

INSTITUTO TECNOLÓGICO DE CHIHUAHUA
DIVISIÓN DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN

***“ALGORITMO PARA RECONSTRUCCIÓN DE
OBJETOS TRIDIMENSIONALES POR
ESTIMACIÓN DE PARÁMETROS DE FRANJAS
SINUSOIDALES”***

TESIS

QUE PARA OBTENER EL GRADO DE

***DOCTOR EN CIENCIAS
EN INGENIERÍA ELECTRÓNICA***

PRESENTA:

EDGAR ZENDEJAS HERNÁNDEZ

**DIRECTOR DE LA TESIS:
*DR. GERARDO TRUJILLO SCHIAFFINO***

**CO-DIRECTOR DE LA TESIS:
*DR. MARCELINO ANGUIANO MORALES***

CHIHUAHUA, CHIH., JUNIO 2024



EDUCACIÓN
SECRETARÍA DE EDUCACIÓN PÚBLICA



**TECNOLÓGICO
NACIONAL DE MÉXICO**



Chihuahua, Chih., 10 de junio de 2024

**EDGAR ZENDEJAS HERNÁNDEZ
PRESENTE**

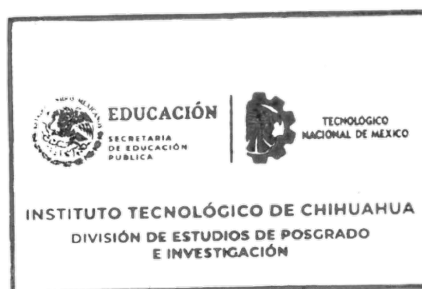
En cumplimiento con los requerimientos para la obtención del grado de Doctor en Ciencias en Ingeniería Electrónica y a propuesta de su Comité Tutorial, la División de Estudios de Posgrado e Investigación le concede la autorización para imprimir la tesis titulada **"Algoritmo para reconstrucción de objetos tridimensionales por estimación de parámetros de franjas sinusoidales"**, dirigida por el Dr. Gerardo Trujillo Schiaffino con el siguiente contenido de capítulos:

- I Antecedentes
- II Marco teórico
- III Técnica propuesta
- IV Software de simulación y captura
- V Montaje experimental
- VI Análisis de datos
- VII Resultados finales
- VIII Conclusiones y trabajo futuro

ATENTAMENTE
Excelencia en Educación Tecnológica®
"La técnica por el Engrandecimiento de México"



**LEONCIO OCHOA CERVANTES
JEFE DE LA DIVISIÓN DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN**



Av. Tecnológico #2909 Col. 10 de Mayo C.P. 31310, Chihuahua, Chihuahua.

Tel. (614) 201 2000 ext. 2157, e-mail: dir_chihuahua@tecnm.mx tecnm.mx | <https://itchihuahua.mx/>





Chihuahua, Chih., 10 de junio de 2024

**LEONCIO OCHOA CERVANTES
JEFE DE LA DIVISIÓN DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN
PRESENTE**

En cumplimiento con los requerimientos para la obtención del grado de Doctor en Ciencias en Ingeniería Electrónica, le notificamos que el documento de tesis del alumno **EDGAR ZENDEJAS HERNÁNDEZ** titulado **"Algoritmo para reconstrucción de objetos tridimensionales por estimación de parámetros de franjas sinusoidales"**, dirigido por el Dr. Gerardo Trujillo Schiaffino, ha sido aprobado y aceptado para su impresión.

Por lo anterior, proponemos le sea concedida la autorización de impresión correspondiente.

Agradeciendo la atención a la presente, quedamos de usted:

ATENTAMENTE

*Excelencia en Educación Tecnológica®
"La técnica por el Engrandecimiento de México"*


DR. GERARDO TRUJILLO SCHIAFFINO
DIRECTOR DE TESIS


DR. MARCELINO ANGUIANO MORALES
MIEMBRO DEL COMITÉ TUTORIAL


DR. LUIS FRANCISCO CORRAL MARTÍNEZ
MIEMBRO DEL COMITÉ TUTORIAL


DRA. PALOMA GUADALUPE MENDOZA VILLEGAS
MIEMBRO DEL COMITÉ TUTORIAL


DRA. DIDIA PATRICIA SALAS PEIMBERT
MIEMBRO DEL COMITÉ TUTORIAL





CARTA CESIÓN DE DERECHOS

En la ciudad de Chihuahua el día 11 de junio de 2024, la que suscribe, EDGAR ZENDEJAS HERNÁNDEZ con número de control D19060654, del Doctorado en Ciencias en Ingeniería Electrónica adscrito a la División de Estudios de Posgrado e Investigación del Instituto Tecnológico de Chihuahua, manifiesta que es autor intelectual del presente trabajo de Tesis bajo la dirección de el Dr. Gerardo Trujillo Schiaffino y cede los derechos del trabajo titulado **"Algoritmo para reconstrucción de objetos tridimensionales por estimación de parámetros de franjas sinusoidales"**, al Tecnológico Nacional de México y/o Instituto Tecnológico de Chihuahua para su difusión, divulgación, transmisión, reproducción, así como su digitalización con fines académicos y de investigación.

EDGAR ZENDEJAS HERNÁNDEZ





DECLARACIÓN DE ORIGINALIDAD

En la ciudad de Chihuahua el día 11 de junio de 2024, la que suscribe, EDGAR ZENDEJAS HERNÁNDEZ con número de control D19060654, del Doctorado en Ciencias en Ingeniería Electrónica adscrito a la División de Estudios de Posgrado e Investigación del Instituto Tecnológico de Chihuahua, manifiesta ser autor intelectual del presente trabajo de Tesis titulado **"Algoritmo para reconstrucción de objetos tridimensionales por estimación de parámetros de franjas sinusoidales"**, bajo la dirección de el Dr. Gerardo Trujillo Schiaffino; y manifiesta que el contenido es original y que las fuentes de información consultadas para su fundamentación están debidamente citadas y referenciadas.

EDGAR ZENDEJAS HERNÁNDEZ



Chihuahua, Chih., 13 de junio del 2024

Dra. María Elena Álvarez-Buylla Roces
Directora de CONAHCYT

At'n. Edwin Ricardo Triujeque Woods
Director de Becas y Posgrado

P r e s e n t e.

Por este conducto aprovecho la ocasión para saludarla e informarle que a la fecha he obtenido el Grado de Doctorado en Ciencias de Ingeniería Electrónica en la División de Estudios de Posgrado e Investigación del Instituto Tecnológico de Chihuahua. Motivo por el cual agradezco todo el apoyo brindado por esta Institución que Usted representa, el otorgamiento de esta beca permitió dedicarme de tiempo completo a la realización de mis estudios de Posgrado y de esta manera lograr el cumplimiento del objetivo principal del convenio establecido.

Sin otro particular por el momento, me es grato quedar de Usted como su seguro servidor, no sin antes reiterar mi agradecimiento. Muchas Gracias.

A t e n t a m e n t e



Dr. Edgar Zendejas Hernández
Exbecario CONAHCYT No. 615621

c.c.p. M.C. Leoncio Ochoa Cervantes
Jefe de la División de Estudios de Posgrado e Investigación

AGRADECIMIENTOS

En primer lugar, agradezco a mi familia, por su amor incondicional y por darme el apoyo a lo largo de este apasionante viaje. Gracias por creer en mí y por ser una fuente constante de inspiración y motivación. Sus palabras de aliento y su confianza en mis capacidades me han impulsado a seguir adelante.

Agradezco a mi director de tesis el Dr. Gerardo Trujillo Schiaffino por su tiempo, amistad y los conocimientos que me transmitió. Su paciencia y compromiso con mi educación han sido fundamentales en mi desarrollo académico y profesional. Gracias doctor.

A la Dra. Didia Patricia Salas Peimbert, Dra. Paloma Guadalupe Mendoza Villegas, Dr. Luis Francisco Corral Martínez y al Dr. Marcelino Anguiano Morales por el apoyo, la confianza y dedicación que siempre me brindaron a lo largo de este periodo. Gracias por contribuir en mi formación profesional y por ayudar a mejorar este trabajo.

A la M.C. Alma Delia Corral Sáenz por su disponibilidad y apoyo mostrado, que fueron de mucha ayuda.

Gracias a mis amigos del Posgrado: Nadia, Moisés, Misael, Erick, Mario, Carlos, Santiago, Alex, Abimael, Macías, Daniel M., Daniel V. y Kikín por los momentos agradables que compartimos. Sus ideas, discusiones y la colaboración mutua han sido una parte esencial de este proceso. Aprecio profundamente cada momento compartido y cada desafío superado juntos.

A mi amigo Aarón, por el ánimo, la ayuda y por saber que siempre puedo confiar en tí.

A todos ustedes, mi más sincero agradecimiento.

RESUMEN

ALGORITMO PARA RECONSTRUCCIÓN DE OBJETOS TRIDIMENSIONALES POR ESTIMACIÓN DE PARÁMETROS DE FRANJAS SINUSOIDALES

M. C. Edgar Zendejas Hernández

Doctorado en Ciencias en Ingeniería Electrónica

División de Estudios de Posgrado e Investigación del

Instituto Tecnológico de Chihuahua

Chihuahua, Chih. 2024

Director de Tesis: Dr. Gerardo Trujillo Schiaffino

Co-Director de Tesis: Dr. Marcelino Anguiano Morales

El presente trabajo se ubica en el área de la perfilometría por proyección de franjas (PPF), ampliamente utilizada para mediciones tridimensionales. Existen diversas técnicas o algoritmos, clasificables en algoritmos espaciales y temporales, adecuados para objetos dinámicos y estáticos respectivamente. Estas técnicas emplean patrones sinusoidales que se modulan según la topografía del objeto, siendo esencial relacionar esta modulación con la forma del objeto. Debido a su exactitud y rapidez, la PPF ha sido ampliamente adoptada en diversas áreas como la medicina, arqueología, ingeniería y manufactura.

En este trabajo se propone un enfoque diferente para analizar el patrón de franjas modulado. La técnica implica la proyección de un patrón de franjas sinusoidales, seguido por la detección de los máximos y mínimos del perfil sinusoidal para obtener un mapa de fase envuelta. Luego, este mapa se desenvuelve de nuevo mediante la detección de máximos y mínimos, permitiendo así una reconstrucción tridimensional del objeto. Este enfoque evita el uso de técnicas complejas como la transformada de Fourier, la transformada wavelet y la transformada de Riesz.

Se presenta el algoritmo propuesto, su desempeño con imágenes simuladas ideales y con adición de ruido. Adicionalmente, esta técnica fue probada de manera experimental y validada estadísticamente para garantizar que el proceso de medición no introduzca errores significativos. Finalmente, se presentan los resultados obtenidos, logrando un desempeño comparable al de otras técnicas PPF clásicas.

ABSTRACT

The present work is in the area of fringe projection profilometry (FPP), widely used for three-dimensional measurements. There are several techniques or algorithms, classifiable into spatial and temporal algorithms, suitable for dynamic and static objects respectively. These techniques use sinusoidal patterns that are modulated according to the topography of the object, and it is essential to relate this modulation to the shape of the object. Due to its accuracy and speed, FPP has been widely adopted in various areas such as medicine, archaeology, engineering and manufacturing.

In this paper, a different approach to analyze the modulated fringe pattern is proposed. The technique involves the projection of a sinusoidal fringe pattern, followed by the detection of the maxima and minima of the sinusoidal profile to obtain a wrapped phase map. This map is then unwrapped by detecting maxima and minima again, thus allowing a three-dimensional reconstruction of the object. This approach avoids the use of complex techniques such as Fourier transform, wavelet transform and Riesz transform.

The performance of the proposed algorithm, evaluated with ideal simulated images and with the addition of noise, is presented. Additionally, this technique was experimentally tested and statistically validated to ensure that the measurement process does not introduce significant errors. Finally, the results obtained are presented, achieving a performance comparable to that of other classical FPP techniques.

CONTENIDO

RESUMEN	vi
ABSTRACT	viii
LISTA DE FIGURAS	xiii
LISTA DE TABLAS	xv
INTRODUCCIÓN	xvi
CAPÍTULO I ANTECEDENTES	1
1.1 Principios básicos	1
1.2 Análisis de franjas	3
1.3 Proyección de franjas	4
1.4 Adquisición de franjas	5
CAPÍTULO II MARCO TEÓRICO	8
2.1 Técnicas de perfilometría por proyección de franjas (PPF)	8
2.1.1 Técnicas espaciales	8
2.1.1.1 Perfilometría por transformada de Fourier (FTP)	8
2.1.1.2 Transformada de ventana de Fourier	11
2.1.1.3 Perfilometría por transformada de wavelet (WTP)	15
2.1.1.4 Corrimiento de fase digital de cuatro pasos (D4-PS)	16
2.1.2 Técnicas temporales	18
2.1.2.1 PSA de mínimos cuadrados de N pasos (LS-PSA)	18
2.1.2.2 PSA de tres pasos	19
2.1.2.3 PSA doble de tres pasos	20
2.1.2.4 Carré PSA	21

2.1.2.5 PSA de 5 pasos de Hariharan	22
2.1.2.6 PSA 2 + 1 modificado	23
2.1.2.7 Perfilometría de transformada de Fourier con desplazamiento π	24
2.2 Ruido	25
2.2.1 Fuentes de ruido	25
2.2.1.1 Ruido del sensor	25
2.2.1.2 Ruido en el proceso de adquisición de imágenes	27
2.2.1.3 Ruido de procesamiento de señales	27
2.2.2 Modelado de ruido	27
2.2.2.1 Ruido impulsivo	28
2.2.2.2 Ruido Gaussiano	28
2.2.2.3 Ruido de Poisson	29
2.3 Corrección gamma	30
CAPÍTULO III TÉCNICA PROPUESTA	32
3.1 Descripción de la técnica	32
3.2 Descripción de la metodología	34
CAPÍTULO IV SOFTWARE DE SIMULACIÓN Y CAPTURA	36
4.1 Generación, proyección y captura de patrones	37
4.1.1 Corrección de factor gamma	38
4.1.2 Preprocesamiento para píxeles dañados	42
4.2 Filtro pasa bajas	44
4.3 Detección de máximos y mínimos	46
4.4 Interpolación de máximos y mínimos	47
4.5 Estimación de fase envuelta	48

4.6 Algoritmo de desenvolvimiento de fase	49
4.6.1 Detección de máximos y mínimos	49
4.6.2 Estimación de fase desenvuelta	50
CAPÍTULO V MONTAJE EXPERIMENTAL	55
5.1 Descripción general	55
5.2 Características técnicas de los componentes	56
5.3 Alineación de componentes	58
CAPÍTULO VI ANÁLISIS DE DATOS	61
6.1 Validación del proceso de medición	61
6.1.1 Prueba de bondad de ajuste chi-cuadrada X^2	62
6.1.2 Histograma	64
6.1.3 Gráfico de probabilidad normal	64
6.2 Caracterización	66
6.2.1 Exactitud	67
6.2.2 Precisión	67
6.2.3 Resultados obtenidos	67
CAPÍTULO VII RESULTADOS FINALES	68
7.1 Resultados de simulación	68
7.2 Resultados experimentales	73
CAPÍTULO VIII CONCLUSIONES Y TRABAJO FUTURO	77
8.1 Conclusiones generales	77
8.2 Conclusiones sobre el algoritmo de estimación de parámetros	77
8.3 Conclusiones de resultados experimentales	78
8.3.1 Corrección del factor gamma	78

8.3.2 Corrección de píxeles dañados	78
8.3.3 Generación de sombras y oclusiones	78
8.4 Trabajo futuro	79
REFERENCIAS	80
ANEXOS	89
ANEXO 1. ESPECIFICACIONES DEL PROYECTOR ACER K335	90
ANEXO 2. ESPECIFICACIONES DE LA CÁMARA POINT GREY CMLN-13S2C-CS	92
APÉNDICES	94
APÉNDICE 1. Código fuente “Estimación de parámetros”	95
APÉNDICE 2. SUBVI “Captura de Cruz”	112
APÉNDICE 3. SUBVI “Proyección de Cruz”	114
3.1 SUBVI “Generación de Cruz”	116
APÉNDICE 4. “Análisis de datos”	117

LISTA DE FIGURAS

Figura 1.1. Diagrama de flujo de PPF.	2
Figura 1.2. Formación de imágenes en PPF.	4
Figura 1.3. Imagen de franjas moduladas por una superficie.	7
Figura 2.1. Categorización de técnicas espaciales y temporales.	9
Figura 2.2. Diagrama de flujo de WFF.	13
Figura 2.3. Patrón sinusoidal simulado con efecto gamma = 2.2.	31
Figura 3.1. Gráfica de variación de intensidad de un corte transversal en la dirección de las franjas.	34
Figura 4.1. Diagrama de flujo de algoritmo propuesto.	36
Figura 4.2. Corte transversal de un patrón de franjas.	37
Figura 4.3. Patrón de franjas sinusoidales con AWGN.	38
Figura 4.4. Vista de sección transversal de patrón de franjas sin corrección gamma.	39
Figura 4.5. Vista de sección transversal de patrones con diferentes valores de factor gamma.	40
Figura 4.6. Vista de sección transversal de capturas con diferentes valores de gamma.	42
Figura 4.7. Píxeles dañados en la cámara.	43
Figura 4.8. Histograma de captura con la lente ocluida.	43
Figura 4.9. Vista de sección transversal de patrón de franjas antes de la aplicación del filtro pasa bajas.	44
Figura 4.10. Vista de sección del patrón de franjas filtrado y normalizado.	45
Figura 4.11. Detección de máximos y mínimos.	46
Figura 4.12. Interpolación de máximos, mínimos y cálculo de promedio.	47
Figura 4.13. Fase envuelta.	48

Figura 4.14. Detección de máximos y mínimos en la señal de fase envuelta.	49
Figura 4.15. Vista de sección transversal de fase envuelta y desenvuelta.	51
Figura 4.16. Vista de sección transversal de fase envuelta y desenvuelta del objeto.	52
Figura 4.17. Reconstrucción 3D del objeto simulado.	52
Figura 4.18. Vista de sección transversal de fase envuelta y desenvuelta con la adición de ruido.	53
Figura 4.19. Reconstrucción 3D del objeto simulado con ruido.	54
Figura 5.1. Vista lateral del montaje experimental.	55
Figura 5.2. Proyector ACER K335.	56
Figura 5.3. Cámara POINT GREY CMLN-13S2C-CS.	57
Figura 5.4. Lente doble gauss 54857.	58
Figura 5.5. Ubicación del vástago: a) Vista isométrica; b) Vista superior.	59
Figura 5.6. Interfaz Captura de Cruz, se alinea la línea verde con el vástago.	59
Figura 5.7. Interfaz Captura de Cruz, se hace coincidir la cruz proyectada con la generada.	60
Figura 6.1. Resultado de la prueba de bondad de ajuste chi-cuadrada χ^2 .	64
Figura 6.2. Histograma de mediciones.	65
Figura 6.3. Gráfica de probabilidad normal.	66
Figura 7.1. Patrones de franjas simulados: a) Patrón de franjas modulado por la función objeto, b) Patrón de franjas de referencia.	69
Figura 7.2. Resultados de la simulación: a) Mapa de fase desenvuelta; b) Reconstrucción 3D del objeto.	70
Figura 7.3. Patrones de franjas capturados: a) Patrón de franjas modulado por la función objeto, b) Patrón de franjas de referencia.	74
Figura 7.4. Resultados experimentales: a) Mapa de fase sin envolver, b) Reconstrucción 3D del objeto.	75

LISTA DE TABLAS

Tabla 4.1. Análisis de coeficientes de correlación para diferentes valores de gamma.	41
Tabla 6.1. Mediciones obtenidas del patrón de referencia.	63
Tabla 7.1. Resultados de error con entrada simulada (Radianes).	72
Tabla 7.2. Resultados de error con entrada capturada (Radianes).	76

INTRODUCCIÓN

El contenido de esta tesis se encuentra estructurado de la siguiente manera:

En el Capítulo I se realiza una introducción sobre la perfilometría por proyección (PPF) de franjas y el proceso de formación de imágenes.

En el Capítulo II se categorizan técnicas de proyección de franjas en métodos espaciales o temporales, explicando de manera general algoritmos realizados por otros investigadores. Además, se incluye una descripción general de las fuentes y tipos de ruido, así como la afectación de la distorsión gamma en los patrones proyectados.

En el Capítulo III contiene una descripción del proyecto de tesis y la metodología empleada para el desarrollo de este trabajo.

En El Capítulo IV se describe la implementación del algoritmo computacional mediante el software desarrollado para las etapas de proyección, adquisición, preprocesamiento, estimación de parámetros y desenvolvimiento de fase, detallando su configuración y funcionamiento.

El Capítulo V muestra el montaje experimental para la obtención de datos experimentales y el proceso de alineación de componentes.

En el Capítulo VI se presenta la validación estadística de los datos experimentales obtenidos para asegurar que no existe un sesgo introducido por el proceso de medición utilizado.

En el Capítulo VII contiene las pruebas finales para evaluar el desempeño de la técnica propuesta respecto a la determinación de los valores de error.

En el Capítulo VIII se presentan las conclusiones más relevantes del trabajo realizado y propuestas para trabajos futuros.

CAPÍTULO I

ANTECEDENTES

1.1 Principios básicos

Las técnicas tridimensionales para recuperar la forma de un objeto se pueden clasificar como métodos de contacto y no contacto. Los métodos de contacto, como su nombre indica, requieren contacto físico entre el objeto y el palpador para obtener una reconstrucción 3D, y que, dependiendo de la muestra y el tamaño del palpador pueden ser adecuados o no. Malacara et al. [1] mencionan un ejemplo donde el método de contacto no es apropiado, sobre todo al tratarse de objetos elásticos tales como un globo o un diafragma donde el contacto físico del palpador puede dañar o deformar la superficie y crear mediciones erróneas. El dispositivo más común basado en este método, es la máquina de medición de coordenadas (CMM), donde un palpador está normalmente montado en un brazo robótico y se coloca sobre el objeto de prueba, el análisis de la posición del brazo, se realiza a través de cinemática inversa y de esta manera se obtiene la altura. Algunas de las ventajas de este método es la alta precisión y resolución, mientras que las desventajas son el tiempo y la precisión en relación con el tamaño de la aguja [1-2].

Por otro lado, los métodos de no contacto eliminan la necesidad del contacto físico, y son capaces de trabajar bajo diferentes principios, permitiendo así una amplia variedad de alternativas. Se puede elegir el procedimiento más adecuado en función de las limitaciones de cada aplicación, sin embargo, a menudo se prefieren los métodos basados en el principio óptico. Algunas de estas técnicas han sido ampliamente investigadas en metrología óptica como: interferometría [3-4], visión estereoscópica [4-5], tiempo de vuelo y luz estructurada [5-13].

Las técnicas de luz estructurada, donde un patrón sinusoidal es utilizado para recuperar la forma del objeto; son generalmente conocidas como perfilometría por proyección de franjas (PPF). El principio básico de PPF es proyectar un patrón sobre

una superficie, donde el patrón proyectado se modula, captura y luego a través de análisis de franjas, la modulación se relaciona con la altura del objeto. El proceso para obtener una reconstrucción 3D, se describe en los pasos que se muestran en la Figura 1.1.

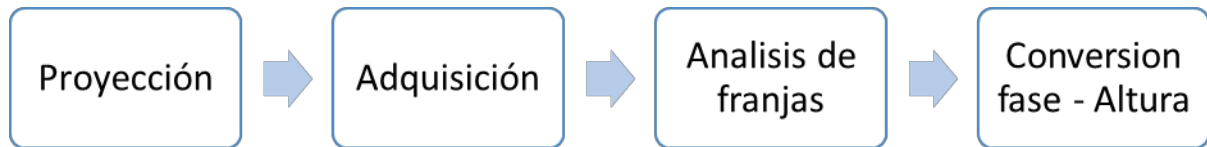


Figura 1.1. Diagrama de flujo de PPF.

El propósito principal de los dos primeros bloques es asegurar que los patrones de franjas sinusoidales proyectados correspondan a una variación de intensidad sinusoidal, donde el proyector agrega un error conocido como factor gamma [12-14], [15-16]. Cabe destacar que no solo se utilizan patrones sinusoidales para PPF, diferentes autores han propuesto una gran variedad de patrones diferentes [17-27].

El análisis de franjas se puede dividir en dos secciones: a) Detección de fase y b) Desenvolvimiento de fase [28].

El objetivo de la detección de fase es analizar las franjas moduladas, el resultado es llamado mapa de fase envuelta, debido al hecho de que los valores obtenidos están dentro de un rango de $(-\pi, \pi)$, de acuerdo a la función tangente inversa. Los algoritmos de análisis franjas son ampliamente investigados y pueden ser categorizados en algoritmos espaciales y temporales. La principal diferencia entre los algoritmos espaciales y temporales es la cantidad de imágenes necesarias para realizar el análisis. Los algoritmos espaciales son adecuados para objetos dinámicos, sólo requieren una captura; mientras que los algoritmos temporales son ideales para objetos estáticos, debido a que el patrón necesita ser desplazado lo que implica capturas múltiples. Un gran número de algoritmos, ya sea espacial o temporal, se han propuesto y analizado por diferentes investigadores [29-33], de modo similar a los métodos de contacto y de no contacto, estos algoritmos presentan ventajas e

inconvenientes, las cuales deben de ser ponderadas de acuerdo a la aplicación particular.

El desenvolvimiento de fase elimina las discontinuidades generadas por la función tangente inversa, obteniendo el mapa de fase desenvuelta. Este proceso no es una tarea trivial y puede complicarse, dependiendo de la calidad del mapa de fase, dicha calidad se puede ver afectada por el ruido, uno de los enfoques para hacer frente a la adición de ruido es filtrar el mapa de fase envuelta [34]. Existen varios algoritmos de desenvolvimiento de fase y se pueden clasificar como: mínimos cuadrados, *branch cut* y *network flow* [35-37].

El último paso es la conversión de fase a altura, donde el objetivo es convertir el mapa de fase desenvuelta a unidades del mundo real. Un sistema PPF típico consta de una unidad de proyector, una unidad de adquisición y una unidad de procesamiento. Para montar un sistema PPF, se han propuesto diferentes configuraciones en la literatura [38]. Takeda *et al.* [39] proponen dos configuraciones: ejes ópticos cruzados y ejes ópticos paralelos. La principal limitación con estas configuraciones es que la unidad de proyección y adquisición deben ser coplanares y paralelos al plano de referencia, sin embargo, estas limitaciones son difíciles de cumplir en condiciones reales. Otras geometrías han sido propuestas, Wen *et al.* [40] proponen una configuración donde la unidad de proyección y captura pueden ser libres.

1.2 Análisis de franjas

Antes de comenzar con el análisis de franjas, es necesario comprender la formación de la imagen, diferentes autores han aportado conocimiento a esta área [8], [41-45]. Para un sistema PPF, los patrones proyectados y capturados se forman siguiendo los pasos que se muestran a continuación, conforme al proceso ilustrado en la Figura 1.2.

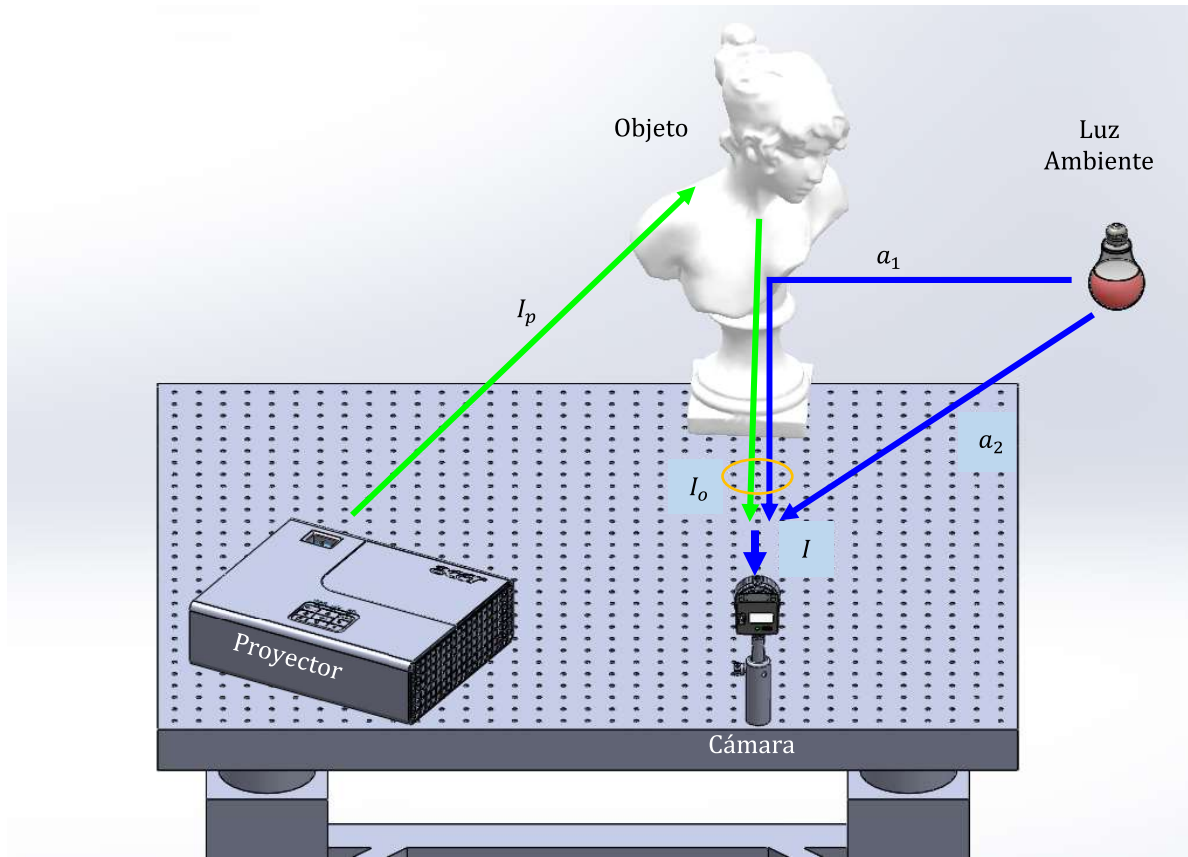


Figura 1.2. Formación de imágenes en PPF.

1.3 Proyección de franjas

Los patrones se generan digitalmente y se proyectan a través de un proyector. Por lo general, las franjas sinusoidales se modelan mediante:

$$I_p(x_p, y_p) = a_p + b_p \cos(2\pi f_0 x_p) \quad (1.1)$$

Donde el subíndice p se utiliza para denotar los términos de proyección, $I(x, y)$ es el valor de intensidad en una cierta ubicación, a es el valor de offset, b es el término modulación de amplitud, f_0 es la frecuencia de franja (período / pixel).

En la práctica, la proyección de un patrón ideal no es posible, ya que el factor gamma introducido por el proyector, provoca distorsiones lineales y no lineales que

distorsionan el patrón ideal de franjas generado. Para disminuir el impacto del factor gamma se genera el patrón de franjas utilizando la siguiente ecuación:

$$I_p(x_p, y_p) = 2c_p \left[\frac{1}{2} + \frac{1}{2} \cos(2\pi f_0 x_p) \right]^{1/\gamma} \quad (1.2)$$

Donde c es el resultado de la igualdad $a = b$, γ denota término de corrección gamma con valores que van desde $2.2 \leq \gamma \leq 2.6$.

1.4 Adquisición de franjas

Los patrones proyectados están modulados de acuerdo a la topografía del objeto $\phi(x, y)$, obteniendo como resultado un patrón modulado. La imagen se forma en la cámara por la luz que entra, donde un porcentaje de la intensidad es reflejado. El primer contribuyente es la luz reflejada por el objeto I_o , la cual se compone por el patrón de franjas proyectado I_p y la luz ambiente $a_1(x, y)$, la cantidad de luz reflejada será proporcional a la reflectividad del objeto $r(x, y)$ y está expresado por:

$$I_o(x, y) = r(x, y)[a_0(x, y) + b_0(x, y) \cos[\phi(x, y)] + a_1(x, y)] \quad (1.3)$$

El segundo contribuyente es la cantidad de luz ambiental que entra directamente en la cámara $a_2(x, y)$ y α la sensibilidad de la cámara. La intensidad capturada $I(x, y)$ se expresa como:

$$I(x, y) = \alpha\{r(x, y)[a_0(x, y) + b_0(x, y) \cos[\phi(x, y)] + a_1(x, y)] + a_2(x, y)\} \quad (1.4)$$

En general, la ecuación 1.4 se puede reescribir por simplicidad como:

$$I(x, y) = a(x, y) + b(x, y) \cos [\phi(x, y)] \quad (1.5)$$

Donde los términos utilizados son: la intensidad media $a(x, y)$ y la modulación de amplitud $b(x, y)$. Estos términos son definidos por las ecuaciones (1.6) y (1.7), respectivamente

$$a(x, y) = \alpha r(x, y)[a_1(x, y) + a_0(x, y)] + \alpha a_2(x, y) \quad (1.6)$$

$$b(x, y) = \alpha r(x, y)b_0(x, y) \quad (1.7)$$

De las ecuaciones anteriores, se pueden sacar varias conclusiones:

- El valor de intensidad promedio capturada viene dado por la ecuación (1.6).
- En la práctica, cuando se proyecta un patrón sinusoidal, ecuación 1.1, es necesario seleccionar los valores de a_p y b_p , normalmente $a_p = b_p = 0.5$. El objetivo es que $a(x, y) \pm b(x, y)$ esté dentro del rango dinámico de $I(x, y)$; si no se cumple esa condición, la intensidad capturada $I(x, y)$ se trunca al estar fuera de los límites.

Otro término que puede ser obtenido es la modulación de datos $C(x, y)$.

$$C(x, y) = \frac{b(x, y)}{a(x, y)} = \frac{r(x, y)b_0(x, y)}{r(x, y)[a_1(x, y) + a_0(x, y)] + a_2(x, y)} \quad (1.8)$$

El término de modulación de datos es un indicador de la calidad del contraste de franjas, que oscila entre $0 \leq C(x, y) \leq 1$; donde 1 representa el valor ideal para un buen contraste de franjas. Los valores por debajo de cierto umbral pueden eliminarse, ya que no tienen información de buena calidad, lo que suele crear problemas en el desenvolvimiento de fase.

En la Figura 1.3 se muestra una imagen de franjas moduladas por la presencia de un objeto, basada en la ecuación (1.5).

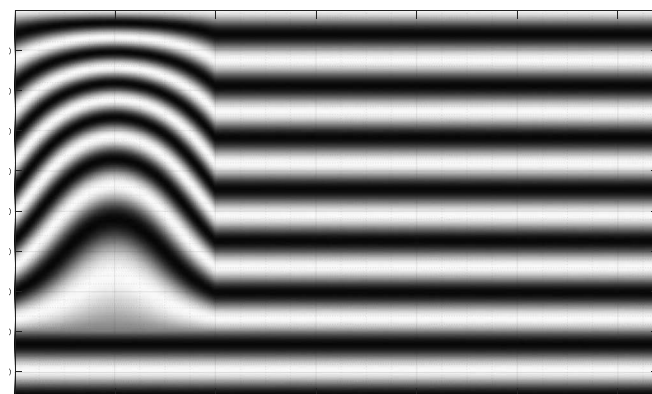


Figura 1.3. Imagen de franjas moduladas por una superficie.

CAPÍTULO II

MARCO TEÓRICO

2.1 Técnicas de perfilometría por proyección de franjas (PPF)

Después de comprender cómo se forma y representa matemáticamente una imagen, a continuación, se clasificarán las técnicas de análisis de franjas, en métodos espaciales y temporales [28].

En general, el análisis espacial requiere solo una imagen de franja para obtener la información de altura de un objeto, y las técnicas temporales suelen estar relacionadas con algoritmos de corrimiento de fase, ya que requieren más de una imagen de franjas para recuperar la forma. Una clasificación de los algoritmos espaciales y temporales [46], se muestra en la Figura 2.1.

2.1.1 Técnicas espaciales

Estos algoritmos son adecuados para objetos dinámicos ya que sólo se requiere una captura para recuperar la topografía del objeto. Aplicaciones como: reconocimiento facial, análisis de marcha, conducción autónoma, entre otras; exploran el uso de estas técnicas donde el tiempo es una limitación. Estos análisis ofrecen una ventaja para muchas aplicaciones ya que el análisis se realiza mediante un solo patrón. Sin embargo, el procesamiento es más complicado en comparación con la mayoría de los métodos temporales.

2.1.1.1 Perfilometría por transformada de Fourier (FTP)

Takeda [29-30], [39], [47-48] es el pionero en el método FTP. Esta técnica se basa en la transformada rápida de Fourier. Un único patrón de franjas capturado se transforma al dominio de la frecuencia, la frecuencia fundamental se filtra y se selecciona un lóbulo desplazándolo hacia el origen. Luego, se aplica la transformada

de Fourier inversa y se logra un patrón filtrado, donde se puede recuperar la fase de una superficie.

El proceso matemático se puede describir mediante el siguiente procedimiento: el patrón de franjas capturado se puede describir mediante una versión ligeramente modificada de la ecuación (1.5) como:

$$I(x, y) = a(x, y) + b(x, y) \cos [\phi(x, y)] \quad (2.1)$$

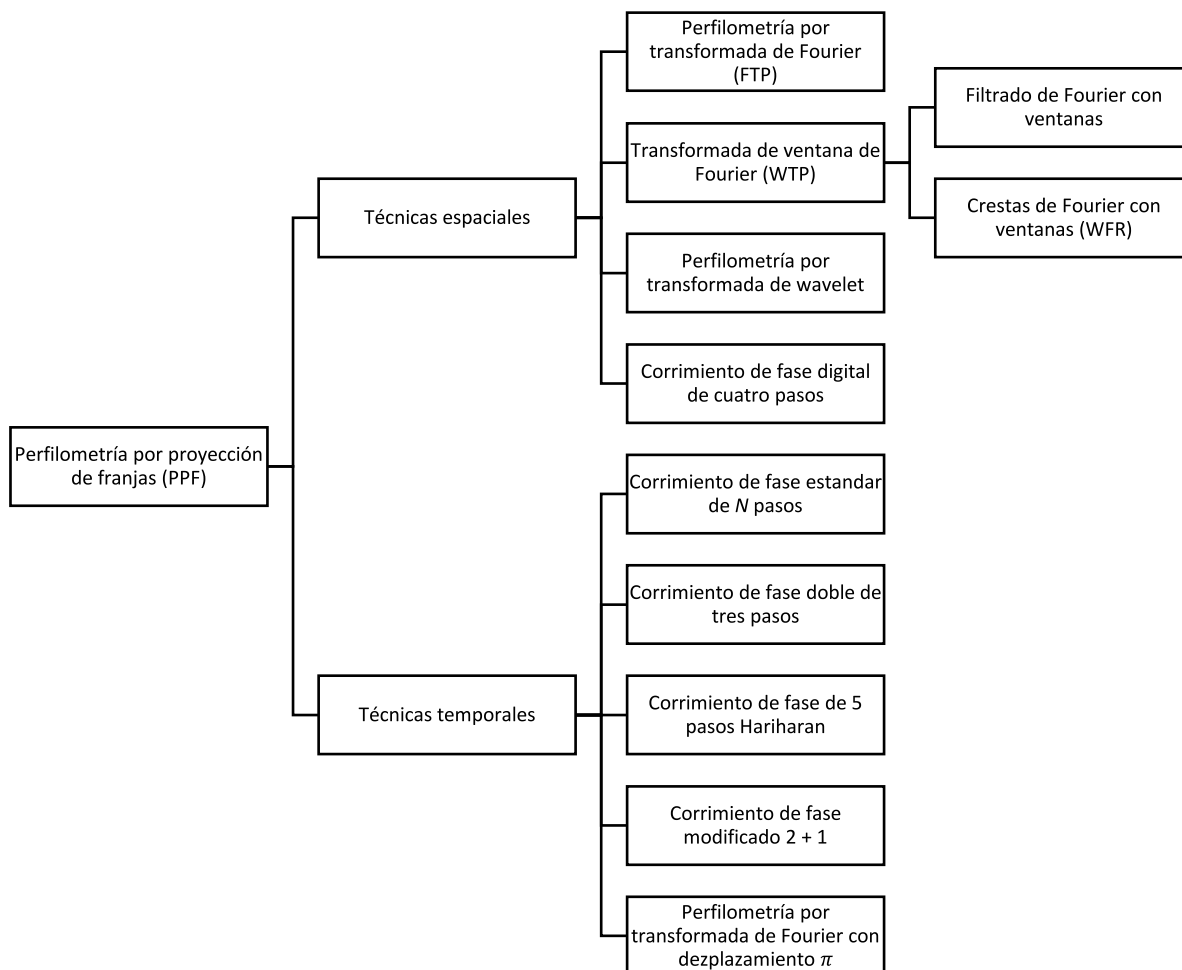


Figura 2.1. Categorización de técnicas espaciales y temporales.

La función se traslada al dominio de la frecuencia, aplicando una transformada de Fourier 2D:

$$I(u, v, \xi, \eta) = \mathcal{F}[I(x, y)] \quad (2.2)$$

Donde $\mathcal{F}$ denota la transformada de Fourier. Como se ha mencionado, la frecuencia fundamental es filtrada; generalmente, se usa un filtro Hanning y se desplaza al origen [30]:

$$\bar{I}(x, y) = \mathcal{F}^{-1}[\bar{I}(x, y, \xi, \eta)] = a(x, y) \exp[j\phi(x, y)] \quad (2.3)$$

Donde $\bar{I}$ denota la función de filtrada, $\mathcal{F}^{-1}$ se refiere a la transformada de Fourier inversa, ξ y η son las frecuencias locales en las direcciones x, y , y $a(x, y)$ es la amplitud del patrón filtrado. El último paso consiste en extraer la fase del objeto del patrón de franjas filtrado:

$$\phi(x, y) = \tan^{-1} \left[\frac{\text{Im } \bar{I}(x, y)}{\text{Re } \bar{I}(x, y)} \right] \quad (2.4)$$

La fase resultante está envuelta, por lo que se debe implementar un algoritmo de desenvolvimiento. El proceso anterior utiliza la transformada de Fourier 2D, pero la transformada de Fourier 1D es también aplicable.

Este método es más tolerante al ruido; sin embargo, como la información de la frecuencia es global, una señal en una sección específica impactará en otras secciones, ya que el contenido espectral de frecuencia se conoce, pero no el tiempo de ocurrencia.

2.1.1.2 Transformada de ventana de Fourier

La WFT también se denomina transformada de Fourier de tiempo reducido. Se basa en el análisis tiempo-frecuencia, su principal característica es determinar el contenido de frecuencia y fase de secciones locales de una señal a medida que cambia con el tiempo. Esta técnica ha sido ampliamente investigada por Kemaio [49-50] para analizar patrones de franjas, y derivar tanto mapas de fase como de derivada de fase. La WFT y la WFT inversa (IWFT) se pueden escribir como:

$$S\{f(u, \xi)\} = \int_{-\infty}^{\infty} f(x)g(x-u) \exp(-j\xi x) dx \quad (2.5)$$

$$f(x) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} S\{f(u, \xi)\}g(x-u) \exp(j\xi x) d\xi du \quad (2.6)$$

Donde $S\{f(u, \xi)\}$ es el espectro de la WFT (S significa transformada de Fourier en tiempo corto), y como convención ξ denota frecuencia; $g(x-u)$ es una función de ventana trasladada y puede elegirse de una amplia variedad; normalmente se utiliza una función gaussiana:

$$g(x) = \exp(-x^2/2\sigma^2) \quad (2.7)$$

El parámetro σ controla la anchura de la ventana. A partir de las ecuaciones (2.5) y (2.6), se observa que la WFT es similar a la transformada de Fourier 1D excepto por la función ventana.

Similar a esto, también se puede hacer una representación 2D WFT, similar a las ecuaciones (2.5) y (2.6), la 2D WFT e IWFT [51] se pueden representar como:

$$S\{f(u, v, \xi, \eta)\} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) g_{u,v,\xi,\eta}^*(x, y) dx dy \quad (2.8)$$

$$f(x, y) = \frac{1}{4\pi^2} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} S\{f(u, v, \xi, \eta)\} g_{u,v,\xi,\eta}(x, y) d\xi d\eta du dv \quad (2.9)$$

Donde ξ, η son las frecuencias locales, * denota conjugado complejo, y $g_{u,v,\xi,\eta}(x, y)$ se conoce como la WFT:

$$g_{u,v,\xi,\eta}(x, y) = g(x - u, y - v) \exp(j\xi x + j\eta y) \quad (2.10)$$

La función de ventana gaussiana debe ser diferente a la ecuación (2.7) ya que el análisis es bidimensional; esta nueva ventana se expresa como:

$$g(x, y) = \exp(-x^2/2\sigma_x^2 - y^2/2\sigma_y^2) \quad (2.11)$$

Basándose en la WFT, pueden aplicarse dos enfoques diferentes para la demodulación de franjas: el método de filtrado de Fourier con ventanas (WFF) y el método de crestas de Fourier con ventanas (WFR).

Método de filtrado de ventana de Fourier (WFF)

El principio del WFF consiste en filtrar los patrones de franjas en el dominio de Fourier con ventanas. Esto es posible porque los componentes espectrales del ruido tienen menor amplitud, debido a la aleatoriedad y la incoherencia, a partir de esta característica se puede seleccionar un umbral para filtrar el ruido del patrón de franjas. El proceso WFF se muestra en la Figura 2.2.

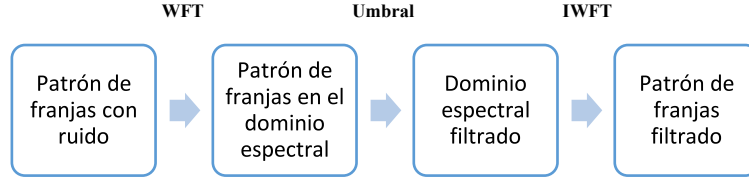


Figura 2.2. Diagrama de flujo de WFF.

El método WFF para 1D puede expresarse a partir de las ecuaciones (2.5) y (2.6) como:

$$\bar{f}(x) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \int_{\xi_l}^{\xi_h} \bar{S}\{f(x, \xi)\} g(x) \exp(j\xi x) dx d\xi \quad (2.12)$$

Donde los límites de integración (ξ_h, ξ_l) están limitados para evitar cálculos innecesarios y $\bar{S}\{f(u, \xi)\}$ es el espectro umbralizado descrito por la ecuación (2.13):

$$\bar{S}\{f(u, \xi)\} = \begin{cases} S\{f(u, \xi)\} & \text{if } |S\{f(u, \xi)\}| \geq Umbral \\ 0 & \text{if } |S\{f(u, \xi)\}| \leq Umbral \end{cases} \quad (2.13)$$

Dado que los patrones de franjas son matrices 2D, la ecuación (2.12) puede ser expandida a 2D y representada como:

$$\bar{f}(x) = \frac{1}{4\pi^2} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \int_{\eta_l}^{\eta_h} \int_{\xi_l}^{\xi_h} \bar{S}\{f(u, v, \xi, \eta)\} g_{u,v,\xi,\eta}(x, y) d\xi d\eta du dv \quad (2.14)$$

Donde los límites de integración $(\eta_h, \eta_l, \xi_h, \xi_l)$ están limitados para acelerar el tiempo de cálculo, y $\bar{S}\{f(u, v, \xi, \eta)\}$ es el espectro umbralizado, definido por:

$$\begin{aligned} & \bar{S}\{f(u, v, \xi, \eta)\} \\ = & \begin{cases} S\{f(u, v, \xi, \eta)\} \text{ if } |S\{f(u, v, \xi, \eta)\}| \geq Umbral \\ 0 \text{ if } |S\{f(u, v, \xi, \eta)\}| \leq Umbral \end{cases} \end{aligned} \quad (2.15)$$

El resultado de cualquiera de las ecuaciones (2.12) o (2.14) es un patrón de franjas suave que puede ser analizado desde cualquier técnica espacial o temporal, WFF puede ser considerada como una técnica de eliminación de ruido.

Método de crestas de Fourier con ventana (WFR)

WFR generalmente se implementa como un algoritmo bidimensional; matemáticamente se puede expresar como:

$$[\omega_x(u, v), \omega_y(u, v)] = \arg \max_{\xi, \eta} |S\{f(u, v, \xi, \eta)\}| \quad (2.16)$$

Dónde ω_x y ω_y son las frecuencias locales en el píxel ubicado en (u, v) . Se basa en el principio de transformada de ventana de Fourier (WFT) en el que se coloca una ventana sobre una determinada parte del patrón de franjas y se calcula la WFT en esa ventana. Se encuentran diferentes coeficientes y un valor óptimo se elige como las frecuencias locales toman el valor de (ξ, η) cuando el espectro de Fourier es máximo $|S\{f(u, v, \xi, \eta)\}|$. Los valores se utilizan luego para caracterizar el patrón de franjas en el centro de la ventana. Para analizar todo el patrón de franjas, la ventana se desplaza píxel a píxel. El valor máximo del espectro o cresta de Fourier es:

$$r(u, v) = |S\{f[u, v, \omega_x(u, v), \omega_y(u, v)]\}| \quad (2.17)$$

A partir del valor de la cresta, la fase se puede extraer mediante:

$$\varphi(u, v) = \tan^{-1} \left\langle \frac{\text{Im } S\{f[u, v, \omega_x(u, v), \omega_y(u, v)]\}}{\text{Re } S\{f[u, v, \omega_x(u, v), \omega_y(u, v)]\}} \right\rangle + \omega_x(u, v)u + \omega_y(u, v)v \quad (2.18)$$

Dado que la fase estimada está envuelta, es necesario un algoritmo de desenvolvimiento.

2.1.1.3 Perfilometría por transformada de wavelet (WTP)

Otra técnica para extraer la forma del objeto, como su nombre lo indica se basa en la conocida transformada wavelet [52], [53], [54], [55], [56], donde el término wavelet se refiere a ondícula u onduleta. A diferencia de la transformada de Fourier, en la que se puede utilizar una suma de senos y cosenos para representar una señal, en la transformada wavelet una señal se puede representar con la suma de wavelets. Existen diferentes wavelets, pero en cada trabajo de campo se prefiere una wavelet específica, ya que presenta excelentes propiedades para esa zona concreta. Si se compara la WTP con la WFT, su principal ventaja es el cambio del tamaño de la ventana en función de la frecuencia. La fase puede extraerse mediante:

$$\varphi(b) = \tan^{-1} \left[\frac{\text{Im } WI(a_{r,b}, b)}{\text{Re } WI(a_{r,b}, b)} \right] \quad (2.19)$$

Donde $a_{r,b}$ se refiere al parámetro de dilatación de escala a en la ubicación de la cresta en la posición b , WI denota la transformada wavelet 1D definida como:

$$WI(a, b) = \frac{1}{\sqrt{a}} \int_{-\infty}^{\infty} I(x) \psi^* \left(\frac{x-b}{a} \right) dx \quad (2.20)$$

Donde ψ^* denota el complejo conjugado, $I(x)$ se refiere al patrón de franjas capturado en la ecuación (2.1), donde las franjas varían en la dirección x ; y ψ es la ondícula madre, en este caso, la ondícula Morlet 1D que se utiliza habitualmente en el procesamiento de imágenes, y es expresada por:

$$\psi(x) = \frac{1}{\pi^{1/4}} \left(\frac{2\pi}{\gamma} \right)^{1/2} \exp \left[-\frac{(2\pi/\gamma)^2 x^2}{2} + j2\pi x \right] \quad (2.21)$$

Donde γ se define como $\gamma = \pi\sqrt{2/\ln 2}$.

2.1.1.4 Corrimiento de fase digital de cuatro pasos (D4-PS)

El corrimiento de fase digital de cuatro pasos [57-58] se basa en la transformada de Riesz; es una técnica novedosa y considera el hecho de que una técnica temporal se utiliza como un método espacial. Esto es posible gracias a la captura de un patrón de franjas modulado y digitalmente generar otros patrones de franjas; en otras palabras, desplazando el patrón sinusoidal digitalmente. La ecuación (2.1) se toma como el patrón capturado de franjas para el análisis matemático. Los kernel de primer orden de Riesz se calculan mediante:

$$r_x = \frac{-x}{2\pi(x^2 + y^2)^{3/2}} \quad (2.22)$$

$$r_y = \frac{-y}{2\pi(x^2 + y^2)^{3/2}} \quad (2.23)$$

Donde r_x y r_y son núcleos en la dirección x y y , respectivamente. Las transformadas de Riesz de primer, segundo y tercer orden se calculan mediante:

$$R^1\{f(x, y)\} = \begin{cases} f_1(x, y) = r_x * f(x, y) \\ f_2(x, y) = r_y * f(x, y) \end{cases} \quad (2.24)$$

$$R^2\{f(x, y)\} = \begin{cases} f_3(x, y) = r_x * [r_x * f(x, y)] \\ f_4(x, y) = r_y * [r_x * f(x, y)] \\ f_5(x, y) = r_y * [r_y * f(x, y)] \end{cases} \quad (2.25)$$

$$R^3\{f(x, y)\} = \begin{cases} f_6(x, y) = r_x * r_x * [r_x * f(x, y)] \\ f_7(x, y) = r_x * r_x * [r_y * f(x, y)] \\ f_8(x, y) = r_x * r_y * [r_y * f(x, y)] \\ f_9(x, y) = r_y * r_y * [r_y * f(x, y)] \end{cases} \quad (2.26)$$

Donde $f_1(x, y) - f_9(x, y)$ se obtienen los componentes de Riesz, $*$ denota el operador de convolución. Los componentes Riesz se combinan en las siguientes ecuaciones para generar digitalmente los tres patrones de franjas necesarios para el uso de un algoritmo de cuatros pasos:

$$I_1(x, y) = I'(x, y) + I''(x, y) \cos[\varphi(x, y) + \pi/2] = \text{sgn}(u) \cdot \sqrt{f_1(x, y)^2 + f_2(x, y)^2} \quad (2.27)$$

$$I_2(x, y) = I'(x, y) + I''(x, y) \cos[\varphi(x, y) + \pi] = \text{sgn}(v) \cdot [f_3(x, y) + f_5(x, y)] \quad (2.28)$$

$$I_3(x, y) = I'(x, y) + I''(x, y) \cos[\varphi(x, y) + 3\pi/2] = \text{sgn}(w) \cdot \sqrt{f_6(x, y)^2 + f_9(x, y)^2} \quad (2.29)$$

Donde $u = f_1(x, y), f_2(x, y)$, $v = f_3(x, y), f_4(x, y), f_5(x, y)$, $w = f_6(x, y), f_7(x, y), f_8(x, y), f_9(x, y)$ y sgn es la función signo definido como:

$$\text{sgn} = \begin{cases} -1 & \text{if } x < 0 \\ 0 & \text{if } x = 0 \\ 1 & \text{if } x > 0 \end{cases} \quad (2.30)$$

Siguiendo estos pasos se tendrá la captura original y tres nuevas imágenes creadas digitalmente, permitiendo el uso de un algoritmo de corrimiento de fase (PSA) de cuatro

pasos. Esta técnica es interesante ya que permite la aplicación de los PSAs convencionales que son difíciles de implementar en objetos dinámicos, por el tiempo de proyección, adquisición y calibración del desplazamiento de fase.

2.1.2 Técnicas temporales

Las técnicas temporales se conocen como algoritmos de desplazamiento de fase (PSA), con el tiempo, se han presentado diferentes revisiones de algoritmos [1], [3-4], [32], [59-61]. Su principio es generar y proyectar patrones de franjas sinusoidales con desplazamientos de fase entre ellos. La ecuación (1.1) se puede modificar para incluir el desplazamiento angular en la proyección, el patrón de franjas digitales se puede expresar como:

$$I_{p,n}(x_p, y_p) = a_p + b_p \cos(2\pi f_0 x_p - 2\pi n/N) \quad (2.31)$$

Donde el subíndice n se refiere al número de patrón, y N es el número total de desplazamientos. De manera similar, la ecuación 1.5 se modifica para introducir el desplazamiento de fase en la captura, y el patrón capturado de franjas sinusoidales se puede expresar mediante:

$$I_n(x, y) = a(x, y) + b(x, y) \cos[\varphi(x, y) - 2\pi n/N] \quad (2.32)$$

En la sección siguiente, se presenta y analiza una descripción general de los algoritmos de PSA, junto con sus respectivas propiedades. La ecuación (2.32) se utiliza ampliamente para el análisis matemático de los siguientes algoritmos.

2.1.2.1 PSA de mínimos cuadrados de N pasos (LS-PSA)

Este algoritmo fue propuesto por Brunning *et al.* [62], la fase se puede extraer mediante la siguiente ecuación [63-64] :

$$\varphi(x, y) = \tan^{-1} \left[\frac{\sum_{n=0}^{N-1} I_n(x, y) \sin(2\pi n/N)}{\sum_{n=0}^{N-1} I_n(x, y) \cos(2\pi n/N)} \right] \quad (2.33)$$

La ecuación anterior es un caso particular donde los desplazamientos de fase están uniformemente espaciados, el resultado es una función tangente inversa, generando un mapa de fase envuelta. Se requieren al menos tres patrones de franjas desplazadas, ya que hay tres incógnitas por determinar. Cuando este algoritmo se implementa en una computadora, se recomienda utilizar una función tangente inversa de 4 cuadrantes para permitir el cálculo en los cuatro cuadrantes.

LS-PSA tiene tres características principales:

- a) Es aplicable para un número N de patrones desplazados.
- b) Ofrece una mayor relación señal-ruido.
- c) Se obtienen buenos resultados, aun con desplazamientos que no son uniformemente espaciados.

En esta técnica se pueden utilizar N pasos. En teoría, un mayor número de pasos ofrecen una alta exactitud ya que la intensidad del ruido aleatorio disminuye por un factor $1/\sqrt{N}$.

2.1.2.2 PSA de tres pasos

Como se mencionó anteriormente, hay tres incógnitas en la ecuación (1.5). Por lo tanto, tres patrones de franjas son necesarios para encontrar esos valores. En lugar de resolver para un desplazamiento específico, supongamos que el patrón de fase capturado se expresa mediante:

$$I_0(x, y) = a(x, y) + b(x, y) \cos[\varphi(x, y) - \alpha] \quad (2.34)$$

$$I_1(x, y) = a(x, y) + b(x, y) \cos[\varphi(x, y)] \quad (2.35)$$

$$I_2(x, y) = a(x, y) + b(x, y) \cos[\varphi(x, y) + \alpha] \quad (2.36)$$

Al resolver el sistema de tres ecuaciones, la fase en cada punto se calcula mediante la ecuación (2.37).

$$\varphi(x, y) = \tan^{-1} \left\{ \frac{(1 - \cos \alpha)}{(\sin \alpha)} \frac{[I_2(x, y) - I_0(x, y)]}{[I_0(x, y) - 2I_1(x, y) + I_2(x, y)]} \right\} \quad (2.37)$$

Como se puede observar, el cambio de fase se puede seleccionar libremente. En general, el PSA de tres pasos no es robusto contra el error de desfase en por desplazamientos no uniformemente espaciados. Algunos casos, como el descrito por Creath [65], suponen desplazamientos de fase que están igualmente espaciados por $\alpha = \pi/2$, que posee una relación señal-ruido más baja en comparación con LS-PSA.

2.1.2.3 PSA doble de tres pasos

El PSA doble de tres pasos propuesto por Huang *et al.* [66], es una versión mejorada de la PSA de tres pasos. Los patrones de franjas desplazadas se expresan como un sistema de dos conjuntos de patrones, el primer conjunto:

$$I_0(x, y) = a(x, y) + b(x, y) \cos[\varphi(x, y)] \quad (2.38)$$

$$I_1(x, y) = a(x, y) + b(x, y) \cos[\varphi(x, y) - 2\pi/3] \quad (2.39)$$

$$I_2(x, y) = a(x, y) + b(x, y) \cos[\varphi(x, y) - 4\pi/3] \quad (2.40)$$

El segundo conjunto de patrones:

$$I_3(x, y) = a(x, y) + b(x, y) \cos[\varphi(x, y) - \pi/3] \quad (2.41)$$

$$I_4(x, y) = a(x, y) + b(x, y) \cos[\varphi(x, y) - \pi] \quad (2.42)$$

$$I_5(x, y) = a(x, y) + b(x, y) \cos[\varphi(x, y) - 5\pi/3] \quad (2.43)$$

Los mapas de fase deben calcularse dos veces, uno para cada conjunto de patrones con el uso de PSA de tres pasos. Posteriormente, se promedian los mapas de dos fases; el error se elimina basándose en la suposición de que cuando se calculan mapas de dos fases con una diferencia de $\pi/3$, la diferencia de fase es casi 180° . La principal ventaja de este algoritmo es la reducción del error de fase generado por la no linealidad de la intensidad [66].

2.1.2.4 Carré PSA

Este algoritmo fue propuesto por Carré [67], se puede pensar como el caso general de PSA de cuatro pasos [68]. Los patrones de franjas capturados se describen como:

$$I_0(x, y) = a(x, y) + b(x, y) \cos\left[\varphi(x, y) - \frac{3}{2}\alpha\right] \quad (2.44)$$

$$I_1(x, y) = a(x, y) + b(x, y) \cos\left[\varphi(x, y) - \frac{1}{2}\alpha\right] \quad (2.45)$$

$$I_2(x, y) = a(x, y) + b(x, y) \cos\left[\varphi(x, y) + \frac{1}{2}\alpha\right] \quad (2.46)$$

$$I_3(x, y) = a(x, y) + b(x, y) \cos\left[\varphi(x, y) + \frac{3}{2}\alpha\right] \quad (2.47)$$

La fase en cada punto se puede calcular mediante:

$$\varphi(x, y) = \tan^{-1} \left\{ \frac{\sqrt{[3(I_2 - I_3) - (I_1 - I_4)][(I_2 - I_3) + (I_1 - I_4)]}}{(I_2 + I_3) - (I_1 + I_4)} \right\} \quad (2.48)$$

En la ecuación (2.48), se omitió la dependencia espacial por motivos de claridad. La principal ventaja es que no es necesario calibrar el sistema, ya que no hay que conocer el desplazamiento de fase de cada patrón específico.

2.1.2.5 PSA de 5 pasos de Hariharan

El PSA de cinco pasos de Hariharan es similar al PSA de Carré, también con un desplazamiento de fase constante desconocido [69]. Considerando un desplazamiento de fase lineal entre capturas, las cinco imágenes de franjas se describen mediante:

$$I_1(x, y) = a(x, y) + b(x, y) \cos[\varphi(x, y) - 2\alpha] \quad (2.49)$$

$$I_2(x, y) = a(x, y) + b(x, y) \cos[\varphi(x, y) - \alpha] \quad (2.50)$$

$$I_3(x, y) = a(x, y) + b(x, y) \cos[\varphi(x, y)] \quad (2.51)$$

$$I_4(x, y) = a(x, y) + b(x, y) \cos[\varphi(x, y) + \alpha] \quad (2.52)$$

$$I_5(x, y) = a(x, y) + b(x, y) \cos[\varphi(x, y) + 2\alpha] \quad (2.53)$$

La fase en cada punto se puede calcular mediante:

$$\tan[\varphi(x, y)] = \frac{\sin[\varphi(x, y)]}{\cos[\varphi(x, y)]} = \frac{[1 - \cos(2\alpha)]}{\sin \alpha} \frac{(I_2 - I_4)}{(2I_3 - I_1 - I_5)} \quad (2.54)$$

Hay otros PSA de cinco pasos [70], pero Hariharan PSA está diseñado para ser insensible a los errores de cambio de fase. Suponiendo que los cambios de fase varíen entre 86° y 94° , e introduciendo esos valores en el factor $\frac{[1-\cos(2\alpha)]}{\sin \alpha}$, se puede observar que el resultado no es superior a 0,001. Por lo tanto, se concluye que puede tolerar pequeños errores de calibración por desplazamiento de fase.

2.1.2.6 PSA 2 + 1 modificado

Estas técnicas se han propuesto [71-72] para permitir mediciones de formas 3D en tiempo real. Esta flexibilidad permite la medición de objetos dinámicos y el reconocimiento de marcha. El PSA 2 + 1 modificado propuesto por Zhang y Yau [73] es un buen enfoque ya que requiere tres capturas: dos de imágenes de franjas y una tercera imagen plana sin modulación de las franjas. Las imágenes están expresadas por:

$$I_0(x, y) = a(x, y) \quad (2.55)$$

$$I_1(x, y) = a(x, y) + b(x, y) \cos(\varphi(x, y)) \quad (2.56)$$

$$I_1(x, y) = a(x, y) + b(x, y) \cos(\varphi(x, y) - \pi/2) = a(x, y) + b(x, y) \sin(\varphi(x, y)) \quad (2.57)$$

La fase en cada punto se puede calcular mediante:

$$\varphi(x, y) = \tan^{-1} \left[\frac{I_2(x, y) - I_0(x, y)}{I_1(x, y) - I_0(x, y)} \right] \quad (2.58)$$

Las ventajas PSA 2 + 1 modificado son los siguientes:

- a) Como se puede ver de la ecuación (2.55), la imagen se puede utilizar para la textura.

b) De acuerdo a Zhang y Yau [73] , el PSA 2 + 1 modificado es dos veces más rápido en comparación con el PSA de tres pasos.

c) El error de medición debido al movimiento es menor, debido a que sólo se requieren dos imágenes. Sin embargo, el pequeño número de capturas, hace a este algoritmo más susceptible al ruido y no linealidad de la intensidad.

2.1.2.7 Perfilometría de transformada de Fourier con desplazamiento π

Diferentes autores proponen una combinación de PSA y FTP [74-75]. Como lo sugiere por su nombre, se utiliza desplazamiento de π . Por lo tanto, es necesario proyectar dos patrones de franjas.

$$I_{p,0}(x_p, y_p) = a_p(x, y) + b_p(x, y) \cos(2\pi f_0 x_p) \quad (2.59)$$

$$I_{p,1}(x_p, y_p) = a_p(x, y) + b_p(x, y) \cos(2\pi f_0 x_p - \pi) = a_p(x, y) - b_p(x, y) \cos(2\pi f_0 x_p) \quad (2.60)$$

Los patrones de franjas capturados son:

$$I_0 = a(x, y) + b(x, y) \cos[2\pi f_0 x + \varphi(x, y)] \quad (2.61)$$

$$I_1 = a(x, y) - b(x, y) \cos[2\pi f_0 x + \varphi(x, y)] \quad (2.62)$$

Conociendo la diferencia entre los desplazamientos, la información de frecuencia fundamental se duplica y el término $I'(x, y)$ se elimina:

$$I_d(x, y) = I_0(x, y) - I_1(x, y) = 2b(x, y) \cos[2\pi f_0 x + \varphi(x, y)] \quad (2.63)$$

Luego, se aplica FTP para extraer la información de la fase. La ventaja de esta técnica es que el espectro fundamental modulado por el objeto, teóricamente, puede extenderse de 0 a $2f_0$, eliminando el lóbulo del término de DC.

2.2 Ruido

El ruido es un fenómeno común dentro de las imágenes. Este consiste en información parásita que introduce perturbaciones aleatorias en la captura adquirida, alterándola o agregándole información [76].

2.2.1 Fuentes de ruido

Existen diferentes tipos de fuentes de ruido. Algunas están estrechamente relacionadas al sensor, otras con el proceso de adquisición y algunas más están asociadas con la etapa de procesamiento de la señal, donde la señal de salida del sensor es transformada en una imagen para poder ser visualizada.

La influencia del ruido en una imagen depende de los parámetros de adquisición: nivel de iluminación de la escena, sensibilidad ISO, temperatura, entre otros.

2.2.1.1 Ruido del sensor

Este tipo de ruido es introducido durante la transformación de una señal luminosa en una señal eléctrica. Este ruido toma la forma de ruido de electrones y ruido de fotones. Hay muchos tipos de ruido relacionados con el sensor y algunos de ellos están relacionados con la tecnología utilizada. Los tipos de ruido que tienen un mayor impacto en las imágenes, se mencionan a continuación.

Ruido térmico o ruido Johnson-Nyquist

Es generado por el movimiento térmico natural de los electrones. Cuando una imagen es capturada por una cámara, la electrónica dentro de la cámara calentará al sensor y esto incrementará la energía térmica, generando una acumulación de electrones térmicos en el sensor. Los electrones térmicos son independientes de los fotoelectrones generados debido a los fotones que inciden en el sensor. Desafortunadamente, el sensor de la cámara no puede diferenciar entre estos dos tipos de electrones y la lectura final del sensor será un acumulado de los electrones térmicos y los fotoelectrones. Este efecto se conoce como corriente oscura y su distribución espacial no es uniforme. Este ruido es proporcional a la raíz cuadrada del número de electrones producidos a través de la agitación térmica en la ausencia de

iluminación externa. Por lo tanto, es independiente del número de electrones incidiendo en el elemento fotosensible.

Ruido sal y pimienta

Es ruido impulsivo que se muestra como la presencia de píxeles blancos y negros distribuidos aleatoriamente en la imagen. Este puede provenir desde fallas en ciertos elementos del sensor, hasta errores en la transmisión de datos o la presencia de polvo sobre la superficie del sensor.

Como la sensibilidad de los fotorreceptores varía lentamente comparada con su eficiencia cuántica en función del factor de llenado, la respuesta no uniforme de los foto-receptores introduce ruido. La eficiencia cuántica es un parámetro importante en los sensores y mide el grado en el que se convierten los fotones entrantes en señales eléctricas. El factor de llenado mide el grado en el cual el sensor convierte la energía eléctrica generada en potencia utilizable y es definido como la medida del área efectiva del sensor que es sensible a la luz.

Ruido de disparo o ruido cuántico

Es el ruido fotónico que se produce cuando el número de fotones incidentes es suficientemente bajo para ser visible. Esta distribución de fotones se modela como un proceso de Poisson. Por esto el número de fotones arribando a un foto-receptor tiene una dispersión en torno a un valor medio λ , con una desviación estándar σ , los cuales son definidos como:

$$\lambda = I A T \quad (2.64)$$

$$\sigma = \sqrt{\lambda} \quad (2.65)$$

Donde I es la intensidad de la fuente de luz, A es el área del foto-receptor y T el tiempo de integración. No hay manera física de eliminarlo, ya que es intrínseco a la naturaleza cuantizada de la luz.

Ruido de lectura

Es producido por los componentes electrónicos antes de la digitalización. Este ruido es de origen térmico y procede del amplificador de salida. Cuanto mayor sea la amplificación, mayor será este ruido. Es independiente del número de fotones incidentes.

2.2.1.2 Ruido en el proceso de adquisición de imágenes

Esta fuente de ruido es debida a la proyección de una señal luminosa sobre una superficie fotosensible, a través de un sistema de lentes. Durante esta fase, pueden introducirse distorsiones geométricas, especialmente alrededor de los bordes del sensor, así como aberración cromática, desenfoque o viñeteado. Este tipo de error de adquisición es clasificado como ruido, debido a que degrada la calidad de la imagen, pero estrictamente no es una forma de ruido, ya que no es aleatorio y es completamente predecible.

2.2.1.3 Ruido de procesamiento de señales

Ruido de cuantificación o ruido de conversión analógica/digital es un proceso no lineal. Cada foto-receptor envía una señal analógica como salida, la cual debe ser transformada en su valor digital más cercano tomado de un conjunto finito de valores discretos. Esta aproximación genera ruido de cuantificación de la señal. Este ruido de cuantificación sigue una distribución uniforme, con un valor máximo que depende del tamaño de cuantificación. Este puede ser despreciable incrementando el número de bits utilizados para codificar los valores cuantizados.

2.2.2 Modelado de ruido

Para realizar el modelado del ruido, una de las teorías más sencillas y utilizadas es la de considerar que el ruido es aditivo, con un valor medio de cero e independiente del valor medido. El ruido se considera como un fenómeno estacionario y ergódico. Este es caracterizado por una densidad de potencia, una varianza y una función de correlación. En la práctica, el ruido es diferente para cada pixel, por lo tanto, es difícil determinar un modelo universal. Estrictamente hablando, el ruido debe de adaptarse

a cada sensor y sistema de adquisición de imágenes, pero en términos generales, el ruido asociado con un sensor se modela en términos de ruido aditivo, blanco, estacionario y Gaussiano.

Para modelar las formas más comunes de ruido, la densidad de probabilidad asociada utilizada tiene la forma:

$$f(c) = Ce^{-K|c|^\alpha} \quad (2.66)$$

Donde C , K y α son constantes y c representa la intensidad medida por el foto-receptor. Por ejemplo:

- $\alpha = 1$: Ruido exponencial.
- $\alpha = 2$: Ruido Gaussiano.
- $\alpha \rightarrow \infty$: Ruido uniforme.

2.2.2.1 Ruido impulsivo

El ruido impulsivo es caracterizado por una densidad de probabilidad f , que representa la probabilidad de que un píxel contenga ruido. La característica impulsiva de este ruido es asociada con la aparición de valores anómalos, que representan una desviación significativa del rango de valores encontrados dentro de un rango dado del píxel que está siendo filtrado.

2.2.2.2 Ruido Gaussiano

La densidad de probabilidad de una distribución normal con media μ y desviación estándar σ es descrita por la ecuación (2.67) y puede ser utilizada para modelar ruido Gaussiano. En el caso donde $\mu = 0$, se tiene ruido blanco Gaussiano aditivo AWGN:

$$f(c) = \frac{1}{\sigma\sqrt{2\pi}} e^{\left[-\frac{1}{2}\left(\frac{c-\mu}{\sigma}\right)^2\right]} \quad (2.67)$$

El AWGN, es un tipo de ruido usado comúnmente en sistemas de comunicación y procesamiento de señales para modelar los efectos de interferencia aleatoria o ruido de fondo en un sistema. Es la manera típica para modelar ruido.

Los componentes que lo conforman, se describen a continuación:

1. Aditivo: Es un ruido que se suma a la señal original.
2. Blanco: Ruido blanco que tiene una densidad espectral de potencia constante en todas las frecuencias, es decir, tiene la misma intensidad en todas las frecuencias. Esta propiedad es deseable para modelar muchos tipos de procesos aleatorios.
3. Gaussiano: La distribución de amplitud sigue una distribución Gaussiana. La distribución Gaussiana es utilizada porque muchos procesos naturales exhiben este tipo de comportamiento estadístico.

2.2.2.3 Ruido de Poisson

Los fotones que llegan a un foto-receptor siguen una distribución de Poisson y la probabilidad de detectar un número de fotones n cuando en promedio N sería la cantidad esperada está dada por la siguiente ecuación:

$$p(n) = \frac{N^n e^{-N}}{n!} \quad (2.68)$$

La ecuación representa una distribución de Poisson con media N y desviación estándar $\sqrt{N}$. Sin embargo, para valores grandes de N , $p(n)$ es del orden de $e^{\left[-\frac{(n-N)^2}{2N}\right]}$, la cual representa una distribución Gaussiana con media N con desviación estándar $\sqrt{N}$. Esto justifica la utilización del ruido blanco Gaussiano como modelo de ruido.

2.3 Corrección gamma

Para PPF se supone que el patrón proyectado es un patrón sinusoidal ideal, pero en la práctica no es posible. Los patrones son distorsionados por el factor gamma. El factor gamma es una corrección introducida por los proyectores digitales, la cual se aplica a las imágenes y videos para mejorar su efecto visual, debido a la incapacidad de los proyectores para generar y/o proyectar el color negro [15]. Este efecto puede modelarse de la siguiente manera:

$$I_p(x, y) = [I(x, y)]^\gamma \quad (2.69)$$

Donde I_p es la intensidad del patrón proyectado, I es la intensidad del patrón ideal y γ es el coeficiente gamma. El efecto gamma genera una distorsión no sinusoidal del patrón de franjas, lo cual se refleja introduciendo armónicos de alto orden, como se muestra en la Figura 2.3, el efecto se observa como una atenuación de los valles en el patrón sinusoidal.

Un método para corregir este fenómeno es el mencionado por Zhaoyang Wang [15], en donde se propone generar patrones a partir de la siguiente ecuación:

$$I(x, y) = 2I_0 \left[\frac{1}{2} + \frac{1}{2} \cos \left(2\pi k \frac{x}{w} + \delta \right) \right]^{1/\gamma} \quad (2.70)$$

Donde I_0 es el valor máximo que se desee en el patrón, w es el ancho del patrón de franjas, k es el número de franjas en el patrón y δ es el desplazamiento de fase. Introduciendo la ecuación (2.69) en la ecuación (2.70), se observa que el efecto del factor gamma es cancelado. En la Figura 2.3, se muestra un patrón sinusoidal generado a partir de la ecuación (2.70), se observa una atenuación de los valles, lo cual cancela el efecto que introduce el proyector.

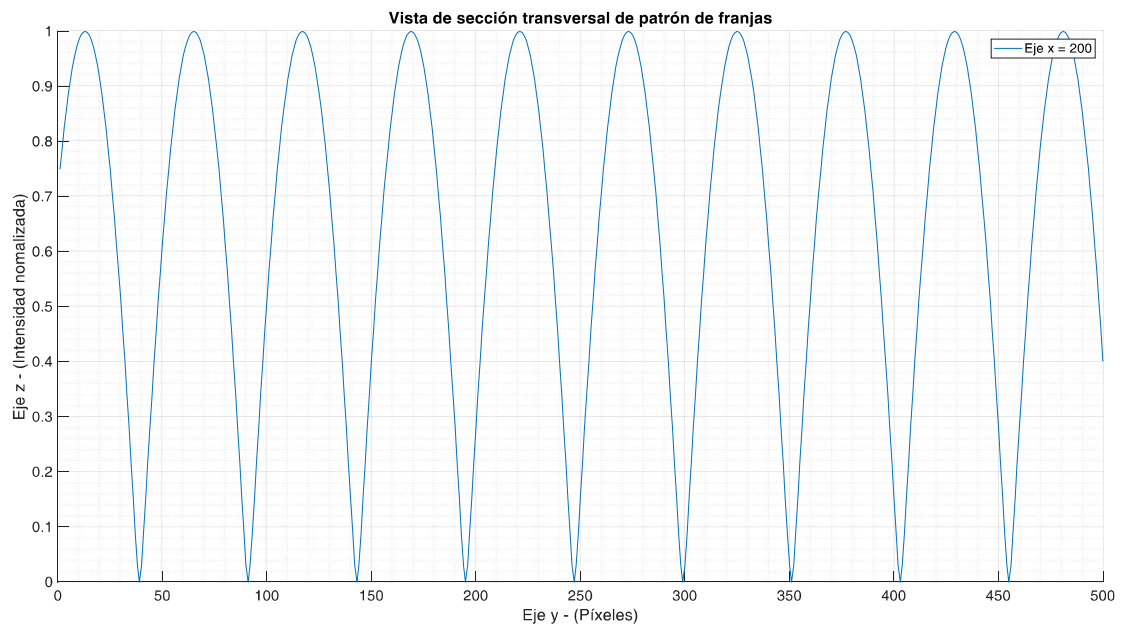


Figura 2.3. Patrón sinusoidal simulado con efecto gamma = 2.2.

CAPÍTULO III

TÉCNICA PROPUESTA

3.1 Descripción de la técnica

Sin duda alguna, dentro de la literatura existe una gran variedad de técnicas de proyección de patrones [18], [77], así como una amplia cantidad de algoritmos de análisis para cada tipo de patrón [8], [28].

Realizando un enfoque particular en los diversos métodos de análisis de franjas, se observa que algunos de ellos involucran métodos matemáticos complejos, los cuales algunas veces requieren complicadas implementaciones.

Dejando un poco de lado esta complejidad existe otra posible limitante, la cual consiste en su aplicabilidad de acuerdo al tipo de objeto que se esté analizando: estáticos o dinámicos.

Como se ha mencionado anteriormente, los algoritmos temporales ofrecen mejores resultados de precisión, exactitud y susceptibilidad a ruido, puesto que existe una mayor información de un mismo objeto y no se sacrifica información espacial. Pero solo son aplicables en objetos estáticos.

Por otro lado, los algoritmos espaciales si bien no tienen tan buena precisión y exactitud, además de ser más susceptibles al ruido, son aplicables en objetos dinámicos.

En este trabajo de tesis se propone un nuevo algoritmo de análisis espacial en el cual, basado en una sola captura de franjas sinusoidales proyectadas sobre un objeto, se pueda obtener su modelo 3D.

Partiendo de la ecuación (2.1), la cual representa un patrón de franjas cosenoidales moduladas por la información del objeto:

$$I(x, y) = a(x, y) + b(x, y)\cos [\varphi(x, y)]$$

La forma del objeto se encuentra en el término $\varphi(x, y)$ que modula la fase del coseno. Despejando este término se obtiene

$$\varphi(x, y) = \cos^{-1} \left[\frac{I(x, y) - a(x, y)}{b(x, y)} \right] \quad (3.1)$$

Entonces, para encontrar la forma del objeto es necesario resolver la función coseno inverso del argumento. Donde $I(x, y)$ es un valor conocido que corresponde a la Intensidad del punto (x, y) en la imagen capturada, pero los valores $a(x, y)$ y $b(x, y)$ deberán ser obtenidos mediante un análisis más detallado.

La Figura 3.1 muestra una vista de la sección transversal del patrón de franjas correspondiente, donde se indican los valores $a(x, y)$ y $b(x, y)$.

Se plantea resolver este problema, a partir del concepto de formación de imágenes, mostrado en la sección 1.3. En la Figura 3.1, se puede observar la contribución de cada una de las variables que ayudan a la formación de la imagen, en este trabajo a estas variables se les hará referencia como parámetros.

Analizando esta figura, se puede observar que el valor de intensidad promedio $a(x, y)$ puede ser estimado como el valor medio $\bar{x}$ del patrón cosenoidal, y de la misma manera se puede visualizar que el término modulación de amplitud $b(x, y)$ puede ser estimado a través del valor pico del patrón cosenoidal menos el parámetro de intensidad promedio.

Una vez estimados ambos valores se puede obtener la fase del objeto mediante la ecuación (3.1).

El planteamiento de esta técnica puede parecer muy simple, pero conlleva retos que deberán ser resueltos para demostrar la validez del algoritmo propuesto, así como su robustez al ser aplicado bajo condiciones reales.

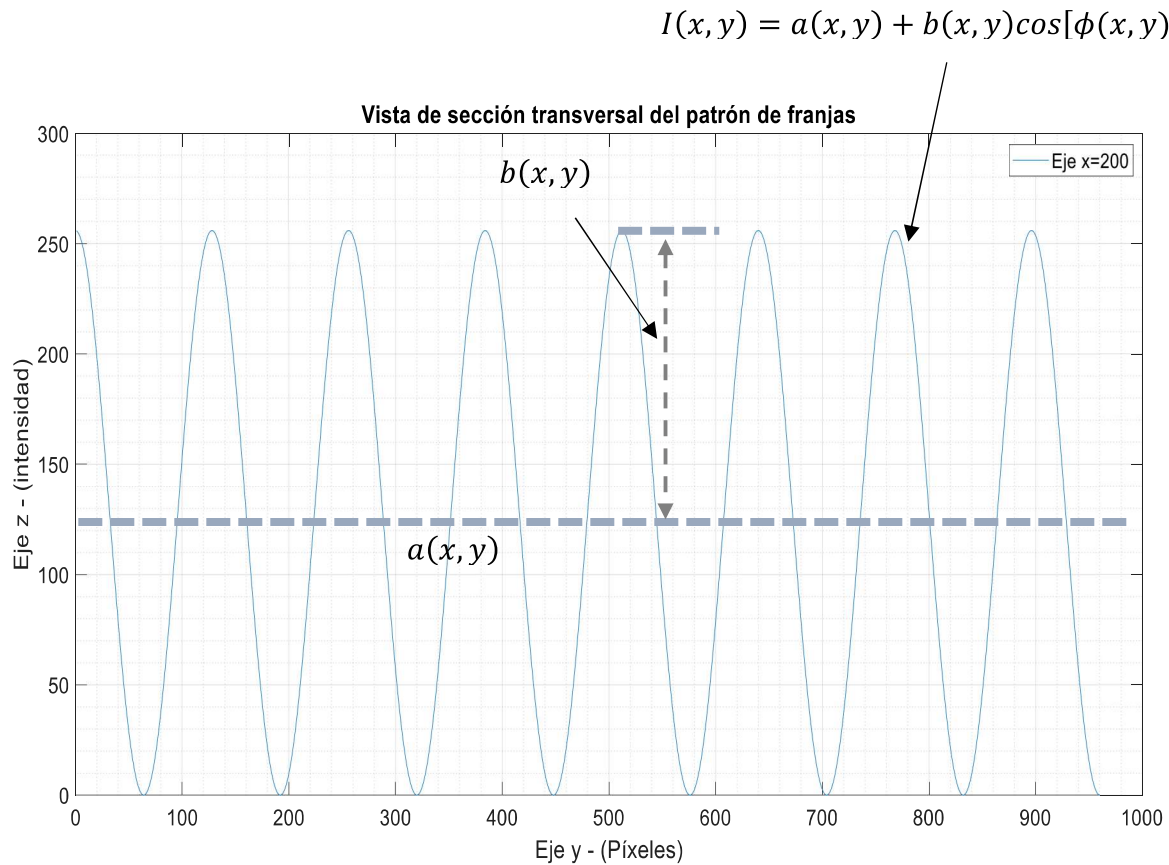


Figura 3.1. Gráfica de variación de intensidad de un corte transversal en la dirección de las franjas.

En resumen, se plantea la creación de una nueva técnica computacional que a partir de una sola captura permita la obtención de la topografía de un objeto, haciendo uso de la estimación de los parámetros $a(x, y)$ y $b(x, y)$ de franjas sinusoidales.

3.2 Descripción de la metodología

La metodología empleada para el desarrollo de este trabajo, se describe de manera general a continuación, pero se describe a detalle en los siguientes capítulos.

Como primer paso se desarrolló un software de simulación con un algoritmo computacional que implementa la estimación de los valores $a(x, y)$ y $b(x, y)$ a partir del análisis de un patrón sinusoidal simulado. Las primeras simulaciones se realizaron bajo condiciones ideales lo cual permitió realizar de manera iterativa las modificaciones

y ajuste necesarios. Después de demostrar que las simulaciones bajo condiciones ideales arrojaron resultados favorables, se procedió a introducir condiciones no ideales donde se adicionaron distintos niveles de ruido blanco para poder evaluar la susceptibilidad a ruido del algoritmo propuesto. Luego se agregó también distorsión gamma a los patrones simulados lo cual es una condición inherente en los sistemas de proyección de franjas

El siguiente paso fue evaluar el desempeño del algoritmo bajo condiciones reales, para lo cual se agregó al software de simulación la capacidad de correr el algoritmo sobre imágenes reales capturadas de una proyección sobre una pantalla. Aquí surgieron algunos retos interesantes que requirieron de nuevos ajustes y modificaciones para obtener datos experimentales consistentes y válidos.

El proceso de desarrollo de este software de simulación y captura, así como los resultados obtenidos se detallan en el Capítulo IV.

Justamente, para utilizar la nueva capacidad de captura de imágenes reales adicionada al software, se desarrolló un montaje experimental de laboratorio. Este montaje consistió en un proyector multimedia, una cámara digital y una pantalla de proyección. El Capítulo V contiene la descripción correspondiente.

Finalmente, se realizó una comparación y evaluación final de los datos simulados y los datos experimentales obtenidos, para poder determinar la viabilidad del algoritmo propuesto mediante un cálculo y análisis de error. Esto se muestra a detalle en el Capítulo VI.

CAPÍTULO IV

SOFTWARE DE SIMULACIÓN Y CAPTURA

Para realizar pruebas dentro de un ambiente controlado, originalmente se implementó un software de simulación de franjas para realizar las simulaciones con el modelo matemático propuesto. Este software puede generar imágenes en condiciones ideales, con adición de ruido y con adición de distorsión gamma. Con el fin de obtener datos experimentales se agregó al software la capacidad de proyectar y capturar los patrones generados, adicionando además la capacidad de corregir errores por pixeles dañados en la cámara. El código fuente de este software se incluye en el Apéndice 1.

El software completo está dividido en las siguientes etapas:

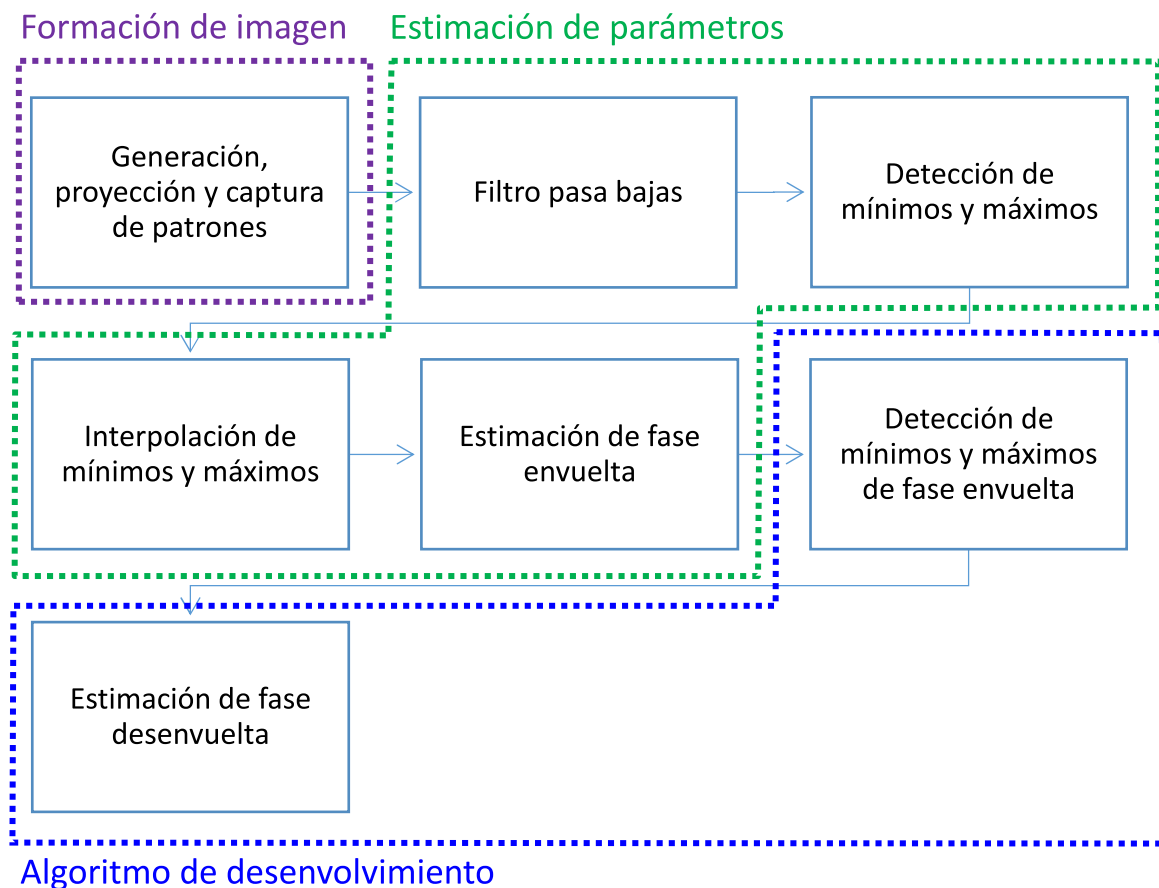


Figura 4.1. Diagrama de flujo de algoritmo propuesto.

4.1 Generación, proyección y captura de patrones

En la etapa de generación de patrones, se implementó un software capaz de generar n patrones sinusoidales con un desfase constante θ entre ellos. Estos patrones se generan como imágenes en escala de grises de 8 bits y el tamaño de las imágenes generadas es ajustable, en las pruebas de simulación la resolución utilizada fue de 1280 x 960 píxeles, este valor fue seleccionado por ser una resolución cercana a la de la cámara que se utilizará en la fase experimental.

Inicialmente, la expresión matemática utilizada para generar estos patrones es:

$$I(x, y) = a(x, y) + b(x, y) \cos[\phi(x, y) + \theta] \quad (4.1)$$

En la Figura 4.2 se muestra una vista de la sección transversal, en ella se observan los términos de intensidad promedio $a(x, y)$, modulación de amplitud $b(x, y)$ y desfase θ .

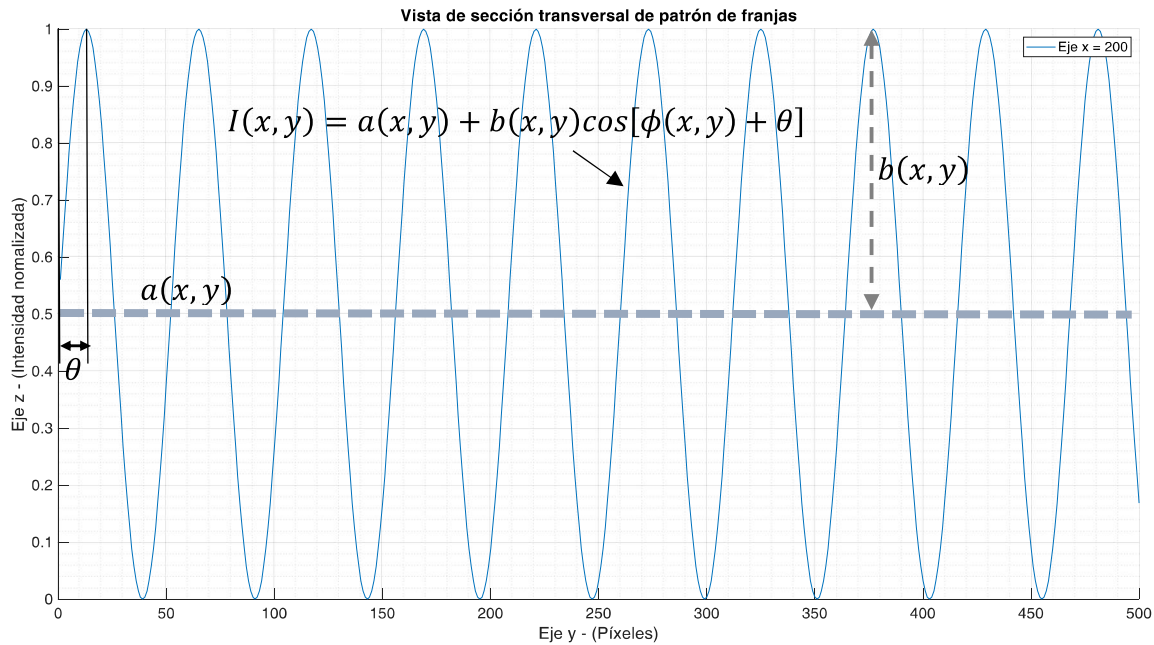


Figura 4.2. Corte transversal de un patrón de franjas.

Las imágenes generadas a partir de la ecuación (4.1) son patrones ideales. Para realizar una mejor evaluación del algoritmo, el software adiciona ruido al patrón de

franjas. Se seleccionó ruido blanco Gaussiano aditivo (AWGN) y es agregado como muestra la siguiente ecuación:

$$I(x, y) = a(x, y) + b(x, y) \cos[\phi(x, y) + \theta] + \eta(x, y) \quad (4.2)$$

Donde η es el ruido. En la Figura 4.3 se observa un patrón de franjas generado con la ecuación (4.2).

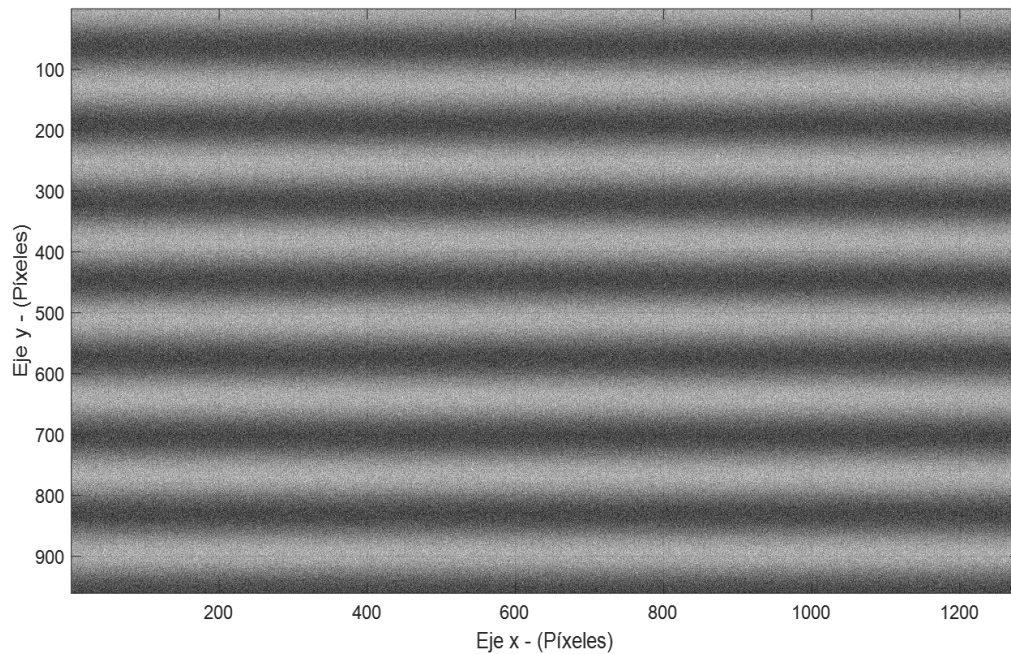


Figura 4.3. Patrón de franjas sinusoidales con AWGN.

4.1.1 Corrección de factor gamma

Otro parámetro que se incluye en la generación de imágenes sintéticas es el factor gamma γ , como se mencionó anteriormente, debido a la limitante que tienen los proyectores multimedia para proyectar color negro, en los patrones sinusoidales proyectados, se puede observar una acentuación de sus valles. En la Figura 4.4, se muestra este fenómeno, en una captura real sin corrección gamma.

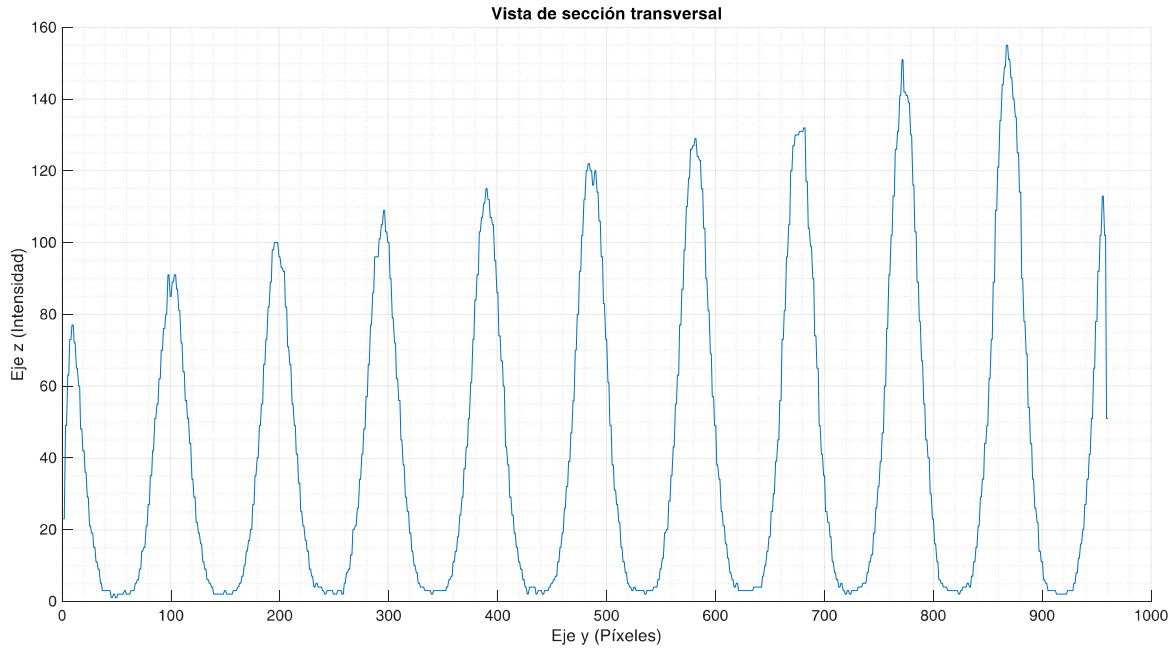


Figura 4.4. Vista de sección transversal de patrón de franjas sin corrección gamma.

De acuerdo a lo mencionado, la ecuación (2.69) modela el efecto del factor gamma sobre una imagen. Con fines demostrativos, se realizó una simulación para observar la adición del factor gamma a un patrón sinusoidal, como se muestra en la Figura 4.5, en esta se observan dos vistas de la sección transversal, que corresponden a diferentes simulaciones de factor gamma, en color azul se asigna $\gamma = 1$, lo que corresponde a una no deformación y en color naranja se asigna $\gamma = 3$, donde se hace observable la deformación de los valles, el efecto es similar al mostrado en la Figura 4.4, concluyendo que es correcto el modelo para este factor.

Para corregir la deformación en los patrones, se decidió generar patrones de acuerdo a lo mencionado por Zhaoyang Wang [15]. Se incluyó esto en la ecuación 4.2 y dando como resultado la siguiente ecuación:

$$I(x, y) = 2I_0 \left[\frac{1}{2} + \frac{1}{2} \cos \left(2\pi k \frac{x}{w} + \theta \right) \right]^{1/\gamma} + \eta(x, y) \quad (4.3)$$

A continuación, todos los patrones sinusoidales que se mencionen en este documento serán generados con la ecuación (4.3).

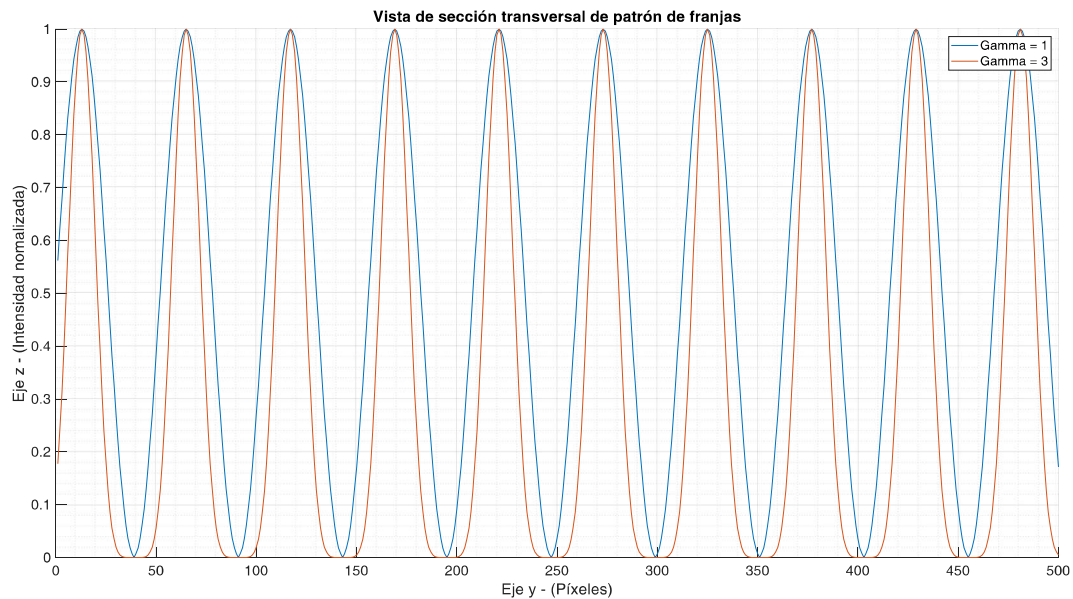


Figura 4.5. Vista de sección transversal de patrones con diferentes valores de factor gamma.

En la literatura se recomiendan valores de corrección para el factor gamma entre $2.2 \leq \gamma \leq 2.6$, la selección de este valor dependerá del proyector que se esté utilizando. En el presente proyecto de investigación, se creó un algoritmo para determinar el valor correcto de gamma asociado al proyector.

El algoritmo funciona de la siguiente manera:

- i. Proyección y captura de patrones sinusoidales de franjas con diferentes valores de gamma.
- ii. Determinación de la amplitud y frecuencia de la frecuencia fundamental.
- iii. Generación de un patrón sinusoidal de referencia con la frecuencia y amplitud determinada.
- iv. Extracción, una a una, de cada columna de cada patrón sinusoidal y se calcula el coeficiente de correlación con respecto de la señal generada en el punto anterior. El cálculo de la correlación se realiza n veces por columna,

esto con el propósito de desplazar el patrón de referencia y obtener diferentes valores de correlación por columna.

- v. Análisis de los coeficientes de correlación de cada patrón para obtener el valor mínimo, máximo, promedio y RMS.

En la Tabla 4.1, se muestra el análisis realizado a patrones capturados con diferentes valores de factor gamma, utilizando el algoritmo descrito anteriormente.

Tabla 4.1. Análisis de coeficientes de correlación para diferentes valores de gamma.

Factor gamma γ \ Coeficiente de correlación	Valor mínimo	Valor máximo	Valor promedio	Valor RMS
1	-0.5011	0.7596	0.3631	0.4845
2	-0.5111	0.7746	0.3657	0.4884
2.1	-0.5109	0.7769	0.3662	0.4892
2.2	-0.5159	0.7767	0.3612	0.4851
2.3	-0.5169	0.7761	0.3585	0.4830
2.4	-0.5174	0.7750	0.3600	0.4851
2.5	-0.5207	0.7782	0.3588	0.4845
2.6	-0.5228	0.7765	0.3570	0.4834
2.7	-0.5189	0.7758	0.3556	0.4812
2.8	-0.5193	0.7750	0.3544	0.4792

Con los datos obtenidos, se seleccionó el valor gamma de 2.1, debido a que tiene el valor RMS más alto, indicando que el patrón proyectado con ese valor gamma, se asemeja más al patrón sinusoidal de referencia generado por el algoritmo.

En la Figura 4.6, se muestra una vista de la sección transversal de dos capturas de patrones, la de color azul corresponde a una captura sin corrección gamma ($\gamma = 1$) y

la de color naranja cuenta con una corrección gama ($\gamma = 2.1$), comparándolas se observa que el patrón corregido se parece más a una onda sinusoidal.

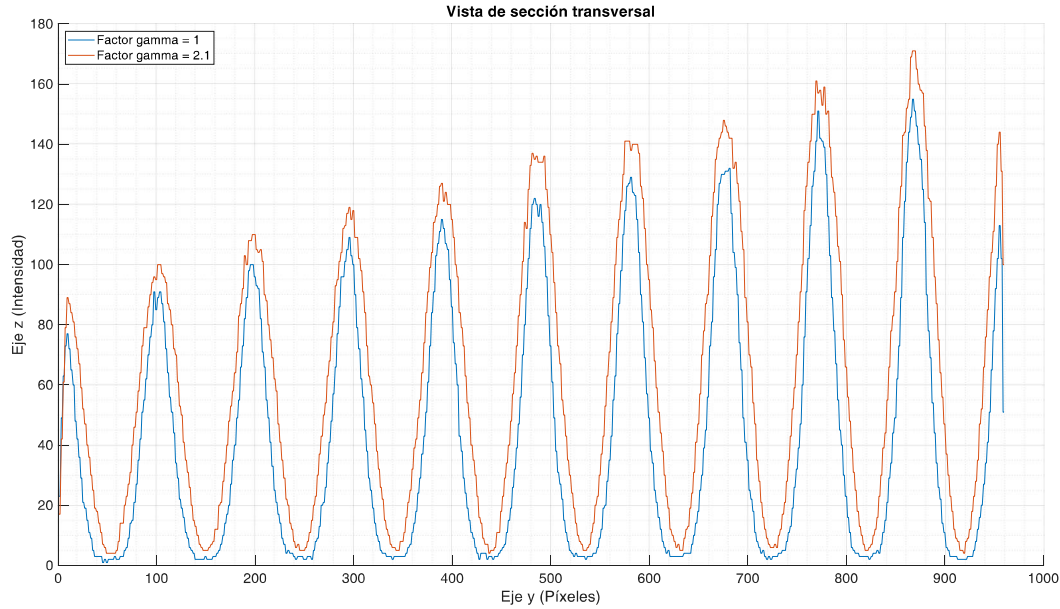


Figura 4.6. Vista de sección transversal de capturas con diferentes valores de gamma.

4.1.2 Preprocesamiento para píxeles dañados

Ya con el patrón capturado y corregido del factor gamma, se realiza un preprocesamiento para corregir los píxeles dañados en la cámara.

Para esta actividad se define el siguiente algoritmo:

- i. Captura de imagen con la lente ocluida $I(x, y)$. La Figura 4.7 muestra algunos de los píxeles dañados de la cámara.
- ii. Cálculo del histograma de la captura con la lente ocluida, Figura 4.8.
- iii. Definición de valor umbral y detección de píxeles dañados. Para seleccionar el valor de umbral se utiliza la función de umbralizado automático *Clustering* de NI Vision Assistant®. En la Figura 4.8, se observa el histograma obtenido.

if $I(x,y) > \text{valor umbral}$

Píxel dañado = (x,y)

iv. Preprocesamiento de la imagen.

a. Adquisición de imagen $I_p(x,y)$

b. Promediado de píxel dañado

$$I_{c(x,y)} = \left\lfloor \frac{I_p(x, y-1) + I_p(x, y+1)}{2} \right\rfloor$$

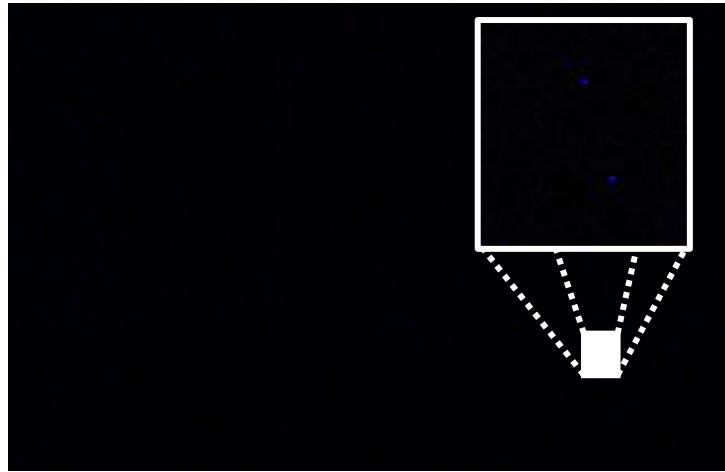


Figura 4.7. Píxeles dañados en la cámara.

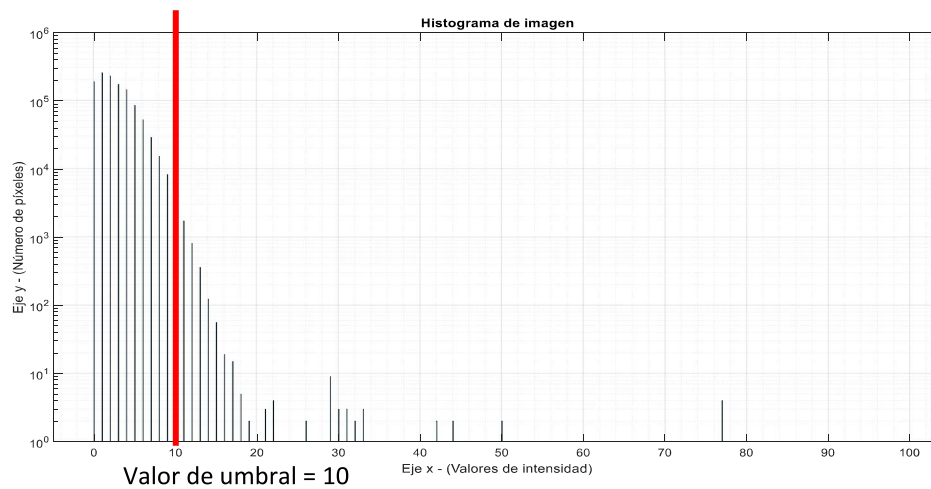


Figura 4.8. Histograma de captura con la lente ocluida.

4.2 Filtro pasa bajas

Como se puede observar en la Figura 4.9, en la región de los semiciclos positivos, aparece ruido adicionado, lo cual hizo necesario aplicar un filtro pasa bajas.

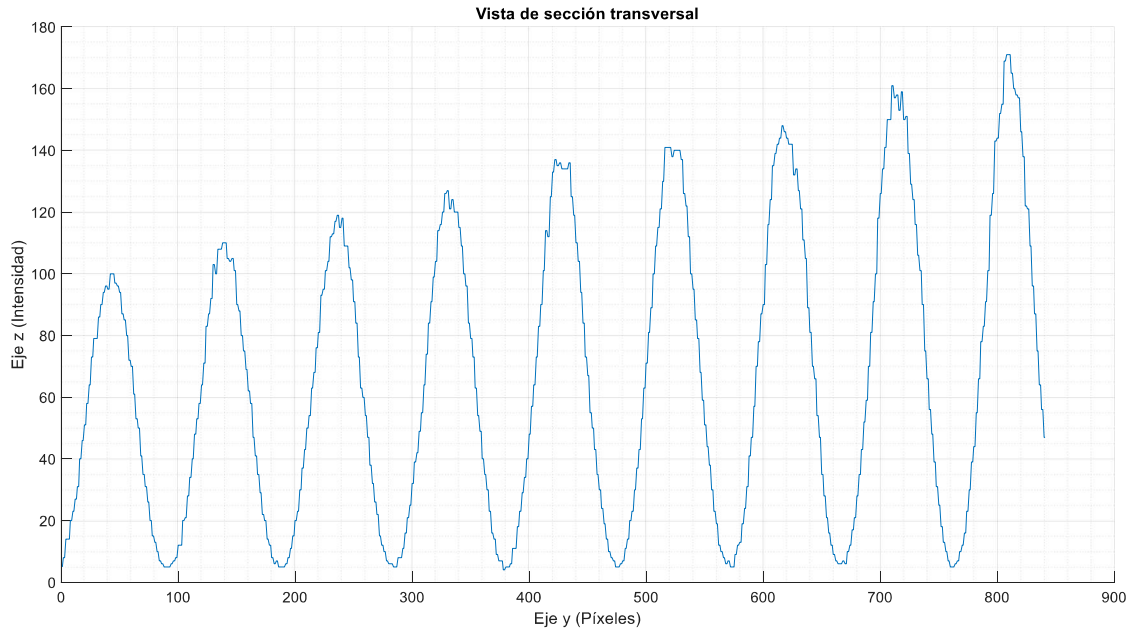


Figura 4.9. Vista de sección transversal de patrón de franjas antes de la aplicación del filtro pasa bajas.

La Figura 4.10, muestra el patrón sinusoidal después de aplicar el filtrado en donde los valores de intensidad son normalizados.

Los parámetros del filtro utilizado son los siguientes:

- i. El periodo del patrón sinusoidal generado es $T = 96$ píxeles o una frecuencia $f = \frac{1}{96} \approx 0.0104 \left[\frac{\text{ciclos}}{\text{píxel}} \right]$. Debido a que la relación entre el área de proyección y captura es aproximadamente 1:1, los valores de frecuencia y periodo son similares en la captura.
- ii. La frecuencia de muestreo $f_s = 1000f = 10.4 \frac{\text{ciclos}}{\text{píxel}}$.

- iii. La frecuencia de corte $f_c = 5 \frac{\text{ciclos}}{\text{píxel}}$.
- iv. El tipo de filtro es seleccionado por la función *lowpass* de MATLAB®, esta función calcula un filtro de un orden mínimo con una atenuación de banda de parada de 60 dB. La función diseña un filtro FIR de orden mínimo si la señal de entrada es suficientemente larga, y un filtro IIR de orden mínimo en caso contrario. En concreto, la función sigue estos pasos:

Calcula el orden mínimo que debe tener un filtro FIR para cumplir las especificaciones. Si la señal es al menos el doble de larga que el orden de filtro requerido, diseña y utiliza ese filtro.

Si la señal no es lo suficientemente larga, calcula el orden mínimo que debe tener un filtro IIR para cumplir las especificaciones. Si la señal es al menos tres veces más larga que el orden de filtro requerido, diseña y utiliza ese filtro.

Si la señal no es lo suficientemente larga, trunca el orden a un tercio de la longitud de la señal y diseña un filtro IIR de ese orden. La reducción del orden se produce a expensas de la inclinación de la banda de transición. Filtra la señal y compensa el retardo.

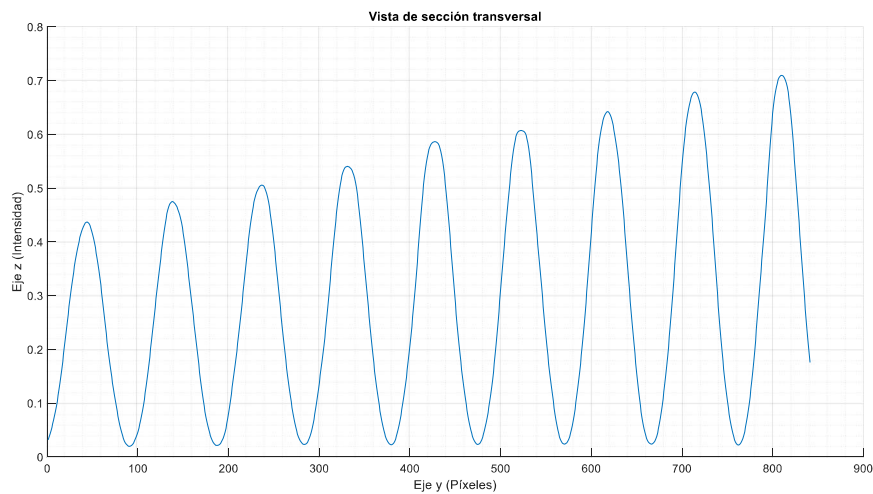


Figura 4.10. Vista de sección del patrón de franjas filtrado y normalizado.

4.3 Detección de máximos y mínimos

Como se planteó en el modelo matemático, es necesario conocer los parámetros de intensidad promedio y modulación de amplitud. Para esto se utiliza un algoritmo de detección de mínimos y máximos. La detección de los mínimos y máximos se muestra en la Figura 4.11.

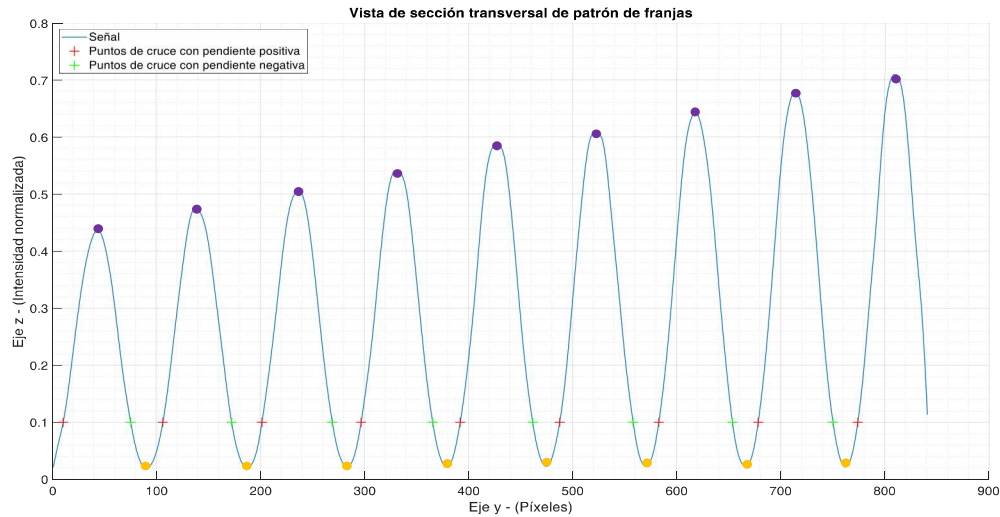


Figura 4.11. Detección de máximos y mínimos.

El algoritmo para la detección de máximos y mínimos funciona de la siguiente manera:

- i. Se extrae una columna de la imagen.
- ii. Figura 4.11, el valor de umbral es de 0.1.
- iii. Los valores de intensidad iguales al valor de umbral son detectados y se les asigna un marcador rojo o verde, dependiendo si la pendiente es positiva o negativa.
- iv. Se generan parejas de marcadores rojos o verdes contiguos, o viceversa. Revisando los índices de la pareja de marcadores, si el índice del marcador rojo es menor al del marcador verde, entre esa pareja se busca un máximo,

en el caso contrario, se busca un mínimo. Esta operación es recursiva para todas las parejas de la columna bajo análisis.

- v. Se crea un arreglo con cada valor mínimo o máximo y su respectivo índice.

4.4 Interpolación de máximos y mínimos

En los valores máximos obtenidos se observa que a mayor índice del píxel, mayor es el valor del máximo, este comportamiento es debido a que esos píxeles se encuentran próximos al eje óptico del proyector donde existe una mayor intensidad. Debido a esto se hace necesario la necesidad de interpolar los valores de máximos y mínimos. Se utiliza la interpolación lineal, evitando la aparición de sobre impulsos. Una vez que estos valores son interpolados se calcula un valor promedio entre cada máximo y mínimo para cada píxel.

En la Figura 4.12, se observan la interpolación de los valores máximos, mínimos y cálculo del valor promedio. Para el valor promedio, se lleva a cabo su cálculo solo cuando exista valor de máximo y mínimo para ese píxel, con esto se descartan los extremos del patrón sinusoidal.

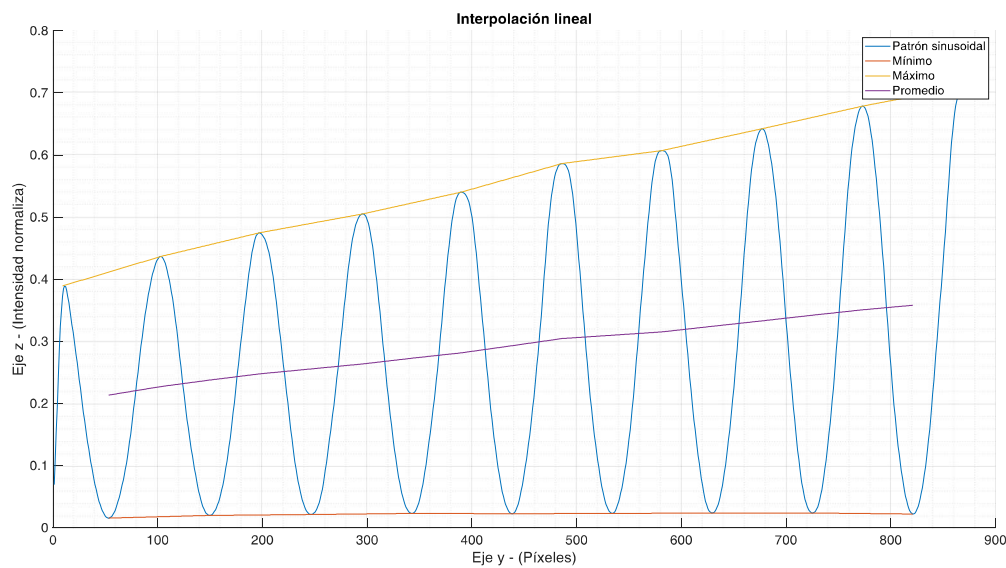


Figura 4.12. Interpolación de máximos, mínimos y cálculo de promedio.

4.5 Estimación de fase envuelta

En esta sección, se calcula el término $\varphi(x, y)$ de la ecuación (3.1). Para llevar a cabo esta tarea se necesita conocer tres incógnitas, la cuales son:

- i. Intensidad de píxel $I(x, y)$, es la intensidad del patrón capturado para un cierto píxel, la cual es obtenida automáticamente con la captura.
- ii. Intensidad promedio $a(x, y)$, esta es desconocida, pero con los valores interpolados en la sección anterior, es estimada, siendo igual al valor promedio.
- iii. Modulación de amplitud $b(x, y)$, también es desconocida y se calcula cómo la diferencia entre el valor máximo y promedio.

Con los parámetros de intensidad promedio y modulación de amplitud estimados, se calcula la fase, como se muestra en la Figura 4.13, la fase se encuentra envuelta.

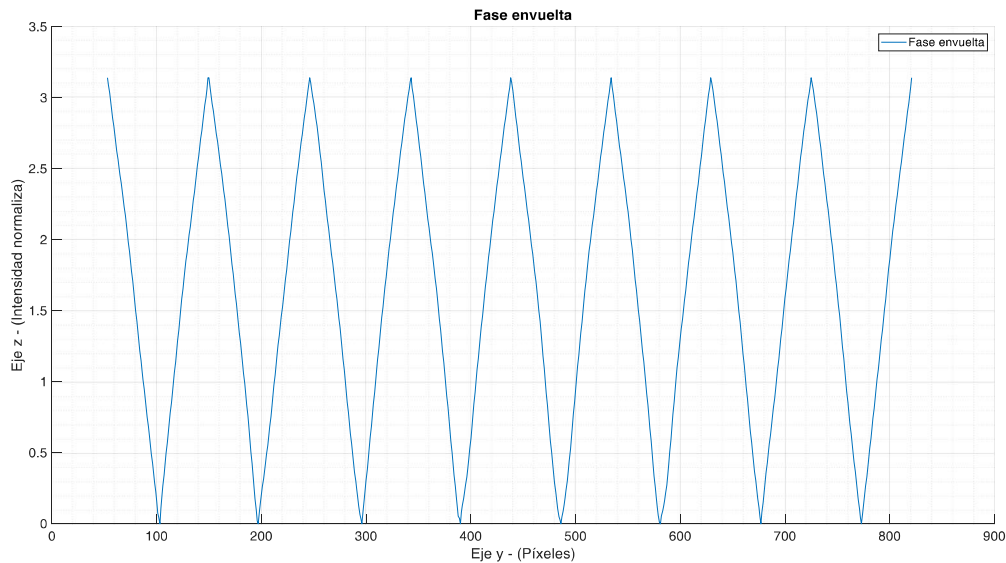


Figura 4.13. Fase envuelta.

4.6 Algoritmo de desenvolvimiento de fase

Como se puede observar en la Figura 4.13, la fase presenta discontinuidades, esto debido a que se obtiene a partir de una función coseno inverso. Para resolver este problema, es necesario la creación e implementación de un algoritmo de desenvolvimiento para una función coseno inverso, para generar una fase desenvuelta o continua.

Para la implementación de este algoritmo es necesario conocer el dominio $[-1 \leq x \leq 1]$ y rango $[0 \leq y \leq \pi]$ de la función coseno inverso.

Para hacer más clara la implementación del algoritmo de desenvolvimiento se dividirá en dos secciones:

4.6.1 Detección de máximos y mínimos

Para la señal de fase envuelta, se reutiliza el algoritmo de detección de máximos y mínimos, debido a que la fase envuelta es similar a una señal triangular. En la Figura 4.14, se muestra la detección de máximos y mínimos en la señal de fase envuelta.

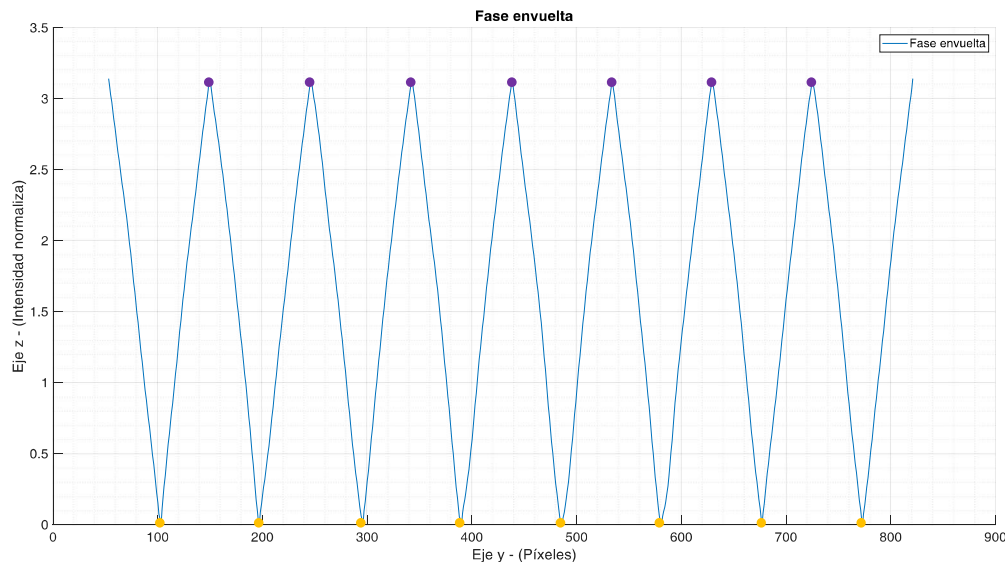


Figura 4.14. Detección de máximos y mínimos en la señal de fase envuelta.

4.6.2 Estimación de fase desenvuelta

Una vez que los mínimos y máximos son identificados, se aplica el algoritmo de desenvolvimiento, los pasos del algoritmo son los siguientes:

- i. Se crea un arreglo de la misma longitud que el arreglo de fase envuelta y se inicializa el apuntador $t = 0$.
- ii. Se recorre la señal de fase envuelta y con la siguiente lógica se calculan los elementos del arreglo de fase desenvuelta:
 - Si se detecta un máximo, para cada elemento anterior, se realizan las siguientes operaciones:

- Se calcula la fase desenvuelta:

$$\phi(x, y)_{desenvuelta} = \phi(x, y)_{envuelta} + 2\pi t$$

- Se incrementa el apuntador:

$$t = t + 1$$

- Si se detecta un mínimo, para cada elemento anterior, solo se calcula la fase desenvuelta:

$$\phi(x, y)_{desenvuelta} = 2\pi t - \phi(x, y)_{envuelta}$$

En la Figura 4.15, se muestran los intervalos en los que el apuntador t se va incrementando, además de mostrar la señal de fase envuelta y desenvuelta.

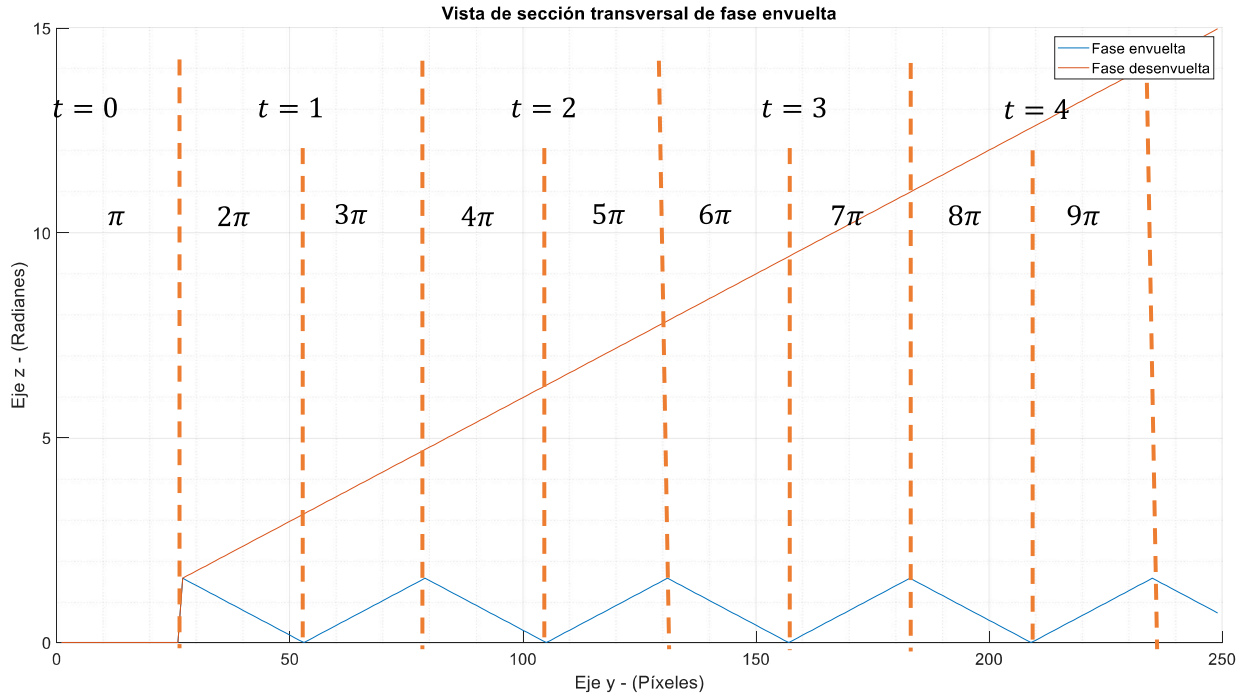


Figura 4.15. Vista de sección transversal de fase envuelta y desenvuelta.

En la Figura 4.16, se muestra la vista de la sección transversal de la fase envuelta y desenvuelta del objeto, se observa como el algoritmo planteado entrega buenos resultados y soluciona el inconveniente de la función coseno inverso.

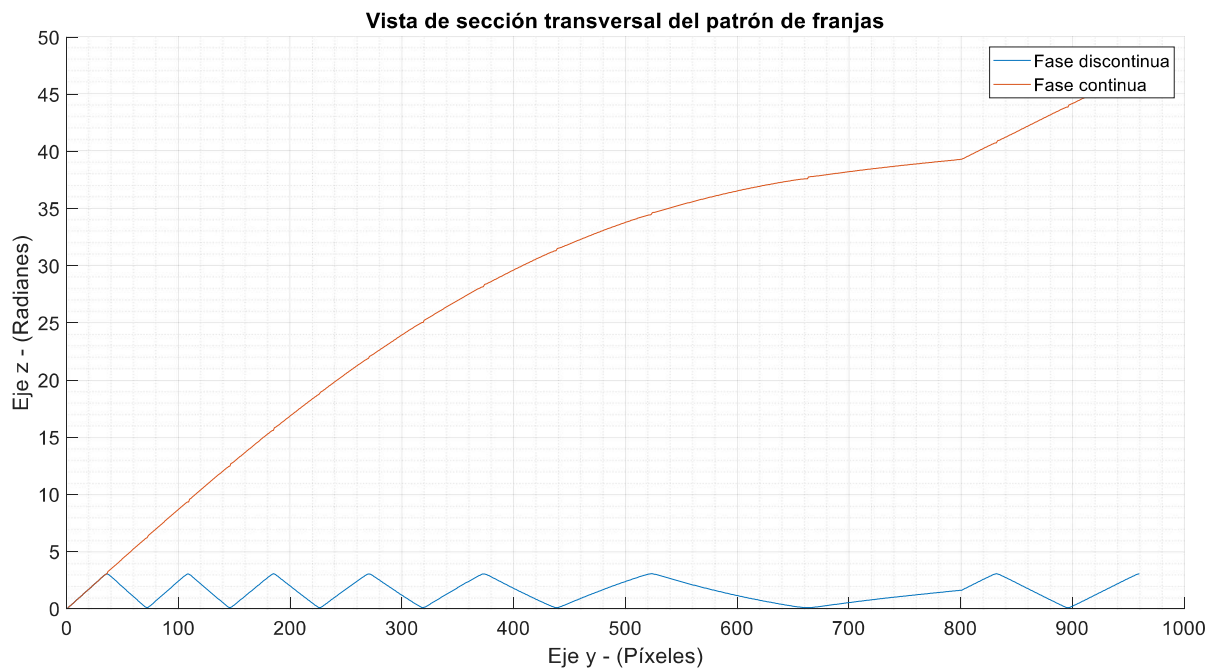


Figura 4.16. Vista de sección transversal de fase envuelta y desenvuelta del objeto.

En la Figura 4.17, se muestra una reconstrucción 3D del objeto después de ser aplicado el algoritmo planteado a todo el patrón de franjas.

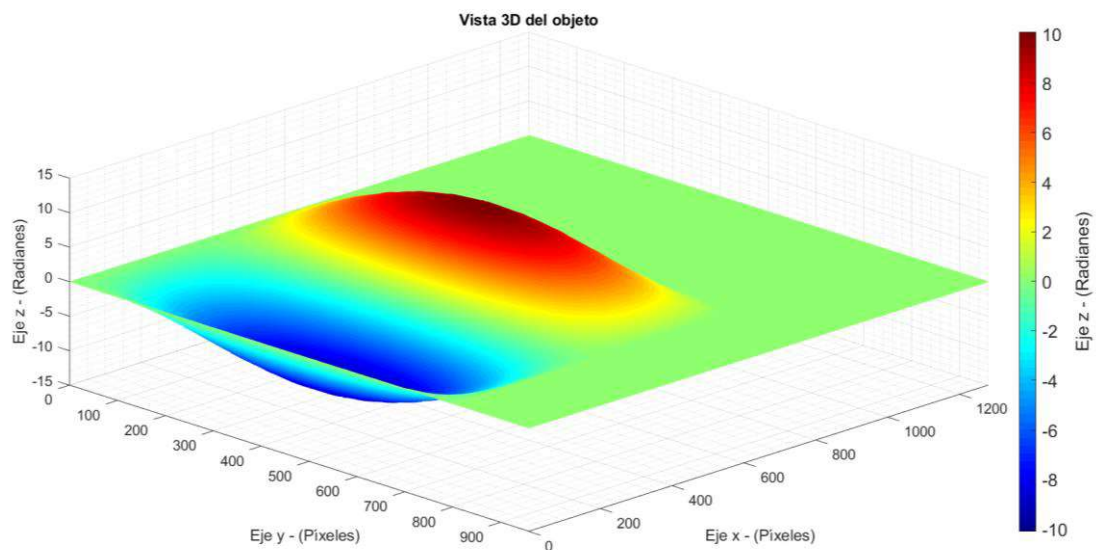


Figura 4.17. Reconstrucción 3D del objeto simulado.

La Figura 4.18, muestra una vista de sección transversal de la fase envuelta y desenvuelta en donde se agregó ruido, se observa como el algoritmo de desenvolvimiento es capaz de desenvolver correctamente la fase envuelta. A continuación, en la Figura 4.19, se muestra la reconstrucción 3D del objeto simulado con ruido agregado, se observan algunas imperfecciones sobre la superficie. En el siguiente capítulo se cuantificará el error asociado a este nuevo método de reconstrucción de objetos.

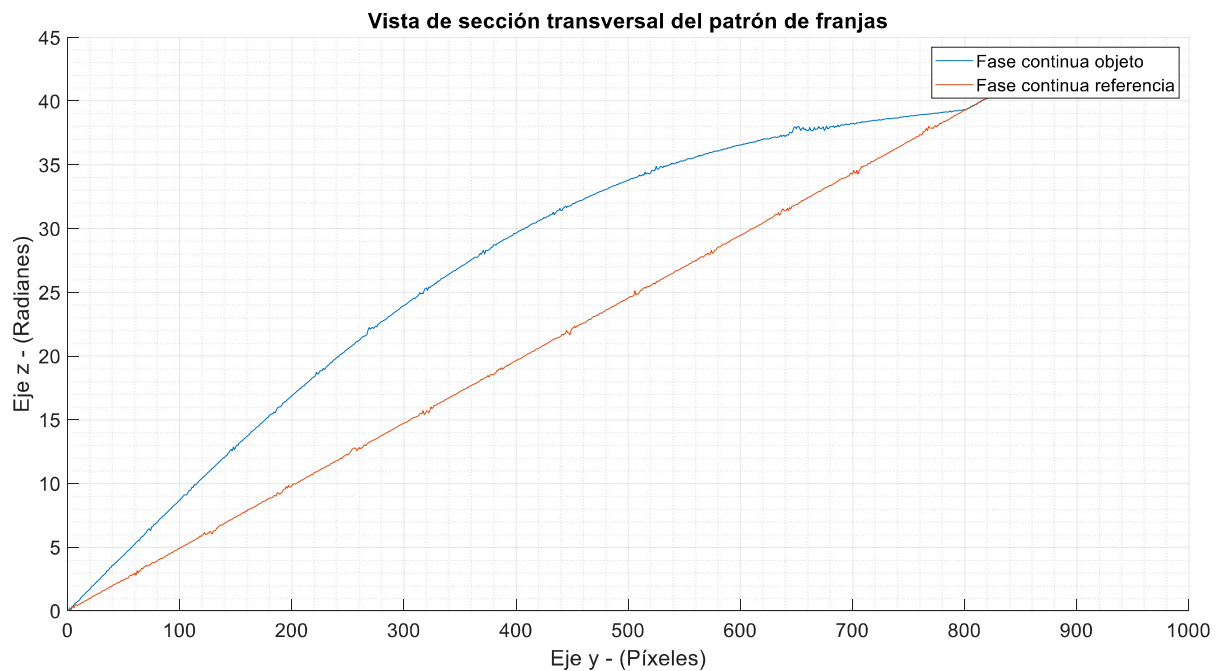


Figura 4.18. Vista de sección transversal de fase envuelta y desenvuelta con la adición de ruido.

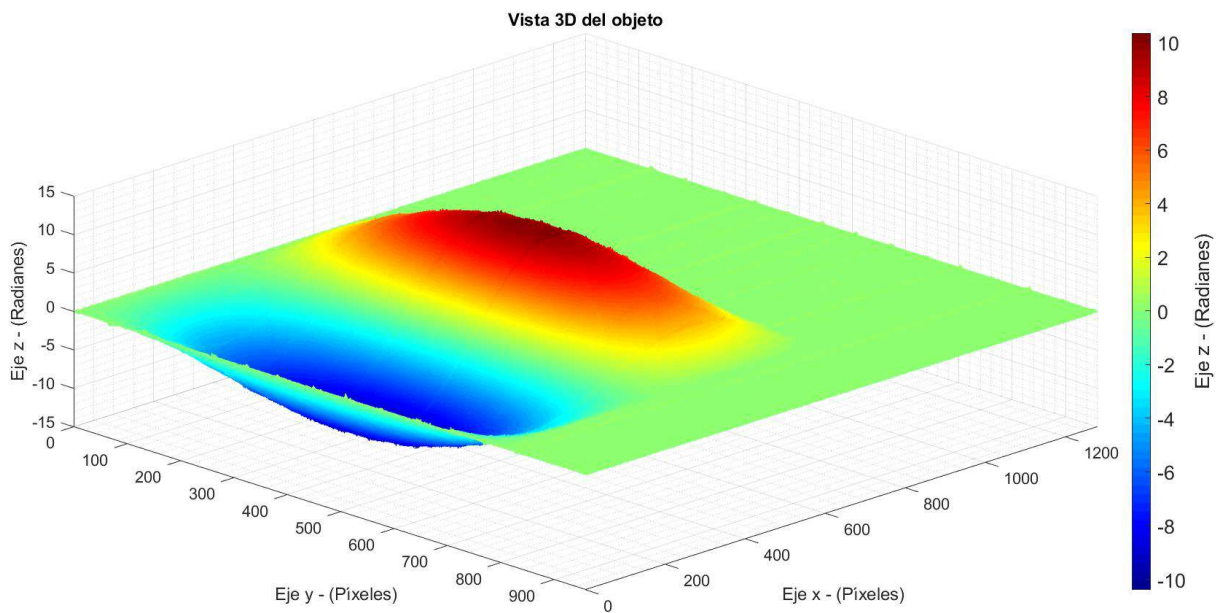


Figura 4.19. Reconstrucción 3D del objeto simulado con ruido.

CAPÍTULO V

MONTAJE EXPERIMENTAL

Como se mencionó anteriormente, con el objetivo de obtener datos experimentales para validar el algoritmo propuesto y caracterizar su desempeño ante condiciones reales, fue necesario desarrollar un montaje de laboratorio. En este capítulo se describe el montaje experimental implementado con este fin.

5.1 Descripción general

Se utilizó un montaje de laboratorio previamente desarrollado [78], el cual consiste en un sistema de luz estructurada para proyección de franjas formado por un proyector multimedia, una cámara digital, una lente formadora de imagen y una pantalla de proyección usada como plano de referencia.

Estos dos componentes son posicionados sobre una mesa óptica como se muestra en la Figura 5.1, los ejes ópticos del proyector y la cámara son paralelos y perpendiculares al plano de referencia, con esas distancias se logra, que el área de proyección sea igual al área de captura.

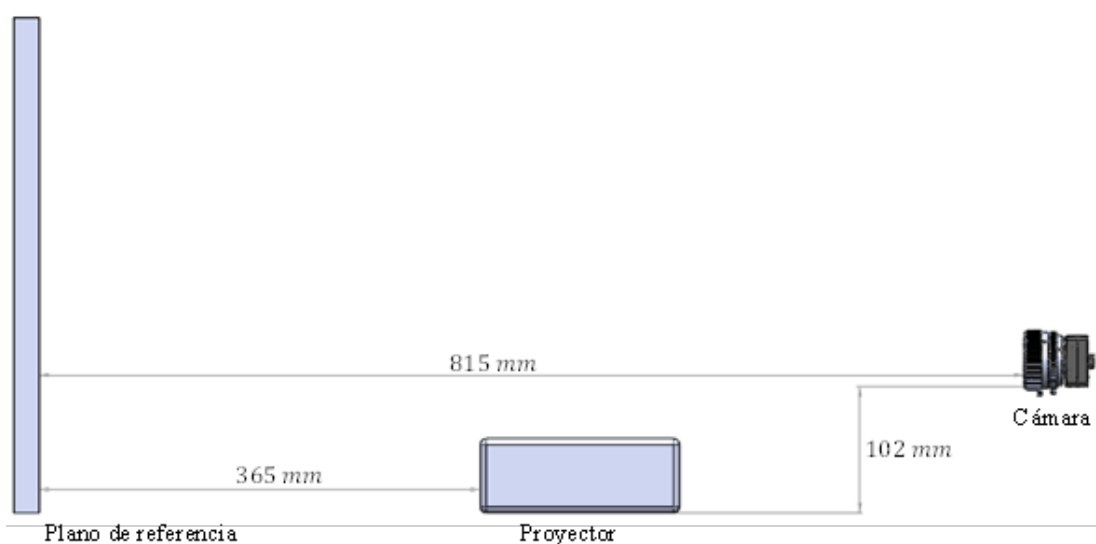


Figura 5.1. Vista lateral del montaje experimental.

5.2 Características técnicas de los componentes

En esta sección se presenta solo una breve descripción de las características técnicas de cada uno de los componentes utilizados, las hojas de datos completas se encuentran en los Anexos 1 y 2.

El sistema de proyección utilizado consiste en un proyector ACER modelo K335 LED, el cual muestra en la Figura 5.2 y cuenta con las siguientes especificaciones técnicas:

- Resolución nativa 1280 x 800 píxeles y resolución máxima 1920 x 1080 píxeles.
- Tecnología DLP.
- Dimensiones: largo 165.1 mm x ancho 231.14 mm x altura 45.72 mm.
- Lente de proyección con enfoque manual de $f = 1.5$ mm a $f = 13.92$ mm.
- Distancia mínima de proyección 899.9 mm.
- Lámpara LED.
- Puertos USB, HDMI, VGA.



Figura 5.2. Proyector ACER K335.

El sistema de adquisición está formado por una cámara y una lente. La cámara utilizada es una cámara digital marca POINT GREY número de parte CMLN-13S2C-CS, ver Figura 5.3 y cuenta con las siguientes especificaciones técnicas:

- Resolución de 1296 x 964 píxeles a 18 cuadros por segundo.
- Sensor Sony ICX445 CCD.
- Tamaño de píxel 3.75 μm .
- Dimensiones: largo 44 mm x ancho 25.5 mm x altura 47 mm.
- Formato de sensor 1/3".
- Montura para lente CS.
- Conexión USB 2.0.
- Captura imágenes en color.



Figura 5.3. Cámara POINT GREY CMLN-13S2C-CS.

La lente utilizada para formar la imagen es una lente marca Edmund Optics tipo doble gauss con número de parte 54857, como la mostrada en la Figura 5.4, la cual cuenta con las siguientes especificaciones técnicas:

- Longitud focal 18 mm.
- Montura tipo C.

- Distancia de trabajo de 210 mm - ∞ .



Figura 5.4. Lente doble gauss 54857.

El plano de referencia consistió en una placa de vidrio plana de 355 mm de ancho por 280 mm de alto recubierta con una película de PVC blanda ORACAL 631, que le proporcionó un acabado blanco mate a la superficie de la lámina, esto para asegurar un plano de referencia de color uniforme y reducir reflexiones indeseadas que pudieran afectar en la etapa de adquisición de imágenes.

5.3 Alineación de componentes

Una etapa fundamental cuando se implementan montajes experimentales de luz estructurada es la correcta alineación de los componentes, en el caso específico del montaje propuesto en esta tesis, la cámara y el proyector deben estar alineados correctamente para evitar errores de perspectiva.

Con el fin de garantizar una correcta alineación se desarrolló y siguió la siguiente metodología.

- i. La cámara con su lente y el proyector deben estar ubicados de acuerdo al esquema mostrado en la Figura 5.1, respetando estrictamente las distancias indicadas.
- ii. Se coloca un vástago frente a la cámara, como se muestra en la Figura 5.5.

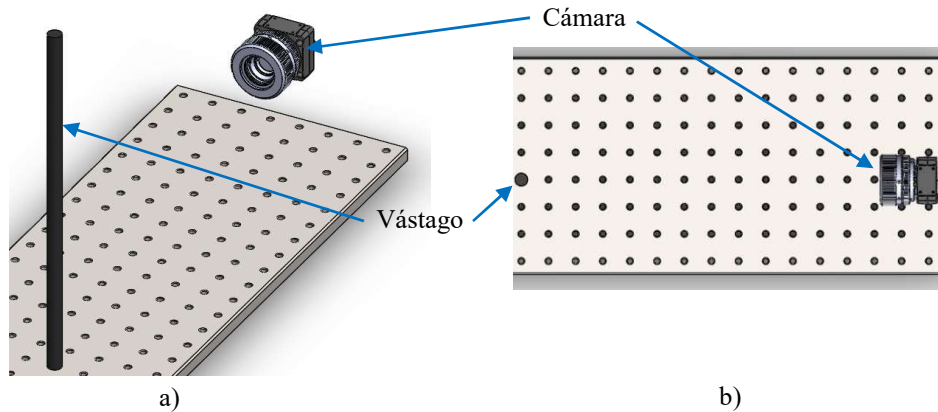


Figura 5.5. Ubicación del vástago: a) Vista isométrica; b) Vista superior.

- iii. Se realiza una captura, la cual se segmenta para separar el vástago del fondo y se superpone a la captura una línea horizontal y otra vertical formando una cruz al centro de la imagen capturada, ver Figura 5.7. Lo anterior a través de la interfaz denominada Captura de Cruz desarrollada en LabVIEW®, ver Apéndice 2. La finalidad es que la línea vertical superpuesta se encuentre al centro del vástago, para lo cual es necesario girar la cámara y con ello eliminar el error de perspectiva.

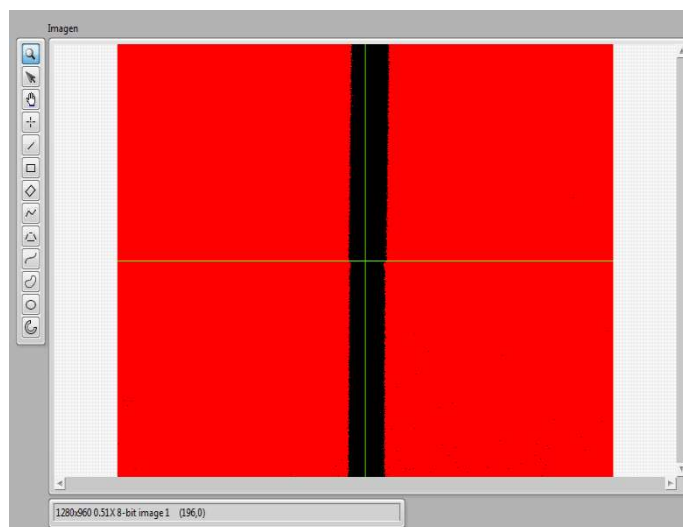


Figura 5.6. Interfaz Captura de Cruz, se alinea la línea verde con el vástago.

- iv. Retirar el vástago y colocar el plano de referencia. Con la interfaz Proyección de Cruz, ver Apéndice 3, se proyecta un patrón de una cruz formado por una línea horizontal y otra vertical al centro de la proyección, después a través de la interfaz Captura de Cruz, se hace coincidir la cruz proyectada con la capturada, ver Figura 5.7. Para lograrlo solo debe de moverse el proyector y si fuera necesario, mover la cámara hacia arriba o abajo para hacer coincidir la línea horizontal.

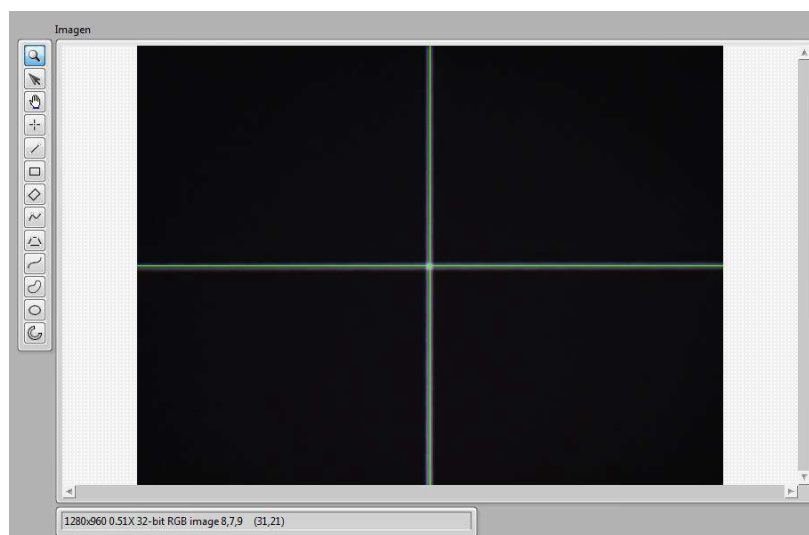


Figura 5.7. Interfaz Captura de Cruz, se hace coincidir la cruz proyectada con la generada.

- v. Los movimientos en los que se giran y desplazan la cámara y el proyector, son solo ajustes finos debido a que los elementos se encuentran posicionados de acuerdo a lo mostrado en la Figura 5.1.

Después de que los componentes han sido alineados correctamente se procede a la captura de imágenes para probar el algoritmo bajo condiciones reales.

CAPÍTULO VI

ANÁLISIS DE DATOS

Con el fin de garantizar que el proceso de medición no introduce errores significativos a la medición, se realizó un análisis estadístico de los datos experimentales obtenidos.

6.1 Validación del proceso de medición

La validación de los datos obtenidos puede ser realizada a través de métodos cualitativos y/o cuantitativos, para de esta manera interpretar las mediciones obtenidas. Para un instrumento de medición, los resultados obtenidos deben de presentar una distribución normal o gaussiana.

Se debe definir una metodología a seguir en el proceso de medición para asegurar que todas las mediciones fueron realizadas bajo las mismas condiciones. El proceso de medición consiste en realizar mediciones siguiendo una serie de pasos e instrucciones previamente definidos. Estas instrucciones suponen que, si el proceso es repetido una y otra vez, las condiciones bajo las que son realizadas las mediciones son idénticas. La información generada es útil para estimar parámetros importantes del proceso de medición, así como el error asociado.

La metodología a seguir fue la siguiente:

- i. Encender el proyector.
- ii. Conectar la cámara.
- iii. Abrir el programa de proyección y captura.
- iv. Apagar la iluminación del laboratorio.
- v. Ejecutar el programa hasta obtener la medición.
- vi. Encender la iluminación.

- vii. Cerrar el programa de proyección y captura.
- viii. Desconectar la cámara.
- ix. Apagar el proyector.

Este proceso se repitió 30 veces, para que sea posible determinar si la distribución es normal.

El patrón de referencia a reconstruir, fue un objeto modulado en los patrones de franjas proyectados, esto por dos importantes razones:

- i. Evitar problemas de discontinuidades, por ejemplo, oclusiones o cambios en la reflectancia generadas por un objeto real.
- ii. Eliminar errores de conversión a unidades del mundo real. El objeto modelado por la ecuación (4.4), tiene unidades de radianes y el mapa de fase envuelta entrega resultados en radianes, evitando el agregar errores de conversión de unidades.

6.1.1 Prueba de bondad de ajuste chi-cuadrada χ^2

La prueba de bondad de ajuste es un método estadístico cuantitativo, el cual se basa en qué tan buen ajuste se tiene entre la frecuencia de ocurrencia de las observaciones de una muestra observada y las frecuencias esperadas que se obtienen de una distribución hipotética [79][80]. Se calcula con la ecuación (6.1):

$$\chi^2 = \sum_{i=1}^k \frac{(o_i - e_i)^2}{e_i} \quad (6.1)$$

Donde los términos o_i y e_i representan las frecuencias observada y esperada, respectivamente y el término k los grupos o clases.

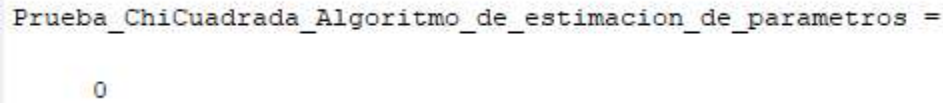
Si la diferencia entre las frecuencias observadas y las frecuencias esperadas es pequeña, el valor de χ^2 será pequeño, lo que indica un buen ajuste. En caso contrario, si la diferencia entre las frecuencias observadas y las frecuencias esperadas es grande, el valor de χ^2 será grande, indicando que el ajuste es pobre. Un buen ajuste conduce a la aceptación de la hipótesis nula, mientras que un ajuste pobre conduce al rechazo, para un cierto nivel de significancia α que define el criterio de decisión a utilizar.

La Tabla 6.1 muestra las 30 mediciones realizadas al patrón de referencia con la técnica de proyección de franjas y proyección de borde, respectivamente. Para validar las mediciones obtenidas utilizando la prueba de bondad de ajuste chi-cuadrada χ^2 , se generó un archivo de texto con las mediciones de altura, las cuales son analizadas con el código fuente que se muestra en el Apéndice 4, la función `chi2gof` de MATLAB®. Cuando el resultado obtenido por la función es igual a 1 la prueba rechaza la hipótesis nula y es cero; todo esto para un nivel de significancia menor al 5%.

Tabla 6.1. Mediciones obtenidas del patrón de referencia.

Número de medición	Altura (Radianes)	Número de medición	Altura (Radianes)
1	4.8382	16	4.9408
2	5.0048	17	4.9494
3	4.9956	18	4.9618
4	5.0005	19	4.9910
5	5.0025	20	5.0139
6	4.9915	21	5.0081
7	4.9809	22	4.9500
8	5.0202	23	4.9184
9	5.0149	24	4.9400
10	4.9753	25	4.9548
11	4.9591	26	4.9567
12	4.9537	27	4.9582
13	4.9440	28	4.9834
14	4.9859	29	4.9768
15	4.9683	30	4.9607

Los resultados obtenidos se muestran en la Figura 6.1. Se observa que se acepta la hipótesis nula, por lo tanto, no hay evidencia suficiente para rechazar el proceso de medición.



```
Prueba_ChiCuadrada_Algoritmo_de_estimacion_de_parametros =  
0
```

Figura 6.1. Resultado de la prueba de bondad de ajuste chi-cuadrada X^2 .

6.1.2 Histograma

Un histograma es una gráfica de la distribución de un conjunto de datos de una variable cuantitativa la cual brinda información sobre las variaciones del proceso de medición. Si el histograma correspondiente a un conjunto de datos cumple los siguientes requerimientos, se dice que corresponde a una distribución normal [79].

- La máxima altura se alcanza en el intervalo central.
- Si se mueve del intervalo central hacia la izquierda o derecha, la altura deberá decrecer.
- Es simétrico respecto del intervalo central.

La Figura 6.2 muestra el histograma correspondiente a las 30 mediciones, donde podemos ver que para nuestro caso corresponde a una distribución normal.

6.1.3 Gráfico de probabilidad normal

Otro método de prueba para revisar si los datos obtenidos se aproximan a una distribución normal involucra el uso de papel cuadriculado de probabilidad, este método es cualitativo debido a que a simple vista se observa si la distribución de puntos se ajusta a una línea recta [79]. Se apoya en la distribución acumulativa y debe cumplir los siguientes requerimientos:

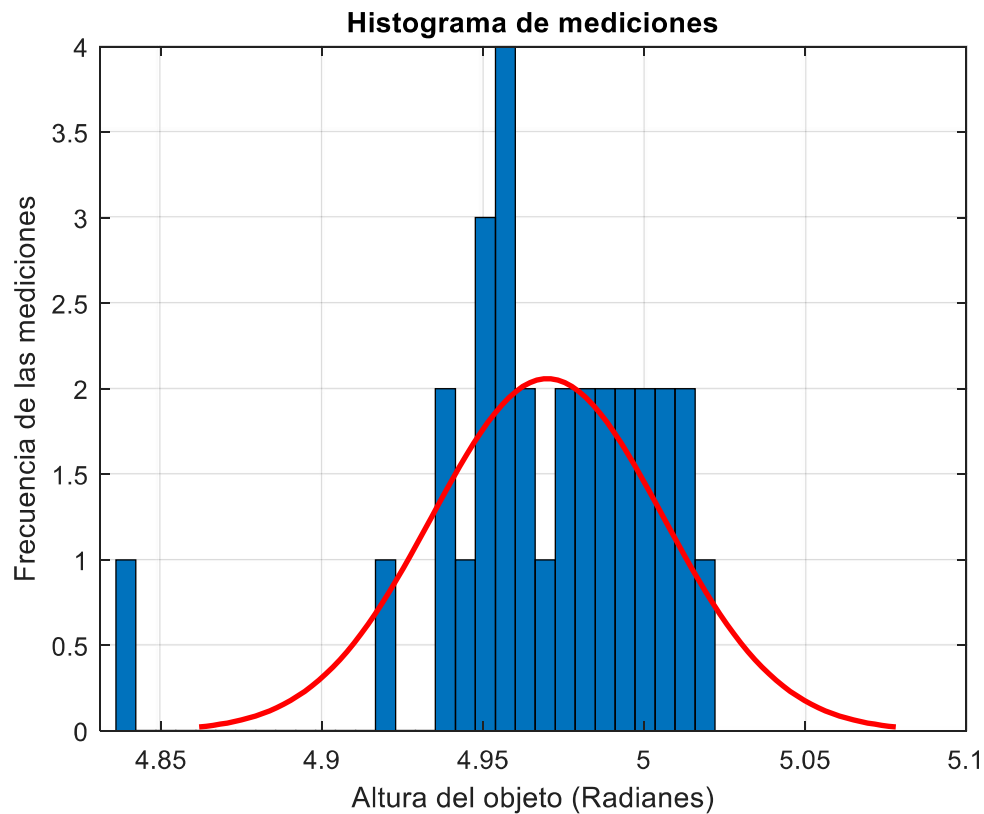


Figura 6.2. Histograma de mediciones.

- Todos los puntos deben caer sobre una línea recta.
- El promedio de todos los puntos de datos debe intersectar esta línea en la ordenada en el 50%.

En la Figura 6.3 se muestra la gráfica de probabilidad normal correspondiente a las 30 mediciones, se observa que las mediciones forman aproximadamente una línea recta, con esto se puede concluir que los datos siguen una distribución normal.

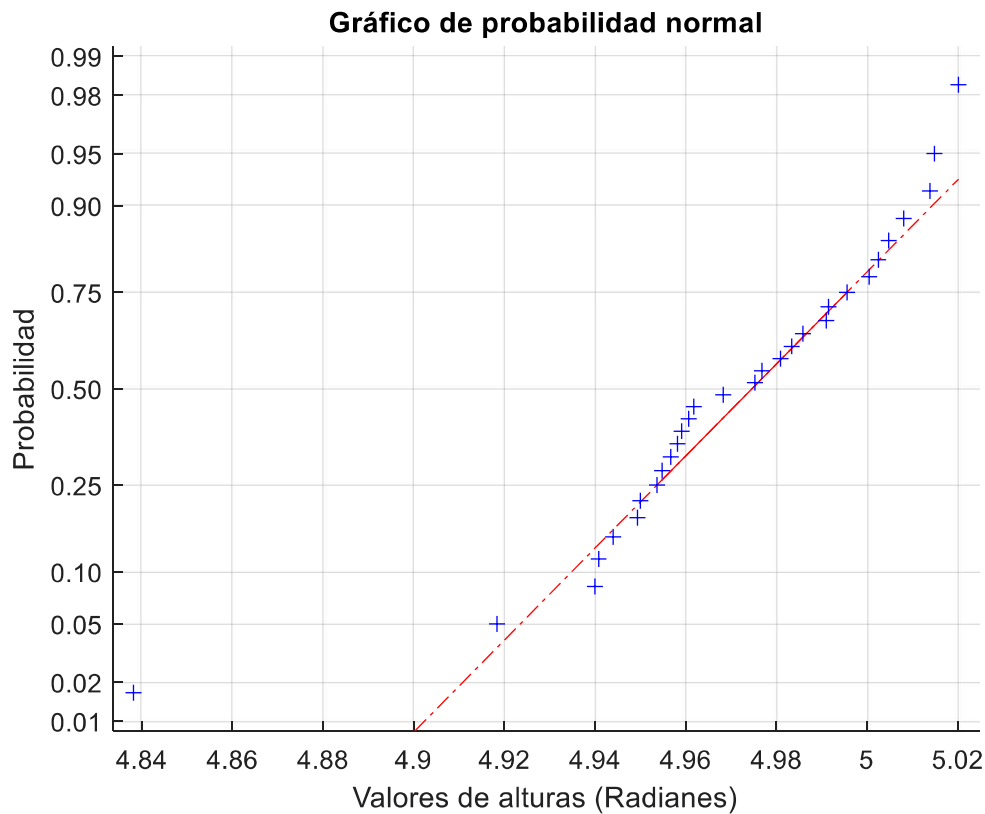


Figura 6.3. Gráfica de probabilidad normal.

6.2 Caracterización

Al realizar la medición de alguna variable física, se debe de tomar en cuenta que no puede ser cuantificada de manera exacta, debido a que sería imposible conocer su valor verdadero, definiendo el valor verdadero como un valor de referencia obtenido cuando la cantidad es medida a través de un método ejemplar, es decir, un método aceptado por expertos el cual es lo suficientemente exacto.

Los instrumentos realizan mediciones de variables físicas entregando valores numéricos que las cuantifican. A esos valores o mediciones realizadas es necesario determinarles dos características básicas como lo son la exactitud y la precisión, para determinar el nivel de confiabilidad del instrumento.

6.2.1 Exactitud

La exactitud expresa la desviación que tiene la medición con respecto del valor verdadero, puede ser expresada mediante la media del error [79], como se muestra en la ecuación 6.2.

$$\bar{X}_{error} = \frac{\sum_{i=1}^N x_{ierror}}{N} \quad (6.2)$$

Donde N es el número de mediciones, $\bar{X}_{error}$ es la media de los errores y x_{ierror} el error de cada medición, calculado con la ecuación (6.3).

$$x_{ierror} = x_{medicion} - Valor\ verdadero \quad (6.3)$$

6.2.2 Precisión

La precisión expresa la medida de repetibilidad de las mediciones de un instrumento con respecto del valor verdadero [79], puede ser calculada con la ecuación (6.4).

$$\sigma_{error} = \sqrt{\frac{\sum_{i=1}^N (x_{ierror} - \bar{X}_{error})^2}{N - 1}} \quad (6.4)$$

Donde σ_{error} es la desviación estándar, N es el número de mediciones, $\bar{X}_{error}$ es la media de lo error y x_{ierror} el error de cada medición.

6.2.3 Resultados obtenidos

En base a los datos experimentales obtenidos y aplicando las ecuaciones (6.2) y (6.4), los resultados obtenidos para la caracterización del algoritmo de estimación de parámetros sinusoidales propuesto fueron:

$$Exactitud = \pm 0.03002 \text{ radianes}$$

$$Precisión = \pm 0.165632 \text{ radianes}$$

CAPÍTULO VII

RESULTADOS FINALES

Después de haber realizado todos los ajustes y modificaciones necesarios en la implementación del algoritmo, como se ha detallado en el Capítulo IV; contar con un montaje experimental de laboratorio sencillo y robusto, descrito en el Capítulo V; y contar con un proceso de medición validado estadísticamente, como quedó demostrado en el Capítulo VI; se procedió entonces a realizar algunas pruebas finales con el fin de evaluar el desempeño de esta técnica con respecto a la determinación de los valores correspondientes de error en la reconstrucción del objeto para poder comparar con las técnicas existentes.

7.1 Resultados de simulación

Primero se trabajó con patrones de franjas simuladas. Para ello se crearon dos patrones de franjas basados en la ecuación (2.71).

$$I(x, y) = 2I_0 \left[\frac{1}{2} + \frac{1}{2} \cos \left(2\pi k \frac{x}{w} + \theta \right) \right]^{1/\gamma} + \eta(x, y)$$

Los parámetros empleados para la generación de estos patrones fueron:

- Intensidad promedio $I_0 = 127$.
- Ancho del patrón $w = 960$.
- Número de franjas en el patrón $k = 18$.
- Factor gamma $\gamma = 2.1$
- Ruido η , generado con una relación señal a ruido $SNR = 10000$, lo cual no agrega ruido al patrón proyectado.

En uno de estos patrones se moduló para simular el objeto descrito por la función mostrada a continuación en la ecuación (7.1), esta función objeto se suma en el argumento de la ecuación (2.71).

$$\text{Objeto}(x, y) = r \sin x \cos y \quad (7.1)$$

$$\text{Dominio: } \{r \in \mathbb{R} \mid r = 10\}, \left\{x \in \mathbb{R} \mid \frac{-\pi}{2} \leq x \leq \frac{3\pi}{2}\right\}, \{y \in \mathbb{R} \mid 0 \leq y \leq \pi\}$$

La Figura 7.1 muestra el resultado de ambos patrones simulados, la Figura 7.1 a) corresponde al patrón de franjas modulado por la función objeto y la Figura 7.1 b) corresponde al patrón de franjas de referencia sin modulación.

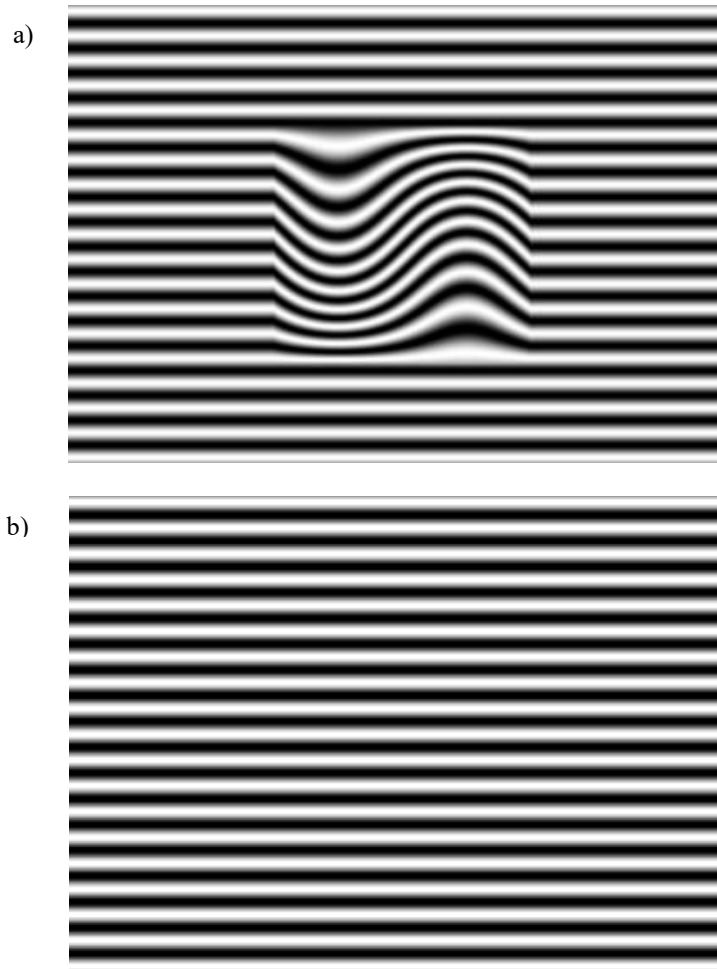


Figura 7.1. Patrones de franjas simulados: a) Patrón de franjas modulado por la función objeto, b) Patrón de franjas de referencia.

Tomando este patrón de franjas con objeto simulado como entrada, al ejecutar el software con el algoritmo desarrollado, arrojó los resultados mostrados en la Figura 7.2, la Figura 7.2a muestra el mapa de fase desenvuelta mientras que la Figura 7.2b muestra la reconstrucción tridimensional del objeto simulado.

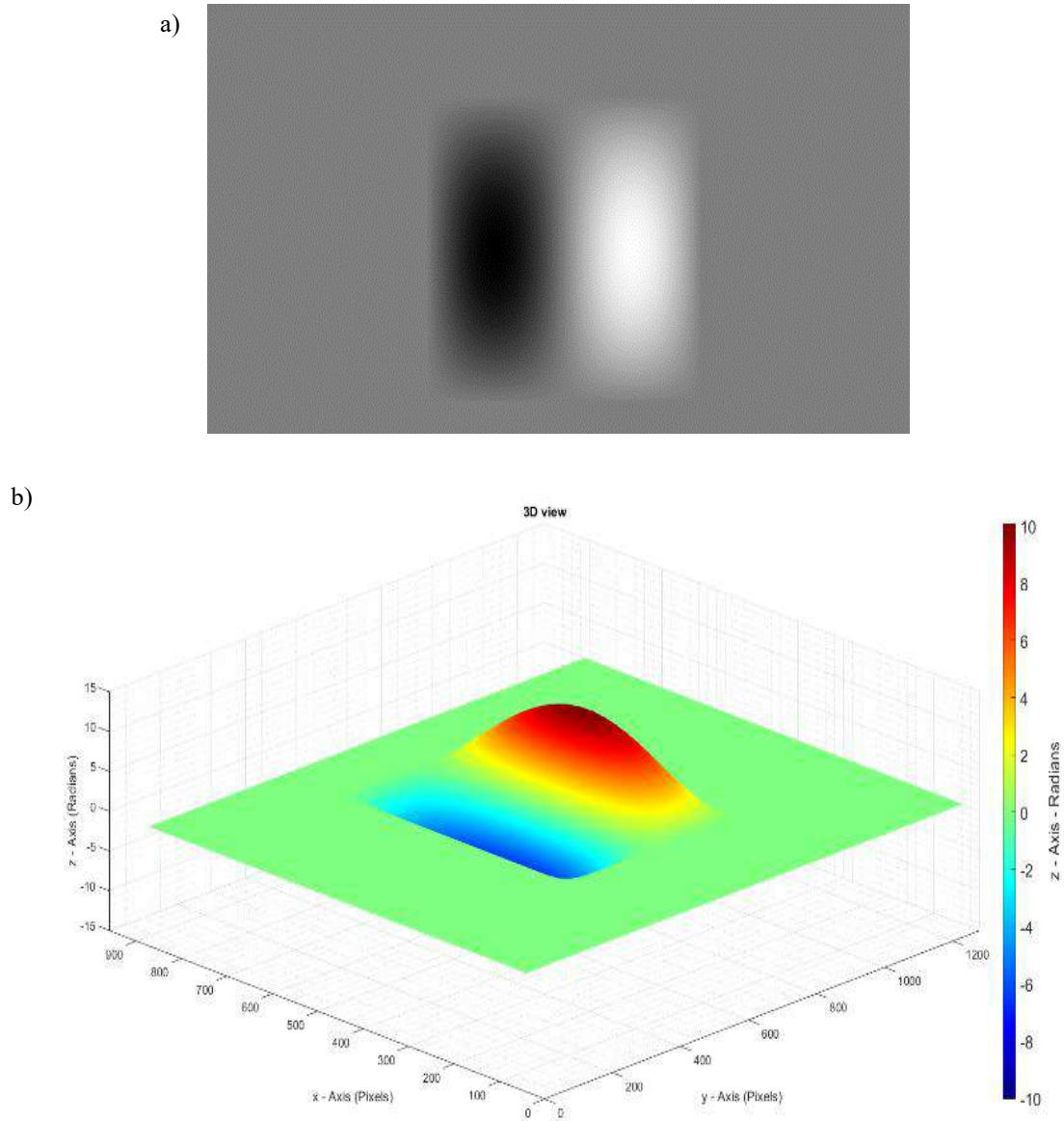


Figura 7.2. Resultados de la simulación: a) Mapa de fase desenvuelta; b) Reconstrucción 3D del objeto.

En términos cualitativos, se puede observar un resultado satisfactorio en la reconstrucción del objeto. Sin embargo, resulta conveniente evaluar el desempeño en términos cuantitativos, para lo cual se procedió a realizar un análisis de error mediante el cálculo de los valores de error de la Raíz Media Cuadrática (RMSE) y error medio (ME) en la reconstrucción.

La determinación del RMSE se realiza de la siguiente manera:

1. El $RMSE_{col}$ se calcula píxel a píxel por columna con la ecuación (7.2).

$$RMSE_{col} = \sqrt{\frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2} \quad (7.2)$$

Donde y_i es valor de fase desenvuelta obtenida por el algoritmo, $\hat{y}_i$ es el valor de fase obtenido por la función objeto para ese píxel y N es el número de píxeles en la columna.

2. Se calcula el RMSE del $RMSE_{col}$ con los datos obtenidos por columna, con la ecuación (7.3).

$$RMSE = \sqrt{\frac{1}{M} \sum_{i=1}^M RMSE_{col}^2} \quad (7.3)$$

Donde M es el número de columnas.

La determinación del ME se realiza de la siguiente manera:

1. El ME_{col} calcula píxel a píxel por columna con la ecuación (7.4).

$$ME_{col} = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i) \quad (7.4)$$

Donde y_i es valor de fase desenvuelta obtenida por el algoritmo, $\hat{y}_i$ es el valor de fase obtenido por la función objeto para ese píxel y N es el número de píxeles en la columna.

2. Se calcula el ME del ME_{col} con los datos obtenidos por columna, con la ecuación 7.5.

$$ME = \frac{1}{M} \sum_{i=1}^M (ME_{col}) \quad (7.5)$$

Donde M es el número de columnas.

La resolución espacial de las imágenes simuladas es de 1296 x 964 píxeles, lo que equivale a aproximadamente 1.3 MP; es decir, se calcularon 1.3 x 10⁶ valores de error para cada imagen. Por motivos de espacio, no se presentan en este documento tablas o apéndices con dichos valores, solo se proporcionan los valores calculados RMS y promedio.

Con el propósito de evaluar la técnica propuesta por si sola, evitando en lo posible la introducción de otras fuentes error en los cálculos, los valores de error están expresados en radianes y no en unidades del mundo real, para evitar los posibles errores inducidos por una mala calibración en el proceso fase-altura [30].

Los resultados del cálculo de error con imagen de entrada simulada, se muestran en la Tabla 7.1.

Tabla 7.1. Resultados de error con entrada simulada (Radianes).

Imagen de entrada	Error promedio ME	Error RMS RMSE
Simulación	-0.0035	0.0178

7.2 Resultados experimentales

Finalmente, se procedió a realizar la evaluación del desempeño de esta técnica bajo condiciones reales de captura de un patrón de franjas por medio de una cámara digital.

Para ello se utilizó el mismo patrón de franjas moduladas con objeto simulado de la sección anterior, ver Figura 7.1a), solo que ahora este patrón fue proyectado y capturado utilizando el montaje de laboratorio descrito en el Capítulo V.

El hecho de utilizar la proyección de las franjas moduladas en lugar de un objeto real sobre la pantalla de referencia tiene su motivación en evitar la aparición de sombras y oclusiones, para poder evaluar solo bajo condiciones de ruido, distorsión gamma y pixeles dañados que se introducen en un proceso de captura real.

La Figura 7.1 muestra las imágenes capturadas, la Figura 7.3 a) corresponde al patrón de franjas modulado por la función objeto y la Figura 7.3 b) corresponde al patrón de franjas de referencia sin modulación.

Tomando ahora este patrón capturado de franjas como entrada, al ejecutar de nuevo el software con el algoritmo desarrollado, arrojó los resultados mostrados en la Figura 7.4, la Figura 7.4 a) muestra el mapa de fase desenvuelta mientras que la Figura 7.4 b) muestra la reconstrucción tridimensional del objeto simulado.

En términos cualitativos, se puede observar un buen resultado en la reconstrucción del objeto.

Pero de nuevo resulta conveniente evaluar el desempeño en términos cuantitativos, para lo cual se procedió a realizar el mismo análisis de error mediante el cálculo de los valores de error RMS y error medio en la reconstrucción.

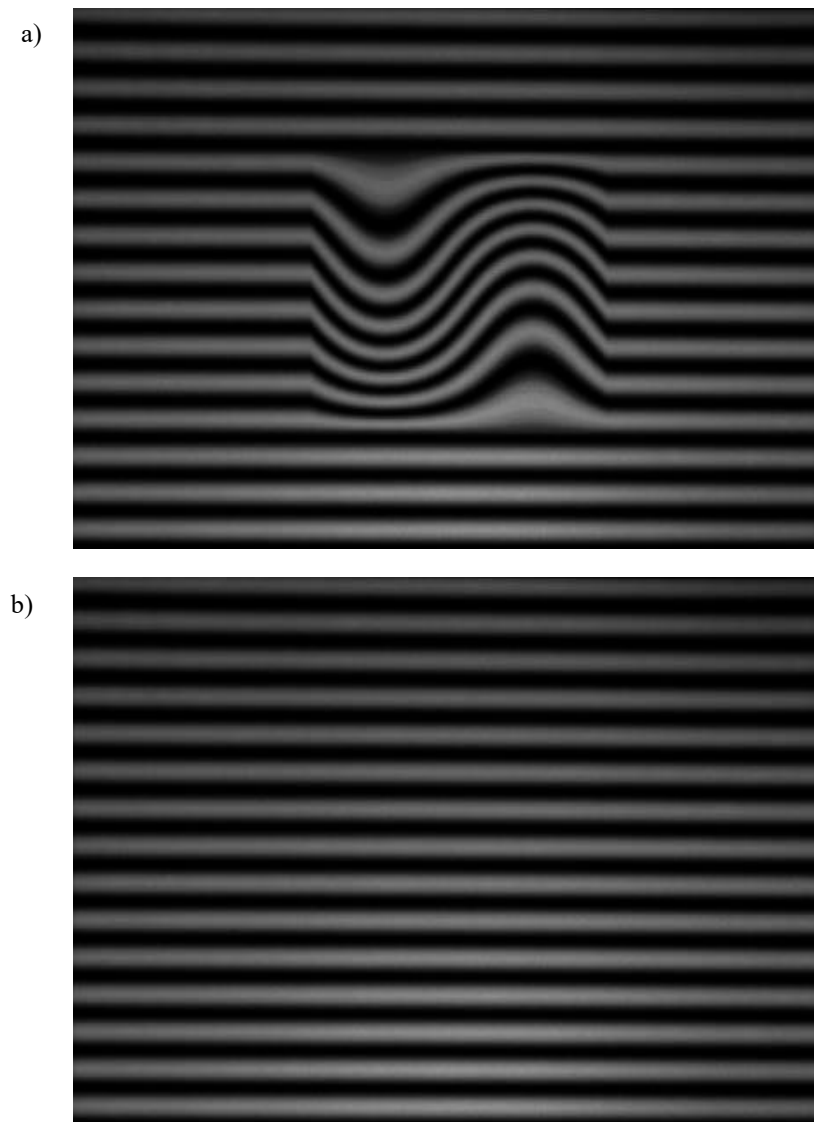


Figura 7.3. Patrones de franjas capturados: a) Patrón de franjas modulado por la función objeto, b) Patrón de franjas de referencia.

Diferente a lo que sucede en la simulación, el proceso de proyectar y capturar un patrón de franjas, genera que la ubicación del objeto en la captura, difiera de su localización en la proyección. Para solucionar este inconveniente fue necesario la creación de un algoritmo para desplazar y escalar el objeto de referencia en la reconstrucción 3D generada.

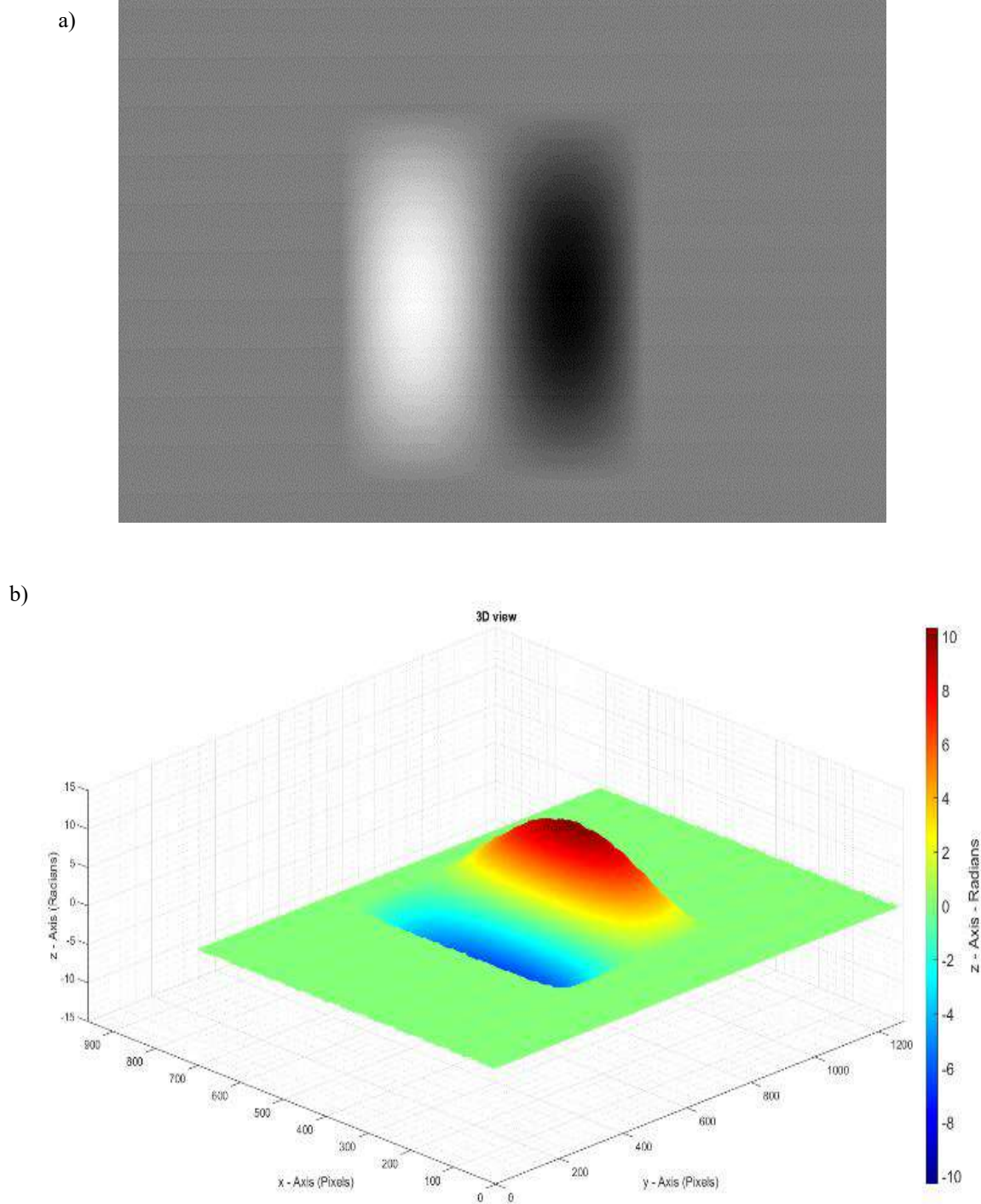


Figura 7.4. Resultados experimentales: a) Mapa de fase sin envolver, b) Reconstrucción 3D del objeto.

El algoritmo funciona de la siguiente manera:

1. Se definen rangos de valores de escalamiento, desplazamiento horizontal y desplazamiento vertical.
2. El mapa de fase de la función objeto se escala utilizando la función *imresize* de MATLAB®.
3. Para cada conjunto de valor de escalamiento, desplazamiento horizontal y desplazamiento vertical, se calcula la correlación lineal entre la imagen escalada y desplazada y la reconstrucción 3D obtenida.
4. Se analizan los valores de correlación lineal obtenidos, se selecciona el valor mayor y se obtienen los valores de escalamiento, desplazamiento horizontal y desplazamiento vertical.

Ya con los valores de escalamiento, desplazamiento horizontal y desplazamiento vertical, se escala y desplaza la función objeto.

Una vez hechas estas correcciones, se repite el procedimiento previo para obtener los resultados del cálculo de error con imagen de entrada capturada que se muestran en la Tabla 7.2.

Tabla 7.2. Resultados de error con entrada capturada (Radianes).

Imagen de entrada	Error promedio ME	Error RMS RMSE
Capturada	0.0044	0.2738

De acuerdo a los resultados obtenidos se puede concluir que la técnica propuesta puede ser considerada un nuevo método espacial para el análisis de franjas sinusoidales.

CAPÍTULO VIII

CONCLUSIONES Y TRABAJO FUTURO

A continuación, se presentan las conclusiones referentes al desarrollo de este trabajo de tesis, así como algunas recomendaciones de trabajo a futuro que podrían optimizar el desempeño del algoritmo propuesto y su aplicación en la medición de objetos reales.

8.1 Conclusiones generales

Se desarrolló una nueva técnica espacial para reconstrucción de objetos tridimensionales, basada en la estimación de los parámetros de un patrón de franjas sinusoidales moduladas por una función objeto de prueba, la cual representa una innovación en el análisis de los sistemas de proyección de franjas y abre nuevas posibilidades para aplicaciones en diversas áreas, desde la ingeniería y la arquitectura hasta la medicina y el entretenimiento, debido a que puede utilizarse, tanto en objetos estáticos como en dinámicos.

8.2 Conclusiones sobre el algoritmo de estimación de parámetros

Se desarrolló un algoritmo computacional basado en el modelo matemático propuesto, logrando la reconstrucción tridimensional de un objeto, simplemente estimando los valores locales de intensidad promedio y modulación de amplitud del perfil cosenoidal presente en el patrón de franjas moduladas por la forma del objeto.

Con la estimación de esos parámetros se calcula la fase envuelta de acuerdo a la ecuación (3.1) y posteriormente se desenvuelve la fase para obtener la forma del objeto.

El algoritmo propuesto, presenta un modelo matemático sencillo pero eficiente, sin embargo, para su correcta implementación, se debe dar una solución a casos especiales del cálculo de fase en la periferia del patrón.

Una de las etapas más importantes en este algoritmo es la interpolación, la cual permite estimar los valores entre crestas. Se realizaron pruebas con diferentes tipos de interpolación, se concluyó que la interpolación lineal, fue la más confiable para evitar problemas en la estimación de la fase envuelta.

Este algoritmo representa una técnica totalmente diferente a las técnicas actualmente utilizadas que se describen en el Capítulo II.

8.3 Conclusiones de resultados experimentales

Al realizar pruebas con capturas reales aparecen nuevas condiciones, que deben tomarse en cuenta para evitar la aparición de errores en la reconstrucción.

Algunas de estas condiciones incluyen distorsión gamma del patrón de franjas introducida por el proyector, píxeles dañados en el sensor de la cámara y la generación de sombras y oclusiones derivadas de la forma misma del objeto.

8.3.1 Corrección del factor gamma

Para mejorar los resultados de las reconstrucciones 3D, se implementó el algoritmo de cálculo de factor gamma, algo que es necesario corregir de acuerdo a la literatura. Se formuló una metodología para poder calcular el valor óptimo de factor gamma asociado a nuestro montaje.

8.3.2 Corrección de píxeles dañados

La cámara utilizada presenta problemas de píxeles muertos, se pueden haber generado por el envejecimiento del sensor, defectos de fabricación o exposición a altas temperaturas. Esto genera un cambio súbito de intensidad en el patrón de franjas, se generó un algoritmo de preprocesamiento para píxeles dañados, mejorando este inconveniente.

8.3.3 Generación de sombras y oclusiones

El problema de las oclusiones generadas al utilizar objetos reales es un problema para cualquier técnica de reconstrucción 3D de objetos.

Debido a que en este trabajo se presentó una técnica nueva y diferente a lo que se realiza actualmente, se optó por utilizar objetos sintéticos modulados en las franjas proyectadas, para evitar la generación de sombras y oclusiones.

8.4 Trabajo futuro

Como los resultados de la presente tesis lo determinaron, existen áreas de mejora en la técnica propuesta. A continuación, se enlistan las áreas de mejora:

- Mejorar el algoritmo de detección de mínimos y máximos para no eliminar la información en los extremos de la captura.
- Evaluar el algoritmo con objetos reales e incluir una técnica para corregir el problema que se genera en las áreas con sombras y oclusiones del patrón.
- Explorar la utilización de un patrón de proyección de variación triangular de intensidad para simplificar el proceso de desenvolvimiento debido a la función coseno inverso.
- En el algoritmo de cálculo del factor gamma, para mejorar el coeficiente de correlación, tomar en cuenta la modulación de amplitud en el patrón de referencia.

REFERENCIAS

- [1] D. Malacara, *Optical Shop Testing: Third Edition*. Wiley, 2006. doi: 10.1002/9780470135976.
- [2] K. J. Stout and L. Blunt, *Three dimensional surface topography*, Second. Penton Press, 2000.
- [3] D. Malacara, Z. Malacara, and Manuel. Servn, *Interferogram analysis for optical testing*, Second Edition. Taylor & Francis, 2005. Accessed: May 27, 2019. [Online]. Available: <https://www.crcpress.com/Interferogram-Analysis-For-Optical-Testing/Malacara-Servn/p/book/9781574446821>
- [4] P. Hariharan, *Optical interferometry*. Academic Press, 2003.
- [5] C. Jiang, B. Lim, and S. Zhang, "Three-dimensional shape measurement using a structured light system with dual projectors," *Appl Opt*, vol. 57, no. 14, pp. 3983–3990, May 2018, doi: 10.1364/AO.57.003983.
- [6] 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, Oct. 2018, doi: 10.1364/OE.26.028544.
- [7] Y. Li, M. Kästner, and E. Reithmeier, "Triangulation-based edge measurement using polyview optics," *Opt Lasers Eng*, vol. 103, pp. 71–76, Apr. 2018, doi: 10.1016/J.OPTLASENG.2017.11.015.
- [8] C. Zuo, S. Feng, L. Huang, T. Tao, W. Yin, and Q. Chen, "Phase shifting algorithms for fringe projection profilometry: A review," *Opt Lasers Eng*, vol. 109, pp. 23–59, Oct. 2018, doi: 10.1016/J.OPTLASENG.2018.04.019.
- [9] X. Li, Z. Zhang, and C. Yang, "Reconstruction method for fringe projection profilometry based on light beams," *Appl Opt*, vol. 55, no. 34, pp. 9895–9906, Dec. 2016, doi: 10.1364/AO.55.009895.
- [10] C. Faber, M. Strohmeier, and H. Liang, "How to Exploit Prior Knowledge in Industrial 3D-Metrology," in *Imaging and Applied Optics 2019 (COSI, IS, MATH,*

- pcAOP*) (2019), *paper CTh2A.6*, The Optical Society, Jul. 2019, p. CTh2A.6. doi: 10.1364/cosi.2019.cth2a.6.
- [11] M. Strohmeier and C. Faber, "Harnessing inverse fringe patterns for actual industrial inspection applications," in *Optics InfoBase Conference Papers*, OSA - The Optical Society, Jun. 2019, p. IW1C.4. doi: 10.1364/ISA.2019.IW1C.4.
- [12] J. A. N. Buytaert and J. J. J. Dirckx, "Study of the performance of 84 phase-shifting algorithms for interferometry," *Journal of Optics (India)*, vol. 40, no. 3, pp. 114–131, Sep. 2011, doi: 10.1007/s12596-011-0044-y.
- [13] C. Li, Y. Cao, C. Chen, Y. Wan, G. Fu, and Y. Wang, "Computer-generated Moiré profilometry," *Opt Express*, vol. 25, no. 22, p. 26815, Oct. 2017, doi: 10.1364/oe.25.026815.
- [14] S. Zhang, "Active versus passive projector nonlinear gamma compensation method for high-quality fringe pattern generation," K. G. Harding and T. Yoshizawa, Eds., May 2014, p. 911002. doi: 10.1117/12.2050534.
- [15] Z. Wang, D. A. Nguyen, J. C. Barnes, Z. W. Ā, D. A. Nguyen, and J. C. Barnes, "Some practical considerations in fringe projection profilometry," *Opt Lasers Eng*, vol. 48, no. 2, pp. 218–225, 2010, doi: 10.1016/j.optlaseng.2009.06.005.
- [16] X. Yu, Y. Liu, N. Liu, M. Fan, and X. Su, "Flexible gamma calculation algorithm based on probability distribution function in digital fringe projection system," *Opt Express*, vol. 27, no. 22, p. 32047, Oct. 2019, doi: 10.1364/oe.27.032047.
- [17] J. Geng, "Structured-light 3D surface imaging: a tutorial," *Adv Opt Photonics*, vol. 3, no. 2, pp. 128–160, Jun. 2011, doi: 10.1364/AOP.3.000128.
- [18] J. Salvi, J. Pagès, and J. Batlle, "Pattern codification strategies in structured light systems," *Pattern Recognit*, vol. 37, no. 4, pp. 827–849, Apr. 2004, doi: 10.1016/J.PATCOG.2003.10.002.
- [19] N. Tornero Martínez, G. Trujillo-Schiaffino, M. Anguiano-Morales, P. G. Mendoza-Villegas, D. P. Salas-Peimbert, and L. F. Corral-Martínez, *Color profilometry techniques: A review*, vol. 51, no. 4. Sociedad Española de Optica, 2018. Accessed: Jun. 04, 2019. [Online]. Available: <https://dialnet.unirioja.es/servlet/articulo?codigo=6805221>

- [20] H. Zhao, C. Zhang, C. Zhou, K. Jiang, and M. Fang, "Circular fringe projection profilometry," *Opt Lett*, vol. 41, no. 21, pp. 4951–4954, Nov. 2016, doi: 10.1364/OL.41.004951.
- [21] C. Zhang *et al.*, "Three-dimensional measurement based on optimized circular fringe projection technique," *Optics Express*, Vol. 27, Issue 3, pp. 2465-2477, vol. 27, no. 3, pp. 2465–2477, Feb. 2019, doi: 10.1364/OE.27.002465.
- [22] W.-H. Su, C.-Y. Kuo, and F.-J. Kao, "Three-dimensional trace measurements for fast-moving objects using binary-encoded fringe projection techniques," *Appl Opt*, vol. 53, no. 24, p. 5283, Aug. 2014, doi: 10.1364/ao.53.005283.
- [23] K. Chen, J. Xi, Y. Yu, S. Tong, and Q. Guo, "Three-dimensional measurement of object surfaces with complex shape and color distribution based on projection of color fringe patterns," *Appl Opt*, vol. 52, no. 30, pp. 7360–7366, Oct. 2013, doi: 10.1364/AO.52.007360.
- [24] W.-H. Su, "Color-encoded fringe projection for 3D shape measurements," *Opt Express*, vol. 15, no. 20, p. 13167, Oct. 2007, doi: 10.1364/oe.15.013167.
- [25] R. Legarda-Saenz, J. L. Flores, J. A. Ferrari, A. Silva, and G. G. Torales, "Color-fringe pattern profilometry using a generalized phase-shifting algorithm," *Applied Optics*, Vol. 54, Issue 30, pp. 8827-8834, vol. 54, no. 30, pp. 8827–8834, Oct. 2015, doi: 10.1364/AO.54.008827.
- [26] L. Wang, Y. Ma, H. Zhang, Y. Xin, C. Yuan, and T.-C. Poon, "Accurate phase-shift estimation for fringe-pattern profilometry," *Appl Opt*, vol. 58, no. 34, p. G358, Dec. 2019, doi: 10.1364/ao.58.00g358.
- [27] Y. Hu, J. Xi, E. Li, J. Chicharo, and Z. Yang, "Three-dimensional profilometry based on shift estimation of projected fringe patterns," *Appl Opt*, vol. 45, no. 4, pp. 678–687, Feb. 2006, doi: 10.1364/AO.45.000678.
- [28] S. S. Gorthi, "Fringe projection techniques: Whither we are?," *Opt Lasers Eng*, vol. 48, no. 2, pp. 133–140, Feb. 2010, doi: 10.1016/J.OPTLASENG.2009.09.001.

- [29] M. Takeda, H. Ina, and S. Kobayashi, "Fourier-transform method of fringe-pattern analysis for computer-based topography and interferometry," *J Opt Soc Am*, vol. 72, no. 1, pp. 156–160, Jan. 1982, doi: 10.1364/JOSA.72.000156.
- [30] L. Huang, Q. Kemao, B. Pan, and A. K. Asundi, "Comparison of Fourier transform, windowed Fourier transform, and wavelet transform methods for phase extraction from a single fringe pattern in fringe projection profilometry," *Opt Lasers Eng*, vol. 48, no. 2, pp. 141–148, Feb. 2010, doi: 10.1016/j.optlaseng.2009.04.003.
- [31] C. Quan, C. J. Tay, and L. J. Chen, "A study on carrier-removal techniques in fringe projection profilometry," *Opt Laser Technol*, vol. 39, no. 6, pp. 1155–1161, Sep. 2007, doi: 10.1016/J.OPTLASTEC.2006.09.003.
- [32] M. Servin, J. A. Quiroga, and M. Padilla, *Fringe Pattern Analysis for Optical Metrology*, 1st ed., vol. 1. Weinheim, Germany: Wiley-VCH Verlag GmbH & Co. KGaA, 2014. doi: 10.1002/9783527681075.
- [33] D. W. Phillion, "General methods for generating phase-shifting interferometry algorithms," *Appl Opt*, vol. 36, no. 31, pp. 8098–8115, Nov. 1997, doi: 10.1364/AO.36.008098.
- [34] Q. Kemao, L. T. H. Nam, L. Feng, and S. H. Soon, "Comparative analysis on some filters for wrapped phase maps," *Appl Opt*, vol. 46, no. 30, pp. 7412–7418, Oct. 2007, doi: 10.1364/AO.46.007412.
- [35] A. No, L. Ying, and A. No, "Phase unwrapping," *Wiley Encyclopedia of Biomedical Engineering*, vol. 24, pp. 1–11, 2006, doi: 10.1002/9780471740360.ebs1356.
- [36] K. Itoh, "Analysis of the phase unwrapping algorithm," *Appl Opt*, vol. 21, no. 14, p. 2470, Jul. 1982, doi: 10.1364/AO.21.002470.
- [37] D. C. Ghiglia and M. D. Pritt, *Two-dimensional phase unwrapping: theory, algorithms, and software*. Wiley, 1998. Accessed: Jun. 04, 2019. [Online]. Available: https://books.google.com.mx/books/about/Two_Dimensional_Phase_Unwrapping.html?id=pQtTAAAAMAAJ&source=kp_book_description&redir_esc=y

- [38] K. J. Gåsvik, *Optical Metrology*. Wiley, 2002. doi: 10.1002/0470855606.
- [39] M. Takeda and K. Mutoh, “Fourier transform profilometry for the automatic measurement of 3-D object shapes,” *Appl Opt*, vol. 22, no. 24, pp. 3977–3982, Dec. 1983, doi: 10.1364/AO.22.003977.
- [40] Y. Wen, S. Li, H. Cheng, X. Su, and Q. Zhang, “Universal calculation formula and calibration method in Fourier transform profilometry,” *Appl Opt*, vol. 49, no. 34, pp. 6563–6569, Dec. 2010, doi: 10.1364/AO.49.006563.
- [41] C. Zuo, Q. Chen, G. Gu, S. Feng, and F. Feng, “High-speed three-dimensional profilometry for multiple objects with complex shapes,” *Opt Express*, vol. 20, no. 17, pp. 19493–19510, Aug. 2012, doi: 10.1364/OE.20.019493.
- [42] H. Jiang, H. Zhao, and X. Li, “High dynamic range fringe acquisition: A novel 3-D scanning technique for high-reflective surfaces,” *Opt Lasers Eng*, vol. 50, no. 10, pp. 1484–1493, Oct. 2012, doi: 10.1016/J.OPTLASENG.2011.11.021.
- [43] S.-T. Yau, “High dynamic range scanning technique,” *Optical Engineering*, vol. 48, no. 3, p. 033604, Mar. 2009, doi: 10.1117/1.3099720.
- [44] R. Zhang, H. Guo, and A. K. Asundi, “Geometric analysis of influence of fringe directions on phase sensitivities in fringe projection profilometry,” *Appl Opt*, vol. 55, no. 27, pp. 7675–7687, Sep. 2016, doi: 10.1364/AO.55.007675.
- [45] H. Sheng, J. Xu, and S. Zhang, “Dynamic projection theory for fringe projection profilometry,” *Appl Opt*, vol. 56, no. 30, p. 8452, Oct. 2017, doi: 10.1364/ao.56.008452.
- [46] E. Zendejas-Hernández, G. Trujillo-Schiaffino, M. Anguiano-Morales, D. P. Salas-Peimbert, L. F. Corral-Martínez, and N. Tornero-Martínez, “Spatial and temporal methods for fringe pattern analysis: a review,” *Journal of Optics* 2023 52:2, vol. 52, no. 2, pp. 888–899, Apr. 2023, doi: 10.1007/S12596-023-01166-1.
- [47] M. Takeda, “Spatial-carrier fringe-pattern analysis and its applications to precision interferometry and profilometry: An overview,” *Industrial Metrology*, vol. 1, no. 2, pp. 79–99, Jun. 1990, doi: 10.1016/0921-5956(90)80019-R.

- [48] B. Li, Y. An, and S. Zhang, "Single-shot absolute 3D shape measurement with Fourier transform profilometry," *Appl Opt*, vol. 55, no. 19, p. 5219, Jul. 2016, doi: 10.1364/ao.55.005219.
- [49] Q. Kemao, "Windowed Fourier transform for fringe pattern analysis," *Appl Opt*, vol. 43, no. 13, pp. 2695–2702, May 2004, doi: 10.1364/AO.43.002695.
- [50] Q. Kemao, H. Wang, and W. Gao, "Windowed Fourier transform for fringe pattern analysis: theoretical analyses," *Appl Opt*, vol. 47, no. 29, pp. 5408–5419, Oct. 2008, doi: 10.1364/AO.47.005408.
- [51] Q. Kemao, "Two-dimensional windowed Fourier transform for fringe pattern analysis: Principles, applications and implementations," *Opt Lasers Eng*, vol. 45, no. 2, pp. 304–317, Feb. 2007, doi: 10.1016/J.OPTLASENG.2005.10.012.
- [52] J. Zhong and J. Weng, "Spatial carrier-fringe pattern analysis by means of wavelet transform: wavelet transform profilometry," *Appl Opt*, vol. 43, no. 26, pp. 4993–4998, Sep. 2004, doi: 10.1364/AO.43.004993.
- [53] M. A. Gdeisat *et al.*, "Spatial and temporal carrier fringe pattern demodulation using the one-dimensional continuous wavelet transform: Recent progress, challenges, and suggested developments," *Opt Lasers Eng*, vol. 47, no. 12, pp. 1348–1361, Dec. 2009, doi: 10.1016/J.OPTLASENG.2009.07.009.
- [54] S. Li, W. Chen, and X. Su, "Reliability-guided phase unwrapping in wavelet-transform profilometry," *Appl Opt*, vol. 47, no. 18, pp. 3369–3377, Jun. 2008, doi: 10.1364/AO.47.003369.
- [55] Z. Wang and H. Ma, "Advanced continuous wavelet transform algorithm for digital interferogram analysis and processing," *Optical Engineering*, vol. 45, no. 4, p. 045601, Apr. 2006, doi: 10.1117/1.2188399.
- [56] R. Zou, S. Cui, F. Tian, Q. Li, and M. Yao, "Phase retrieval of optical fringe pattern using wavelet ridge section and adaptive bandpass filter," *Optical Engineering*, vol. 55, no. 9, p. 093103, Sep. 2016, doi: 10.1117/1.OE.55.9.093103.
- [57] Y. Tounsi, M. Kumar, A. Siari, F. Mendoza-Santoyo, A. Nassim, and O. Matoba, "Digital four-step phase-shifting technique from a single fringe pattern using

- Riesz transform,” *Opt Lett*, vol. 44, no. 14, pp. 3434–3437, Jul. 2019, doi: 10.1364/OL.44.003434.
- [58] Y. Tounsi, M. Kumar, A. Siari, F. Mendoza-Santoyo, O. Matoba, and A. Nassim, “Riesz transform for fringes pattern analysis: advantages and limitations,” p. 23, Oct. 2020, doi: 10.1117/12.2574757.
- [59] T. Yoshizawa, *Handbook of optical metrology : principles and applications*. 2017.
- [60] K. Creath, “Comparison Of Phase-Measurement Algorithms,” K. Creath, Ed., International Society for Optics and Photonics, Mar. 1987, p. 19. doi: 10.1117/12.939587.
- [61] G. Cloud, *Optical Methods of Engineering Analysis*. Cambridge: Cambridge University Press, 1995. doi: 10.1017/CBO9780511575013.
- [62] J. H. Bruning, D. R. Herriott, J. E. Gallagher, D. P. Rosenfeld, A. D. White, and D. J. Brangaccio, “Digital Wavefront Measuring Interferometer for Testing Optical Surfaces and Lenses,” *Appl Opt*, vol. 13, no. 11, pp. 2693–2703, Nov. 1974, doi: 10.1364/AO.13.002693.
- [63] C. J. Morgan, “Least-squares estimation in phase-measurement interferometry,” *Opt Lett*, vol. 7, no. 8, pp. 368–370, Aug. 1982, doi: 10.1364/OL.7.000368.
- [64] V. Srinivasan, H. C. Liu, and M. Halioua, “Automated phase-measuring profilometry of 3-D diffuse objects,” *Appl Opt*, vol. 23, no. 18, pp. 3105–3108, Sep. 1984, doi: 10.1364/AO.23.003105.
- [65] K. Creath, “V Phase-Measurement Interferometry Techniques,” *Progress in Optics*, vol. 26, no. C, pp. 349–393, Jan. 1988, doi: 10.1016/S0079-6638(08)70178-1.
- [66] P. S. Huang, Q. J. Hu, and F.-P. Chiang, “Double three-step phase-shifting algorithm,” *Appl Opt*, vol. 41, no. 22, pp. 4503–4509, Aug. 2002, doi: 10.1364/AO.41.004503.

- [67] P. Carré, "Installation et utilisation du comparateur photoélectrique et interférentiel du Bureau International des Poids et Mesures," *Metrologia*, vol. 2, no. 1, pp. 13–23, Jan. 1966, doi: 10.1088/0026-1394/2/1/005.
- [68] D. Li, C. Liu, and J. Tian, "Telecentric 3D profilometry based on phase-shifting fringe projection," *Opt Express*, vol. 22, no. 26, p. 31826, Dec. 2014, doi: 10.1364/oe.22.031826.
- [69] P. Hariharan, "Digital phase-stepping interferometry: effects of multiply reflected beams," *Appl Opt*, vol. 26, no. 13, pp. 2506–2507, Jul. 1987, doi: 10.1364/AO.26.002506.
- [70] J. Schwider, R. Burow, K.-E. Elssner, J. Grzanna, R. Spolaczyk, and K. Merkel, "Digital wave-front measuring interferometry: some systematic error sources," *Appl Opt*, vol. 22, no. 21, pp. 3421–3432, Nov. 1983, doi: 10.1364/AO.22.003421.
- [71] P. L. Wizinowich, "Phase shifting interferometry in the presence of vibration: a new algorithm and system," *Appl Opt*, vol. 29, no. 22, pp. 3271–3279, Aug. 1990, doi: 10.1364/AO.29.003271.
- [72] P. L. Wizinowich, "System For Phase Shifting Interferometry In The Presence Of Vibration," J. E. Greivenkamp and M. Young, Eds., International Society for Optics and Photonics, Dec. 1989, p. 25. doi: 10.1117/12.962804.
- [73] S. Zhang and S.-T. Yau, "High-speed three-dimensional shape measurement system using a modified two-plus-one phase-shifting algorithm," *Optical Engineering*, vol. 46, no. 11, p. 113603, Nov. 2007, doi: 10.1117/1.2802546.
- [74] X.-Y. Su, J. Li, L.-R. Guo, and W.-Y. Su, "An Improved Fourier Transform Profilometry," C. P. Grover, Ed., International Society for Optics and Photonics, Jan. 1989, p. 241. doi: 10.1117/12.947595.
- [75] L. Guo, X. Su, and L. Guo, "Improved Fourier transform profilometry for the automatic measurement of 3D object shapes," *Optical Engineering*, vol. 29, no. 12, p. 1439, 1990, doi: 10.1117/12.55746.

- [76] Christine. Fernández-Maloigne, F. Robert-Inacio, and Ludovic. Macaire, “Digital color imaging,” 2012, Accessed: Jan. 08, 2024. [Online]. Available: <https://www.wiley.com/en-ca/Digital+Color+Imaging-p-9781848213470>
- [77] J. Salvi, S. Fernandez, T. Pribanic, and X. Llado, “A state of the art in structured light patterns for surface profilometry,” *Pattern Recognit*, vol. 43, no. 8, pp. 2666–2680, Aug. 2010, doi: 10.1016/j.patcog.2010.03.004.
- [78] E. Zendejas Hernández, “Perfilómetro tridimensional basado en la fusión de datos obtenidos por corrimiento de fase y triangulación,” Tesis de maestría, Instituto Tecnológico de Chihuahua, Chihuahua, 2016.
- [79] E. O. Doebelin, *Measurement systems : application and design*. McGraw-Hill, 1990.
- [80] R. E. Walpole, R. H. Myers, S. L. Myers, and R. Cruz, *Probabilidad y estadística para ingenieros*. Prentice Hall, 1999. Accessed: Jun. 03, 2024. [Online]. Available: <https://latam.casadellibro.com/libro-probabilidad-y-estadistica-para-ingenieros/9789701702642/844241>

ANEXOS

ANEXO 1.
ESPECIFICACIONES DEL PROYECTOR ACER K335

Image	
Native Resolution	1280 x 800
Maximum Resolution	1920 x 1080
Standard Mode Brightness	1000 lm
Native Aspect Ratio	16:10
Compatible Aspect Ratio	16:09
	04:03
Contrast Ratio	10,000:1
Throw Ratio	1.4:1 (66"@78.74")
Digital Zoom	1.7x
Vertical Keystone Correction	-40°/+40°
Maximum Vertical Sync	120 Hz
Maximum Horizontal Sync	100 kHz
Minimum Vertical Sync	50 Hz
Minimum Horizontal Sync	30 kHz
Color Supported	1.07 Billion Colors (30-bit)
Projection Lens	
Lens Type	Manual Focus
Maximum Lens Aperture	F/1.5
Maximum Focal Length	13.92 mm
Minimum Diagonal Image Size	30"

Maximum Diagonal Image Size	100"
Minimum Projection Distance	35.43"
Maximum Projection Distance	118.11"

ANEXO 2.
ESPECIFICACIONES DE LA CÁMARA POINT GREY CMLN-13S2C-CS

Resolution	1296 x 964
Frame Rate	18 FPS
Megapixels	1.3 MP
Chroma	Color
Sensor Name	Sony ICX445
Sensor Type	CCD
Readout Method	Global shutter
Sensor Format	1/3"
Pixel Size	3.75 μ m
Lens Mount	CS-mount
ADC	12-bit
Quantum Efficiency Blue (% at 470 nm)	46
Quantum Efficiency Green (% at 525 nm)	50
Quantum Efficiency Red (% at 640 nm)	48
Temporal Dark Noise (e-)	7.65
Saturation Capacity (e-)	6365
Dynamic Range (dB)	57.85
Gain Range	0 dB to 24 dB

Exposure Range	0.01 ms to 10 seconds
Trigger Modes	Standard, bulb, skip frames, overlapped
Partial Image Modes	Pixel binning, ROI
Image Processing	Color/Greyscale conversion, gamma, white balance
User Sets	2 memory channels for custom camera settings
Flash Memory	256 KB non-volatile memory
Non-isolated I/O Ports	2 bi-directional
Serial Port	1 (over non-isolated I/O)
Auxiliary Output	3.3 V, 150 mA maximum
Interface	USB 2.0
Power Requirements	4.745 to 5.25 V via Mini-B USB 2.0 interface or JST 7-pin GPIO connector
Power Consumption (Maximum)	2 W
Dimensions	25.5 mm x 44 mm x 41 mm
Mass	37 grams
Machine Vision Standard	IIDC v1.31
Compliance	CE, FCC, KCC, RoHS. The ECCN for this product is: EAR099.
Temperature (Operating)	0° to 45°C
Temperature (Storage)	-30° to 60°C
Humidity (Operating)	20 to 80% (no condensation)
Humidity (Storage)	20 to 95% (no condensation)

APÉNDICES

APÉNDICE 1.

Código fuente “Estimación de parámetros”

```
clear all; close all; clc;
tic

%mex run_persistence1d.cpp

% Sinusoidal Fringe pattern
A=128; %Bias
Value
B=127;
%Amplitude
T=52; %Pixels
per period

g = im2double(imread('vp1.tif'));
g=g(60:900,50:1230);
% mesh(g)
h = im2double(imread('vpref.tif'));
h=h(60:900,50:1230);
% g = im2double(imread('obj_simulated.tif'));
%
% h = im2double(imread('ref_simulated.tif'));

% figure
% title('Fringe pattern's cross-sectional view','fontsize',36);
% xlabel(['y - Axis (Pixels)'],'fontsize',18);
% ylabel('z - Axis (Intensity)','fontsize',18);
%
% grid on
% grid minor
% hold on
% % plot(Disc_phase_obj(:,1))
%
% %plot(Disc_phase_obj2(:,1))
% plot(255*g(1:200,200))
% % plot(Minimum_obj)
% % plot(Maximum_obj)
% % plot(Avg_obj)
% hold off
% c.FontSize = 20;
% set(gca,'FontSize',16)
% legend('Axis x = 200','Location','northeast');

yy=g(:,200);

[ysize xsize] =size(g);

fs=1000*2*pi*(1/T); %Frequency sample
```

```

[gF,D]=lowpass(g,5,fs);      %Filtering Noise (object)
hF=lowpass(h,5,fs);        %Filtering Noise (Reference)
%% Display example of filtering
yy=g(:,285,1);

yyy=gF(:,285,1);
% plot(yy)
% hold on
% plot(yyy)
% hold off
%%
%% Power spectrum as a function of frequency (Noisy signal)
y = fft(yy);
n = length(yy);            % number of samples
f = (0:n-1)*(fs/n);        % frequency range
power = abs(y).^2/n;        % power of the DFT

% plot(f,power)
% xlabel('Frequency')
% ylabel('Power')

%% Power spectrum centered at 0 frequency (Noisy signal)
y0 = fftshift(y);          % shift y values
f0 = (-n/2:n/2-1)*(fs/n);  % 0-centered frequency range
power0 = abs(y0).^2/n;     % 0-centered power

% plot(f0,power0)
% xlabel('Frecuencia ciclos/píxeles')
% ylabel('Potencia')
% grid on
% grid minor

%% %% Power spectrum as a function of frequency (Filtered signal)
%% y1 = fft(yy1);
%% n = length(yy1);        % number of samples
%% f1 = (0:n-1)*(fs/n);    % frequency range
%% power1 = abs(y).^2/n;   % power of the DFT
%%
%% plot(f1,power1)
%% xlabel('Frequency')
%% ylabel('Power')
%%
%% %% Power spectrum centered at 0 frequency (Filtered signal)
%% y01 = fftshift(y1);     % shift y values
%% f01 = (-n/2:n/2-1)*(fs/n); % 0-centered frequency range
%% power01 = abs(y01).^2/n; % 0-centered power
%%
%% plot(f01,power01)
%% xlabel('Frequency')
%% ylabel('Power')
%% %% %%
%% Values above the threshold (object)

```

```

% mesh(gF)

xx=(1:ysize);

for x=1:xsize
    max1=max(gF(:,x));
    min1=min(gF(:,x));
    %     threshold_max = ((max1+min1)/2)-0.1; % your value here
    %     threshold_min = ((max1+min1)/2)-0.1; % your value here
    threshold_max = 0.1; % your value here
    threshold_min = 0.1; % your value here
    [t0_pos,s0_pos,t0_neg,s0_neg]= crossing_V7(gF(:,x),xx,threshold_max,'linear');
% positive (pos) and negative (neg) slope crossing points
% ind => time index (samples)
% t0 => corresponding time (x) values
% s0 => corresponding function (y) values , obviously they must be equal to
"threshold"

%     figure(1)
%
plot(xx,gF(:,x),t0_pos,s0_pos,'+r',t0_neg,s0_neg,'+g','linewidth',2,'markersize',1
2);grid on
%     legend('signal envelope','positive slope crossing points','negative slope
crossing points');

    max_pos_obj=ceil(t0_pos);
    max_neg_obj=floor(t0_neg);

% if x==400
%
%     [t0_pos,s0_pos,t0_neg,s0_neg]=
crossing_V7(gF(:,x),xx,threshold_max,'linear');
%     figure
%     title('Vista de sección transversal de patrón de
franjas','fontsize',18);
%     xlabel('Eje y - (Píxeles)','fontsize',18);
%     ylabel('Eje z - (Intensidad normalizada)','fontsize',18);
%
%     grid on
%     grid minor
%     hold on
%
plot(xx,gF(:,x),t0_pos,s0_pos,'+r',t0_neg,s0_neg,'+g','linewidth',1,'markersize',1
2);
%     hold off
%     c.FontSize = 20;
%     set(gca,'FontSize',16)
%     legend('Señal','Puntos de cruce con pendiente positiva','Puntos de cruce
con pendiente negativa','Location','northwest');
% end

%% Values below the threshold (object)

```

```

    threshold_min = 0.1; % your value here
    [t0_pos,s0_pos,t0_neg,s0_neg]= crossing_V7(gF(:,x),xx,threshold_min,'linear');
% positive (pos) and negative (neg) slope crossing points
% ind => time index (samples)
% t0 => corresponding time (x) values
% s0 => corresponding function (y) values , obviously they must be equal to
"threshold"

%     figure(2)
%
plot(xx,gF(:,x),t0_pos,s0_pos,'+r',t0_neg,s0_neg,'+g','linewidth',2,'markersize',1
2);grid on
%     legend('signal envelope','positive slope crossing points','negative slope
crossing points');

    min_pos_obj=floor(t0_pos);
    min_neg_obj=ceil(t0_neg);

%% Sine fit for maximum and minimum detection (Object)

max_pos=length(max_pos_obj);
max_neg=length(max_neg_obj);

if max_pos<max_neg
    max_length=max_pos;
else
    max_length=max_neg;
end
min_obj=0;
idx_min_obj=0;
max_obj=0;
idx_max_obj=0;
for i=1:max_length %Index for Max or min arrays
    if (i==1 && max_pos_obj(i) < max_neg_obj(i))
        x_interval=(max_pos_obj(i):1:max_neg_obj(i));

    elseif (i==1 && max_pos_obj(i) > max_neg_obj(i))
        x_interval=(1:1:max_neg_obj(i));

    else
        x_interval=(min_neg_obj((i-1)):1:max_neg_obj(i));
    end

    if (x==285)
        tttt=0;
    end
    tf=isempty(x_interval);
    y_to_fit=transpose(gF(x_interval,x));
    [SineP]=sineFit(x_interval,y_to_fit,0);
    y_fit=SineP(1)+SineP(2)*sin(2*pi*SineP(3)*x_interval+SineP(4));

```

```
if i==1
    idx_max_obj(i)=find(y_to_fit==max(y_to_fit));
    max_obj(i)=(max(y_to_fit));
else
    idx_max_obj(i)=min_neg_obj((i-1))+find(y_to_fit==max(y_to_fit));
    max_obj(i)=(max(gF(idx_max_obj(i)-1,x)));

    idx_min_obj(i-1)=min_neg_obj((i-1))+find(y_to_fit==min(y_to_fit))-1;
    min_obj(i-1)=min(y_to_fit);

end

end

if (x==1009)
    tttt=0;
end
xq = 1:1:ysize;

x2=idx_max_obj(:)-1;
v2=max_obj(:);

x1=idx_min_obj(:);
v1=min_obj(:);

Minimum_obj = interp1(x1,v1,xq,'linear');
Maximum_obj = interp1(x2,v2,xq,'linear');
Avg_obj=(Maximum_obj+Minimum_obj)/2;

phase_estimated2=(transpose(gF(:,x))-Avg_obj)./(Maximum_obj-Avg_obj);

Disc_phase_obj(:,x)=abs(asin(phase_estimated2));
Disc_phase_obj2(:,x)=abs(acos(phase_estimated2));

% if (x == 400)
%     figure
%     title('Interpolación lineal','fontsize',18);
%     xlabel('Eje y - (Píxeles)','fontsize',18);
%     ylabel('Eje z - (Intensidad normaliza)','fontsize',18);
%
%     grid on
%     grid minor
%     hold on
%     plot(Disc_phase_obj(:,1))
%
%     plot(Disc_phase_obj2(:,400))
%     plot(gF(1:870,400))
%     plot(Minimum_obj)
%     plot(Maximum_obj)
%     plot(Avg_obj)
%     hold off
%     c.FontSize = 20;
%     set(gca,'FontSize',16)
```

```

%      legend('Patrón sinusoidal','Mínimo','Máximo','Promedio','northwest');
%      %legend('Fase envuelta','northwest');
%      dummy=1;
%      end

%      plot(Disc_phase_obj(:,x))
end
save('Disc_phase_obj.mat','Disc_phase_obj')

gF=hF;
%% Values above the threshold (Reference)
for x=1:xsize
    max1=max(gF(:,x));
    min1=min(gF(:,x));
%      threshold_max = ((max1+min1)/2)-0.1; % your value here
%      threshold_min = ((max1+min1)/2)-0.1; % your value here
    threshold_max = 0.15; % your value here
    threshold_min = 0.1; % your value here
    [t0_pos,s0_pos,t0_neg,s0_neg]= crossing_V7(gF(:,x),xx,threshold_max,'linear');
% positive (pos) and negative (neg) slope crossing points
% ind => time index (samples)
% t0 => corresponding time (x) values
% s0 => corresponding function (y) values , obviously they must be equal to
"threshold"
%      if x==1
%          figure(1)
%
plot(xx,gF(:,x),t0_pos,s0_pos,'+r',t0_neg,s0_neg,'+g','linewidth',2,'markersize',1
2);grid on
%          legend('signal envelope','positive slope crossing points','negative
slope crossing points');
%      end

    max_pos_obj=ceil(t0_pos);
    max_neg_obj=floor(t0_neg);

%% Values below the threshold (Reference)

    [t0_pos,s0_pos,t0_neg,s0_neg]= crossing_V7(gF(:,x),xx,threshold_min,'linear');
% positive (pos) and negative (neg) slope crossing points
% ind => time index (samples)
% t0 => corresponding time (x) values
% s0 => corresponding function (y) values , obviously they must be equal to
"threshold"

%      figure(2)
%
plot(xx,gF(:,x),t0_pos,s0_pos,'+r',t0_neg,s0_neg,'+g','linewidth',2,'markersize',1
2);grid on

```

```
% legend('signal envelope','positive slope crossing points','negative slope
crossing points');

min_pos_obj=floor(t0_pos);
min_neg_obj=ceil(t0_neg);

%         if (x == 1040)
%             www=0;
%         end

%% Sine fit for maximum and minimum detection (Reference)

max_pos=length(max_pos_obj);
max_neg=length(max_neg_obj);

if max_pos<max_neg
    max_length=max_pos;
else
    max_length=max_neg;
end

min_obj=0;
idx_min_obj=0;
max_obj=0;
idx_max_obj=0;

for i=1:max_length      %Index for Max or min arrays

    if (i==1 && max_pos_obj(i) < max_neg_obj(i))
        x_interval=(max_pos_obj(i):1:max_neg_obj(i));

    elseif (i==1 && max_pos_obj(i) > max_neg_obj(i))
        x_interval=(1:1:max_neg_obj(i));

    else
        x_interval=(min_neg_obj((i-1)):1:max_neg_obj(i));
    end

    tf=isempty(x_interval);
    y_to_fit=transpose(h(x_interval,x));
    [SineP]=sineFit(x_interval,y_to_fit,0);

%     if (x == 1040)
%         [SineP]=sineFit(x_interval,y_to_fit,1)
%     end

    y_fit=SineP(1)+SineP(2)*sin(2*pi*SineP(3)*x_interval+SineP(4));

%         if (x == 1040)
% %             figure
% %
% %             plot(y_fit)
```

```
%                end

    if i==1
        idx_max_obj(i)=find(y_fit==max(y_fit));
        max_obj(i)=(max(y_fit));
    else
        idx_max_obj(i)=min_neg_obj((i-1))+find(y_fit==max(y_fit));
        max_obj(i)=(max(y_fit));

        idx_min_obj(i-1)=min_neg_obj((i-1))+find(y_fit==min(y_fit));
        min_obj(i-1)=min(y_fit);

    end

end

xq = 1:1:ysize;

x2=idx_max_obj(:);
v2=max_obj(:);

x1=idx_min_obj(:);
v1=min_obj(:);

Minimum_obj = interp1(x1,v1,xq,'linear');
Maximum_obj = interp1(x2,v2,xq,'linear');
Avg_obj=(Maximum_obj+Minimum_obj)/2;

phase_estimated2=(transpose(gF(:,x))-Avg_obj)./(Maximum_obj-Avg_obj);

Disc_phase_ref(:,x)=abs(asin(phase_estimated2));
%     if (x == 61)
%         plot(Disc_phase_ref(:,x));
%         hold on;
%         plot(Avg_obj);
%         plot(Minimum_obj);
%         plot(Maximum_obj);
%     end
%     plot(Disc_phase_obj(:,x))
end

save('Disc_phase_ref.mat','Disc_phase_ref')
toc
beep
%%

clear all; close all; clc;

m=matfile('Disc_phase_obj.mat');
dsc_obj_init = m.Disc_phase_obj;
m=matfile('Disc_phase_ref.mat');
dsc_ref_init = m.Disc_phase_ref;
```

```
% m=matfile('Disc_phase_obj.mat');
% dsc_obj = m.Disc_phase_obj;
% m=matfile('Disc_phase_ref.mat');
% dsc_ref = m.Disc_phase_ref;

% dsc_obj = dsc_obj1(100:800,:);
% dsc_ref = dsc_ref1(100:800,:);

dsc_obj_init(isnan(dsc_obj_init))==0;
dsc_ref_init(isnan(dsc_ref_init))==0;

% T=52;
% fs=1000*2*pi*(1/T);           %Frequency sample
%
%
% % dsc_obj=lowpass(dsc_obj_init,5,fs);      %Filtering Noise (object)
% % dsc_ref=lowpass(dsc_ref_init,5,fs);      %Filtering Noise (Reference)
% %
% % mesh(dsc_ref)
% % mesh(dsc_obj)

dsc_obj=dsc_obj_init;
dsc_ref=dsc_ref_init;

[ysize xsize]=size(dsc_obj);

% plot(dsc_obj(:,28))

plot(dsc_obj(:,1))
hold on
plot(dsc_ref(:,1))
mesh(dsc_obj)

%% Values above the threshold (object)

xx=(1:ysize);
threshold_max = 0.8; % your value here
threshold_min = 0.25;
for x=1:xsize
[t0_pos,s0_pos,t0_neg,s0_neg]=
crossing_V7(dsc_obj(:,x),xx,threshold_max,'linear'); % positive (pos) and negative
(neg) slope crossing points
% ind => time index (samples)
% t0 => corresponding time (x) values
% s0 => corresponding function (y) values , obviously they must be equal to
"threshold"
if x==400
    figure
    title('Vista de sección transversal de fase discontinua','fontsize',18);
    xlabel('Eje y - (Píxeles)','fontsize',18);
    ylabel('Eje z - (Radianes)','fontsize',18);
```

```
    grid on
    grid minor
    hold on

plot(xx,dsc_obj(:,x),t0_pos,s0_pos,'+r',t0_neg,s0_neg,'+g','markersize',12);
    hold off
    c.FontSize = 20;
    set(gca,'FontSize',16)
    legend('Señal','Puntos de cruce con pendiente positiva','Puntos de cruce
con pendiente negativa');
    dummy=1;
end
max_pos_obj=ceil(t0_pos);
max_neg_obj=floor(t0_neg);

[t0_pos,s0_pos,t0_neg,s0_neg]=
crossing_V7(dsc_obj(:,x),xx,threshold_min,'linear'); % positive (pos) and negative
(neg) slope crossing points

if x==730
    figure
    title('Vista de sección transversal de fase discontinua','fontsize',18);
    xlabel('Eje y - (Píxeles)','fontsize',18);
    ylabel('Eje z - (Radianes)','fontsize',18);

    grid on
    grid minor
    hold on

plot(xx,dsc_obj(:,x),t0_pos,s0_pos,'+r',t0_neg,s0_neg,'+g','linewidth',2,'markersi
ze',12);
    hold off
    c.FontSize = 20;
    set(gca,'FontSize',16)
    legend('Señal','Puntos de cruce con pendiente positiva','Puntos de cruce
con pendiente negativa','Location','southeast');
end

% figure(1)
%
plot(xx,dsc_obj(:,x),t0_pos,s0_pos,'+r',t0_neg,s0_neg,'+g','linewidth',2,'markersi
ze',12);grid on
% legend('signal envelope','positive slope crossing points','negative slope
crossing points');

min_pos_obj=floor(t0_pos);
min_neg_obj=ceil(t0_neg);

if length(max_pos_obj)<=length(max_neg_obj)
    max_length=length(max_pos_obj);
else
    max_length=length(max_neg_obj);
```

```
end

for i=1:max_length      %Index for Max or min arrays
    if max_pos_obj(i) < max_neg_obj(i)
        x_interval=(max_pos_obj(i):1:max_neg_obj(i));

        y=dsc_obj(x_interval,x);
        idx_max_obj(i)=max_pos_obj(i)+find(y==max(y))-1;

    elseif i+1 <= length(max_neg_obj)

        x_interval=(max_pos_obj(i):1:max_neg_obj(i+1));

        y=dsc_obj(x_interval,x);
        idx_max_obj(i)=max_pos_obj(i)+find(y==max(y))-1;
    end

end

end

if length(min_pos_obj)<=length(min_neg_obj)
    min_length=length(min_pos_obj);
else
    min_length=length(min_neg_obj);
end

% min_length=length(min_neg_obj)-1;

for i=1:min_length      %Index for Max or min arrays
    if min_neg_obj(i) < min_pos_obj(i)
        x_interval=(min_neg_obj(i):1:min_pos_obj(i));

        y=dsc_obj(x_interval,x);
        idx_min_obj(i)=min_neg_obj(i)+find(y==min(y))-1;

    elseif i+1 <= length(min_pos_obj)
        x_interval=(min_neg_obj(i):1:min_pos_obj(i+1));

        y=dsc_obj(x_interval,x);
        idx_min_obj(i)=min_neg_obj(i)+find(y==min(y))-1;
    end

end

end

idx_max_obj_sz=size(idx_max_obj);
idx_min_obj_sz=size(idx_min_obj);

if idx_min_obj(1)<idx_max_obj(1) % Slope 0 = negative slope --- Slope 1 =
positive slope
    idx_min_obj=idx_min_obj(2:idx_min_obj_sz(2));
end
```

```
if idx_min_obj(1)<idx_max_obj(1) % Slope 0 = negative slope --- Slope 1 =
positive slope
    slope=0;
    i=1;
else
    slope=1;
    i=0;
end
for y=1:ysize
    if x==11
        kkkk=1;
    end
    if(find(idx_min_obj==y))
        slope=1;

    end
    if(find(idx_max_obj==y))
        slope=0;
        i=i+1;
    end
    if slope==1
        p_est(y,x)=dsc_obj(y,x)+(i*pi);
    end
    if slope==0
        p_est(y,x)=(i*pi)-dsc_obj(y,x);
    end
end
%
if x==400
    figure % Objeto
    hold on
    title('Vista de sección transversal de fase envuelta','fontsize',18);

    xlabel('Eje y - (Píxeles)','fontsize',18);
    ylabel('Eje z - (Radianes)','fontsize',18);

    grid on
    grid minor
    plot(dsc_obj(52:300,x))
    plot(p_est(52:300,x))

    c.FontSize = 20;
    set(gca,'FontSize',16)
    legend('Fase envuelta','Fase desenvuelta');

    dummy=1;

end
end
```

```

%% Values above the threshold (Ref)
[ysize xsize]=size(dsc_ref);
xx=(1:ysize);
for x=1:xsize
[t0_pos,s0_pos,t0_neg,s0_neg]=
crossing_V7(dsc_ref(:,x),xx,threshold_max,'linear'); % positive (pos) and negative
(neg) slope crossing points
% ind => time index (samples)
% t0 => corresponding time (x) values
% s0 => corresponding function (y) values , obviously they must be equal to
"threshold"

% % figure(1)
% %
plot(xx,dsc_obj(:,x),t0_pos,s0_pos,'+r',t0_neg,s0_neg,'+g','linewidth',2,'markersi
ze',12);grid on
% % legend('signal envelope','positive slope crossing points','negative slope
crossing points');

max_pos_ref=ceil(t0_pos);
max_neg_ref=floor(t0_neg);

[t0_pos,s0_pos,t0_neg,s0_neg]=
crossing_V7(dsc_ref(:,x),xx,threshold_min,'linear'); % positive (pos) and negative
(neg) slope crossing points

if x==0
figure
title('Vista de sección transversal de fase discontinua','fontsize',18);
xlabel('Eje y - (Píxeles)','fontsize',18);
ylabel('Eje z - (Radianes)','fontsize',18);

grid on
grid minor
hold on

plot(xx,dsc_ref(:,x),t0_pos,s0_pos,'+r',t0_neg,s0_neg,'+g','linewidth',2,'markersi
ze',12);
hold off
c.FontSize = 20;
set(gca,'FontSize',16)
legend('Señal','Puntos de cruce con pendiente positiva','Puntos de cruce
con pendiente negativa');
end

% figure(1)
%
plot(xx,dsc_obj(:,x),t0_pos,s0_pos,'+r',t0_neg,s0_neg,'+g','linewidth',2,'markersi
ze',12);grid on

```

```
% legend('signal envelope','positive slope crossing points','negative slope
crossing points');

min_pos_ref=floor(t0_pos);
min_neg_ref=ceil(t0_neg);

max_length=length(max_pos_ref);

if length(max_pos_ref)<=length(max_neg_ref)
    max_length=length(max_pos_ref);
else
    max_length=length(max_neg_ref);
end

for i=1:max_length      %Index for Max or min arrays
    if i==30
        test_value=0;
    end

    if max_pos_ref(i) < max_neg_ref(i)
        x_interval=(max_pos_ref(i):1:max_neg_ref(i));

        y=dsc_ref(x_interval,x);
        idx_max_ref(i)=max_pos_ref(i)+find(y==max(y))-1;

    elseif i+1 <= length(max_neg_ref)
        x_interval=(max_pos_ref(i):1:max_neg_ref(i+1));

        y=dsc_ref(x_interval,x);
        idx_max_ref(i)=max_pos_ref(i)+find(y==max(y))-1;
    end

end

if length(min_pos_ref)<=length(min_neg_ref)
    min_length=length(min_pos_ref);
else
    min_length=length(min_neg_ref);
end

% min_length=length(min_neg_ref)-1;
for i=1:min_length      %Index for Max or min arrays
    if min_neg_ref(i) < min_pos_ref(i)
        x_interval=(min_neg_ref(i):1:min_pos_ref(i));

        y=dsc_ref(x_interval,x);
        idx_min_ref(i)=min_neg_ref(i)+find(y==min(y))-1;

    elseif i+1 <= length(min_pos_ref)
```

```
        x_interval=(min_neg_ref(i):1:min_pos_ref(i+1));

        y=dsc_ref(x_interval,x);
        idx_min_ref(i)=min_neg_ref(i)+find(y==min(y))-1;
    end

end

idx_max_ref_sz=size(idx_max_ref);
idx_min_ref_sz=size(idx_min_ref);

if idx_min_ref(1)<idx_max_ref(1) % Slope 0 = negative slope --- Slope 1 =
positive slope
    idx_min_ref=idx_min_ref(2:idx_min_ref_sz(2));
end

if idx_min_ref(1)<idx_max_ref(1)
    slope=0;
    i=1;
else
    slope=1;
    i=0;
end
for y=1:ysize
    if(find(idx_min_ref==y))
        slope=1;

    end
    if(find(idx_max_ref==y))
        slope=0;
        i=i+1;
    end
    if slope==1
        p_est_ref(y,x)=dsc_ref(y,x)+(i*pi);
    end
    if slope==0
        p_est_ref(y,x)=(i*pi)-dsc_ref(y,x);
    end
end

end

object=p_est-p_est_ref;

plot(p_est_ref(:,393))
hold on
plot(p_est(:,393))
plot(object(:,393))
```

```

%% Figura español
% figure
% % mesh(object(100:800,1:xsize));
% mesh(object(:,1:xsize));
% % contourf(unwrap)
% % shading interp;
% colormap jet; %%
https://www.mathworks.com/help/matlab/ref/colormap.html
% title('Vista 3D del objeto','fontsize',18);
% xlabel('Eje y - (Píxeles)','fontsize',18);
% ylabel('Eje x - (Píxeles)','fontsize',18);
% zlabel('Eje z - (Radianes)','fontsize',18);
% xlim([0 1280])
% ylim([0 960])
% % zlim([0 1])
% grid on
% grid minor
% colorbar
% c=colorbar;
% c.Label.String = 'Eje z - Radianes';
% c.FontSize = 16;
% set(gca,'FontSize',13)
%
% view(135,45);
%%
%% Figura english
figure('DefaultAxesFontSize',30)
mesh(object(100:800,1:xsize));
% mesh(object(:,1:xsize));
% contourf(unwrap)
% shading interp;

colormap jet; %%
https://www.mathworks.com/help/matlab/ref/colormap.html
title('3D view','fontsize',30);

hold on
xlabel('y - Axis (Pixels)','fontsize',24);
ylabel('x - Axis (Pixels)','fontsize',24);
zlabel('z - Axis (Radians)','fontsize',24);

xlim([0 1280])
ylim([0 960])
% zlim([0 1])
grid on
grid minor
colorbar
c=colorbar;
c.Label.String = 'z - Axis - Radians';
c.FontSize = 18;
set(gca,'FontSize',13)

view(-45,45);

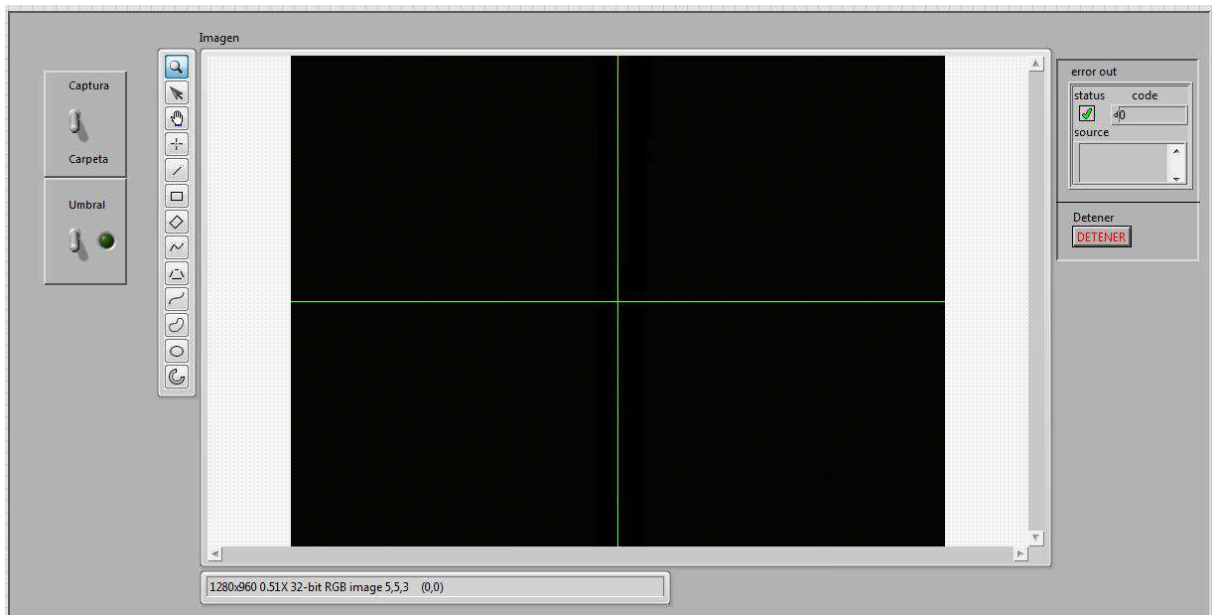
```

```
%%  
obj=object(100:800,1:xsize);  
h=obj(420,580)+1.21  
mesh(obj)  
obj_gray=mat2gray(obj);  
  
imshow(obj_gray)  
imwrite(obj_gray,'obj2.tif');  
save('obj2.mat','obj')  
beep
```

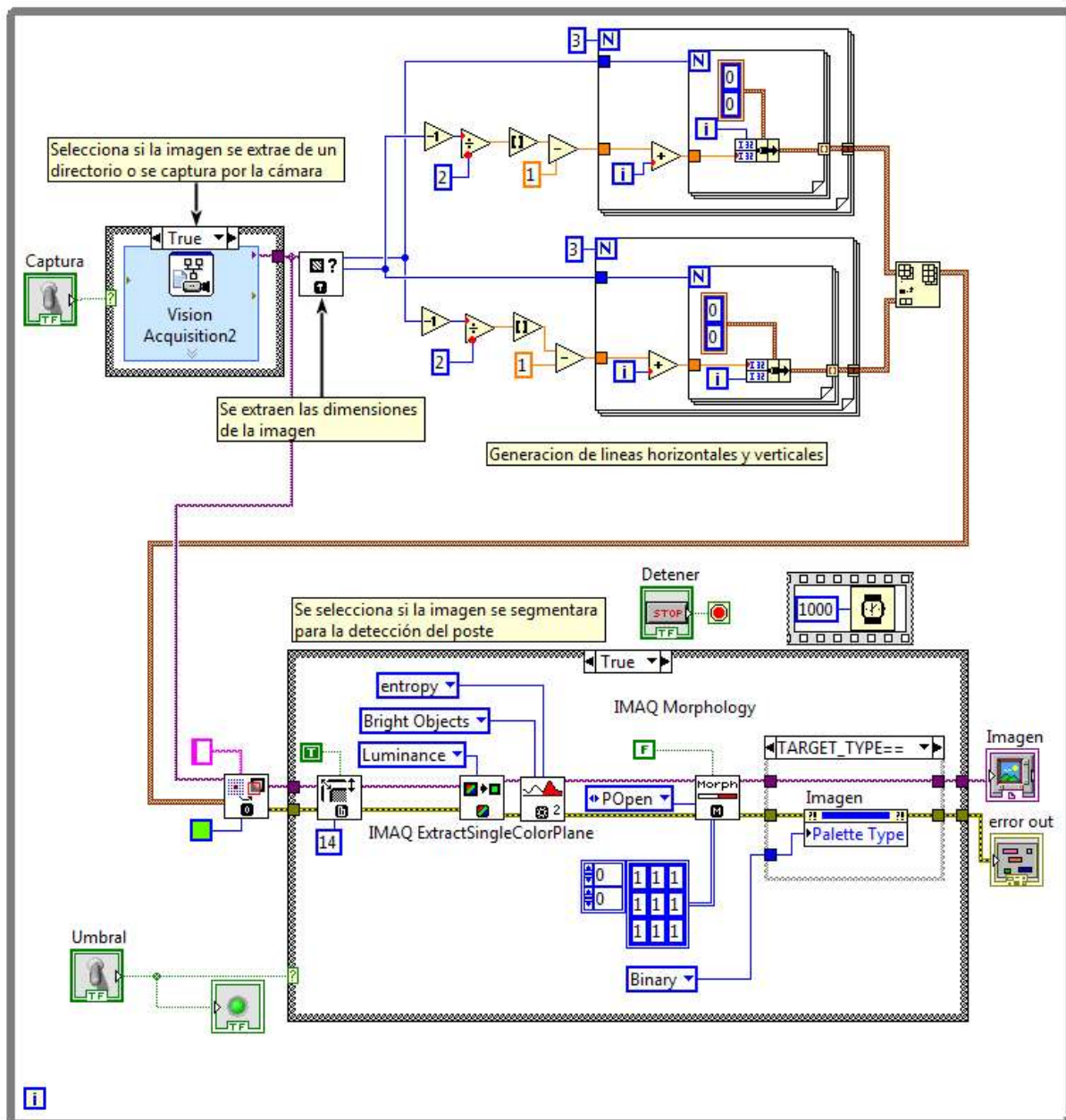
APÉNDICE 2.

SUBVI “Captura de Cruz”

Interfaz de usuario

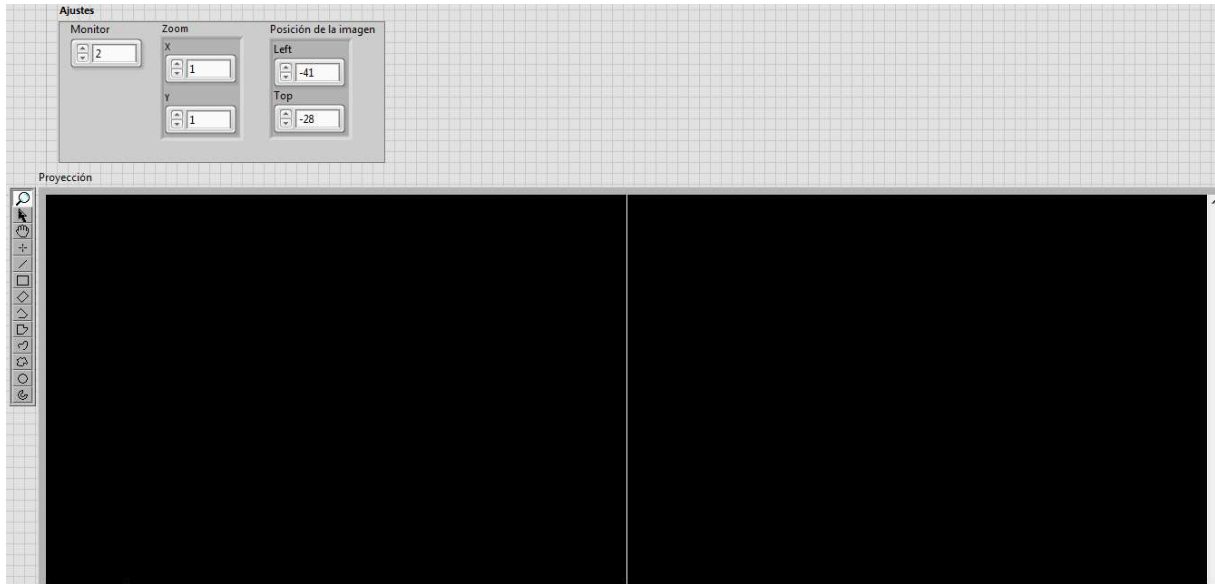


Código fuente

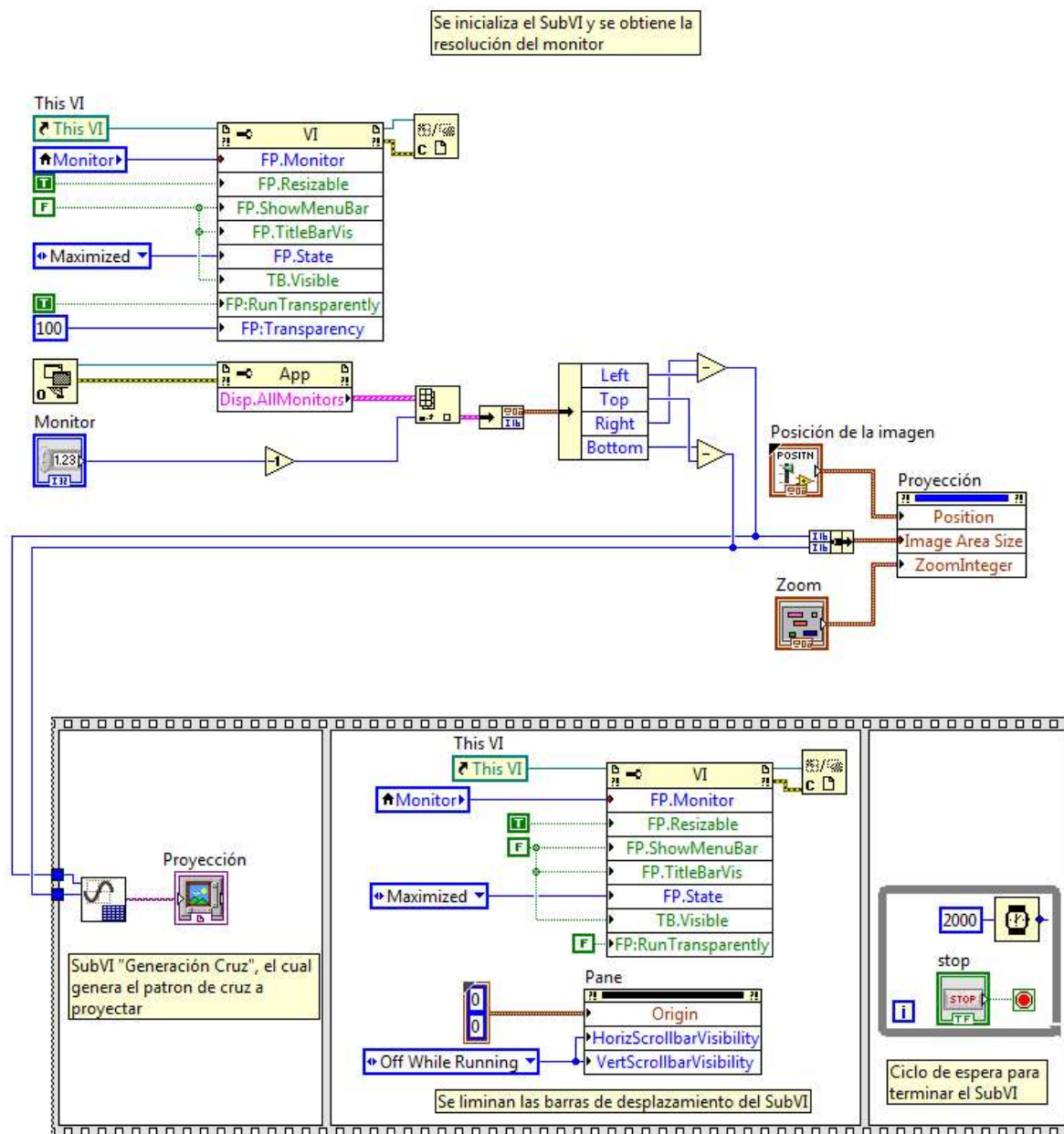


APÉNDICE 3. SUBVI “Proyección de Cruz”

Interfaz de usuario

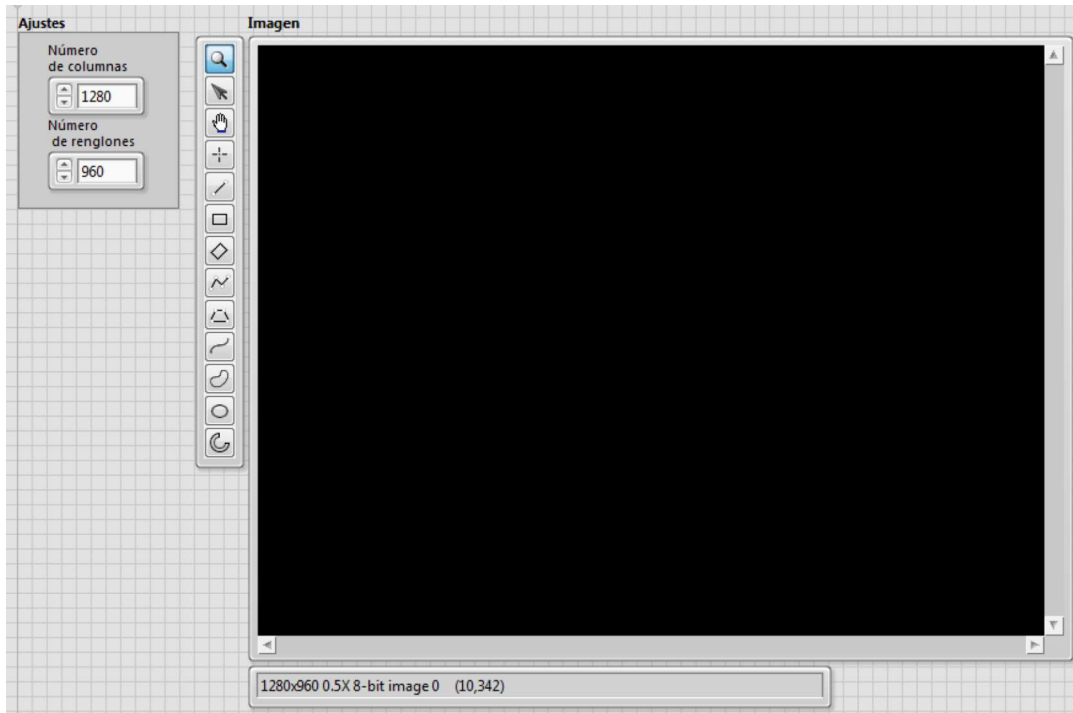


Código fuente

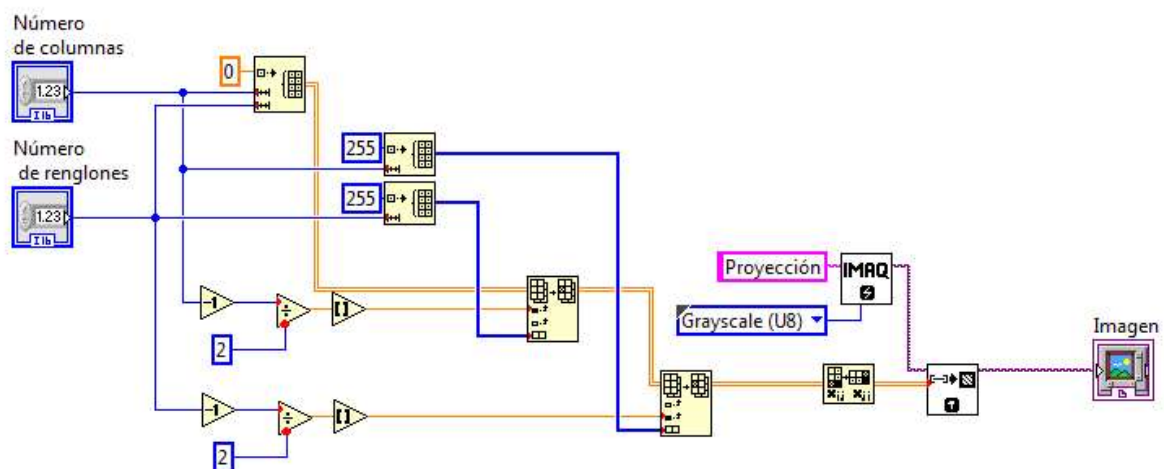


3.1 SUBVI “Generación de Cruz”

Interfaz de usuario



Código fuente



APÉNDICE 4.

“Análisis de datos”

```

clear all; close all; clc                                     %Se
inicializa Matlab borrando los registros

Z=[4.8382, 5.0048, 4.9956, 5.0005, 5.0025, 4.9915, 4.9809, 5.0202, 5.0149...
    , 4.9753, 4.9591, 4.9537, 4.9440, 4.9859, 4.9683, 4.9408, 4.9494, 4.9618...
    , 4.9910, 5.0139, 5.0081, 4.9500, 4.9184, 4.9400, 4.9548, 4.9567, 4.9582,
    4.9834, 4.9768, 4.9607];

figure('Name','Venatana de Gráfica de Histograma',...         %Se da
formato a la ventana cambiando el nombre, desactivando número de ventana
    'NumberTitle','off','color','white')                       %Color
de fondo blanco
nbins=30;
%Numero de barras del Histograma
histfit(Z,nbins,'normal')                                     %Se
genera y despliega el histograma
ylabel('Frecuencia de las mediciones');                       %Se
asigna nombre al eje vertical
xlabel('Altura del objeto (Radianes)');
%Se asigna nombre al eje horizontal
title('Histograma de mediciones');                             %Se
agrega título al Histograma
grid on
E=0;

Media = mean(Z);                                              %Se
calcula la media

mediana = median(Z);                                          %Se
calcula la mediana

moda = mode(Z);                                               %Se
calcula la moda

varianza = var(Z);                                            %Se
calcula la varianza de la muestra

varianza1=var(Z,1);                                           %Se
calcula la varianza de la población

Desv_est=std(Z);                                              %Se
calcula la desviación estándar

% Media,mediana,moda,varianza,varianza1,Desv_est             %Se
despliegan los valores de media, mediana, moda, varianza y desviación estandar

Orden_Medicion_Asc=sort(Z);

```

```

grafica=normcdf(Orden_Medicion_Asc,Media,Desv_est);
figure('Name','Conjunto de datos',...           %Se da
formato a la ventana cambiando el nombre, desactivando número de ventana
    'NumberTitle','off','color','white')         %Color
de fondo blanco
plot(Orden_Medicion_Asc,grafica)
normplot(Z)
ylabel('Probabilidad');                         %Se
asigna nombre al eje vertical
xlabel('Valores de alturas (Radianes)');
%Se asigna nombre al eje horizontal
title('Gráfico de probabilidad normal');        %Se
agrega titulo al Histograma
Prueba_ChiCuadrada_Algoritmo_de_estimacion_de_parametros=chi2gof(Z)

```