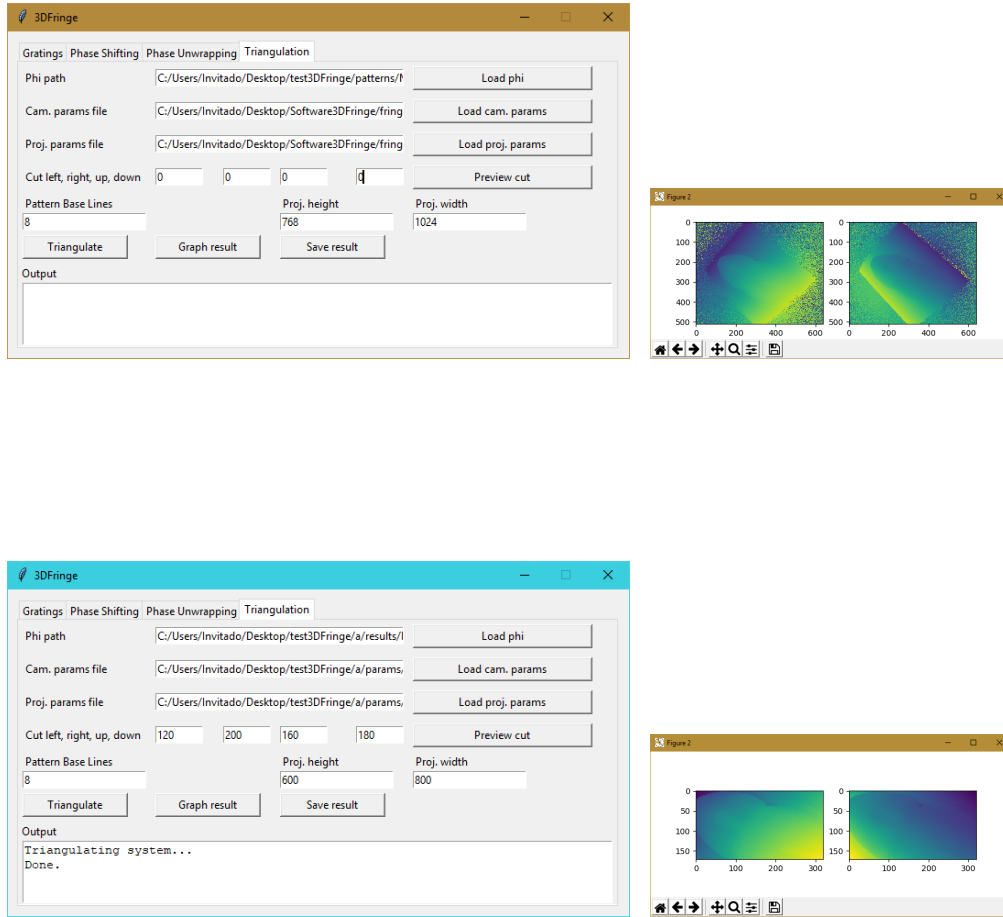


Figura 3.17: Recorte de una imagen usando las cajas **Left**, **right**, **up**, **down**. (Arriba) Especificando imagen con su tamaño original (configuración [0,0,0,0]). (Abajo) Solicitando recortar la imagen 120 píxeles a la izquierda, 200 píxeles a la derecha, 160 píxeles arriba, y 180 píxeles abajo (configuración [120, 200, 160, 180]).

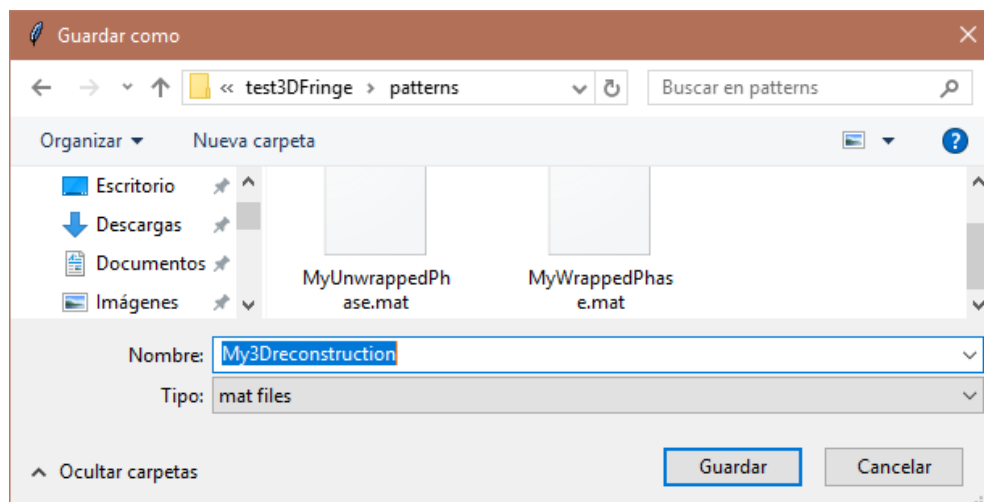
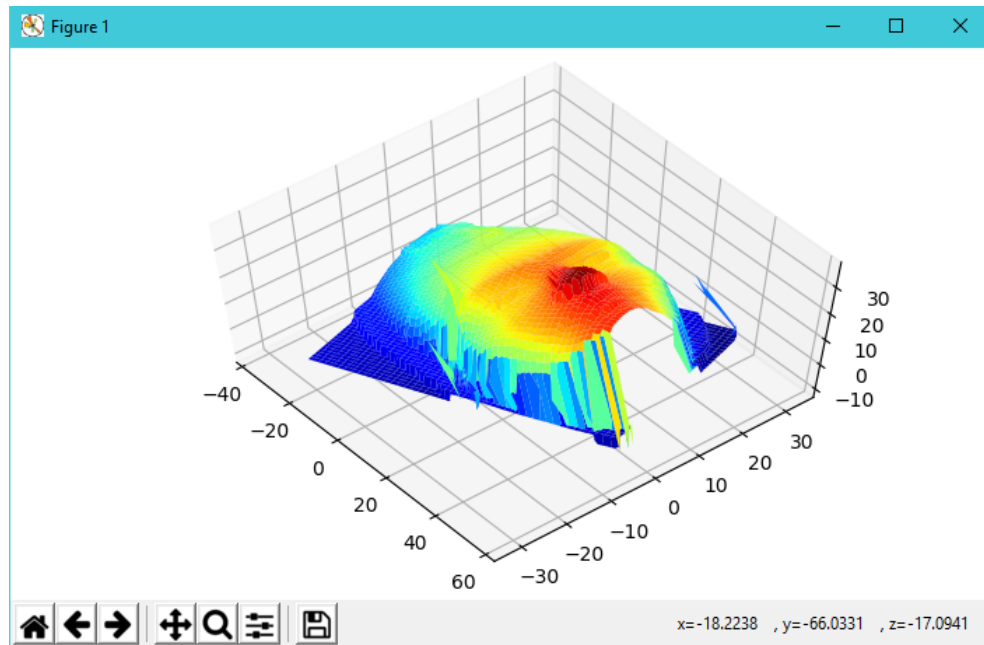


Figura 3.18: (Arriba) Visualización en una gráfica  $xyz$  del objeto tridimensional reconstruido. (Abajo) Ventana para indicar la dirección donde se guardará la información tridimensional calculada.

### 3.7. Ejemplo con datos de prueba

Para fines de referencia, a continuación se ilustra la operación del software *3DFringe* para la digitalización de un cono. Para esto, se procesarán los datos experimentales proporcionados con el software. El procesamiento de datos se realizó con el siguiente procedimiento.

1. Generación de rejillas en la sección **Gratings**. Como se muestra en la figura 3.19, se especificó una resolución de  $800 \times 600$  para generar 5 rejillas en ambos ejes del proyector, con 10 píxeles por franja y 4 corrimientos de fase. Se guardaron los multiplicadores de frecuencia. Al dar clic en el botón **Projection**, se desplegaron las rejillas y se capturaron automáticamente 40 patrones de franjas.
2. Extracción de fase envuelta en la sección **Phase Shifting**. Se carga el directorio donde se encuentran almacenados los patrones de franjas capturados previamente. En el campo **kstep** se indica el número de corrimientos de fase (cuatro). Se guarda la fase envuelta calculada por el sistema. En la figura 3.20 se muestra el resultado de este procedimiento.
3. La fase envuelta contiene discontinuidades debido a la naturaleza periódica de las rejillas proyectadas. Para eliminar estas discontinuidades, es necesario realizar el desenvolvimiento de fase usando la sección **Phase Unwrapping**. Como se muestra en la figura 3.21, solo se deben cargar los archivos guardados previamente (multiplicadores de frecuencia y fase envuelta). Los datos proporcionados son procesados y el resultado se guarda en un archivo **\*.mat**.
4. Triangulación usando un sistema cámara-proyector calibrado. Como se muestra en la figura 3.22, solo se debe cargar la fase desenvuelta, los parámetros de calibración, la resolución del proyector, y el número de líneas base del patrón de calibración usado. Adicionalmente, se puede configurar un recorte en los mapas de fase con la finalidad de omitir los bordes de los mapas de fase donde no hay información útil.

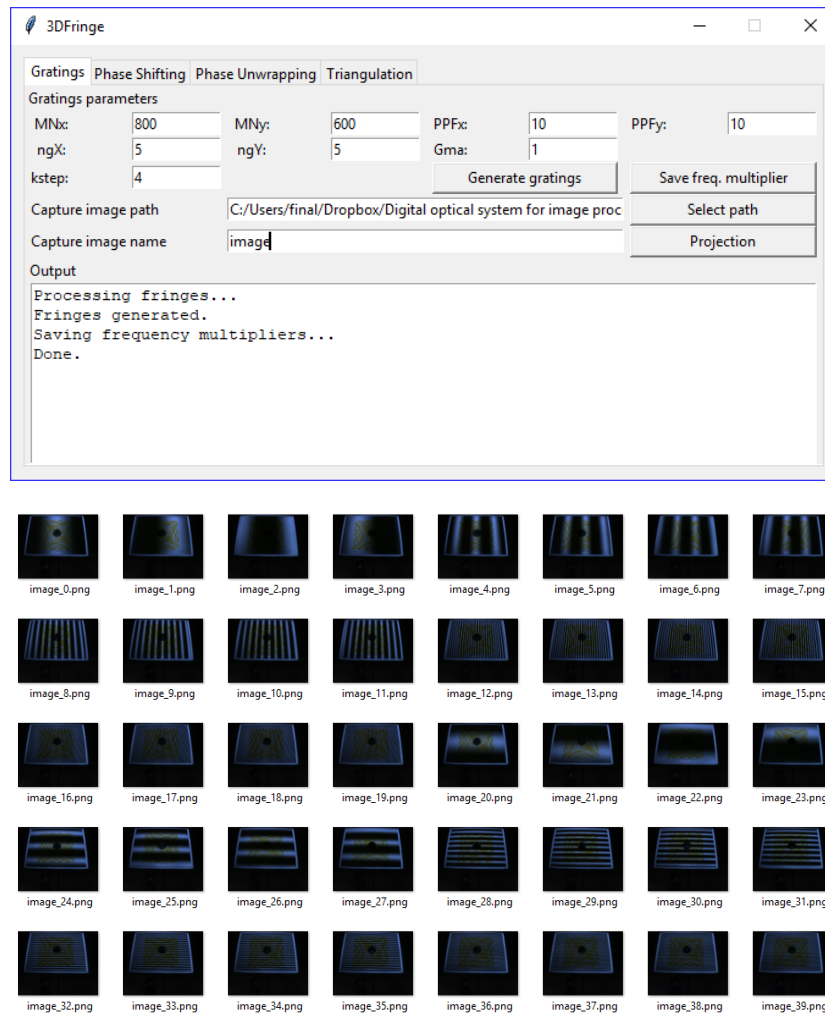


Figura 3.19: Generación de rejillas. (Arriba) Parámetros utilizados en la sección *Gratings*. (Abajo) Patrones de franjas capturados.

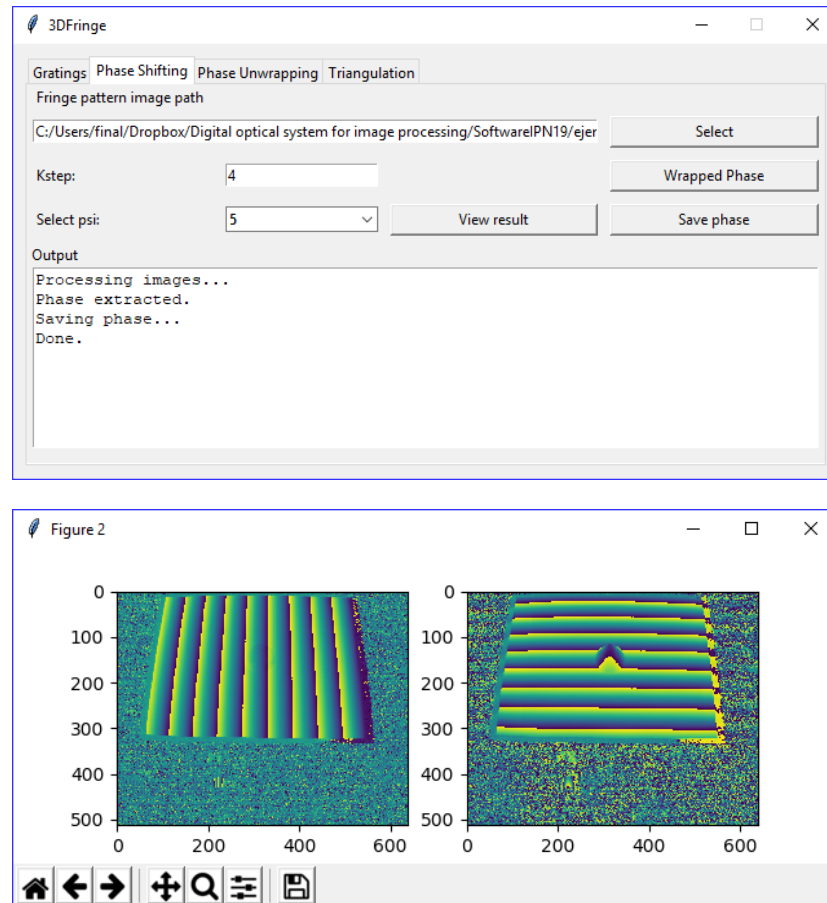


Figura 3.20: Extracción de fase envuelta. (Arriba) Parámetros utilizados en la sección Phase Shifting. (Abajo) Visualización de la fase envuelta calculada.

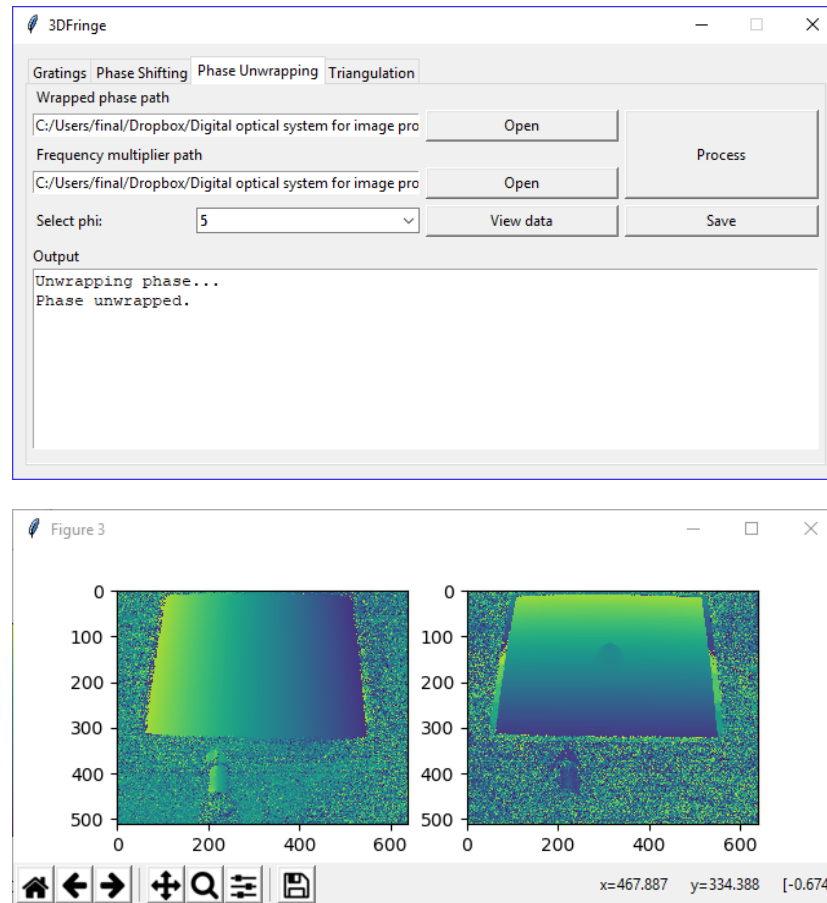


Figura 3.21: Desenvolvimiento de fase. (Arriba) Parámetros utilizados en la sección Phase Unwrapping. (Abajo) Visualización de la fase desenvuelta calculada.

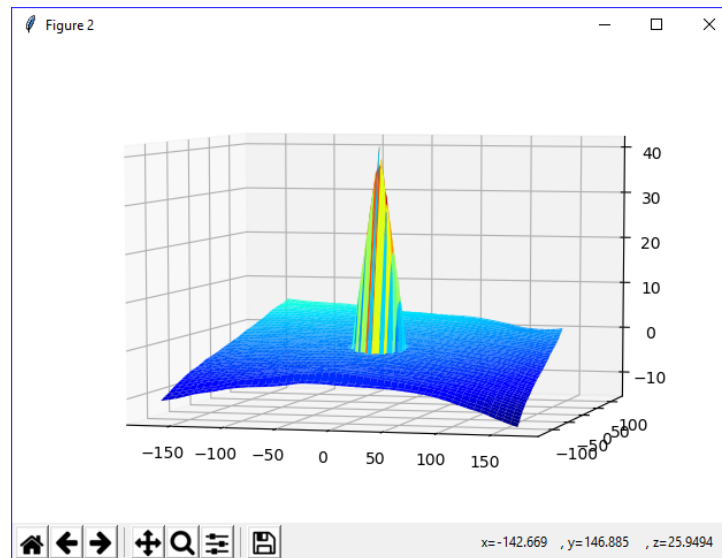
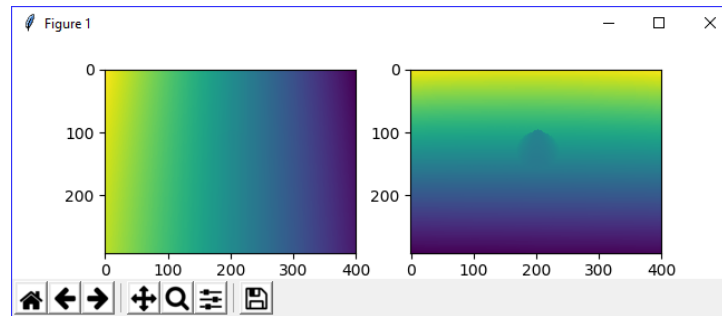
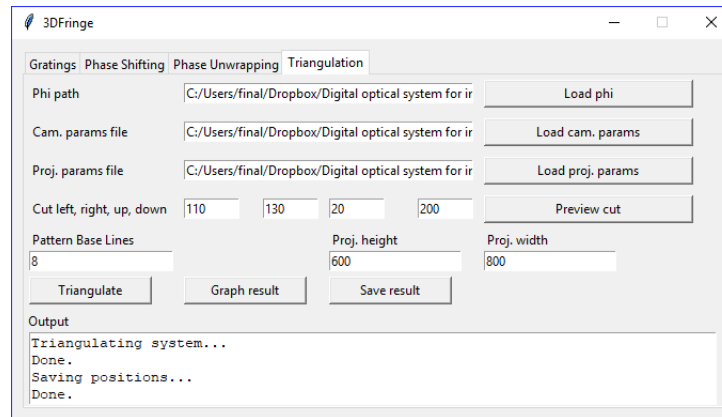


Figura 3.22: Triangulación. (Arriba) Parámetros utilizados en la sección Triangulation. (Medio) Fase desenvuelta recortada para eliminar el ruido en los bordes. (Abajo) Resultado final del proceso de triangulación.

# Capítulo 4

## Manual técnico

### Contenido

|                                           |    |
|-------------------------------------------|----|
| 4.1. Preámbulo . . . . .                  | 45 |
| 4.2. Descripción de campos . . . . .      | 46 |
| 4.2.1. Sección Gratings . . . . .         | 46 |
| 4.2.2. Sección Phase Shifting . . . . .   | 48 |
| 4.2.3. Sección Phase Unwrapping . . . . . | 50 |
| 4.2.4. Sección Triangulation . . . . .    | 52 |

### 4.1. Preámbulo

El software *3DFringe* proporciona al usuario las herramientas necesarias para digitalizar la superficie de un objeto tridimensional usando el método de proyección de franjas. Sin embargo, para que el sistema opere adecuadamente, es necesario conocer las restricciones de este software. Asimismo, en este manual se abordarán los detalles técnicos de la implementación para mantenimiento futuro del código fuente.

## 4.2. Descripción de campos

### 4.2.1. Sección Gratings

En esta sección, el sistema requerirá que estén conectados por lo menos una cámara y un proyector a la computadora donde se ejecutará el software *3DFringe*. Además se requiere que la cámara y el proyector estén calibrados; es decir, el usuario debe tener disponibles los archivos que contienen los parámetros de cada dispositivo. Estos parámetros serán requeridos por el software en la sección *Triangulation*.

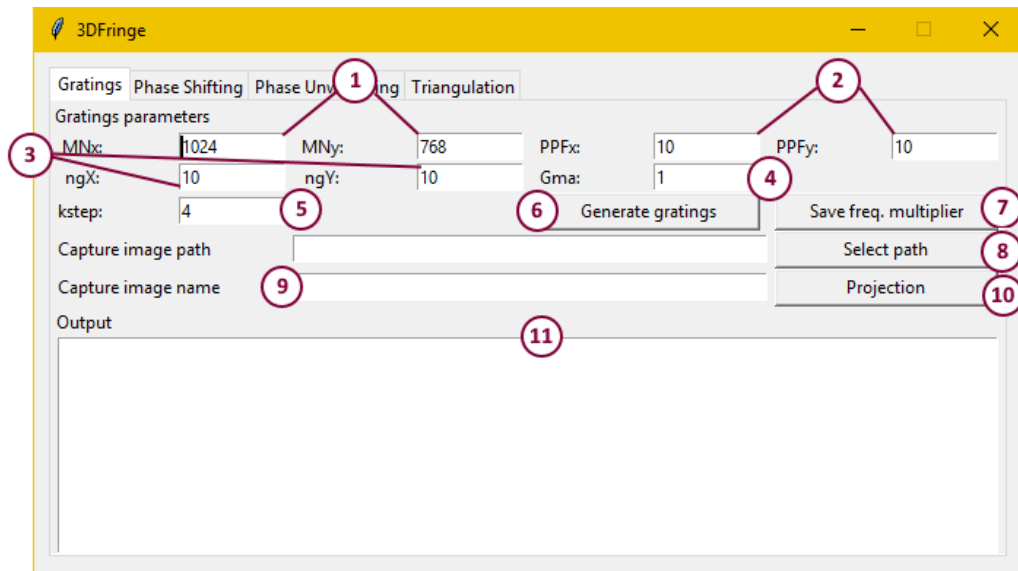


Figura 4.1: Campos de la sección **Gratings**.

- En las cajas de texto marcadas con (1), (2), (3), y (5), el sistema solo admite números enteros positivos (dato tipo *integer*). Cualquier otro carácter provocará un mensaje de error. Este mensaje será desplegado en la salida del sistema (*Output*).
- En la caja de texto marcada con (4), el sistema solo admite números racionales positivos (dato tipo *float*). Cualquier otro carácter provocará un mensaje de error. Este mensaje será desplegado en la salida del sistema (*Output*).

- Con el botón marcado con (6), el sistema llamará la función generadora de rejillas si los datos introducidos anteriormente son válidos. En caso contrario, el sistema mostrará un mensaje de error en la salida del sistema (*Output*).
- Al dar clic sobre el botón marcado con (7), el sistema mostrará un cuadro de dialogo para guardar los resultados en un archivo con extensión \*.mat. El archivo guardado contendrá dos variables de tipo matriz: alpx y alpy.
- En el botón marcado con (8), el sistema solo permite elegir una ubicación en la computadora (carpeta).
- En la caja de texto marcada con (9), el usuario puede asignar un nombre raíz para las imágenes que serán capturadas por la cámara durante la proyección de franjas. Si el usuario deja esta caja de texto en blanco, el sistema asignará por defecto el nombre `image`.
- El botón marcado con (10) inicia un bucle donde toma una rejilla, la despliega en el proyector, y captura con la cámara el patrón de franjas correspondiente. Este proceso se repite por cada rejilla disponible.
- Los patrones de franjas capturados se almacenarán como imágenes en la carpeta indicada en formato \*.png.
- Si el sistema no detecta una cámara conectada, entonces aparecerá un mensaje error en la salida del sistema (*Output*). En este mensaje se indica la ausencia de cámaras conectadas.
- La caja de texto marcada con (11), mostrará al usuario el estatus de los procesos en ejecución, mensajes de error, entre otros.

### 4.2.2. Sección Phase Shifting

En esta sección el usuario deberá conocer la ruta en la computadora donde se encuentran los patrones de franjas capturados previamente en la sección **Gratings**.

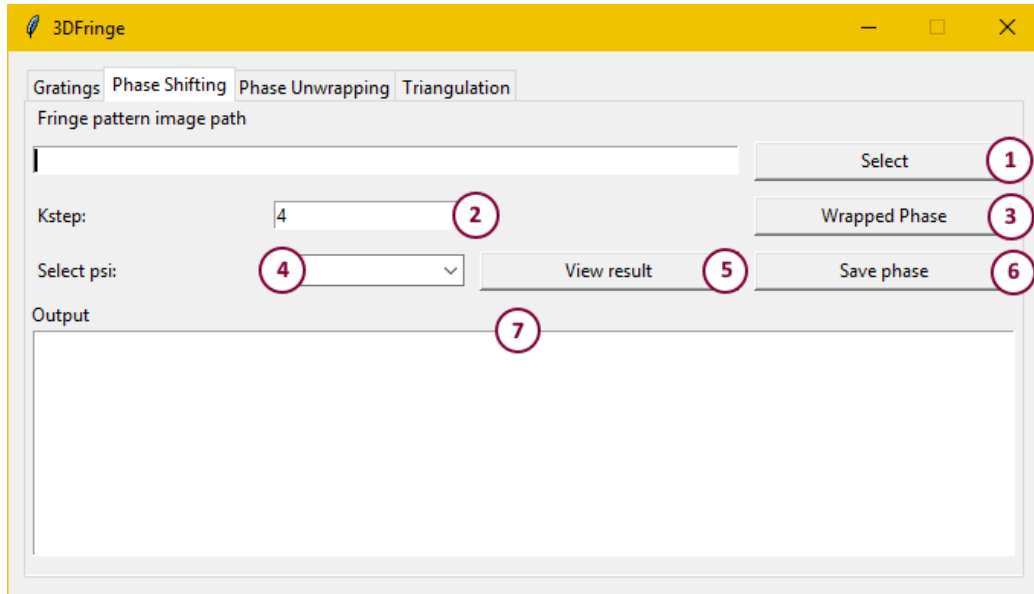


Figura 4.2: Campos de la sección **Phase Shifting**.

- El botón marcado con (1) permite al usuario seleccionar solo ubicaciones (carpetas), no archivos.
- En la caja de texto marcada con (2), el usuario debe introducir el mismo número **Kstep** (número de corrimientos de fase) usado en la sección **Gratings**.
- En el botón marcado con (3), si el sistema encuentra errores en los parámetros al no encontrar imágenes, o tener un valor de **Kstep** incorrecto, mostrará el siguiente mensaje de error

Given parameters are not correct.

Si no se encuentran errores, el sistema iniciará el proceso de extracción de fase y notificará al usuario el momento en que finalice con un mensaje de salida (*Output*).

- En el menú desplegable marcado con (4), se cargarán automáticamente los índices correspondientes a los mapas de fase **psi** calculados. Este control permite elegir un mapa de fase para visualización entre todos los mapas de fase disponibles.
- Con el botón marcado con (5), el sistema mostrará en otra ventana un gráfico con el mapa de fase **psi** seleccionado usando el control marcado con (4). Si se da clic sobre el botón (5) antes de que el sistema haya calculado los mapas de fase, se mostrará un mensaje de error en la salida del sistema (*Output*).
- El botón marcado con (6) generará una nueva ventana para guardar los mapas de fase calculados en un archivo con extensión **\*.mat**. El archivo guardado contendrá las siguientes dos variables de tipo matriz: **psix** y **psiy**. Estas variables contendrán los mapas de fase envueltos en valores de tipo flotante entre  $-\pi$  y  $+\pi$  radianes.

### 4.2.3. Sección Phase Unwrapping

En esta sección, el usuario deberá tener disponibles los siguientes archivos.

1. Archivo `*.mat` donde están almacenados los mapas de fase envueltos (generados en la sección **Phase Shifting**).
2. Archivo `*.mat` donde están almacenados los multiplicadores de frecuencia asociados a las rejillas usadas en la sección **Gratings**.

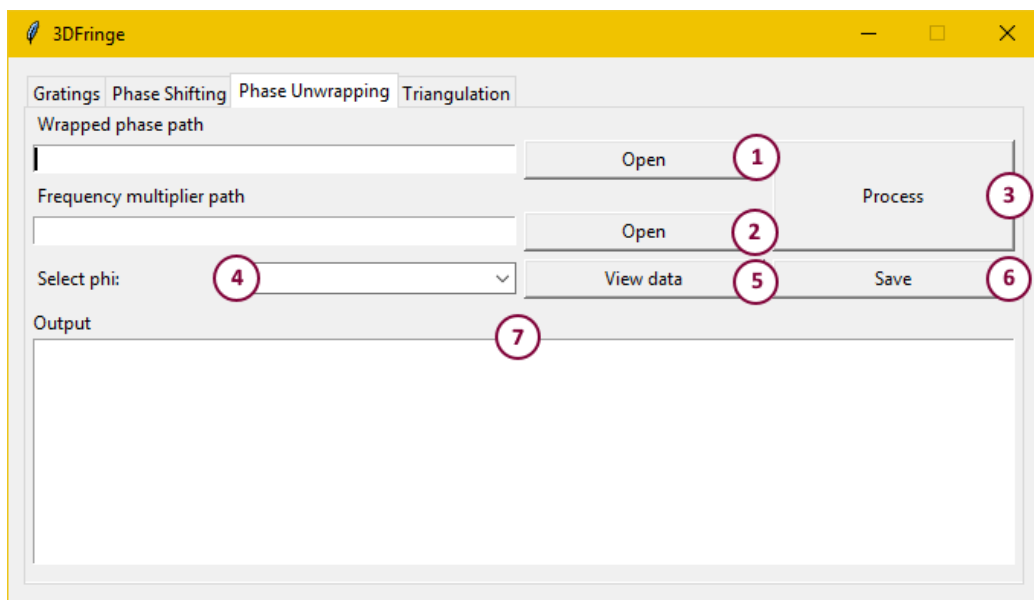


Figura 4.3: Campos de la sección **Phase Unwrapping**.

- En los botones marcados con (1) y (2), el usuario solo podrá seleccionar ubicaciones (carpetas), no archivos.
- El botón marcado con (3) iniciará el proceso de desenvolvemento de fase solo si las rutas especificados por el usuario contienen archivos `*.mat` con las variables requeridas (`psix`, `psiy`, `alpx`, y `alpy`). El software notificará al usuario, con mensajes en la salida (*Output*), si encontró algún error o si finalizó el proceso con éxito.

Si las rutas especificadas no contiene los archivos con las variables requeridas, el sistema mostrará el mensaje de error correspondiente en la salida del sistema (*Output*).

- Después de que el sistema finaliza el proceso de desenvolvimiento de fase, el menú desplegable marcado con (4) se cargará automáticamente con los índices de los mapas de fase **phi** disponibles. El usuario podrá seleccionar algún mapa de fase disponible para posterior visualización usando el número de índice correspondiente.
- Con el botón marcado con (5), el sistema mostrará en otra ventana un gráfico desplegando el mapa de fase **phi** seleccionado. Si el sistema aún no ha realizado el desenvolvimiento de fase (no hay ningún mapa de fase disponible), el sistema mostrará un mensaje de error en la salida del sistema (*Output*).
- El botón marcado con (6) ordenará al sistema guardar en un archivo **\*.mat** las siguientes variables de tipo matriz: **phix**, y **phiy**. Estas variables contendrán los mapas de fase calculados por el sistema.

#### 4.2.4. Sección Triangulation

En esta sección, el usuario deberá tener disponibles los siguientes archivos.

1. Archivo \*.mat que contiene la fase desenvuelta.
2. Archivo \*.mat que contiene los parámetros de la cámara usada para capturar los patrones de franjas durante el proceso de proyección (sección **Gratings**).
3. Archivo \*.mat que contiene los parámetros del proyector usado para desplegar rejillas (sección **Gratings**).

Además, el usuario deberá proporcionar la siguiente información.

- a). El número de líneas base del patrón usado para calibrar la cámara y el proyector.
- b). La resolución del proyector.

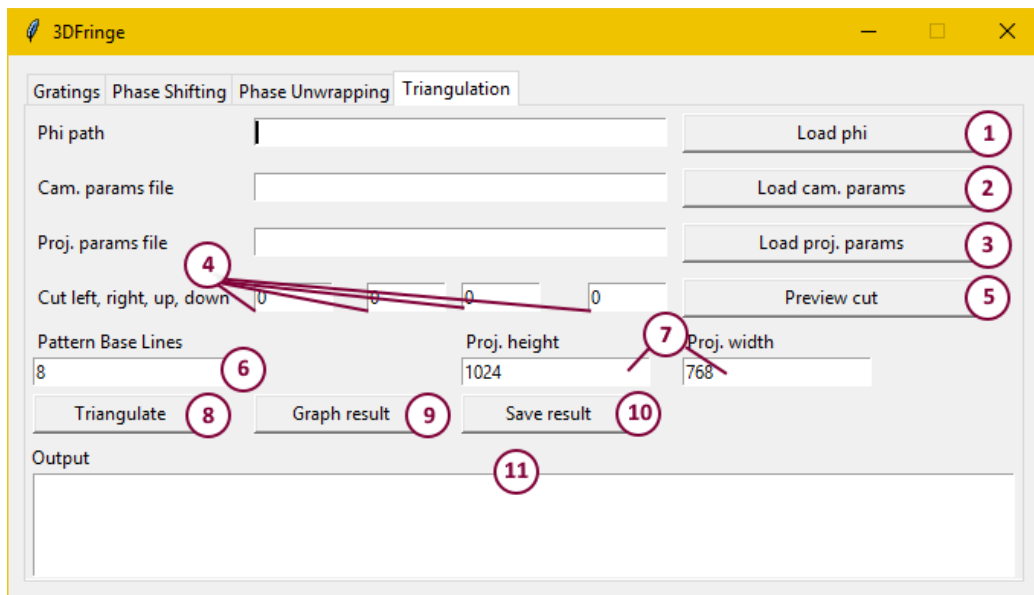


Figura 4.4: Campos de la sección Triangulation.

- El botón marcado con (1) permite seleccionar archivos solo con extensión `*.mat`. El archivo seleccionado debe contener las variables `phix` y `phiy` que se generaron en la sección **Phase Unwrapping**.
- El botón marcado con (2) permite seleccionar archivos solo con extensión `*.mat`. El archivo seleccionado debe contener las variables `Kc`, `Rc` y `tc`. Estas variables representan los parámetros intrínsecos,  $K_c$ , y extrínsecos,  $(R_c, \mathbf{t}_c)$ , de la cámara usada en la sección **Gratings** para capturar los patrones de franjas.
- El botón marcado con (3) permite seleccionar archivos solo con extensión `*.mat`. El archivo seleccionado debe contener las variables `Kp`, `Rp` y `tp`. Estas variables representan los parámetros intrínsecos,  $K_p$ , y extrínsecos,  $(R_p, \mathbf{t}_p)$ , del proyector usado en la sección **Gratings** para proyectar rejillas sobre el objeto analizado.
- En las cajas de texto marcadas con (4), el sistema solo admite la entrada de números enteros positivos. Cualquier carácter introducido fuera de esta restricción causará un mensaje de error en los datos que se mostrará en la salida del sistema (*Output*).
- El botón marcado con (5) mostrará una ventana emergente con la pre-visualización del corte especificado solo si los valores de corte son correctos. De lo contrario, se mostrará un mensaje de error en la salida del sistema (*Output*).
- En las cajas de texto marcadas con (6) y (7) solo se admite la entrada de números enteros positivos. Cualquier carácter introducido fuera de esta restricción causará un mensaje de error en los datos que se desplegará en la salida del sistema (*Output*).

- El botón marcado con (8) iniciará el proceso de triangulación solo si los datos introducidos en los campos anteriores son válidos. En caso contrario, se mostrará un mensaje de error en la salida del sistema (*Output*).
- El botón marcado con (9) mostrará en otra ventana un gráfico *xyz* con los resultados calculados por el sistema. En el caso de dar clic sobre este botón antes de que el sistema halla finalizado con éxito el proceso de triangulación, se desplegará un mensaje de error en la salida del sistema (*Output*).
- El botón marcado con (10) permite guardar, en un archivo con extensión *\*.mat*, las variables de tipo matriz: **Px**, **Py**, y **Pz**. Estas matrices contienen las coordenadas  $(x, y, z)$  correspondientes a cada punto en la superficie del objeto observado por la cámara. Esta información es suficiente para reconstruir digitalmente el objeto tridimensional analizado.

# Apéndice **A**

## Diagramas de flujo del sistema

### Contenido

|                                   |    |
|-----------------------------------|----|
| A.1. Preámbulo . . . . .          | 55 |
| A.2. Diagramas de flujo . . . . . | 56 |

### A.1. Preámbulo

En ese apéndice se proporcionan los diagramas de flujo de los procesos principales que componen el software *3DFringe*.

## A.2. Diagramas de flujo

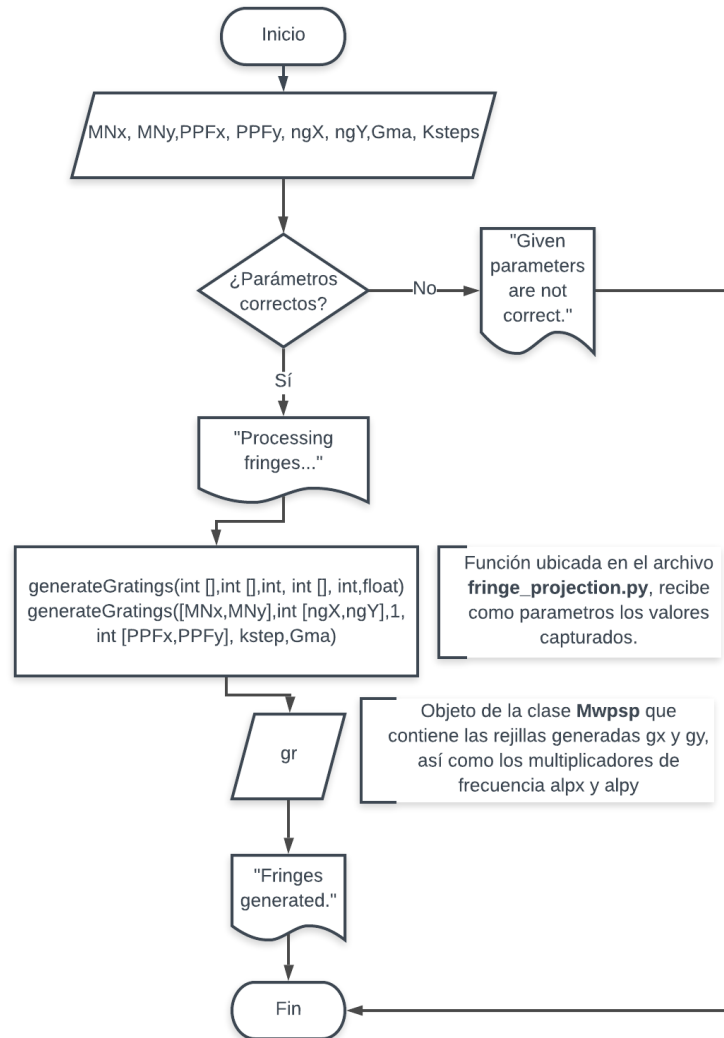


Figura A.1: Diagrama de flujo para el botón *Generate gratings*.

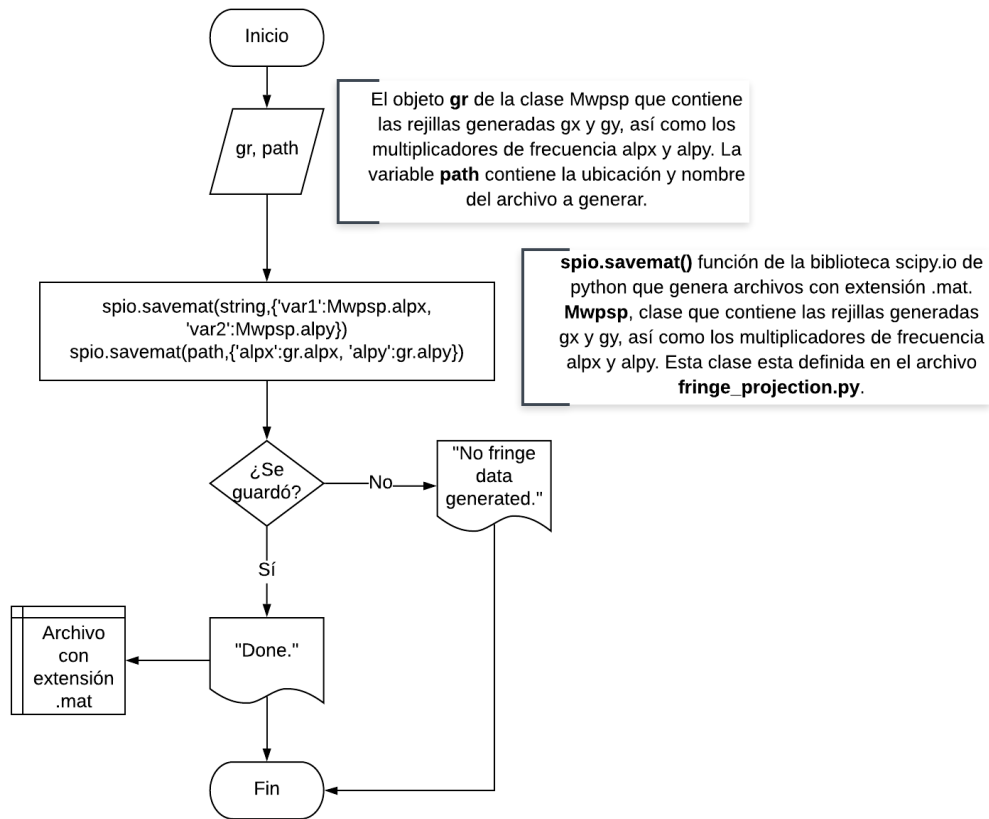
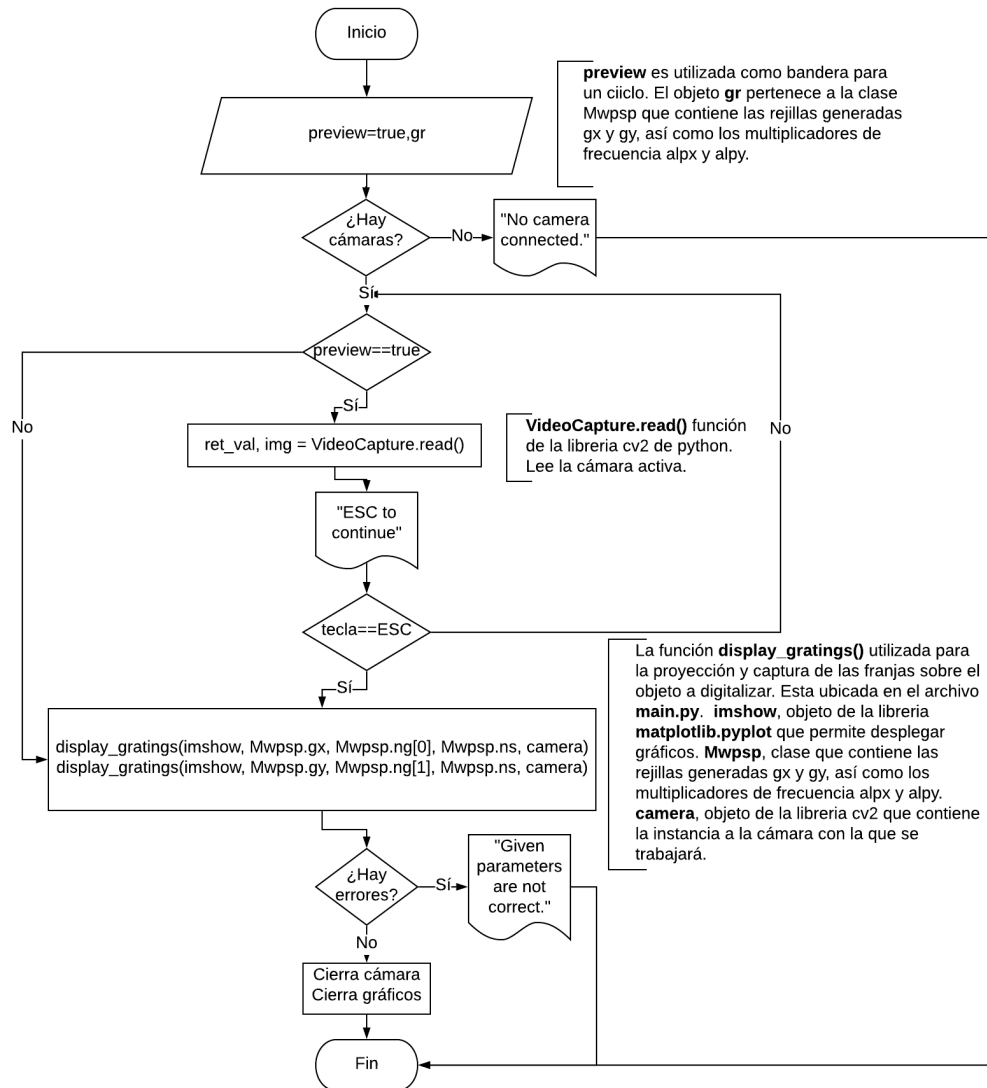


Figura A.2: Diagrama de flujo para el botón *Save freq. multiplier*.

Figura A.3: Diagrama de flujo para el botón *Projection*.

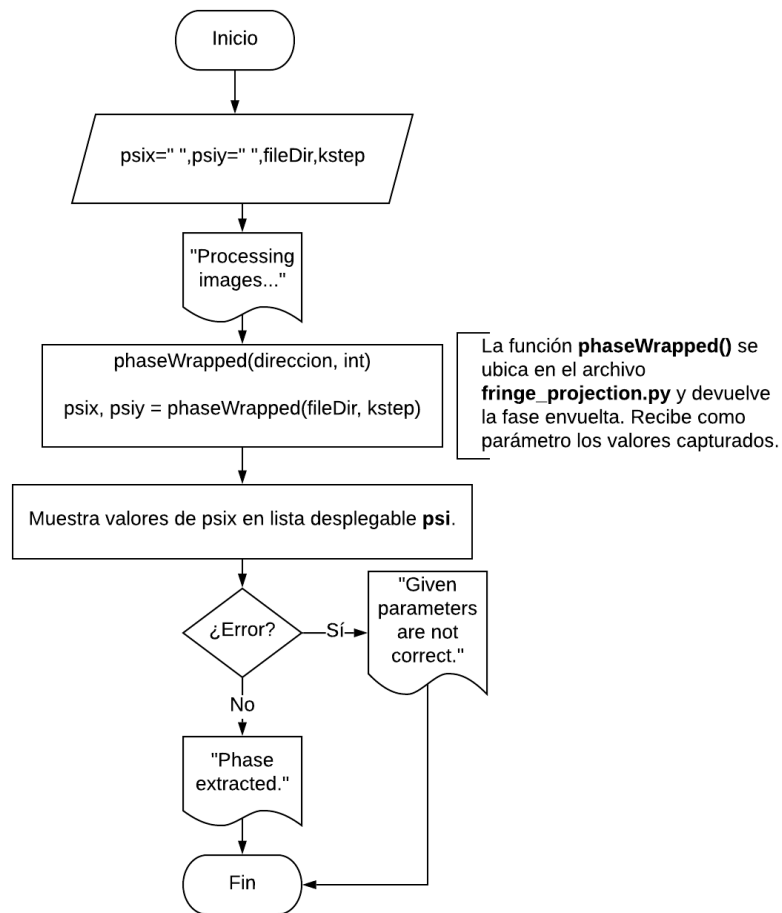
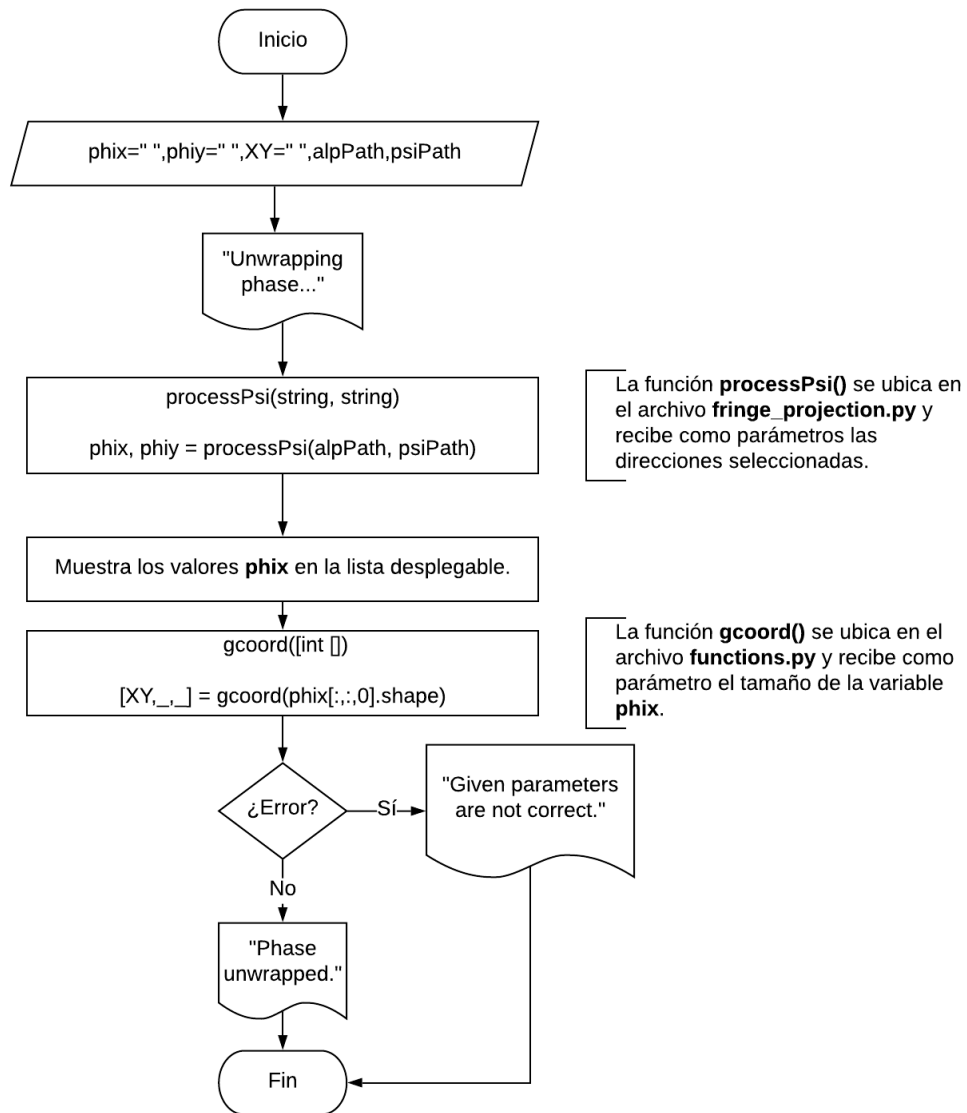
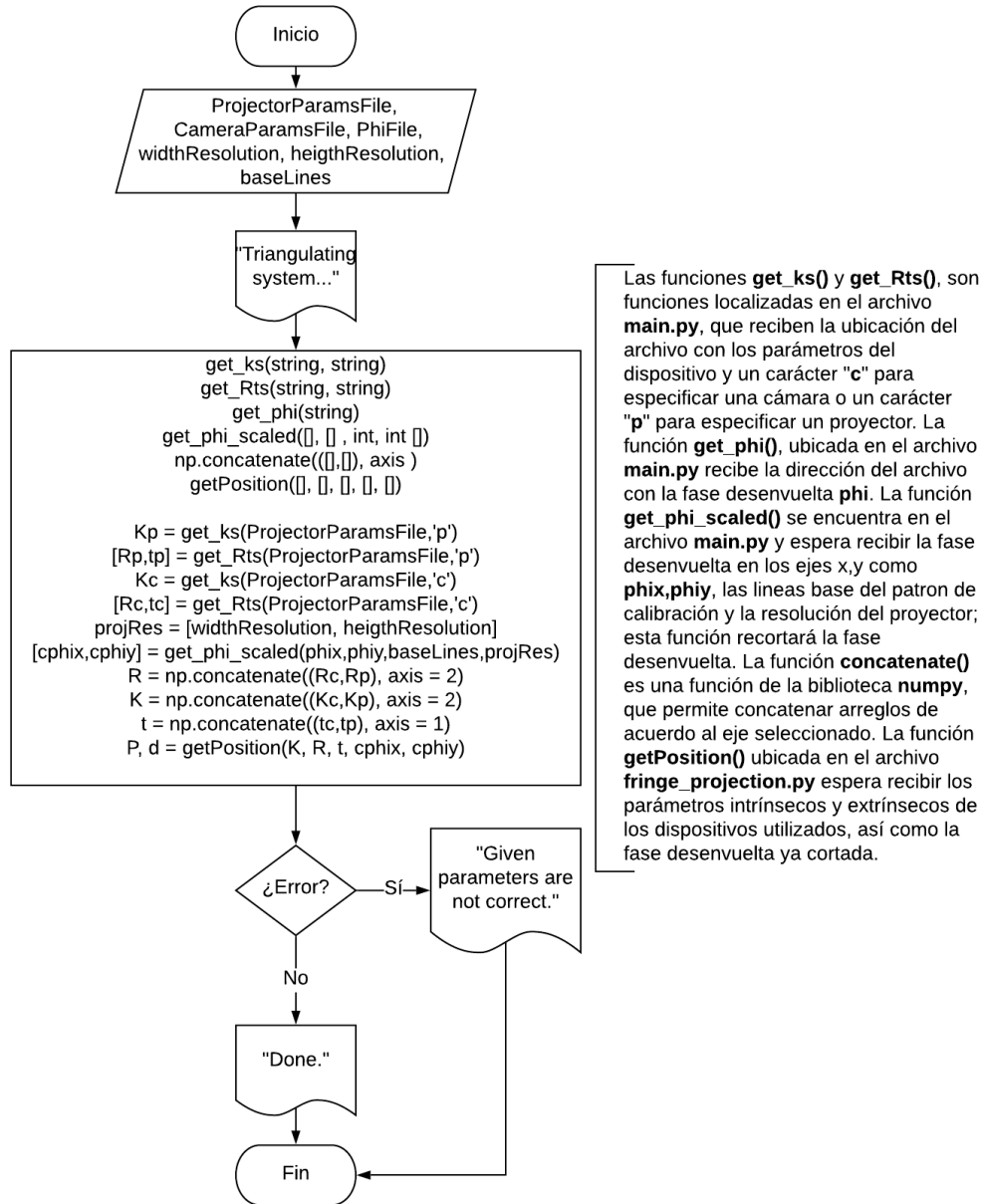


Figura A.4: Diagrama de flujo para el botón *Wrapped phase*.

Figura A.5: Diagrama de flujo para el botón *Process*.

Figura A.6: Diagrama de flujo para el botón *Triangulate*.

# Apéndice **B**

## Funciones y código fuente

### Contenido

|                                          |    |
|------------------------------------------|----|
| B.1. Preámbulo . . . . .                 | 62 |
| B.2. Funciones y código fuente . . . . . | 63 |

### B.1. Preámbulo

En este apéndice se describen las funciones de los procesos principales del software *3DFringe*. El código fuente asociado a las funciones principales es proporcionado.

## B.2. Funciones y código fuente

| Módulo   | Función          | Entrada |                                          | Salida |                                                                                             | Descripción                                                        |
|----------|------------------|---------|------------------------------------------|--------|---------------------------------------------------------------------------------------------|--------------------------------------------------------------------|
| Gratings | generateGratings | mn      | Resolución de las rejillas               | Mwpsp  | Clase que tiene como constructor la generación de rejillas y su mutiplicador de frecuencias | Crea una instancia de la clase Mwpsp con los parámetros de entrada |
|          |                  | ng      | Número de rejillas a generar             |        |                                                                                             |                                                                    |
|          |                  | sel     | Selector de multiplicador de frecuencias |        |                                                                                             |                                                                    |
|          |                  | ppf     | Número de franjas por píxeles            |        |                                                                                             |                                                                    |
|          |                  | kstep   | Número de corrimiento de fase            |        |                                                                                             |                                                                    |
|          |                  | gma     | Corrección de gamma                      |        |                                                                                             |                                                                    |

Figura B.1: Funciones para generar rejillas en la sección Gratings.

| Módulo         | Función      | Entrada  |                                                       | Salida |                             | Descripción                                                                                 |
|----------------|--------------|----------|-------------------------------------------------------|--------|-----------------------------|---------------------------------------------------------------------------------------------|
| Phase shifting | ksteps       | I        | Vector de imágenes                                    | wphi   | Fase envuelta               | Extracción de fase                                                                          |
|                |              | delta    | Frecuencia espacial                                   |        |                             |                                                                                             |
|                | phaseWrapped | filePath | Directorio que contiene todas las imágenes capturadas | psix   | Fases envueltas en el eje x | Carga las imagenes del directorio y se acomodan de acuerdo al número de corrimiento de fase |
|                |              | nstep    | Número de corrimiento de fase                         | psiy   | Fases envueltas en el eje y |                                                                                             |

Figura B.2: Funciones para realizar extracción de fase envuelta en la sección Phase shifting.

| Módulo           | Función    | Entrada |                                                                         | Salida |                                | Descripción                                                                                                             |
|------------------|------------|---------|-------------------------------------------------------------------------|--------|--------------------------------|-------------------------------------------------------------------------------------------------------------------------|
| Phase unwrapping | m wps      | psi     | Fases envueltas de ambos ejes                                           | phi    | Fases desenvueltas             | Corrección de la fase extraída                                                                                          |
|                  |            | alp     | Multiplicadores de frecuencia de ambos ejes                             |        |                                |                                                                                                                         |
|                  | processPsi | alpPath | Directorio que contiene el archivo de los multiplicadores de frecuencia | phix   | Fases desenvueltas en el eje x | Carga los archivos de la fase desenvuelta y los multiplicadores de frecuencia al sistema. Manda llamar la función m wps |
|                  |            | psiPath | Directorio que contiene el archivo de las fases envueltas               | phiy   | Fases desenvueltas en el eje y |                                                                                                                         |

Figura B.3: Funciones para realizar el desenvolvimiento de fase en la sección Phase unwrapping.

| Módulo        | Función     | Entrada |                                                            | Salida |                                                            | Descripción                                                                               |
|---------------|-------------|---------|------------------------------------------------------------|--------|------------------------------------------------------------|-------------------------------------------------------------------------------------------|
| Triangulation | getDs       | Rs      | Matriz de rotación de los dispositivos                     | d      | Direcciones de los puntos observados para cada dispositivo | Calcula las direcciones de los puntos observados para cada dispositivo                    |
|               |             | Ks      | Parámetros intrínsecos de los dispositivos                 |        |                                                            |                                                                                           |
|               |             | PHIx    | Fase desenvuelta en el eje x                               |        |                                                            |                                                                                           |
|               |             | PHIy    | Fase desenvuelta en el eje y                               |        |                                                            |                                                                                           |
|               | triang      | d       | Direcciones de los puntos observados para cada dispositivo | p      | Vector de posiciones                                       | Encuentra los puntos más cercanos de las líneas al punto observado entre los dispositivos |
|               |             | t       | Vector de posición de los dispositivos                     |        |                                                            |                                                                                           |
|               | getPosition | K       | Parámetros intrínsecos de los dispositivos                 | P      | Vector de posiciones                                       | Manda llamar las funciones getDs y triang con dichos parámetros requeridos                |
|               |             | R       | Matriz de rotación de los dispositivos                     | d      | Direcciones de los puntos observados para cada dispositivo |                                                                                           |
|               |             | t       | Vector de posición de los dispositivos                     |        |                                                            |                                                                                           |
|               |             | cphix   | Fase desenvuelta recortada en el eje x                     |        |                                                            |                                                                                           |
|               |             | cphiy   | Fase desenvuelta recortada en el eje y                     |        |                                                            |                                                                                           |

Figura B.4: Funciones que implementan el método de triangulación en la sección Triangulation.

```

def __mwpssp(self, MN, ng, sel, ppf, ns, gamma):
    if(np.size(MN) == 1):
        MN = np.array([MN,MN])

    self.x = np.linspace(-1, 1, MN[1])
    self.y = np.linspace(-1, 1, MN[0])
    X, Y = np.meshgrid(self.x, self.y)
    self.MN = MN
    self.gma = gamma

    if(np.size(ppf) == 1):
        ppfx = ppf
        ppfy = ppf
    else:
        ppfx = ppf[0]
        ppfy = ppf[1]

    self.ppf = [ppfx, ppfy]
    self.ns = ns

    if(np.size(ng) == 1):
        ng = [ng, ng]

    self.ng = ng

    self.fnx = MN[1]/ppfx
    self.fny = MN[0]/ppfy

    self.alpx = self.__freq_mult(self.fnx, ng[0], sel)
    self.alpy = self.__freq_mult(self.fny, ng[1], sel)

    self.gx = self.__aux_fringes(X, self.alpx, ng[0], ns, gamma)
    self.gy = self.__aux_fringes(Y, self.alpy, ng[1], ns, gamma)

def __aux_fringes(self, A, alp, ng, ns, gamma):
    alpk = np.insert(alp,0,1)
    I = np.zeros([A.shape[0], A.shape[1], ns, ng]);
    for kf in range(0, ng):
        for ks in range(0, ns):
            I[:, :, ks, kf] = (0.5 + 0.5*np.cos(np.prod(alpk[0:kf+1])*np.pi*A + 2*np.pi*ks/ns))**(gamma)

```

Figura B.5: Constructor de la clase Mwpssp para la generación de rejillas.

```

def ksteps(I,delta=1):
    MNSF = I.shape
    delta = np.arange(0,MNSF[2])
    delta = (delta) * 2*np.pi/MNSF[2]

    l = np.zeros((MNSF[0],MNSF[1]))
    r = np.zeros((MNSF[0],MNSF[1]))

    for k in range(0, MNSF[2]):
        l = l + (I[:, :, k] * np.sin(delta[k]))
        r = r + (I[:, :, k] * np.cos(delta[k]))

    wphi = -np.arctan2(l,r)

    del(l)
    del(r)

    return wphi

```

Figura B.6: Función `ksteps` encargada de realizar la extracción de fase en-vuelta.

```

def mwps(psi,alp,w=1):
    k = np.zeros(psi.shape)#Fringe order

    phi = k #Phase

    phi[:, :, 0] = psi[:, :, 0]
    n = 1
    while(n<psi.shape[2]):
        k[:, :, n] = np.around( (alp[0,n-1]*phi[:, :, n-1] - psi[:, :, n])/(2*np.pi))
        phi[:, :, n] = psi[:, :, n] + 2*np.pi*k[:, :, n]
        n = n + 1

    n = 0
    alpk = np.insert(alp[0], 0, 1)
    while(n<psi.shape[2]):
        phi[:, :, n] = phi[:, :, n]/(w*np.pi*(np.prod(alpk[0:n+1])))

        n = n + 1

    return [phi,k]

```

Figura B.7: Función `mwps` encargada de realizar el desenvolvimiento de fase.

```

def triang(d,t):# d=3xMxK M = number of pixels, K = number of devices | t=3xK
    MK = np.array([d.shape[1],d.shape[2]])
    DD = np.zeros([3,3,MK[1]])
    Dt = np.zeros([3,MK[1],1])
    p = np.zeros((3,MK[0],1))
    i = 0
    while(i<MK[0]):#repeat i number of pixels
        j = 0
        while(j<MK[1]): #repeat j number of devices
            dk = d[:,i,j].reshape(3,1)
            DD[:,j,j] = np.eye(3) - ( (dk@dk.T)/(dk.T@dk) )
            Dt[:,j] = (DD[:,j,j]@t[:,j].reshape(3,1))
            j = j + 1
        p[:,i] = linalg.solve(np.sum(DD, axis = 2),np.sum(Dt, axis = 1))
        i = i + 1

    del MK, DD, Dt
    gc.collect()

    return p # returns p matrix of size 3xM

```

Figura B.8: Función `triang` para calcular coordenadas  $(x, y, z)$  usando el método de triangulación.

# Bibliografía

- [1] J. Geng, “Structured-light 3d surface imaging: a tutorial,” *Adv. Opt. Photon.*, vol. 3, no. 2, pp. 128–160, 2011.
- [2] S. Zhang, “High-speed 3d shape measurement with structured light methods: A review,” *Optics and Lasers in Engineering*, vol. 106, pp. 119 – 131, 2018.
- [3] S. S. Gorthi and P. Rastogi, “Fringe projection techniques: Whither we are?,” *Optics and Lasers in Engineering*, vol. 48, no. 2, pp. 133 – 140, 2010.
- [4] H. Yue, Y. Yu, W. Chen, and X. Wu, “Accurate three dimensional body scanning system based on structured light,” *Opt. Express*, vol. 26, no. 22, pp. 28544–28559, 2018.
- [5] J. R. and et al., “Key concepts for phase-to-coordinate conversion in fringe projection systems,” *Appl. Opt.*, vol. 58, no. 18, pp. 4828–4834, 2019.
- [6] J. R. and D. V., “Flexible camera-projector calibration using superposed color checkerboards,” *Optics and Lasers in Engineering*, vol. 120, pp. 59 – 65, 2019.