

**INSTITUTO TECNOLÓGICO DE CHIHUAHUA**  
**DIVISIÓN DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN**

---

***“CORRECCIÓN DE OCLUSIONES PARA LA  
OBTENCIÓN DE INFORMACIÓN 3D EN OBJETOS  
COMPLEJOS”***

***TESIS***

**QUE PARA OBTENER EL GRADO DE**

***MAESTRO EN CIENCIAS EN INGENIERÍA  
ELECTRÓNICA***

**PRESENTA:**

***DANIEL EDUARDO LUÉVANO PORTILLO***

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

---



**TECNOLÓGICO NACIONAL  
DE MÉXICO**



**CHIHUAHUA, CHIH., SEPTIEMBRE 2019**



Chihuahua, Chih. 19 de septiembre de 2019

**C. DANIEL EDUARDO LUÉVANO PORTILLO  
PRESENTE**

Por este conducto le comunico que a propuesta del Jurado de Examen, la División de Estudios de Posgrado e Investigación ha concedido autorización para la impresión de su tesis para obtener el grado de Maestro en Ciencias en Ingeniería Electrónica, cuyo título es:

***"Corrección de oclusiones para la obtención de información 3D en objetos complejos"***

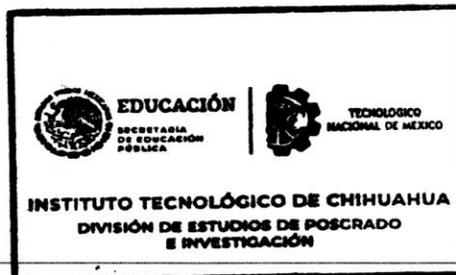
La tesis presenta el siguiente contenido de capítulos:

- I Introducción
- II Técnicas de perfilometría 3D
- III Técnica de proyección de franjas
- IV Imágenes digitales
- V Técnicas de restauración de imágenes
- VI Algoritmo senoidal de inpainting

**ATENTAMENTE**

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

**MTRO. LUIS CARDONA CHACÓN  
JEFE DE LA DIVISIÓN DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN**





# EDUCACIÓN

SECRETARÍA DE EDUCACIÓN PÚBLICA



TECNOLÓGICO  
NACIONAL DE MÉXICO

Instituto Tecnológico de Chihuahua

"2019, Año del Caudillo del Sur, Emiliano Zapata"

Chihuahua, Chih. 16 de agosto de 2019

**MTR. LUIS CARDONA CHACÓN**  
**JEFE DE LA DIVISIÓN DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN**  
**PRESENTE**

Por medio de la presente notificamos a usted que en cumplimiento de los requerimientos para la obtención del grado de Maestro en Ciencias en Ingeniería Electrónica, el documento de tesis del **C. DANIEL EDUARDO LUÉVANO PORTILLO**, ha sido aprobado y aceptado para su impresión. El título de la tesis es:

***"Corrección de oclusiones para la obtención de información 3D en objetos complejos"***

Por lo que 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. MARCELINO ANGUIANO MORALES**  
**DIRECTOR DE TESIS**

**DR. LUIS FRANCISCO CORRAL MARTÍNEZ**  
**MIEMBRO DEL JURADO DE EXAMEN**

**DR. ÓSCAR ARTURO CHÁVEZ LÓPEZ**  
**MIEMBRO DEL JURADO DE EXAMEN**

**MTR. ISMAEL ARTURO GARDUÑO**  
**WILCHES**  
**MIEMBRO DEL JURADO DE EXAMEN**



Ave Tecnológico No. 2909 Col. 10 de Mayo C.P. 31310, Chihuahua, Chih. México  
Tel. 01 (614) 201 2000, (614) 413 5187, Ext. 2150 e-mail: [dir.chihuahua@tecnm.mx](mailto:dir.chihuahua@tecnm.mx)  
[www.tecnm.mx](http://www.tecnm.mx) | [www.itchihuahua.edu.mx](http://www.itchihuahua.edu.mx)





## **CARTA CESIÓN DE DERECHOS**

En la ciudad de Chihuahua el día **26 de septiembre de 2019**, el que suscribe **C. DANIEL EDUARDO LUÉVANO PORTILLO** de la Maestría en Ciencias en Ingeniería Electrónica, con número de control **G16061361**, 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 del **Dr. Marcelino Anguiano Morales** y cede los derechos del trabajo titulado **"Corrección de oclusiones para la obtención de información 3D en objetos complejos"**, 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.

  
**C. DANIEL EDUARDO LUÉVANO PORTILLO**



## RESUMEN

### CORRECCIÓN DE OCLUSIONES PARA LA OBTENCIÓN DE INFORMACIÓN 3D EN OBJETOS COMPLEJOS

Ing. Daniel Eduardo Luévano Portillo  
Maestría en Ciencias en Ingeniería Electrónica  
División de Estudios de Posgrado e Investigación del  
Instituto Tecnológico de Chihuahua  
Chihuahua, Chih., 2019  
Director de Tesis: Dr. Marcelino Anguiano Morales

La reconstrucción de objetos en 3D utilizando perfilometría por proyección de franjas es una técnica óptica de no contacto muy popular que tiene diversos campos de aplicación. En esta técnica, se utilizan algoritmos para obtener la fase desenvuelta, la cual contiene la información topográfica 3D del objeto.

El problema más difícil para la reconstrucción basada en esta técnica es la presencia de oclusiones en regiones de la superficie del objeto debido principalmente a pronunciados cambios de altura del objeto y al ángulo de proyección de franjas. Las oclusiones hacen que la reconstrucción 3D sea ambigua en regiones del objeto que no son visibles en la imagen de entrada, debido principalmente a la generación de singularidades en las franjas proyectadas o por interferencia de iluminación del área de trabajo. Estas regiones deben ser reconstruidas y visualmente aceptables.

En este trabajo, se implementa la técnica de *inpainting* para restaurar regiones discontinuas en el patrón de franjas generado en la superficie del objeto y así aumentar la precisión de la profundidad de recuperación de información. Logrando con ello una reconstrucción 3D más manejable y de mejor calidad. Además, este proceso es más eficiente en recursos y tiempo para combatir las oclusiones a diferencia de hacer capturas adicionales con diferentes ángulos del objeto.

El estudio de mediciones 3D robustas para objetos basados en imágenes con escenas complejas tiene un alto valor de investigación ya que maneja los problemas reales en ambientes prácticos de trabajo.

# ÍNDICE

<b>LISTA DE FIGURAS</b>		<i>viii</i>
<b>LISTA DE TABLAS</b>		<i>xi</i>
<b>CAPÍTULO I</b>	<b>INTRODUCCIÓN</b>	1
<b>CAPÍTULO II</b>	<b>TÉCNICAS DE PERFILOMETRÍA 3D</b>	4
2.1	Técnicas de contacto	5
2.2	Técnicas de no contacto	6
2.2.1	Técnica de visión estereoscópica	6
2.2.2	Técnica de escáner de tiempo de vuelo	7
2.2.3	Técnica de escáner 3D por luz estructurada.	7
<b>CAPÍTULO III</b>	<b>TÉCNICA DE PROYECCIÓN DE FRANJAS</b>	10
3.1	Parámetros geométricos y relación fase-altura	10
3.2	Técnicas de análisis de franjas	12
3.2.1	Técnica por corrimiento de fase	13
3.2.2	Técnica de perfilometría por transformada de Fourier	13
3.2.3	Desenvolvimiento de fase	15
<b>CAPÍTULO IV</b>	<b>IMÁGENES DIGITALES</b>	17
4.1	Imagen digital	17
4.2	Procesamiento digital de imágenes	18
<b>CAPÍTULO V</b>	<b>TÉCNICAS DE RESTAURACIÓN DE IMÁGENES</b>	23
5.1	<i>Partial Differential Equation (PDE)</i>	23
5.2	<i>Texture synthesis</i>	24
<b>CAPÍTULO VI</b>	<b>ALGORITMO SINUSOIDAL DE INPAINTING</b>	26
6.1	Movimiento Armónico simple	26
6.2	Algoritmo sinusoidal	27
6.2.1	Variables en la zona de restauración	31
6.2.1.1	Número de onda	31
6.2.1.2	Amplitud	32
6.2.1.3	Desplazamiento vertical	32

6.2.1.4	Constante de fase	33
6.2.2	Programación del algoritmo sinusoidal	33
<b>CAPÍTULO VII</b>	<b>ARREGLO EXPERIMENTAL</b>	36
7.1	Elementos del sistema	36
7.2	Proyector	37
7.3	Cámara y lente	38
7.4	Arreglo experimental	39
7.5	Software	40
7.5.1	MATLAB	41
7.5.1	NI LabVIEW	41
7.5.1	Uc480Viewer	42
7.5.1	Wolfram Mathematica	42
<b>CAPÍTULO VIII</b>	<b>RESULTADOS EXPERIMENTALES Y CONCLUSIONES</b>	44
8.1	Primera prueba de laboratorio	44
8.1.1	Reconstrucción de superficie compleja sin Inpainting	44
8.1.2	Reconstrucción de superficie compleja con Inpainting	45
8.2	Segunda prueba de laboratorio	47
8.2.1	Reconstrucción de objeto complejo sin Inpainting	47
8.2.2	Reconstrucción de objeto complejo con Inpainting	48
8.3	Conclusiones	49
	<b>REFERENCIAS</b>	51
	<b>ANEXO</b>	53

# LISTA DE FIGURAS

## **CAPÍTULO II TÉCNICAS DE PERFILOMETRÍA 3D**

- Figura 2.1 Técnicas de reconstrucción 3D 4
- Figura 2.2 Sistema de posicionamiento tridimensional explorando un motor de combustión interna. 5
- Figura 2.3 Sistemas de estereovisión comerciales. 7
- Figura 2.4 Escáner láser 3D de tiempo de vuelo utilizados en la industria. 7
- Figura 2.5 Ejemplo de proyección de franjas y la triangulación entre proyector, cámara y objeto. 8

## **CAPÍTULO III TÉCNICA DE PROYECCIÓN DE FRANJAS**

- Figura 3.1 Arreglo geométrico de proyección de franjas. 10
- Figura 3.2 Diagrama de flujo para el análisis de franjas. 12
- Figura 3.3 (a) Espectro de Fourier de un patrón de franjas y; (b) espectro seleccionado y trasladado al origen 14

## **CAPÍTULO IV IMÁGENES DIGITALES**

- Figura 4.1 Interacción de la radiación electromagnética con los objetos. 17
- Figura 4.2 Matriz de una imagen digital donde es señalado un pixel con coordenadas (x,y). 18
- Figura 4.3 Histograma con valores en extremo oscuro (a); histograma con valores en extremo brillante (b); histograma con bajo contraste (c); histograma con alto contraste (d). 20
- Figura 4.4 Filtros: (a) Pasa-baja; (b) Pasa-alto; (c) Pasa-banda. 21

## **CAPÍTULO V TÉCNICAS DE RESTAURACIÓN DE IMÁGENES**

- Figura 5.1 (a) Imagen con pequeñas oclusiones; (b) Imagen restaurada con algoritmo basado en PDE. 24
- Figura 5.2 (a) Imagen con múltiples texturas; (b) Resultado del algoritmo Texture synthesis. 25

## **CAPÍTULO VI ALGORITMO SINUSOIDAL DE INPAINTING**

Figura 6.1	(a) Resorte se estira de modo que el cuerpo tenga su desplazamiento máximo a partir del equilibrio; (b) Posición de equilibrio con el resorte relajado; (c) El resorte está totalmente comprimido.	27
Figura 6.2	Ejemplo de valores por cada pixel antes de la zona de oclusión.	28
Figura 6.3	Ejemplo de gráfica con los valores correspondientes a una fila de pixeles antes de la zona de oclusión con los valles y crestas identificados.	28
Figura 6.4	(a) función sinusoidal anterior a la zona de restauración; (b) función sinusoidal resultante al variar solo el valor de número de onda; (c) función sinusoidal posterior a la zona de restauración.	32
Figura 6.5	(a) función sinusoidal anterior a la zona de restauración; (b) función sinusoidal resultante al variar solo el valor de la amplitud; (c) función sinusoidal posterior a la zona de restauración.	32
Figura 6.6	(a) función sinusoidal anterior a la zona de restauración; (b) función sinusoidal resultante al variar solo el valor del desplazamiento vertical; (c) función sinusoidal posterior a la zona de restauración.	32
Figura 6.7	Ejemplo de gráfica en la zona de restauración con los valores de la función sinusoidal resultante al variar la amplitud, el número de onda, el desplazamiento vertical y la constante de fase variando linealmente.	33
Figura 6.8	Diagrama de flujo del algoritmo sinusoidal.	35
<b>CAPÍTULO VII ARREGLO EXPERIMENTAL</b>		
Figura 7.1	Diagrama de flujo del arreglo experimental	36
Figura 7.2	Proyector Toshiba DLP modelo TDP-FF1	37
Figura 7.3	Vista lateral del proyector donde “a” es la distancia de la lente y la pantalla y H es la altura desde el fondo de la imagen al centro de la lente.	38
Figura 7.4	Cámara CCD marca Thorlabs modelo DCU224C.	39
Figura 7.5	Lente óptica Thorlabs modelo MVL25.	39
Figura 7.6	Esquema del arreglo experimental.	39

Figura 7.7	Distancias del arreglo experimental.	40
Figura 7.8	Interface de la aplicación desarrollada en LabView para la generación de franjas.	41
Figura 7.9	Interface del Software uc480Viewer realizando la calibración del arreglo óptico.	42
Figura 7.10	Software Wolfram Mathematica haciendo una restauración de imagen con la función Inpaint.	43
<b>CAPÍTULO VIII RESULTADOS EXPERIMENTALES Y CONCLUSIONES</b>		
Figura 8.1	Superficie a reconstruir y su zona de oclusión.	44
Figura 8.2	(a) Mapa de fase envuelta del patrón con defecto; (b) Mapa de fase desenvuelta con defecto.	45
Figura 8.3	Objeto a reconstruir y su zona de oclusión restaurada por el algoritmo sinusoidal.	46
Figura 8.4	(a) Mapa de fase envuelta del patrón sin defecto; (b) Mapa de fase desenvuelta sin defecto.	46
Figura 8.5	Reconstrucción final 3D con el algoritmo desarrollado.	46
Figura 8.6	Objeto a reconstruir y su zona de oclusión.	47
Figura 8.7	Defecto en la reconstrucción 3D sin inpainting.	47
Figura 8.8	Perfil de línea del área de oclusión mostrando el error del salto de fase.	48
Figura 8.9	Restauración en la zona de oclusión	48
Figura 8.10	Reconstrucción 3D con inpainting.	49
Figura 8.11	Perfil de línea del área de oclusión mostrando la disminución del error.	49

# LISTA DE TABLAS

## **CAPÍTULO VII ARREGLO EXPERIMENTAL**

Tabla 7.1	Relación entre distancia de proyección y el tamaño de proyección medido en diagonal.	38
-----------	--	----

## CAPÍTULO I

### INTRODUCCIÓN

El estudio de la luz siempre ha sido un tema interesante para la humanidad ya que la vida depende de este fenómeno natural, toda la energía que se recibe en forma de luz alimenta al planeta y, por ende, a todas las especies que habitan en él. Por ello la luz ha sido objeto de estudio científico desde hace ya varios siglos.

Pitágoras (582 - 507 a.C.) planteó que la luz emanaba del ojo en forma de rayos luminosos que se propagaban en línea recta y chocaban con el cuerpo observado. Después Newton (1643-1727 d.C.) estableció el modelo corpuscular de la luz y Huygens (1629-1695 d.C.) formuló su teoría ondulatoria de la luz. Más tarde, Young (1773-1829 d.C.) descubrió la interferencia de la luz y concluyó que la luz es una onda transversal. Faraday (1791-1867 d.C.) descubrió que los efectos de los campos magnético y eléctrico asociados a la materia interactúan con la luz, y gracias a esto Maxwell (1831- 1879 d.C.) logró integrar los fenómenos ópticos, eléctricos y magnéticos en un mismo marco teórico. Esto proporcionó un mecanismo de propagación al campo electromagnético y se acuñó el concepto de ondas electromagnéticas.

Todo este tiempo de estudio ha permitido crear formas artificiales para producir luz, interactuar con ella y analizar sus efectos sobre la materia, lo cual es de especial interés en este trabajo para lograr una reconstrucción digital de un objeto utilizando luz y su interacción con éste mismo.

El escaneo óptico tridimensional (3D) es un campo reciente de investigación con gran importancia. Diversas técnicas han sido aplicadas exitosamente en varias áreas de medición 3D de no contacto como la tomografía médica, reconstrucción 3D de escenas, escaneo de rostro 3D, control de calidad industrial, y reconstrucción 3D de objetos complejos u objetos con fuerte o poca iluminación. Entre las técnicas de escaneo óptico 3D existentes, la perfilometría por proyección de franjas (PPF) destaca por su bajo costo, alta resolución y rápida adquisición de datos. Un procedimiento de PPF típico consiste en un patrón de franjas proyectado en los objetos de interés. Debido a la estructura 3D de los objetos, el patrón de

franjas será distorsionado. Posteriormente, la información 3D del objeto puede ser recuperada midiendo la cantidad de distorsión en el patrón de franjas expresada en términos de cambio de fase.

Mientras que muchos métodos tradicionales de PPF pueden desempeñarse satisfactoriamente en simulaciones, su rendimiento puede degradarse significativamente en la práctica. Esto es debido a la suposición de que los cambios de fase del patrón de franjas pueden ser observados completamente y capturados por un dispositivo de imagen, como una cámara digital. Sin embargo, este no es el caso ya que en realidad la calidad de las imágenes puede ser seriamente afectada por interferencias en el ambiente de trabajo o la misma naturaleza del objeto a analizar.

En esta tesis se propone una solución a la problemática de la reconstrucción 3D para objetos complejos en los que la información del patrón de franjas puede estar distorsionada o faltante debido a oclusiones ocasionadas por la topografía del objeto y la triangulación de la cámara y proyector de franjas, desarrollando un algoritmo que puede estimar la información del patrón de franjas en las áreas afectadas.

El procesamiento de reconstrucción se llevó a cabo mediante la técnica de perfilometría por transformada de Fourier, en la cual se busca la obtención del mapa de fase envuelta del objeto y posteriormente desarrollar ésta misma por el método de Takeda, propuesto en 1982, para finalmente obtener la relación entre fase y altura del objeto escalando los valores a las distancias del arreglo experimental.

Cada capítulo de esta tesis desarrolla alguna parte de este proceso de la siguiente manera:

En el CAPÍTULO II se hace una breve revisión de las principales técnicas de perfilometría 3D existentes.

En el CAPÍTULO III se describe la perfilometría de luz estructurada sus técnicas y configuraciones básicas.

En el CAPÍTULO IV se describe la teoría que hay detrás de las imágenes digitales.

En el CAPÍTULO V se presenta el procesamiento digital que sufren las imágenes al utilizar la técnica de *inpainting* para restaurar imágenes digitales.

En el CAPÍTULO VI se presenta el algoritmo propuesto en este trabajo de tesis y su funcionamiento.

En el CAPÍTULO VII se presenta el arreglo experimental, así como sus componentes.

Finalmente, en el CAPÍTULO VIII se presentan los resultados finales obtenidos de este trabajo de tesis así como conclusiones y trabajos a futuro.

## CAPÍTULO II

## TÉCNICAS DE PERFILOMETRÍA 3D

Gracias al avance en la tecnología las técnicas de perfilometría 3D se han estado desarrollando y perfeccionando. Además, el constante crecimiento de la industria, demanda nuevos procesos los cuales deben ser más veloces y cumplir con los estándares de calidad deseados, donde la repetitividad dimensional es fundamental para asegurar la calidad del producto de aquí la necesidad de medir con rapidez y exactitud la forma 3D de objetos.

La perfilometría 3D permite conocer las dimensiones y forma de una superficie u objeto y tiene la versatilidad para ser adaptado a diferentes áreas y aplicaciones como son: arqueología, medicina, inspección de partes industriales, construcción, ingeniería inversa, mapas topográficos, entre otras [1].

Existe una variedad de técnicas de perfilometría 3D las cuales se puede clasificar de la siguiente manera:

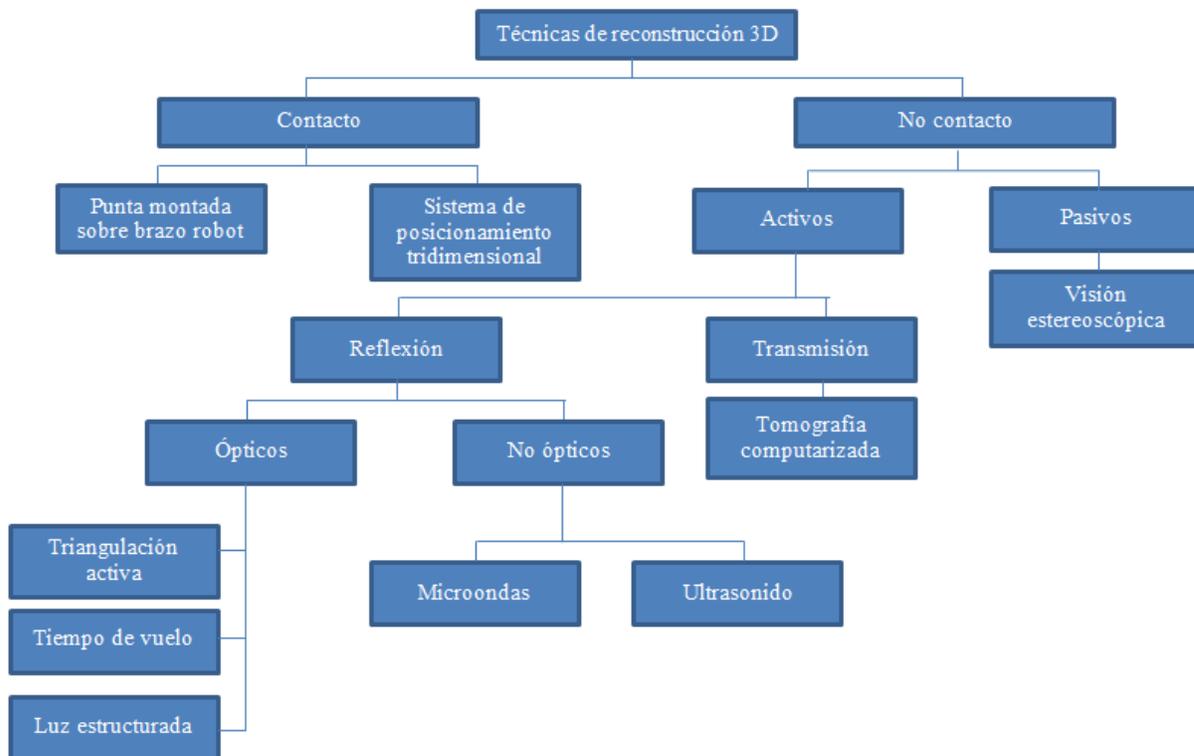


Figura 2.1 Técnicas de reconstrucción 3D.

Estos métodos trabajan con distintos principios por lo que según la aplicación se debe de elegir el método más apropiado tomando en cuenta las limitantes de cada método.

A continuación, se describe el funcionamiento de los métodos de contacto y algunos métodos ópticos, especialmente la técnica de luz estructurada ya que se basó en ella para el presente trabajo de tesis.

## 2.1 Técnicas de contacto

Los sistemas de adquisición de topografías o superficies por contacto físico son aquellos que exploran la superficie de los objetos por medio de dispositivos que tienen contacto directo con la superficie a explorar, este método de contacto consiste en una sonda por medio de la cual se obtienen mediciones referentes a la posición tridimensional de cientos de puntos sobre las superficies de las piezas. Las coordenadas X, Y, Z de cada punto medido son guardadas para procesarlas con software especializado. Las mediciones obtenidas por este método de contacto pueden alcanzar gran precisión; por lo que su empleo generalmente es utilizado frecuentemente en la industria manufacturera. Sin embargo, uno de los problemas de los sistemas por contacto es el tiempo de escaneo que necesitan para recorrer el objeto de estudio punto por punto, lo cual los vuelve lentos en comparación con otros métodos [2].

Otra desventaja de estas técnicas es que, al depender del contacto físico, puede llegar a causar daños o modificaciones a la superficie de los objetos.



Figura 2.2 Sistema de posicionamiento tridimensional explorando un motor de combustión interna.

## 2.2 Técnicas de no contacto

La característica principal de estas técnicas es la rapidez y bajo costo con la que se pueden obtener las mediciones requeridas de un objeto. Además, al no hacer contacto con la superficie o pieza lo cual es una característica importante ya que cualquier contacto entre el escáner y el objeto puede llevar a daños o deformación en la superficie del objeto lo que es inaceptable en muchas aplicaciones. Eso significa que debe haber una distancia entre el objeto y el escáner. En general, no queremos que esta distancia sea muy corta ya que limita la flexibilidad en el sistema, pero tampoco muy alta porque habrá una probabilidad mayor que el sistema sea afectado por interferencia en el ambiente de trabajo [3].

Estas técnicas se pueden subdividir en pasivas y activas. Las primeras consisten en técnicas que no emiten ninguna clase de radiación, pero se enfocan en detectar aquella reflejada en el ambiente como puede ser la luz visible. Las segundas son aquellas técnicas que se caracterizan por emitir alguna clase de radiación. La medición de la forma del objeto se logra analizando la energía reflejada o transmitida por la superficie del objeto.

Las principales técnicas basadas en el principio óptico son: escaneo por visión estereoscópica, escaneo por tiempo de vuelo y escaneo por proyección de luz estructurada.

### 2.2.1 Técnica de visión estereoscópica

Para simular los ojos humanos, un sistema de estereovisión emplea un par de cámaras alineadas y separadas una cierta distancia conocida con exactitud para capturar dos imágenes desde dos diferentes ángulos. Con el conocimiento de los parámetros de la cámara y las imágenes capturadas, las coordenadas 3D de la escena pueden ser obtenidas por triangulación. En la práctica, más de dos cámaras son comúnmente empleadas para obtener con precisión estructuras geométricas 3D de la escena. Sistemas de estereovisión han sido usados en varias aplicaciones como lo son reconstrucción de edificios 3D, reconocimiento de rostro 3D y escáner de cuerpo 3D. La mayoría de los sistemas de estereovisión tienen poca precisión y son computacionalmente intensivos y debido a esto no son recomendables para aplicaciones en tiempo real [4].



Figura 2.3 Sistemas de estereovisión comerciales.

### 2.2.2 Técnica de escáner de tiempo de vuelo.

Un escáner de tiempo de vuelo es un sistema de cámara de imágenes que emite pulsos de luz, por ejemplo, un haz láser, hacia el objetivo y captura la señal de pulsos parcialmente reflejada. Midiendo el tiempo entre la emisión y la recepción de los pulsos reflejados, el sistema puede medir la distancia del objeto con referencia a la velocidad de la luz. Su principal ventaja es que puede obtener una escena 3D completa con un solo disparo a la velocidad de fotogramas de video. La ventaja de estos escáneres es que pueden recopilar datos sobre mediciones a distancias muy lejanas, las cuales pueden llegar al rango de varios cientos de metros. Esta técnica es muy útil para realizar escaneos de grandes estructuras como pueden ser edificios, aunque solo produce un mapa de baja resolución debido a la dispersión e interferencia de la luz. Debido a estos inconvenientes no es recomendable para el modelado 3D de objetos o escenas complejas [5].



Figura 2.4 Escáner láser 3D de tiempo de vuelo utilizados en la industria.

### 2.2.3 Técnica de escáner 3D por luz estructurada.

La configuración más básica para la medición 3D por proyección de luz estructurada consta de un proyector y una cámara. Consiste en proyectar un patrón sobre el objeto a analizar, este patrón es deformado por la geometría del objeto y posteriormente se realiza la

captura de la imagen para analizar la modulación inducida, ya que contiene la información topográfica del mismo. Esta topografía será obtenida aplicando un algoritmo de análisis de patrones que consiste en la recuperación de la fase relacionada con el perfil del objeto bajo estudio y, en la mayoría de los casos, un paso adicional de desenvolvimiento para obtener un mapa de fase continuo y finalmente la reconstrucción 3D [6].

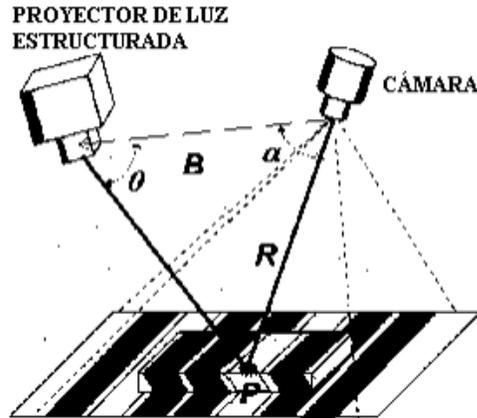


Figura 2.5 Ejemplo de proyección de franjas y la triangulación entre proyector, cámara y objeto.

La técnica que involucra la proyección de un patrón de franjas es probablemente la más eficiente, ya que se puede reconstruir una mayor superficie en una sola imagen. En general, este método es el más veloz, ya que requiere de una sola imagen para reconstruir una superficie. Su desventaja en comparación con otras técnicas es que necesita de la calibración tanto de la cámara como del sistema de proyección, además, está limitado por las características tanto de distancia focal y luminancia del proyector y de la cámara respectivamente. Además, si la superficie del objeto escaneado es brillante o translúcida, el patrón proyectado diferirá del patrón original, dando lugar a valores atípicos en los datos 3D.

La técnica de proyección de patrón de franjas cuenta con la ventaja de que pueden medirse objetos de distintas dimensiones a través de la modificación de la frecuencia de las franjas, con las únicas limitantes de las resoluciones tanto del sistema de proyección como de la cámara [7].

El actual sistema de proyección de franjas utilizado en esta tesis se considera un sistema de luz estructurada que usa un patrón de franjas sinusoidal, obteniendo un mapa de fase para finalmente reconstruir la topología del objeto, además, se hace uso de una técnica de restauración digital de imágenes llamada *inpainting* para mejorar la robustez del método cuando se trabaja en ambientes con dispersión o interferencia de luz.

En el siguiente capítulo se desarrolla el fundamento para la técnica de proyección de franjas, aspecto importante porque se utilizó a lo largo de este trabajo de investigación.

## CAPÍTULO III

## TÉCNICA DE PROYECCIÓN DE FRANJAS

Esta técnica se enfoca en la detección de fase del patrón periódico, comúnmente sinusoidal, para reconstruir la topografía del objeto. La modulación de este patrón a causa del objeto es considerado como el resultado de la suma del patrón original y la información topológica del objeto.

## 3.1 Parámetros geométricos y relación fase-altura

El principio de funcionamiento de la técnica de proyección de franjas se puede observar en la Figura 3.1.

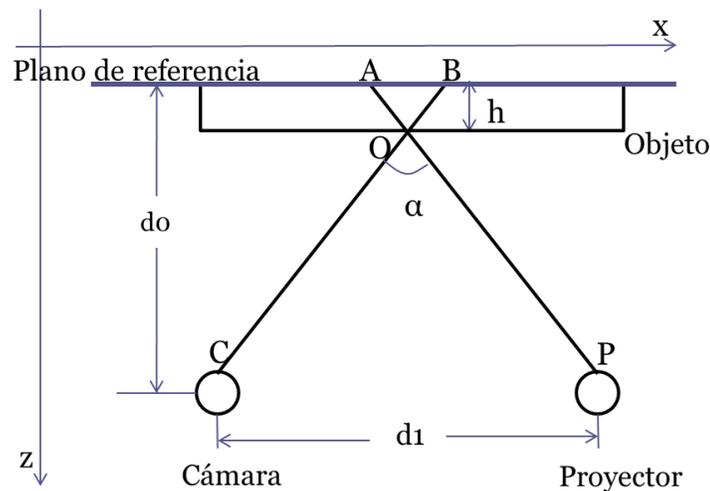


Figura 3.1 Arreglo geométrico de proyección de franjas.

Si se coloca un objeto sobre el plano de referencia, se puede calcular la profundidad del mismo. Si la distancia  $d_0$  es relativamente grande con respecto a  $d_1$ , entonces la rejilla proyectada sobre el objeto será irregular con cambios de frecuencia en la dirección  $x$ , en cuyo caso la fase observada en  $O$  será muy similar a la fase en  $A$  y los puntos  $O$  y  $B$  sobre el objeto se proyectarán en el plano imagen en el mismo punto  $C$ . La distancia  $AB$  puede expresarse entonces como la ecuación 3.1 [8].

$$AB = \frac{\phi_B - \phi_O}{2\pi f} \quad (3.1)$$

Donde:

$\phi_B$  es la fase en el punto B.

$\phi_O$  es la fase en el punto O.

$f$  es la frecuencia espacial de la rejilla sobre el plano de referencia.

Puede hacerse notar de la ecuación (3.1) que la distancia AB es proporcional a la diferencia de fase producida por la presencia del objeto ya que en lugar de aparecer un punto proyectado cualquiera como si estuviera en el plano de referencia (en la posición A), éste se muestra en el punto B. Si se analizan los triángulos POC y AOB en la Figura 3.1, se obtiene la relación:

$$z(x, y) = \frac{(AB)d_O}{d_1 + AB} \quad (3.2)$$

Al sustituir la ecuación (3.1) en (3.2) se obtiene la profundidad en términos de la fase.

$$z(x, y) = \frac{(\phi_B - \phi_O)d_O}{2\pi f d_1 + (\phi_B - \phi_O)} \quad (3.3)$$

Si se expresa el periodo de la rejilla proyectada como  $p = 1/f$  y se hace  $\Delta\phi = \phi_B - \phi_O$ , entonces la ecuación anterior puede expresarse en función del ángulo  $\alpha$  entre el proyector y la cámara [9].

$$z(x, y) = \frac{\Delta\phi(x, y) T}{2\pi + \sin\alpha} \quad (3.4)$$

En la ecuación 3.4 se puede observar la importancia del cálculo de fase a lo largo de toda la imagen para conocer la altura del objeto, para lograr esto previamente se hace un análisis de franjas en la imagen entre los cuales existen varias técnicas, estudiaremos a continuación las más relevantes para nuestra aplicación.

### 3.2 Técnicas de análisis de franjas

Se han desarrollado diversos métodos de análisis de franjas, los cuales podemos clasificar en técnicas temporales o espaciales, la primera consiste en una serie de tomas múltiples de por lo menos tres, en las cuales cada valor de fase por pixel será calculado comparado con las demás tomas dando como resultado el mapa de fase. La segunda consiste en una sola toma o dos si se cuenta la toma del patrón proyectado sin objeto (patrón de referencia), y se logra gracias a que para el cálculo del mapa de fase los valores de fase por pixel serán calculados en base a los valores de sus vecinos, lo cual le da una ventaja importante tanto en velocidad como en captura de objetos en movimiento [10].

Generalmente, estas técnicas siguen una secuencia de pasos que se puede representar en un diagrama de flujo, ver Figura 3.2

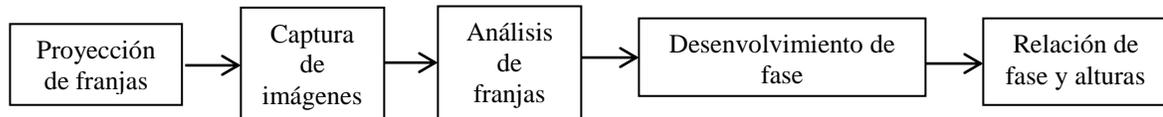


Figura 3.2 Diagrama de flujo para el análisis de franjas.

La técnica temporal más relevante es la técnica por corrimiento de fase mientras que la técnica espacial más relevante es perfilometría por transformada de Fourier esta última de especial importancia por su implementación en el presente trabajo de investigación, además, ambas técnicas parten de la misma relación entre la fase y la intensidad observada en las imágenes de un patrón de perfil cosenoidal y que puede aproximarse por la ecuación 3.5 [11].

$$I(x, y) = a(x, y) + b(x, y)\cos(2\pi f_0(x, y) + \varphi(x, y)) \quad (3.5)$$

Donde:

$a(x, y)$  = Intensidad de la fuente de luz

$b(x, y)$  = Variaciones de reflectancia del objeto (dispersión)

$f_0(x, y)$  = Frecuencia espacial de la portadora

$\varphi(x, y)$  = Término de fase

Una parte fundamental de la topometría por medio de la proyección de un patrón de franjas con perfil cosenoidal, es la correcta adquisición de la fase, ya que ésta se encuentra asociada directamente con la profundidad de un punto en el objeto.

### 3.2.1 Técnica por corrimiento de fase

Esta técnica consiste en una serie de tomas de patrones sinusoidales que son proyectados secuencialmente sobre el objeto, estos patrones tienen un corrimiento de fase la cual en su modelo más común es de  $\pi/2$  radianes, por lo que con cuatro capturas de imagen se completa la información requerida. La siguiente ecuación que parte de la ecuación 3.5 muestra la intensidad de cada uno de estos patrones [12].

$$I_n(x, y) = a(x, y) + b(x, y)\cos(2\pi f_0(x, y) + \varphi(x, y) + \frac{n\pi}{2}) \quad (3.6)$$

Como se mencionó anteriormente el término de fase es el que contiene la información deseada para la reconstrucción 3D por lo que debe ser despejada de la ecuación, para hacer esto se puede considerar como términos comunes  $a(x, y)$  y  $b(x, y)$ , al restar  $I_3 - I_1$  e  $I_0 - I_2$ , tenemos:

$$\varphi(x, y) = \tan^{-1} \left\{ \frac{I_3(x, y) - I_1(x, y)}{I_0(x, y) - I_2(x, y)} \right\} \quad (3.7)$$

Esta distribución de fase  $\varphi(x, y)$  se encuentra contenida dentro del rango de  $-\pi$  a  $\pi$  por lo que se debe aplicar un algoritmo de desenvolvimiento de fase donde se acumulen consecutivamente y así obtener una fase continua. Posteriormente, esta fase continua se somete a la transformación de alturas con la cual se puede realizar el mapa con la reconstrucción 3D del objeto en la imagen.

### 3.2.2 Técnica de perfilometría por transformada de Fourier

Una de las técnicas para recuperar la fase es la técnica de la transformada de Fourier, este método se ha aplicado ampliamente en el procesamiento de señales e imágenes al transformar información de intensidad al dominio de Fourier o dominio de la frecuencia. Para aplicar la transformada de Fourier a señales interferométricas se introduce una señal portadora para

separar la señal de primer orden de los componentes de orden cero en el dominio de la frecuencia [13].

En la técnica de la transformada de Fourier propuesta por Takeda [14], se toma el patrón de franjas adquirido, que parte de la ecuación 3.5 y por conveniencia se reescribe como:

$$g(x, y) = a(x, y) + c(x, y)e^{2\pi i f_0 x} + c^*(x, y)e^{-2\pi i f_0 x} \quad (3.8)$$

Donde,

$$c(x, y) = \frac{1}{2}b(x, y)e^{i\varphi(x, y)} \quad (3.9)$$

Después se aplica la transformada rápida de Fourier (FFT),

$$G(f, y) = A(f, y) + C(f - f_0, y) + C^*(f + f_0, y) \quad (3.10)$$

En donde las letras mayúsculas denotan el espectro de Fourier y  $f$  es la frecuencia espacial en la dirección  $x$ . Los espectros de Fourier se separan como se muestra en la Figura 3.3.

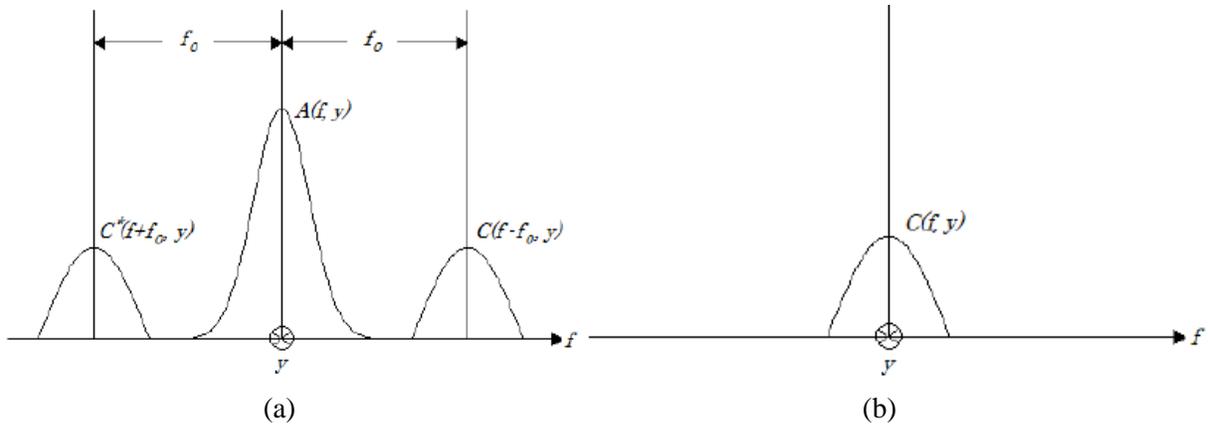


Figura 3.3 (a) Espectro de Fourier de un patrón de franjas  $y$ ; (b) espectro seleccionado y trasladado al origen.

Se hace uso de cualquiera de los dos espectros correspondientes a la señal portadora y se traslada al origen una distancia  $f_0$  para filtrar también la frecuencia portadora y obtener  $C(f, y)$  como se muestra en la Figura 3.2. Ahora se obtiene la transformada inversa de Fourier para  $C(f, y)$  con respecto a  $f$  para obtener  $c(x, y)$  definido por la ecuación 3.9.

Finalmente para obtener la fase  $\varphi(x, y)$  a partir de  $c(x, y)$  se aplica la función arcotangente,

$$\varphi(x, y) = \tan^{-1} \left\{ \frac{\text{Im}(c(x, y))}{\text{Re}(c(x, y))} \right\} \quad (3.11)$$

Esta distribución de fase obtenida, al igual que la anterior técnica, también se encuentra envuelta a un factor  $2\pi$  dentro del rango de  $-\pi$  a  $\pi$  debido a la función arcotangente, por lo que finalmente es necesario aplicar un algoritmo de desenvolvimiento de fase para obtener una fase continua con la cual se puede realizar la reconstrucción 3D.

### 3.2.3 Desenvolvimiento de fase

Por cualquiera que sea el método de análisis de franjas, éste genera un mapa de fase del patrón deformado el cual se encuentra con saltos cada  $2\pi$ . Por este motivo es necesario utilizar un algoritmo para conectar estos saltos de fase y obtener la distribución de fase final que está relacionada con la distribución física del objeto, a este procedimiento se le conoce como desenvolvimiento de fase.

El algoritmo realiza el desenvolvimiento en una trayectoria definida y calcula las diferencias de fase entre los pixeles vecinos a lo largo de la trayectoria para decidir si suma o resta  $2\pi$  para generar el mapa de fase final como se puede observar en las ecuaciones 3.15, realizar esto es algo sencillo siempre que el mapa de fase sea ideal o en otras palabras sin discontinuidad en franjas creadas por oclusiones o ruido, de lo contrario entrega resultados erróneos [15].

$$\begin{aligned} \phi(x_{i+1}) - \phi(x_i) > \pi & \quad \phi(x_{i+1}) - \phi(x_i) \leq -\pi & (3.15) \\ \phi(x_{i+1}) = \phi(x_i) - 2\pi & \quad \phi(x_{i+1}) = \phi(x_i) + 2\pi \end{aligned}$$

Desde luego se han presentado nuevos algoritmos de desenvolvimiento de fase que toman en cuenta estas discontinuidades espaciales y tratan de minimizar el error producido. Aunque también aumentan el tiempo de procesamiento considerablemente por lo que en aplicaciones

en tiempo real o video no son compatibles. Por esta razón la importancia de utilizar un algoritmo de desenvolvimiento de computo ligero.

En este capítulo se llevó a cabo una revisión a las técnicas de análisis de franjas, así como su posterior desenvolvimiento de fase para finalmente hacer la relación fase-altura. Todo esto con la finalidad de conocer el principio de funcionamiento de la perfilometría por luz estructurada y su configuración más utilizada.

## CAPÍTULO IV

### IMÁGENES DIGITALES

#### 4.1 Imagen digital

En general, el propósito de la formación de imágenes es extraer información física sobre la fuente de radiación y/o los objetos con los que ha interactuado. Dichas imágenes pueden ser de naturaleza reflejada, al incidir la luz en la superficie de los objetos esta puede ser emitida, como son las imágenes infrarrojas; o absorbida en el caso de las imágenes por rayos-X. En la Figura 4.1 se observa un esquema del origen de la formación de las imágenes [16].

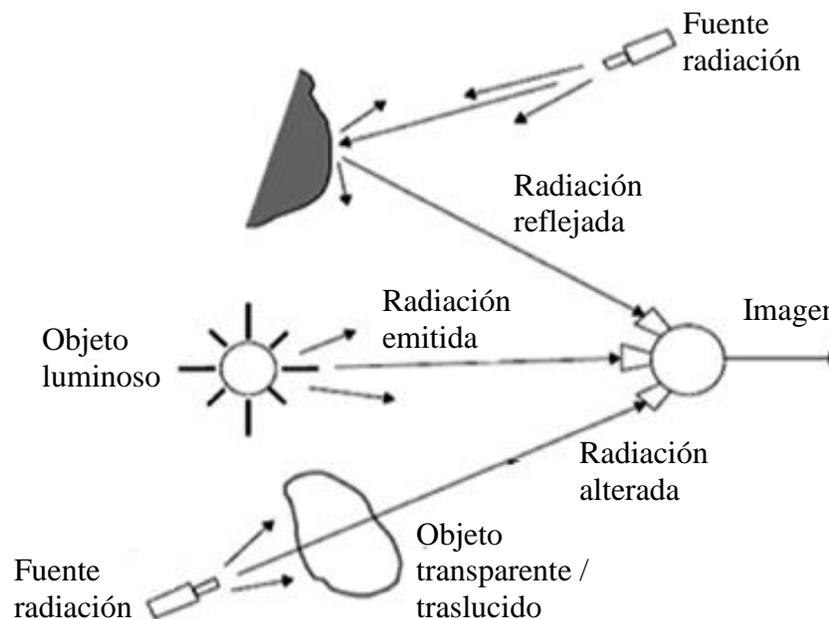


Figura 4.1 Interacción de la radiación electromagnética con los objetos.

Las señales e imágenes que se aprecian en la naturaleza son de carácter analógico, entendiendo como tal, que son señales continuas en el dominio espacio/tiempo. Por lo tanto, una imagen se puede representar como una función bidimensional  $f(x, y)$ , donde  $x$  y  $y$  son las coordenadas espaciales y el valor de  $f$  en un punto cualquiera  $(x, y)$  es proporcional al brillo de la imagen en ese punto, y puede tomar cualquier valor posible. Sin embargo, para el procesamiento digital se hace necesario que una vez que ha sido detectada la imagen  $f(x, y)$ , ésta

debe ser digitalizada para que pueda ser comprensible por un ordenador. Esta nueva imagen será una señal bidimensional con coordenadas espaciales discretas, que puede tomar valores de un conjunto discreto de valores.

Una imagen digital, puede considerarse como una matriz cuyos índices de filas y columnas identifican un punto de la imagen y el valor del correspondiente elemento de la matriz indica el nivel de gris en ese punto. Esto es lo que se conoce como pixel, abreviatura del término en inglés "*picture elements*", ver Figura 4.2.

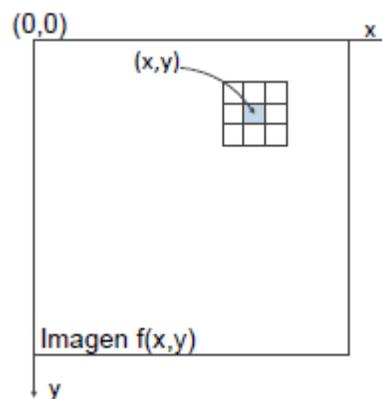


Figura 4.2 Matriz de una imagen digital donde es señalado un pixel con coordenadas (x,y).

## 4.2 Procesamiento digital de imágenes

Es el conjunto de técnicas que se aplican a las imágenes digitales con el objetivo de mejorar la calidad o facilitar la búsqueda de información.

El interés de los métodos de procesamiento de imágenes digitales se fundamenta en dos áreas principales de aplicación: a) Mejor calidad para la interpretación humana y b) Procesamiento de los datos de la escena para la percepción autónoma de las máquinas. En el intento por dotar a las máquinas de un sistema de visión aparece el concepto de visión artificial. En la mayoría de los sistemas, la capacidad sensorial de la visión es complementada con mecanismos sensoriales tales como: detectores de alcance de profundidad o de proximidad. Desde una perspectiva general, la visión artificial es la capacidad de la máquina

para ver el mundo que le rodea, más precisamente para deducir la estructura y las propiedades del mundo tridimensional a partir de una o más imágenes bidimensionales.

Normalmente se procesan imágenes en escala de grises lo que quiere decir que el valor que tiene cada pixel corresponde a algunos de las variaciones de los 256 valores de gris, pasando desde el valor del color negro (cero), hasta el blanco (255). Habitualmente la imagen va mostrando pequeños cambios de tonalidad con la variación de valores de gris. La calidad de una representación gráfica en estos tonos, es decir la nitidez, es lo que marca la pauta para la magnitud de los píxeles y por ende el número total de ellos de una imagen; aparte de ello cuanto más alta sea la resolución, las variaciones de tonalidades de gris serán más sutiles y por lo tanto mayor calidad de la misma, una mejor representación, mejor resolución, pero computacionalmente implica mayor recurso de máquina [17].

Los métodos de procesamiento de imágenes pueden realizarse tanto en el dominio espacial como en el dominio de la frecuencia. Los del primer grupo se caracterizan por operar directamente sobre los píxeles de la imagen, mientras que los del segundo grupo modifican una transformada de ésta (Fourier) para obtener los resultados.

Los métodos de procesamiento en el dominio espacial se pueden clasificar en dos tipos:

- Punto a punto: las operaciones se efectúan sobre el valor del pixel, sin tener en cuenta los píxeles vecinos. Sirve para mejorar condiciones de bajo contraste, baja luminosidad o demasiada claridad.
- Operaciones entre vecinos: con ayuda de una máscara o *kernel*, las operaciones son efectuadas sobre un pixel tomando en cuenta sus píxeles vecinos. Son útiles cuando se busca reducir el ruido, detectar bordes, o generar algún tipo de efecto.

Un ejemplo de procesamiento espacial se logra modificando el histograma de la imagen para mejorar el contraste. Un histograma se refiere al número de píxeles con igual valor de la escala de gris. Este aspecto calcula la probabilidad de distribución  $P(b)$  de diferentes niveles de gris de una representación gráfica. Así mismo, la representación gráfica del histograma muestra una idea del aspecto que posee la imagen, esto es, proporciona información sobre su

intensidad media y la dispersión de los valores de nivel de gris de ésta, siendo por tanto, una medida del contraste de la imagen [18].

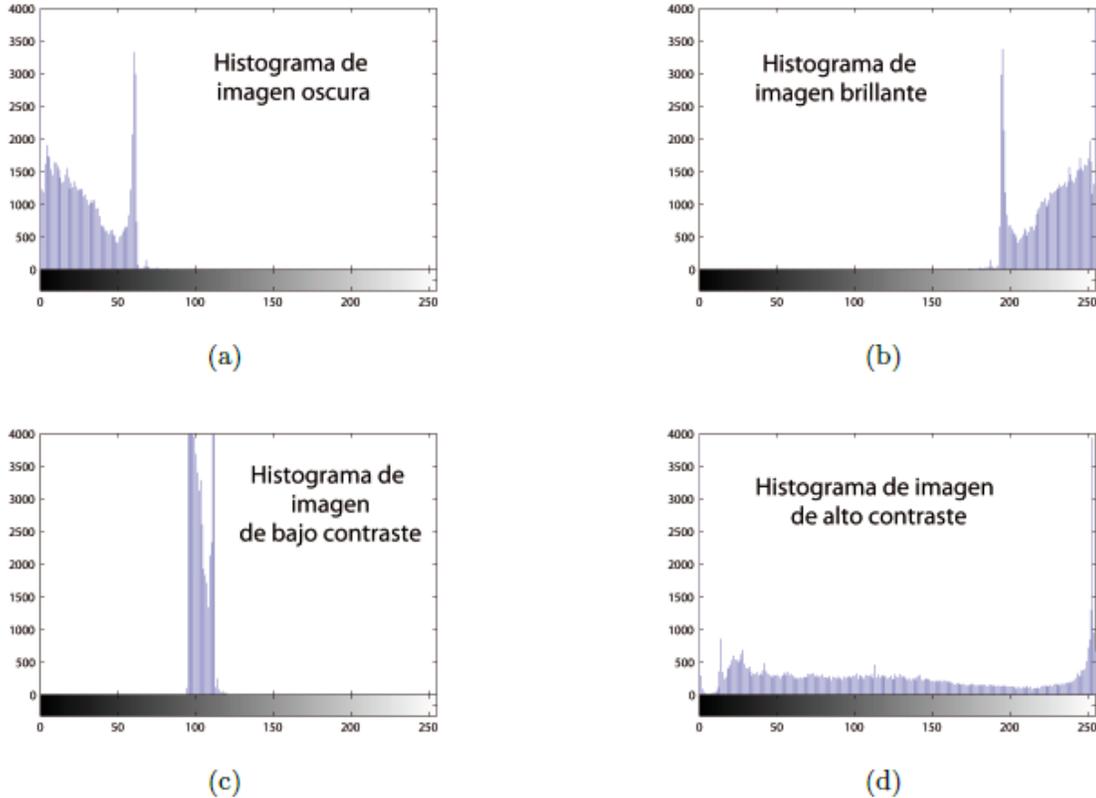


Figura 4.3 Histograma con valores en extremo oscuro (a); histograma con valores en extremo brillante (b); histograma con bajo contraste (c); histograma con alto contraste (d).

Por otro lado, para un procesamiento en el dominio de frecuencia la misma debe ser considerada como una función no periódica, y definirse en otro espacio bidimensional cuyos ejes vengán determinados por la amplitud y frecuencia para cada dirección de la imagen [11].

En imágenes digitales se utiliza la versión bidimensional de la "Transformada de Fourier" por lo que se emplea la Transformada de Fourier Discreta, cuya expresión se muestra en la ecuación (4.1).

$$f(u, v) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f(m, n) e^{-j2\pi(Um+Vn)} \quad (4.1)$$

Una de las ventajas del uso de la transformada de Fourier en el procesado de las imágenes, es poder aplicar el teorema de la convolución que se describe en la ecuación (4.2).

Sea  $g(x, y)$  una imagen formada por la convolución de una imagen  $f(x, y)$  y un filtro invariante de posición  $h(x, y)$ , entonces se cumple que:

$$g(x, y) = f(x, y) * h(x, y) \Leftrightarrow G(u, v) = F(u, v)H(u, v) \quad (4.2)$$

Siendo  $G$ ,  $F$  y  $H$  la transformada de Fourier de  $g$ ,  $f$  y  $h$  respectivamente [11].

Dicho teorema es la base fundamental de las técnicas de filtrado en el dominio frecuencial, por lo que la operación de la convolución de un imagen  $f(x, y)$  y un filtro  $h(x, y)$ , se transforma en la sencilla operación del producto de la imagen transformada  $F(u, v)$  con el filtro  $H(u, v)$ , ver Figura 4.4. Por tanto, el proceso de filtrado en el dominio de la frecuencia consistirá en los siguientes pasos:

1. Calcular  $F(u, v)$ , la Transformada de Fourier Discreta de la imagen original.
2. Multiplicar  $F(u, v)$  por la función de transferencia del filtro  $H(u, v)$ .
3. Calcular la transformada inversa del resultado del paso 2.
4. Obtener la parte real del resultado en 3.

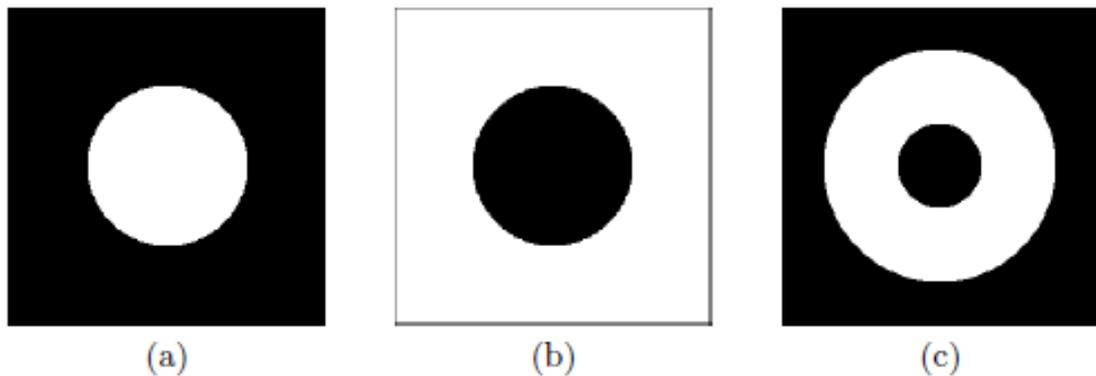


Figura 4.4 Filtros: (a) Pasa-baja; (b) Pasa-alto; (c) Pasa-banda.

En este capítulo se llevó a cabo una revisión a las imágenes digitales, su composición y como es el procesamiento de imágenes digitales sobretodo en el dominio de la frecuencia. Estas bases nos ayudaran a comprender las técnicas de restauración de imágenes lo cual es el núcleo para la elaboración de este trabajo de investigación. A continuación realizaremos una revisión a las principales técnicas de restauración de imágenes existentes y sus aplicaciones.

## CAPÍTULO V

### TÉCNICAS DE RESTAURACIÓN DE IMÁGENES

*Inpainting* es un área activa de investigación en procesamiento de imágenes, también es conocido como restauración de imagen. El principal objetivo de este proceso es llenar la información faltante de la región basándose en la información circundante al área de interés. Los algoritmos de *inpainting* tienen numerosas aplicaciones como lo es reconstruir fotografías y películas dañadas, eliminación de fecha o texto superpuesto en la imagen, eliminación de objetos indeseados, eliminación de ruido, etc.

Los métodos más usados de *inpainting* actualmente son el método basado en *Partial Differential Equation* (PDE) y el método basado en *Texture synthesis*. La idea básica detrás de estos algoritmos que han sido propuestos en la literatura es el rellenar estas regiones con la información disponible de su entorno.

#### 5.1 *Partial Differential Equation* (PDE)

PDE es una ecuación diferencial que contiene una o más variables y usa derivaciones parciales, Bertalmio propuso un método en donde la información se propaga en la área de occlusión, el algoritmo extiende la información geométrica y fotométrica que llega al borde del área dentro de ella misma. El algoritmo es eficiente cuando es aplicado a imágenes con pequeños daños ya que conlleva un efecto difuminado como se puede observar en la Figura 5.1, pero cuando las regiones son grandes este algoritmo tomará mucho tiempo ya que es iterativo y además no producirá buenos resultados.

La mayor desventaja de este método es el hecho que no puede reconstruir texturas. Chan and Shen [19] propusieron el *Total Variational* una evolución del método donde encuentran la línea de *isophote* que es la línea de píxeles de igual valor en escala de grises que contiene la información más prometedora y es usada para completar la región en menos tiempo, sin embargo, adolece el mismo problema al no poder reconstruir grandes regiones de textura.

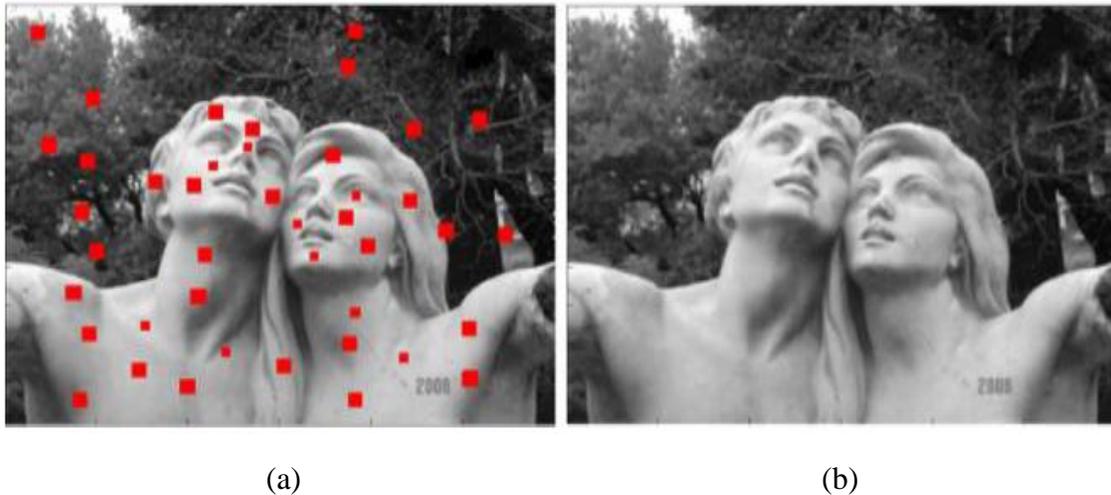


Figura 5.1 (a) Imagen con pequeñas oclusiones; (b) Imagen restaurada con algoritmo basado en PDE.

## 5.2 *Texture synthesis*

El objetivo principal de la restauración basada en *texture synthesis* es generar patrones similares a los que se dan en el patrón de muestra, de esta forma la textura reproducida retiene las propiedades estadísticas de la textura raíz.

La técnica desarrollada por Efros y Leung [20] usa un pixel como punto de partida, ubicado en el borde del área de oclusión, posteriormente define una ventana alrededor de él para encontrar bloques similares en la imagen y restaurar el área de interés. Este método restaura textura pixel por pixel como se puede observar en la Figura 5.2.

Efros y Freeman [21] presentan una variación al método en la que la síntesis de texturas es llevada a cabo usando bloques, y no pixel por pixel, lo cual reduce drásticamente el tiempo de ejecución. El algoritmo ha probado ser más eficiente al copiar un bloque entero cuando un candidato válido es encontrado en la fuente. Aunque el método es mucho más rápido y por lo tanto más eficiente sacrifica un poco en proporcionar buenos resultados en imágenes altamente texturizadas.

Los pasos del algoritmo incluyen primeramente la definición de una máscara y una imagen a restaurar, seguido por la detección de borde del área de oclusión, todos los píxeles sobre el borde serán clasificados en orden descendente por número de píxeles vecinos

conocidos. Una ventana será definida centrada en cada pixel seleccionado para restaurar, esta ventana tiene un tamaño parametrizado y será usada en la búsqueda de bloques similares en la imagen. La medición de similitud es dada por el error cuadrático medio. Para preservar la textura local, un kernel gaussiano es usado, el cual controla la influencia de pixeles localizados muy lejos del área de oclusión.

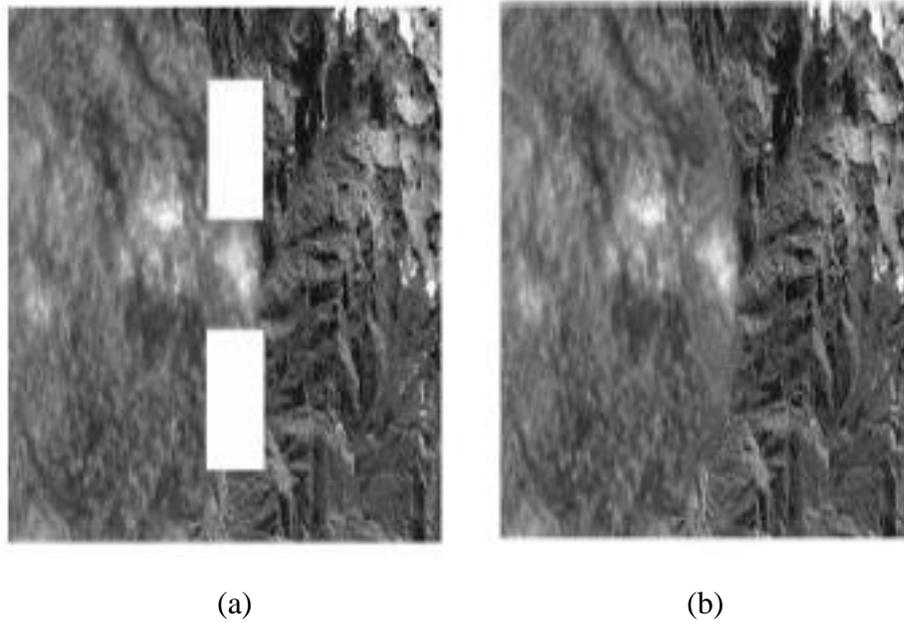


Figura 5.2 (a) Imagen con múltiples texturas; (b) Resultado del algoritmo Texture synthesis.

En este capítulo se pudo observar el efecto que se tiene al trabajar con algoritmos de inpainting sobre una imagen, esto nos ayuda a comprender el funcionamiento y el objetivo principal de estos algoritmos. Teniendo esto como base, en el siguiente capítulo, veremos el algoritmo de inpainting desarrollado en este trabajo de investigación específicamente para la corrección de oclusiones en perfilometría 3D.

## CAPÍTULO VI

### ALGORITMO SINUSOIDAL DE INPAINTING

Tomando como base la naturaleza de las imágenes con las cuales se trabaja en la perfilometría 3D por luz estructurada se llegó a la conclusión que se puede desarrollar un algoritmo para trabajar la zona de oclusión digitalmente y resolver el problema de falta de información en esta área. En base a esto, se desarrolló un nuevo algoritmo de inpainting utilizando el software Matlab pensando en las características propias del programa.

Para la elaboración del algoritmo de restauración se tomó como base la ecuación de movimiento armónico simple ya que por su naturaleza encuadra en nuestro patrón de proyección para la perfilometría 3D. El movimiento armónico simple es un problema importante porque muchos problemas físicos que implican vibraciones mecánicas, acústica, óptica o física atómica se pueden reducir y aproximar al oscilador armónico simple, o una combinación de ellos. Por lo que es importante recordar este concepto.

#### 6.1 Movimiento Armónico Simple

El movimiento de una partícula en un sistema complejo es más fácil de analizar si consideramos que el movimiento es una superposición de oscilaciones armónicas, las cuales pueden describirse en términos de funciones seno y coseno [22].

Si consideramos como base las expresiones de la fuerza y de la energía potencial de un resorte ideal con constante de fuerza  $k$ , comprimido o estirado en una distancia  $x$  y una masa  $m$  unida al final, tenemos que

$$F(x) = -kx \quad (6.1)$$

$$U(x) = \frac{1}{2}kx^2 \quad (6.2)$$

Existe una posición de equilibrio en que el resorte no ejerce ninguna fuerza sobre el cuerpo (Figura 6.1b). Si el cuerpo se desplaza hacia la derecha la fuerza ejercida por el resorte sobre el cuerpo apunta hacia la izquierda (Figura 6.1a). Si el cuerpo se desplaza hacia la izquierda, la fuerza apunta hacia la derecha (Figura 6.1c)

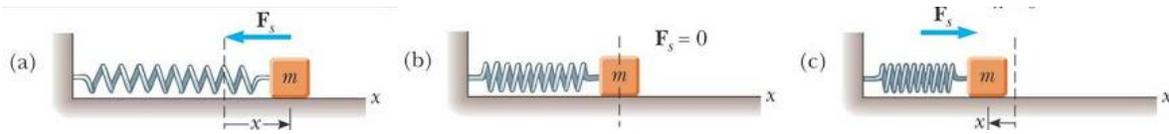


Figura 6.1 (a) Resorte se estira de modo que el cuerpo tenga su desplazamiento máximo a partir del equilibrio; (b) Posición de equilibrio con el resorte relajado; (c) El resorte está totalmente comprimido.

Aplicando la segunda ley de Newton,  $F=ma$ , al movimiento del resorte se sustituye  $F$  por  $ma$  y la aceleración  $a$  se sustituye por  $d^2x/dt^2$ . Esto nos da:

$$m \frac{d^2x}{dt^2} = -kx$$

$$\frac{d^2x}{dt^2} + \frac{k}{m}x = 0 \quad (6.3)$$

La ecuación 6.3 recibe el nombre de ecuación del movimiento del oscilador armónico simple el cual es una relación entre una función del tiempo  $x(t)$  y su segunda derivada con respecto al tiempo, para hallar una función que satisfaga a esta relación podemos reescribir la ecuación 6.3.

$$\frac{d^2x}{dt^2} = -\left(\frac{k}{m}\right)x \quad (6.4)$$

La ecuación 6.4 requiere que  $x(t)$  sea una función cuya segunda derivada sea la negativa de la función misma, excepto por un factor constante  $k/m$ . Las funciones seno y coseno tienen esta propiedad por lo que una solución de la ecuación 6.4 es

$$x = x_{max} \cos(\omega t + \phi) \quad (6.5)$$

## 6.2 Algoritmo sinusoidal

El uso de un patrón sinusoidal que es modulado sobre una superficie u objeto, lo convierte en un candidato natural para utilizar la ecuación de movimiento armónico simple (Ecuación 6.5) ya que se puede aproximar la información faltante utilizando la información de este mismo patrón antes y después del área de oclusión de las imágenes a procesar. En consecuencia, el primer paso es la obtención de información de la onda sinusoidal desde la

propia imagen digital anterior a la zona de oclusión y hacer lo mismo posterior a la zona de oclusión, esto es, encontrar los primeros tres puntos de cambio de signo de la pendiente dentro de los valores discretos por cada fila de pixeles (Figura 6.2) para poder representar su forma sinusoidal.

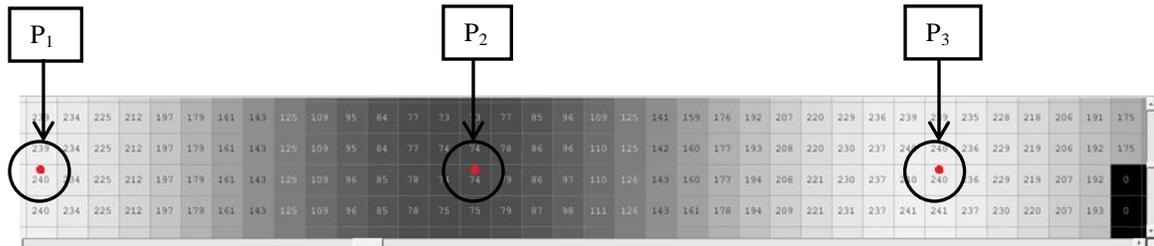


Figura 6.2 Ejemplo de valores por cada pixel antes de la zona de oclusión.

Al detectar estos puntos se denominarían los valles y crestas de la onda sinusoidal anterior y posterior a la zona de oclusión, podemos trabajar gráficamente la onda y calcular sus elementos (Figura 6.3).

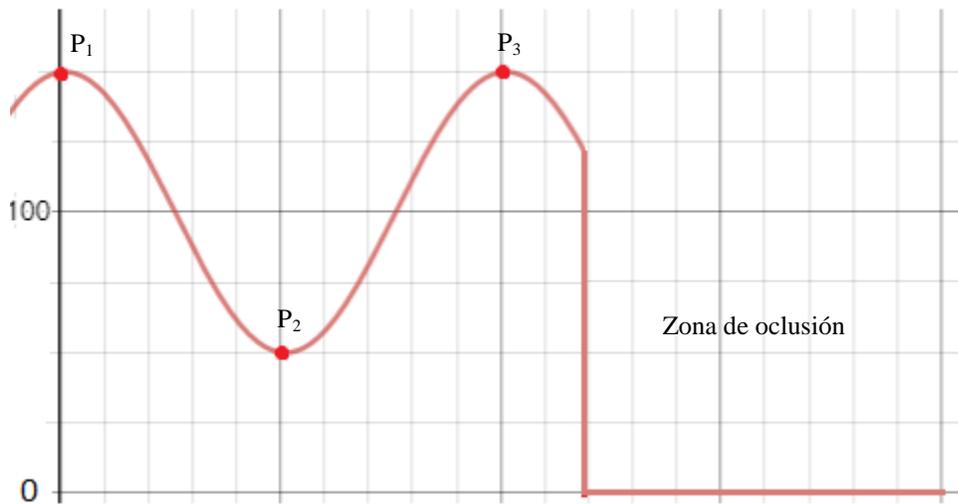


Figura 6.3 Ejemplo de gráfica con los valores correspondientes a una fila de píxeles antes de la zona de oclusión con los valles y crestas identificados.

Para relacionar estos puntos hay que trabajar sobre la función de movimiento armónico simple y con sus variables de amplitud, número de onda, constante de fase y finalmente se

agrega una variable de desplazamiento vertical ya que toma solo valores discretos entre 0 y 255.

La función sinusoidal entonces se expresa:

$$f(x) = a \operatorname{sen}(bx + c) + d \quad (6.6)$$

Donde:

$a$  = Amplitud

$b$  = Número de onda

$c$  = Constante de fase

$d$  = Desplazamiento vertical

La amplitud ( $A$ ) es calculada con la mitad de la diferencia entre el valor de la cresta y el valle, por lo tanto:

$$A = \frac{P_1 - P_2}{2} \quad (6.7)$$

Para el número de onda ( $k$ ) primero es calculada la longitud de onda ( $\lambda$ ) la cual está determinada por el número de píxeles desde el primer valle o cresta ( $P_1$ ) hasta el tercer valle o cresta ( $P_3$ ) y posteriormente con la expresión:

$$k = \frac{2\pi}{\lambda} \quad (6.8)$$

Para la constante de fase ( $\phi$ ) tenemos que tomar en cuenta que según como se exprese la función sinusoidal puede tener una constante horizontal o una constante de fase, para el primer caso tenemos que:

$$f(x) = a \operatorname{sen}(b(x + c)) + d \quad (6.9)$$

Donde “ $c$ ” es la constante horizontal.

Para el segundo caso tenemos que:

$$f(x) = a \operatorname{sen}(bx + c) + d \quad (6.10)$$

Donde “c” es la constante de fase.

Como en nuestro caso usamos la expresión con constante de fase primero calculamos la constante horizontal que está dada por el número de pixeles desde  $P_3$  hasta el inicio de la zona de oclusión y posteriormente usando la expresión:

$$\phi = b * c \quad (6.11)$$

Por último, el desplazamiento vertical es calculado con un promedio entre el valor de  $P_1$  y  $P_2$ , esto es:

$$d = \frac{P_1 + P_2}{2} \quad (6.12)$$

Es importante aclarar que la adición de la variable de desplazamiento vertical es debido a que los pixeles de las imágenes a procesar utilizan valores discretos normalmente de 0 a 255 si es de 8 bits, por lo que no es posible representar valores negativos en una imagen digital, al utilizar un desplazamiento vertical aseguramos que la función sinusoidal se mantenga dentro de este rango.

Los valores de amplitud, número de onda y fase, deben ser calculados tanto anterior como posterior al área a restaurar. Estos son calculados por cada línea de pixel de la imagen, en donde se van detectando los cambios de signo de la pendiente entre los valores consecutivos de la función sinusoidal, al detectar tres cambios de pendiente se obtiene la información de estos puntos y como se demostró anteriormente la de la onda sinusoidal así como sus elementos. Al realizar este procedimiento antes y después del área a restaurar tenemos como resultado dos ondas sinusoidales serán la base para completar la información faltante en cada pixel a restaurar de la zona de oclusión donde cada uno de estos pixeles se considera como un “paso” al aproximarse desde la función sinusoidal anterior a la función sinusoidal posterior de la zona a restaurar. Por lo que si definimos estas funciones sinusoidales como  $f_1$  y  $f_2$  tenemos que la función de restauración  $f_r$  se puede definir por la expresión:

$$f_r(x)_{x \rightarrow k} = f_1(x) + x \left( \frac{f_2(x) - f_1(x)}{k} \right) \quad (6.13)$$

Donde:

$f_r(x)$  es la función sinusoidal de restauración.

$f_1(x)$  es la función sinusoidal anterior al área de restauración.

$f_2(x)$  es la función sinusoidal posterior al área de restauración.

$k$  es la longitud del área de restauración.

Al igual que muchos algoritmos de inpainting, puede desempeñarse de mejor o peor manera dependiendo de varios factores, por ejemplo, la naturaleza de las imágenes a analizar, tamaño de la imagen, tamaño de la zona a restaurar, cambios de alta frecuencia en la imagen, respetabilidad en valores de intensidad de la imagen, contraste de la imagen, etc. Por lo que es de gran importancia encontrar el algoritmo de inpainting que mejor se adapte a la aplicación de interés.

### 6.2.1 Variables en la zona de restauración

El algoritmo se basa en que al aproximarse la función  $f_1$  a la función  $f_2$  las variables de amplitud, número de onda, constante de fase y desplazamiento vertical varían de manera lineal para cada elemento. Así que vamos a observar el efecto que se tiene al pasar de una función sinusoidal a otra función sinusoidal variando linealmente e independientemente cada variable.

#### 6.2.1.1 Número de onda

Para cada línea de píxeles el valor del número de onda en la función sinusoidal varía linealmente conforme avanza en la zona de restauración hasta llegar al valor de la función sinusoidal posterior a la zona de restauración, esto crea un efecto de variación en los ciclos por unidad de distancia, se ejemplifica como se muestra en la Figura 6.4.

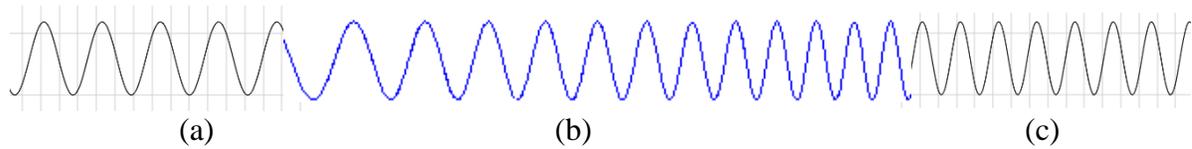


Figura 6.4 (a) función sinusoidal anterior a la zona de restauración; (b) función sinusoidal resultante al variar solo el valor de número de onda; (c) función sinusoidal posterior a la zona de restauración.

### 6.2.1.2 Amplitud

Para cada línea de píxeles el valor de la amplitud en la función sinusoidal varía linealmente conforme avanza en la zona de restauración hasta llegar al valor de la función sinusoidal posterior a la zona de restauración, se ejemplifica en la Figura 6.5.

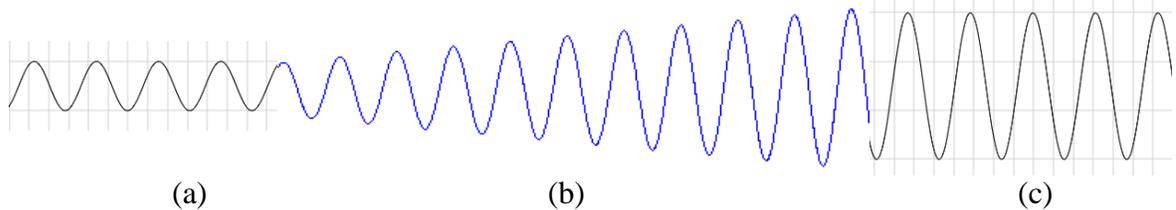


Figura 6.5 (a) función sinusoidal anterior a la zona de restauración; (b) función sinusoidal resultante al variar solo el valor de la amplitud; (c) función sinusoidal posterior a la zona de restauración.

### 6.2.1.3 Desplazamiento vertical

Para cada línea de píxeles el valor del desplazamiento vertical en la función sinusoidal varía linealmente conforme avanza en la zona de restauración hasta llegar al valor de la función sinusoidal posterior a la zona de restauración, se ejemplifica como se muestra en la Figura 6.6.

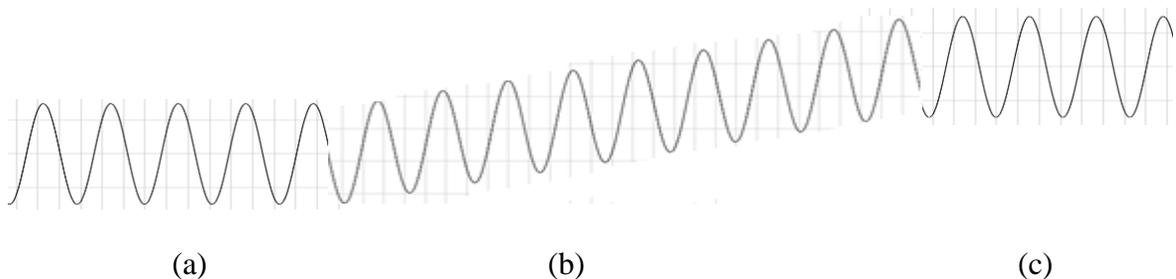


Figura 6.6 (a) función sinusoidal anterior a la zona de restauración; (b) función sinusoidal resultante al variar solo el valor del desplazamiento vertical; (c) función sinusoidal posterior a la zona de restauración.

#### 6.2.1.4 Constante de fase

Para cada línea de píxeles el valor de la constante de fase en la función sinusoidal que varía linealmente conforme avanza en la zona de restauración hasta llegar al valor de la función sinusoidal posterior a la zona de restauración simplemente cambia el valor de fase en toda la función de la zona de restauración y no se visualiza su efecto en la gráfica pero es importante porque es la variable que ayuda a que cada función inicien y terminen en fase entre ellas.

Por lo que el efecto combinado entre la amplitud, el número de onda, el desplazamiento vertical y la constante de fase variando linealmente conforme avanza en la zona de restauración se ejemplifica como se muestra en la Figura 6.7.

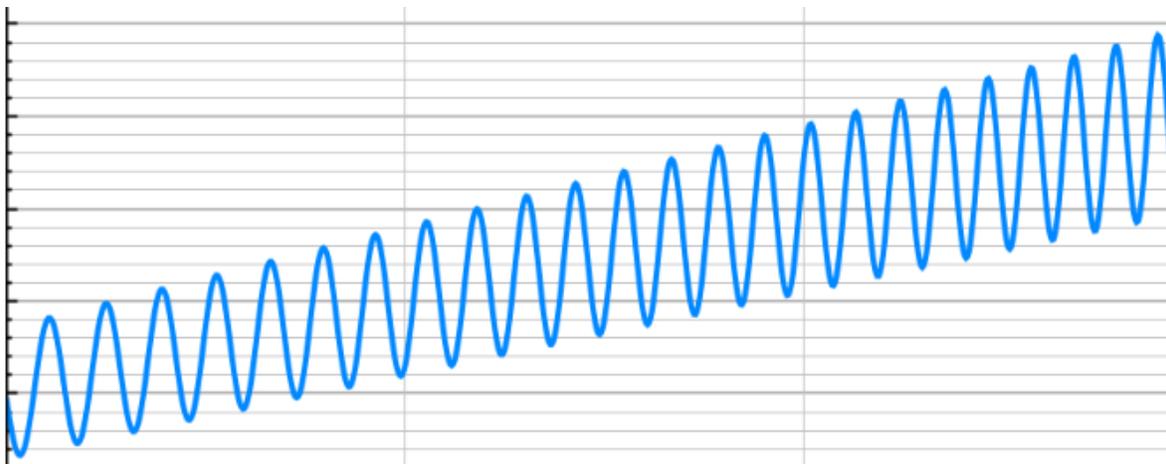


Figura 6.7 Ejemplo de gráfica en la zona de restauración con los valores de la función sinusoidal resultante al variar la amplitud, el número de onda, el desplazamiento vertical y la constante de fase variando linealmente.

#### 6.2.2 Programación del algoritmo sinusoidal

El desarrollo del algoritmo sinusoidal se llevó a cabo en el software Matlab para el cual se utilizó la *graphical user interface* (GUI) que puede implementar de manera gráfica controles y botones para la fácil utilización del programa de restauración como puede ser cargar imágenes y crear gráficas.

El programa de restauración abarca tres etapas principalmente. La primera etapa, consiste en cargar la imagen a analizar y hacer una conversión de formato a escala de grises si es necesario. Posteriormente, realizar una segmentación a la imagen con oclusión, en esta imagen lo que nos interesa es seleccionar la zona con oclusión para generar una nueva imagen de igual tamaño que denominaremos máscara, la cual es una imagen binaria con valor de píxeles de 255 en el área a restaurar y valor de 0 en el resto de la imagen.

La segunda etapa consta de utilizar la imagen original y la máscara previamente obtenida, para realizar una multiplicación matricial entre las imágenes de tal forma que se obtenga un valor de píxel de 0 en zona de oclusión de la imagen original. Además, hacer una detección de filas de estos valores para restaurar posteriormente.

Finalmente, en la tercera etapa utilizamos la imagen obtenida previamente y aplicamos para cada fila una detección de cambios de pendiente de positivo a negativo y viceversa para la detección de valles y crestas en la onda sinusoidal tanto para los píxeles a la izquierda y a la derecha de la zona a restaurar. Posteriormente, identificamos y creamos las funciones sinusoidales con sus variables de igual manera a la izquierda y a la derecha de la zona a restaurar para poder hacer una diferencia entre ellas misma y crear, según el número de píxeles a restaurar, un paso por píxel para cada variable, y con ello, rellenar la información con estos valores discretos en la zona con valores de píxel igual a 0.

Un diagrama de flujo que representa el proceso de restauración se puede observar en la siguiente Figura 6.8 y las principales partes del código de programación en Matlab se pueden consultar en el Anexo al final de este trabajo de investigación.

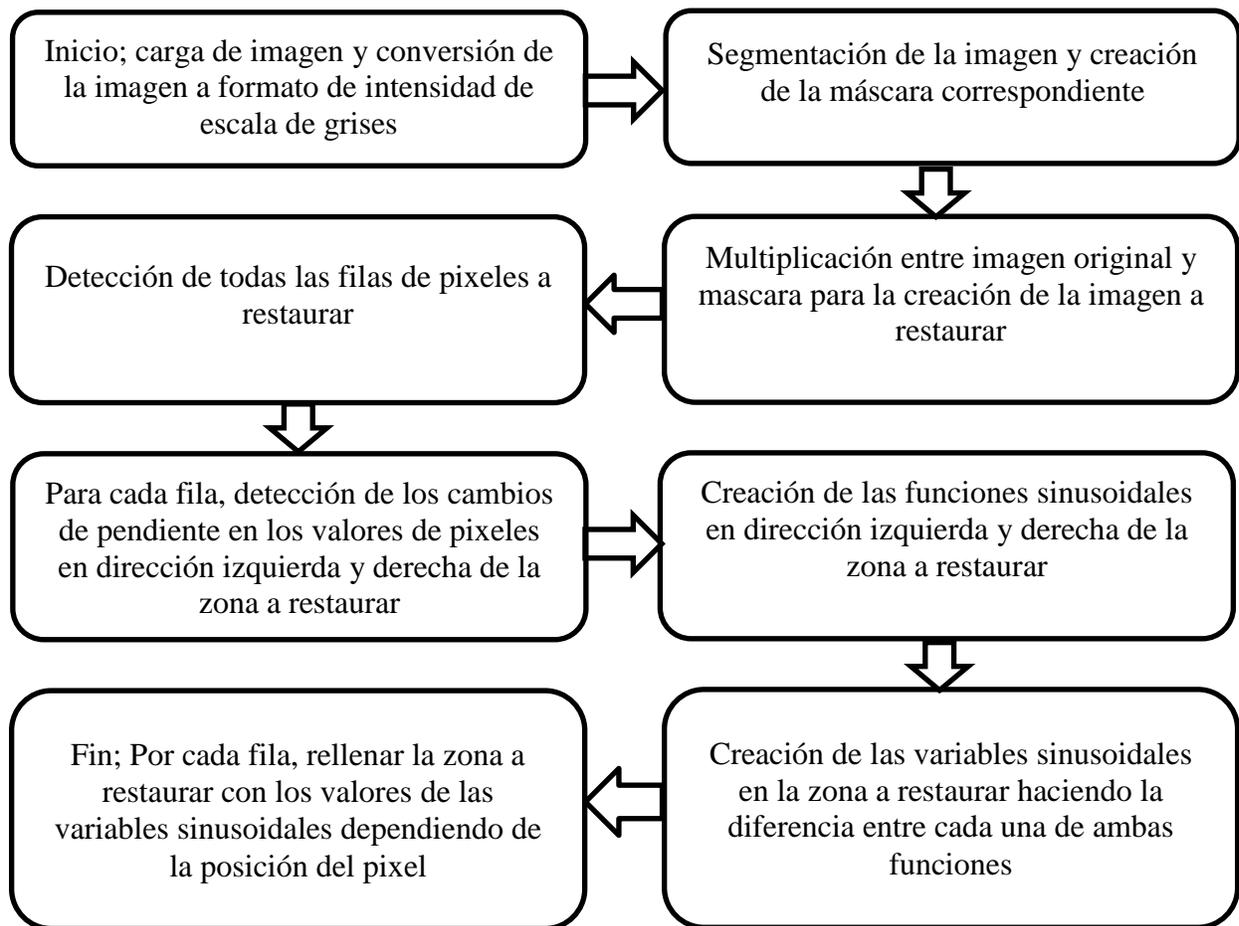


Figura 6.8 Diagrama de flujo del algoritmo sinusoidal.

A continuación, en el siguiente capítulo, se muestra el arreglo experimental que se utilizó para la toma de imágenes y posterior restauración, así como su reconstrucción 3D con oclusiones corregidas gracias al algoritmo sinusoidal desarrollado en este trabajo de investigación.

## CAPÍTULO VII

### ARREGLO EXPERIMENTAL

#### 7.1 Elementos del sistema

Los elementos requeridos para el empleo del algoritmo sinusoidal son: primero un dispositivo que proyecte un patrón de franjas, segundo un objeto a reconstruir, tercero una cámara que capture la imagen del objeto, en nuestro caso utilizamos un proyector DLP (*Digital Light Processing*) y una cámara CMOS (*Complementary Metal-Oxide-Semiconductor*), cuarto una computadora el cual procese la imagen a reconstruir y además implemente el algoritmo sinusoidal. Un diagrama de flujo del arreglo experimental y como relacionan la información se muestra en la figura 7.1.

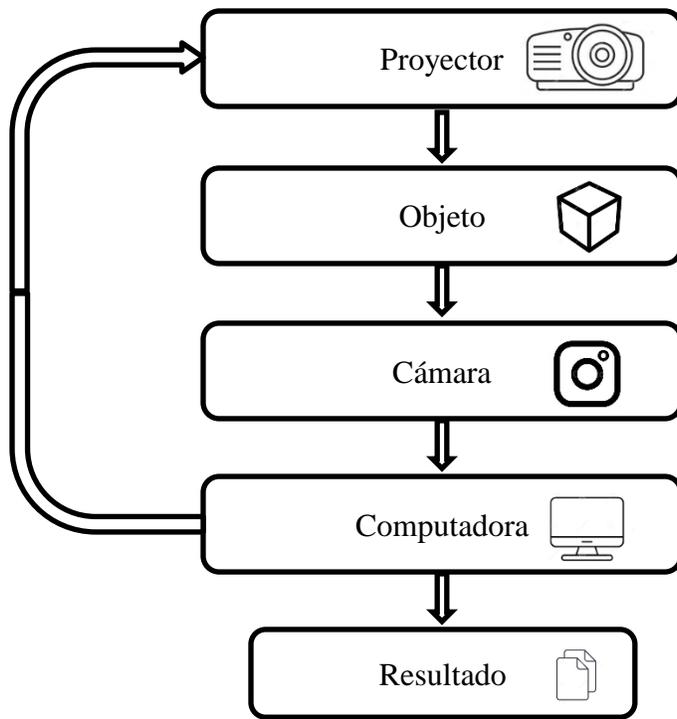


Figura 7.1 Diagrama de flujo del arreglo experimental.

A continuación, una descripción general del proyector y la cámara involucrados en el arreglo experimental.

## 7.2 Proyector

Como fuente de luz se utilizó un proyector Toshiba DLP modelo TDP-FF1, el cual cuenta con una resolución de 800x600 pixeles, con una lámpara tipo LED, contraste de 1500:1, brillo de 400 lúmenes y apertura de lente F/2.6 con una distancia de entre 0.4 m a 2.5 m.

El proyector (Figura 7.2) cuenta con la posibilidad de ajustar el tamaño de la imagen, así como la corrección limitada del efecto trapezoidal que es una distorsión que se presenta cuando se forma un ángulo entre el eje de proyección con respecto a la normal del plano de referencia.

La resolución y potencia, a pesar de su pequeño tamaño, cumple con lo necesario para la distancia propuesta en el arreglo y que el patrón sea lo suficientemente nítido para la captura de la imagen. El proyector es conectado por puerto VGA (*Video Graphics Array*) a la computadora y de esta manera se proyectó el patrón de franjas generado sobre el objeto.



Figura 7.2 Proyector Toshiba DLP modelo TDP-FF1.

Para el cálculo del tamaño de imagen proyectada que tiene que abarcar el objeto entero, se consultó el manual de operación donde muestra la relación de distancia con el tamaño de proyección, pero tomando en cuenta también la nitidez y resolución del patrón de franjas.

La Figura 7.3 muestra la disposición de distancias mientras la Tabla 7.1 muestra la relación entre la distancia de proyección y el tamaño de proyección [23].

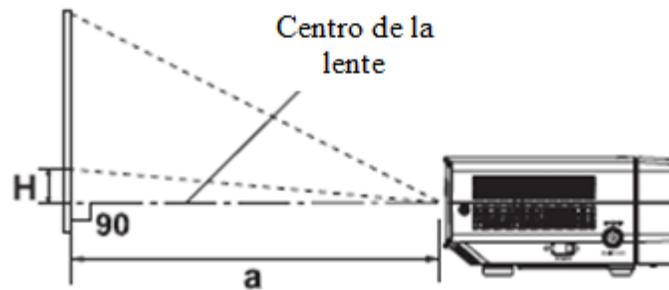


Figura 7.3 Vista lateral del proyector donde “a” es la distancia de la lente y la pantalla y H es la altura desde el fondo de la imagen al centro de la lente.

Tamaño de proyección (pulgadas)	Distancia de proyección (pies)	Altura (H) (pulgadas)
11	1.32	0.3
17	2.04	1
25	3	1.5
30	3.6	1.8
35	4.2	2.1
40	4.8	2.4
45	5.4	2.7
50	6	3
60	7.2	3.6
68	8.16	4.1

Tabla 7.1 Relación entre distancia de proyección y el tamaño de proyección medido en diagonal.

### 7.3 Cámara y lente

Para capturar las imágenes se utilizó una cámara Thorlabs modelo DCU224C (Figura 7.4) que cuenta con un sensor óptico tipo CCD (*Charge-Coupled Device*), una resolución de 1280x1024 píxeles (1.31 Megapíxeles), relación de aspecto 5:4 y diagonal de sensor óptico de 8.52 mm. Dichas características cumplen para obtener buen contraste en las franjas y obtener imágenes con buen nivel de detalle para la obtención de información del arreglo experimental [24].



Figura 7.4 Cámara CCD marca Thorlabs modelo DCU224C.

Además, la cámara es acoplada a una lente óptica Thorlabs modelo MVL25 (Figura 7.5) con distancia focal de 25mm, enfoque manual, apertura de lente F/1.6 y montura tipo C.



Figura 7.5 Lente óptica Thorlabs modelo MVL25.

#### 7.4 Arreglo experimental

El arreglo experimental que fue implementado para hacer pruebas en laboratorio es un arreglo triangular como se muestra en la Figura 7.6.

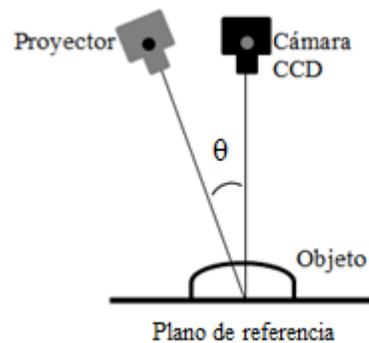


Figura 7.6 Esquema del arreglo experimental.

El arreglo experimental se montó con las siguientes distancias: desde la cámara CCD y del proyector al plano de referencia son colocados a una distancia de 1.64m al plano de referencia ya que siguiendo la Tabla 7.1 obtenemos una proyección de imagen diagonal de 1m lo que es suficiente para cubrir por completo los objetos a analizar. Por último, la separación entre la cámara CCD y el proyector es de 0.55m. Por lo que el ángulo formado por los ejes ópticos de la cámara y el proyector es de  $18.5^\circ$  el cual es un ángulo suficiente para crear oclusiones. El arreglo físico se puede observar en la figura 7.7.

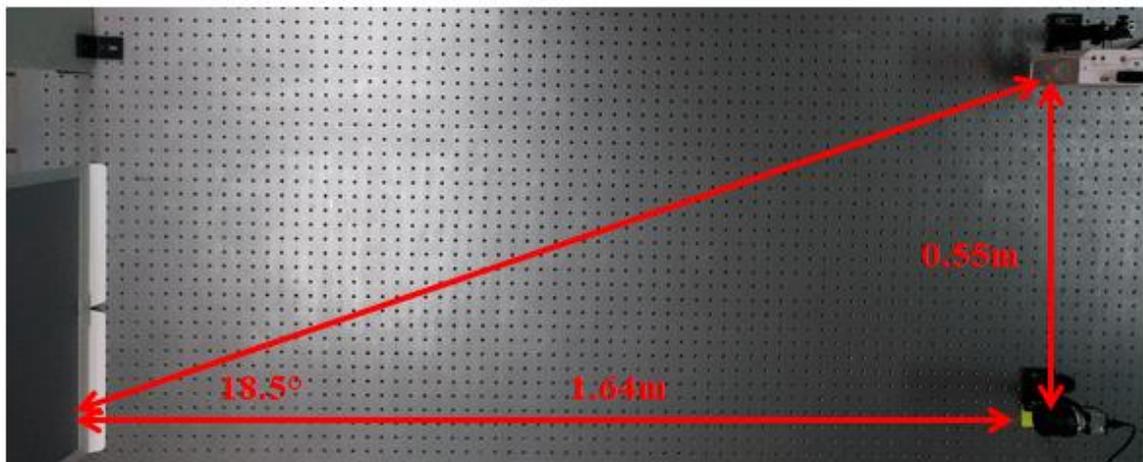


Figura 7.7 Distancias del arreglo experimental.

En la Figura 7.7 se puede observar que el proyector está situado de manera vertical esto es para eliminar por completo el problema del efecto trapezoidal generado por ángulo entre el eje de proyección con respecto a la normal del plano de referencia ya que al estar rotada  $90^\circ$  la imagen de proyección, los bordes trapezoidales quedan en sentido contrario a las franjas proyectadas sin distorsionarlas [25].

## 7.5 Software

Durante la elaboración del trabajo de investigación y posterior arreglo experimental se hicieron uso de varios softwares para diferentes propósitos, los principales se enuncian a continuación con una breve descripción de cada uno de ellos.

### 7.5.1 MATLAB

Para la realización del algoritmo sinusoidal se utilizó el software MATLAB el cual ofrece un entorno de desarrollo integrado en un lenguaje de cálculo técnico de alto nivel para cálculos científicos y de ingeniería. Entre sus principales características está el desarrollo de algoritmos, análisis de datos y su visualización mediante gráficas, aplicaciones para ajustar curvas, clasificar datos y ajustar sistemas de control además de herramientas para crear aplicaciones con interfaces de usuario personalizadas [26].

### 7.5.2 NI LabVIEW

La generación del patrón de franjas sinusoidales se realizó en el software NI LabVIEW ya que ofrece un enfoque de programación gráfica con un entorno amigable, esta visualización hace que sea más fácil representar una lógica compleja en el diagrama, desarrollar algoritmos de análisis de datos y diseñar interfaces de usuario personalizadas [27].

En nuestro caso se diseñó una interface de usuario personalizada con varias botoneras y configuraciones para las franjas sinusoidales, como la cantidad y el tamaño de la resolución de éstas, además de una opción de generación de una imagen de calibración como se muestra en la Figura 7.8.

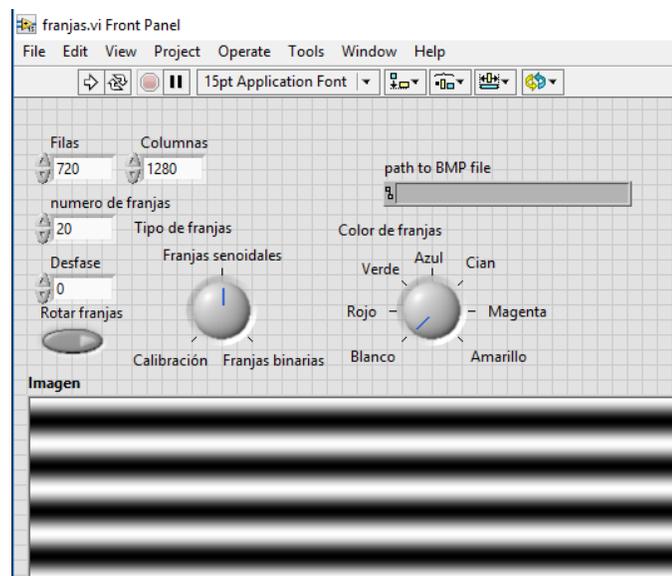


Figura 7.8 Interface de la aplicación desarrollada en LabVIEW para la generación de franjas.

### 7.5.3 Uc480Viewer

El software que controla la cámara es el uc480Viewer suministrado por Thorlabs, se utilizó el software para la captura de imágenes así como para la calibración de la misma, al poder centrar el eje óptico de la cámara con la imagen de calibración suministrada por el proyector que a su vez lo genera LabVIEW. Asiando ajustes manuales se puede llegar a coincidir estos dos ejes con el plano de referencia como se observa en la Figura 7.9

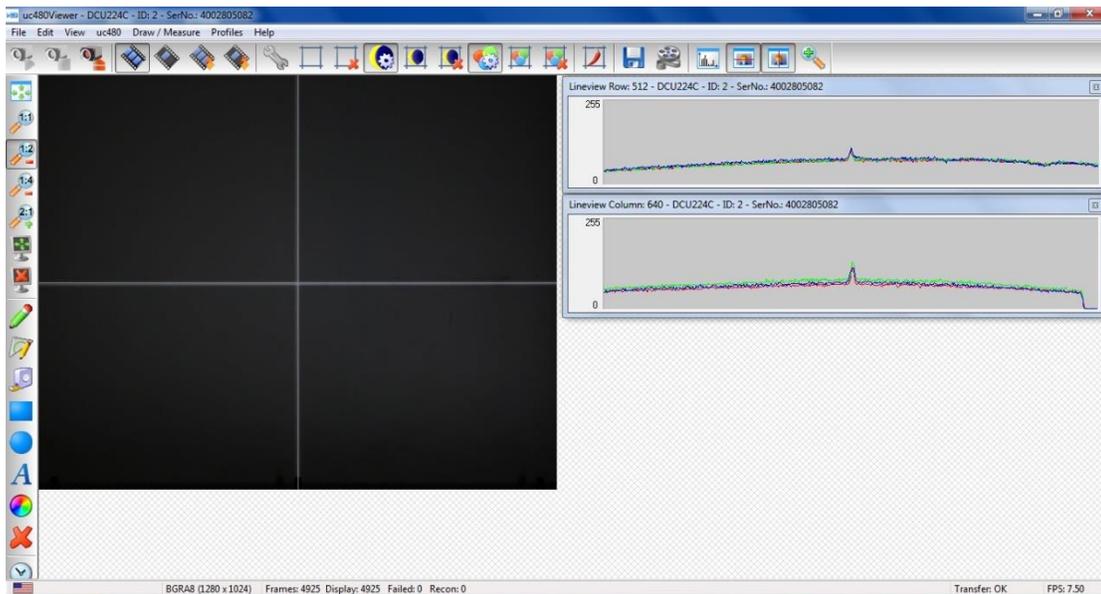


Figura 7.9 Interface del Software uc480Viewer realizando la calibración del arreglo óptico.

### 7.5.3 Wolfram Mathematica

El software de Wolfram Mathematica se utilizó para la comparación del algoritmo sinusoidal con otros algoritmos de inpainting ya conocidos y por lo tanto comerciales. Mathematica proporciona un único sistema integrado, que cubre todos los ámbitos de la computación técnica y por lo tanto con funciones y algoritmos ya desarrollados.

Una función especialmente útil en nuestro proyecto fue la función de Inpaint de Mathematica (Figura 7.10), el proceso es simple e intuitivo se selecciona la imagen a restaurar y una imagen de mascara que indique la región a restaurar y entrega el resultado. Es importante que al usar algoritmos de Inpaint comerciales como por ejemplo el de *Texture*

*synthesis*, entonces, se puede comparar el desempeño de nuestro algoritmo sinusoidal con estos últimos [28].

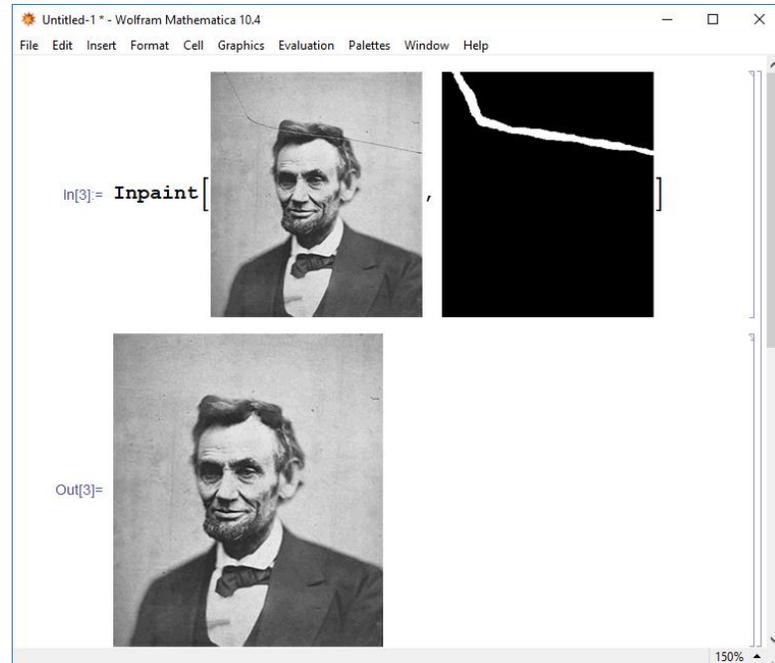


Figura 7.10 Software Wolfram Mathematica haciendo una restauración de imagen con la función `Inpaint`.

## CAPÍTULO VIII

### RESULTADOS EXPERIMENTALES Y CONCLUSIONES

En este capítulo se presenta la implementación del algoritmo sinusoidal desarrollado en este proyecto de tesis, para corregir oclusiones en imágenes de perfilometría 3D.

Primeramente, para demostrar el efecto de la utilización de inpainting en imágenes con zona de oclusión se decidió hacer dos pruebas de laboratorio, la primera consiste en la reconstrucción del objeto de prueba sin utilización de algoritmo de restauración y posteriormente utilizando el algoritmo de restauración desarrollado en este trabajo de investigación. La segunda prueba consiste en el mismo procedimiento pero utilizando el algoritmo de restauración *Texture synthesis* para demostrar que se pueden obtener resultados muy similares con otros algoritmos de *Inpainting* comerciales para la corrección de estas oclusiones, Finalmente se muestran los resultados de ambas pruebas.

#### 8.1 Primera prueba de laboratorio

##### 8.1.1 Reconstrucción de superficie compleja sin Inpainting

Para esta prueba un patrón sinusoidal es proyectado sobre una superficie convexa la cual se le realizó una cavidad en el centro para simular un objeto con superficie compleja y generar una oclusión como se aprecia en la Figura 8.1. Además, el objeto está montado sobre un plano que funcionará como plano de referencia.

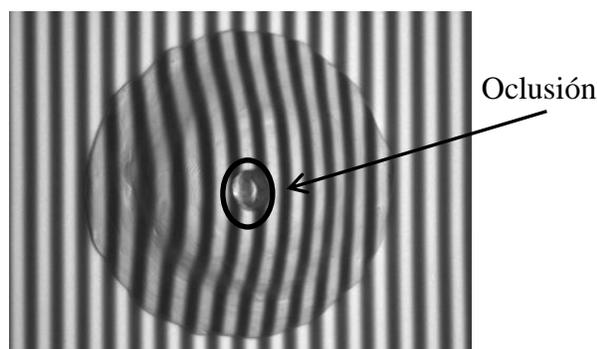


Figura 8.1 Superficie a reconstruir y su zona de oclusión.

Al realizar el proceso de reconstrucción 3D en Matlab por método de Fourier tenemos que al calcular la fase envuelta se tiene un defecto en esta zona lo cual podemos observar en la Figura 8.2 (a). Lo mismo para la fase desenvuelta en la cual podemos observar el salto de fase en la Figura 8.2 (b). Este defecto afecta a la reconstrucción final por lo que es algo que se quiere evitar, una solución recurrente a este problema es enmascarar esta zona y dejarla sin información, lo que nos da como resultado una reconstrucción incompleta y poco estética, pero como se verá en la siguiente sección, si se hace un procesado de restauración en esta zona se puede obtener un mapa de fase continuo para la misma imagen

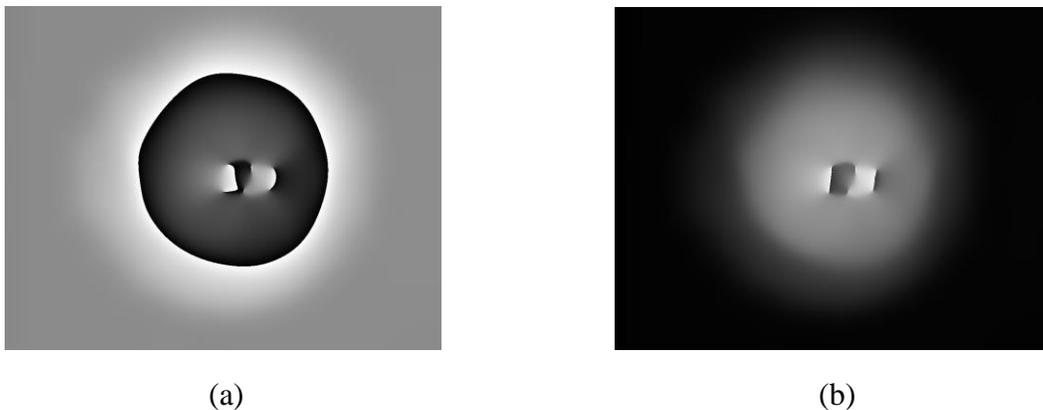


Figura 8.2 (a) Mapa de fase envuelta del patrón con defecto; (b) Mapa de fase desenvuelta con defecto.

### 8.1.2 Reconstrucción de superficie compleja con Inpainting

Continuando con la prueba de laboratorio ahora la misma imagen se sometió a un proceso de inpainting previo a la reconstrucción 3D, el algoritmo utilizado fue el desarrollado en esta investigación el algoritmo sinusoidal de inpainting para este tipo de imágenes, al restaurar esta imagen la discontinuidad de las franjas ha desaparecido como se puede observar en la Figura 8.3. Lo mismo para el mapa de fase envuelta y mapa de fase desenvuelta de la Figura 8.4 (a) y Figura 8.4 (b) respectivamente. Finalmente, al hacer la reconstrucción final se tiene un resultado satisfactorio sin ningún rastro de saltos de fase y por lo tanto sin defectos en esta zona como se puede apreciar en la Figura 8.5.

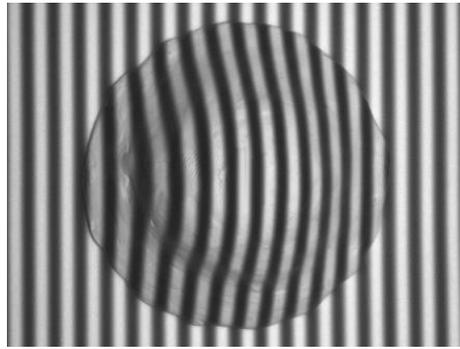
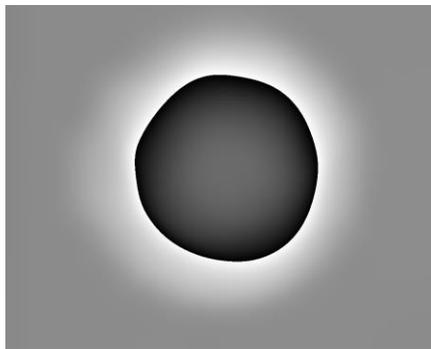


Figura 8.3 Objeto a reconstruir y su zona de oclusión restaurada por el algoritmo sinusoidal.



(a)



(b)

Figura 8.4 (a) Mapa de fase envuelta del patrón sin defecto; (b) Mapa de fase desenvuelta sin defecto.

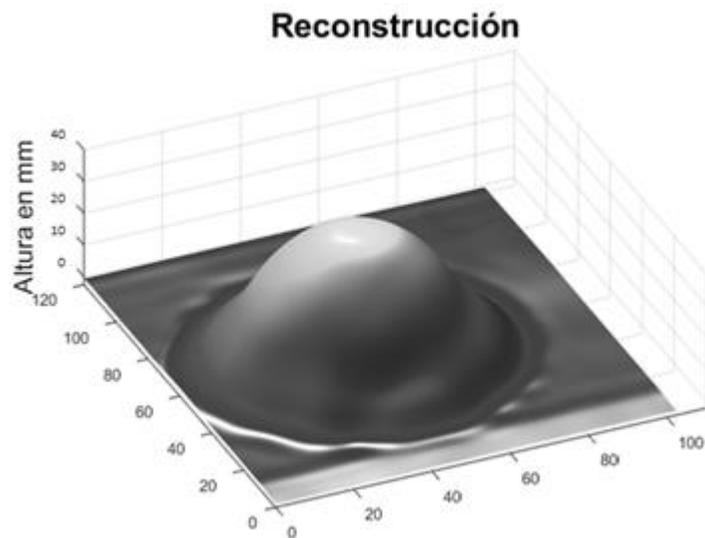


Figura 8.5 Reconstrucción final 3D con el algoritmo desarrollado.

## 8.2 Segunda prueba de laboratorio

### 8.2.1 Reconstrucción de objeto complejo sin Inpainting

En esta prueba un patrón sinusoidal es proyectado sobre un maniquí simulando un perfil humano, éste crea una zona de oclusión justo en la zona de la oreja como se puede apreciar en la Figura 8.6. Nuevamente sin restaurar el área de oclusión y al realizar el proceso de reconstrucción 3D, esta zona de discontinuidad puede ocasionar salto de fase y por lo tanto un error en la reconstrucción final como se muestra en la Figura 8.7, si hacemos un perfil de línea de izquierda a derecha sobre esta área tenemos un defecto debido al salto de fase como lo muestra la Figura 8.8.

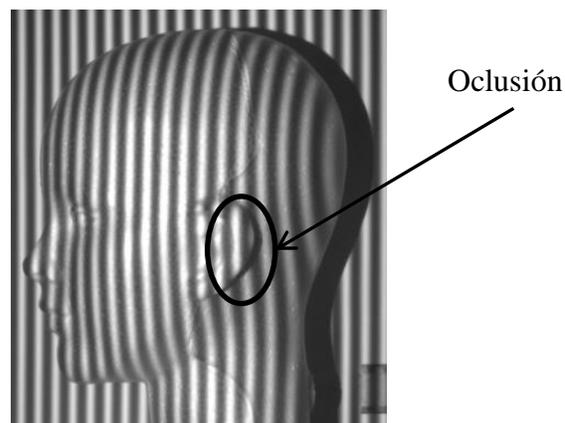


Figura 8.6 Objeto a reconstruir y su zona de oclusión.

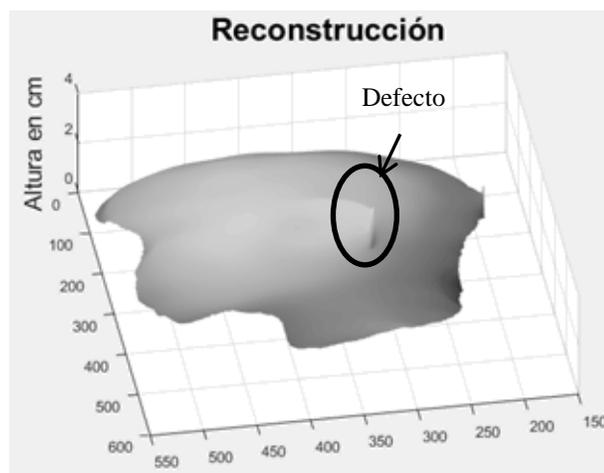


Figura 8.7 Defecto en la reconstrucción 3D sin inpainting.

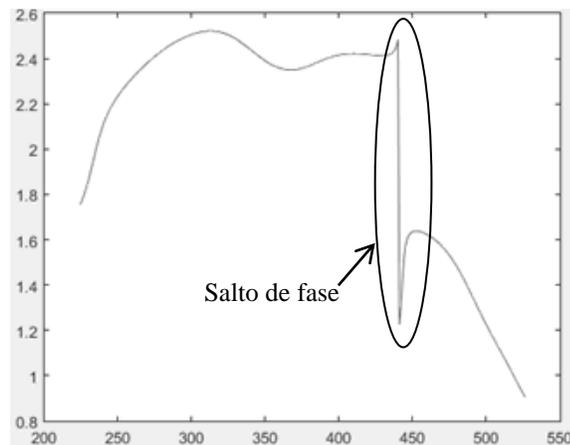


Figura 8.8. Perfil de línea del área de oclusión mostrando el error del salto de fase.

### 8.2.2 Reconstrucción de objeto complejo con Inpainting

Posteriormente, la misma imagen se sometió a un proceso de inpainting previo a la reconstrucción 3D, el algoritmo utilizado fue el *Texture synthesis* para lo cual nos apoyamos en el software Mathematica 10.0 de Wolfram el cual incorpora este algoritmo dentro de su función de inpainting [10]. Al restaurar esta imagen la zona de oclusión disminuyó en gran medida como se puede observar en la Figura 8.9, el proceso de reconstrucción de esta zona tiene una notable mejora y su reconstrucción final es de mejor calidad como lo muestra la Figura 8.10. Finalmente, se realiza un perfil de línea de izquierda a derecha sobre la misma zona tenemos que el defecto debido al salto de fase ha sido corregido satisfactoriamente como se puede observar en la Figura 8.11.

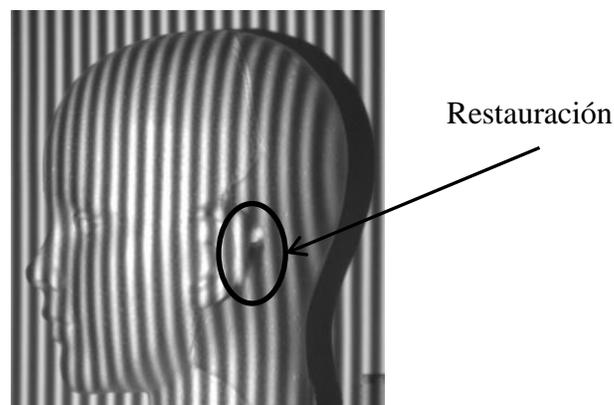


Figura 8.9 Restauración en la zona de oclusión.

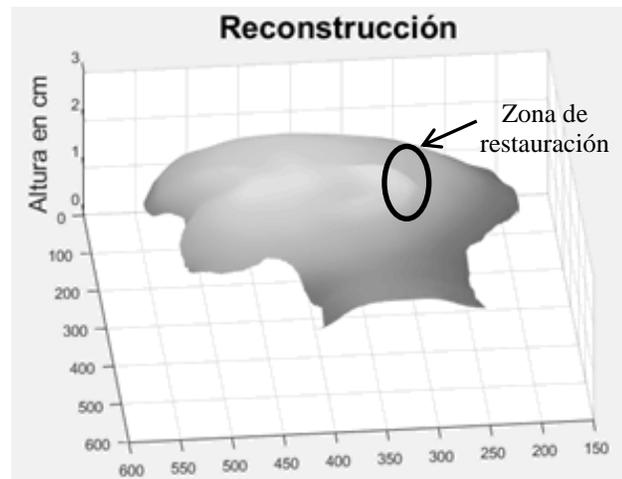


Figura 8.10 Reconstrucción 3D con inpainting.

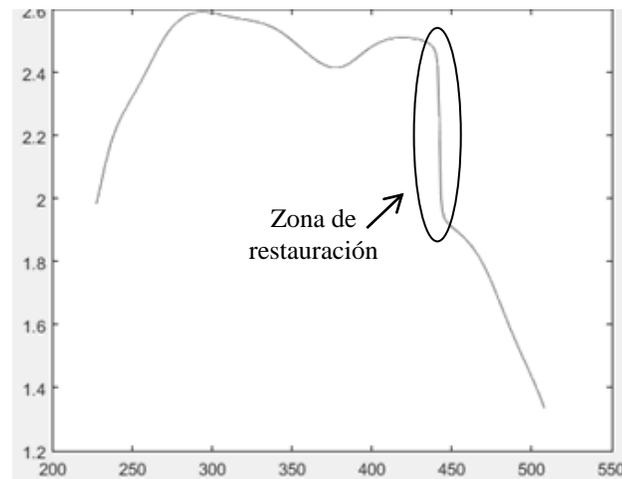


Figura 8.11 Perfil de línea del área de oclusión mostrando la disminución del error.

### 8.3 Conclusiones

El resultado inmediato es que se logró implementar el algoritmo propuesto en este trabajo de tesis, sin embargo, hay margen de mejora en tanto en el proceso de reconstrucción como en la automatización del proceso. Futuros trabajos serian de utilidad para el refinamiento del método propuesto.

Se ha cumplido nuestro objetivo de desarrollar un algoritmo que aprovechara la información presente en la imagen digital y que combinándolo con un método de

perfilometría 3D se diera una respuesta inmediata al problema de oclusiones en este tipo de investigaciones.

Durante el transcurso del proyecto de investigación se presentaron innumerables retos para los cuales fue necesario enfrentarlos desde diferentes ángulos, pero gracias a todas las personas involucradas en este proceso se logró de manera satisfactoria.

La investigación fue exhaustiva en diferentes áreas de matemáticas, física, ingeniería y sistemas computacionales. Por lo que es un proyecto que abarca varias áreas de aprendizaje y esto nos dejó una mayor experiencia de preparación y de vida. El algoritmo queda como conocimiento adquirido y además general para que se pueda seguir desarrollando en un futuro.

Podemos concluir que el proyecto de tesis fue un éxito al crear un algoritmo eficiente que ataca el problema de oclusiones en objetos complejos para la perfilometría 3D.

REFERENCIAS

- [1] Eleázar Jaramillo, Andrés; Flavio Prieto, Pierre Boulanger “Inspección de piezas 3D: Revisión de la literatura” en Revista Ingeniería e Investigación, diciembre, vol. 27, No. 003, Universidad Nacional de Colombia. 2007
- [2] Vaquero, D.A. etal. “A Projector-camera Setup for Geometryinvariant Frequency Demultiplexing.” Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2009, IEEE, 20-25 June 2009.
- [3] E. B. Li, X. Peng, J. Xi, J. F. Chicharo, J. Q. Yao and D. W. Zhang, “Multi-frequency and multiple phase-shift sinusoidal fringe projection for 3D profilometry,” Optical Society of America, Vol. 13, No. 5, 2005, pp. 1561-1569
- [4] Diana Beltrán Guerrero, Luis Basañez Villaluenga, “Técnicas y algoritmos para la adquisición, transmisión y visualización de escenas 3D”, 2008.
- [5] Bastanlar, Y., Grammalidis, N., Zabulis, X., Yilmaz, E., Yardimci, Y. and Triantafyllidis, G., “3DD Reconstruction for a Cultural Heritage Virtual Tour System,” in Proc. Of International Society for Photogrammetry and Remote Sensing, 2008, pp 1023-1028
- [6] Mikkel Damgaard Olsen, “Structured Light in Matlab”, Kongens Lyngby 2010, IMM-B.Sc.-2010-17
- [7] J. Salvi, S. Fernandez, T. Pribanic, and X. Llado, “A State of the Art in Structured Light Patterns for Surface Profilometry,” Pattern Recognition, Vol. 43, No. 8, 2010, pp. 2666-2680.
- [8] S. Im, Y. Lee, J. Kim, and M. Chang, “A Solution for Camera Occlusion Using a Repaired Pattern from a Projector,” International Journal of Precision Engineering and Manufacturing, Vol. 17, No. 11, 2016, pp. 1443-1450.
- [9] Geng, J. “Structured-light 3D surface imaging: a tutorial,” Advances in Optics and Photonics, Vol.3 issue 2, pp. 128-160, 2011.
- [10] Sai Siva Gorthi, Pramod Rastogi. “Fringe projection techniques; Whither we are?,” Optics and Lasers in Engineering, vol. 48, pp. 133-140, 2010
- [11] Abdul-Rahman, H. “Three-Dimensional Fourier Fringe Analysis and Phase Unwrapping,” Ph.D. thesis, Liverpool John Moores University, United Kingdom, 2007
- [12] Toru Yoshizawa. Handbook of Optical Metrology Principles and Applications, Boca Raton, FL, Taylor & Francis Group, 2009, pp 437-489
- [13] M. Takeda and H. Motoh, “Fourier transform profilometry for the automatic measurement of 3-D object shapes,” Appl. Opt. 22, 1983, pp. 3977-3982.

- [14] M. Takeda, H. Ina, and S. Koboyashi, "Fourier-transform method of fringe-pattern analysis for computer-based topography and interferometry," *J. Opt. Soc. Am.* 72, 1982, pp. 156-160
- [15] Sicardi-Segade, A., Estrada, J. C., Martínez-García, A., & Garnica, G. (2015). On axis fringe projection: A new method for shape measurement. *Optics and Lasers in Engineering*, 69, 29-34.
- [16] Diana Vanessa Gómez Trejos. Alejandra Guerrero Guzmán. Ing. Carlos Augusto Meneses Escobar. "Estudio y análisis de técnicas para procesamiento digital de imágenes" Universidad Tecnológica De Pereira. Ingeniería De Sistemas Y Computación. Mayo del 2016.
- [17] Kjell J. Gasvik. *Optical Metrology*. Wiley, third edition, 2002. ISBN 0470843004.
- [18] González, R.C., Wintz, P. "Procesamiento digital de imágenes" Addison-Wesley, (1996), Tema 3,4, pág 89-269
- [19] T. F. Chan, and J. H. Shen, "Non-texture inpainting by curvature driven diffusions," *J. Visual Commun. Image Rep.* 4, 2001, pp. 436-449.
- [20] A. Efros and K. Leung, "Texture Synthesis by Non-parametric Sampling," *IEEE International Conference on Computer Vision (ICCV'99)*, Corfu, Greece, September 1999.
- [21] A. Efros and W. T. Freeman, "Image quilting for texture synthesis and transfer," in *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*, ASSOC Computing Machinery, 2001, pp. 341-346.
- [22] P. Beer, E. Johnston, "Mecanica vectorial para ingenieros estatistica y dinamica," McGrall Hill, novena edicion ISBN 970-106103-9
- [23] Toshiba Owner's Manual LED Projector TDP-FF1A. Toshiba Corporation, Irvine, CA., 2000-2005
- [24] ThorLabs CCD and CMOS cameras Operation Manual and SDK version 4.20 2013
- [25] Kjell J. Gasvik. *Optical Metrology*. Wiley, third edition, 2002. ISBN 0470843004.
- [26] Mathworks documentation center [en línea], disponible: <https://la.mathworks.com/help/matlab/> visitado: 13/Abr/2018.
- [27] LabVIEW Development Guidelines, National Instruments Corporation, April 2003 Edition, Part Numer 321393D-01
- [28] Wolfram Mathematica 10.0 manual and guide documentation center, Image Processing & Analysis, Inpainting [en línea], disponible: <http://reference.wolfram.com/language/guide/ImageProcessing.html> visitado: 26/Jun/2018.

## ANEXO

## SEGMENTAR

```
function [imagenbinaria,imagenquemada]= segmenta(imagen)
fontSize = 16;
I=imagen;
f=figure;
imshow(I);
    axis on;
    title('Imagen a procesar', 'FontSize', fontSize);
    set(gcf, 'Position', get(0,'Screensize'));
    message = sprintf('Click izquierdo sin soltar para comenzar a
segmentar.\nLevantarse para terminar');
    uiwait(msgbox(message));
hFH = imfreehand();
imagenbinaria = hFH.createMask();
xy = hFH.getPosition;
close(f)
imagenquemada = I;
imagenquemada(imagenbinaria) = 255;
end
```

## RESTAURACIÓN

```
function [imagenrestaurada] = restauracion(I,mask)
if isempty(I) || isempty(mask);
    msgbox(sprintf('No hay imagen'),'Error','Error');
else

    mask = logical(mask);

    imagenquemada = I;
    imagenquemada(mask) = NaN;

    [L,n]= bwlabel(mask,8);

    for area = 1:n
        [r, c] = find(L==area); %agregar para más regiones

        [submatrixin] = submatricesin(r,c); %rows all region
        % imtool (imagenquemada);
        [imagenrestaurada]= inpaint(r,c,submatrixin,imagenquemada); %
        imagenquemada = imagenrestaurada;
    end

end

end

function [imagenquemada] = inpaint(r,c,submatrixin,imagenquemada)
    rc = [r c];
```

```
%imtool(imagenquemada); %good

for selector = 1:max(r)
    if ~isempty(submatrixin{selector});
        xy = submatrixin{selector} ;
        x = selector;
        y = xy(:,2);
        rmin = min(r);
        rmax = max(r);

        [senthreepointsizq, submatrixizq]=
submatrixizq(x,y,imagenquemada,rmin,rmax);
        senthreepointsmatxizq{selector}=senthreepointsizq;

        [senthreepointsder, submatrixder]=
submatrixder(x,y,imagenquemada,rmin,rmax);
        senthreepointsmatxder{selector}=senthreepointsder;

        for n = 1:3
            pixelsubmatrixder(n) = submatrixder(senthreepointsder(n));
            pixelsubmatrixizq(n) = submatrixizq(senthreepointsizq(n));
        end

        lengthwaveizq = senthreepointsizq(3) - senthreepointsizq(1)+1;
        lengthwaveder = senthreepointsder(3) - senthreepointsder(1);

        numondader = 2*pi/lengthwaveder;
        numondaizq = 2*pi/lengthwaveizq;

        difamplitudder = diff(pixelsubmatrixder);
        difamplitudizq = diff(pixelsubmatrixizq);
        absdifamplitudder = abs(difamplitudder);
        absdifamplitudizq = abs(difamplitudizq);
        amplitudder= absdifamplitudder(1)/2;
        amplitudizq= absdifamplitudizq(2)/2;
        offsetamplitudder=(pixelsubmatrixder(1)+pixelsubmatrixder(2))/2;
        offsetamplitudizq=(pixelsubmatrixizq(2)+pixelsubmatrixizq(3))/2;

        if difamplitudizq(2)<0
            fasependienteizq = pi;
        else
            fasependienteizq = 0;
        end

        if difamplitudder(2)<0
            fasependienteder = pi;
        else
            fasependienteder = 0;
        end
        distanciarest = numel(y); %%%

        distanciafaseizq = numel(submatrixizq)-senthreepointsizq(3)+1;
```

```

    distanciafaseder = senthreepointsder(1);

    [ondader]= senoidalder(amplitudder, offsetamplitudder, numondader,
    distanciafaseder, fasependienteder, distanciarest);
    [ondaizq]= senoidalizq(amplitudizq, offsetamplitudizq, numondaizq,
    distanciafaseizq, fasependienteizq, distanciarest);

    [ondafinal] = diffbothwaves(amplitudder, amplitudizq,
    offsetamplitudder, offsetamplitudizq,...
    numondader, numondaizq, distanciafaseder,
    distanciafaseizq,fasependienteder, fasependienteizq, distanciarest);

    matrixinpaint(selector) = ondafinal;
    %temporal
    y1 = min(y); %max
    y1 = y1-1; %+
    for p=1:distanciarest
    imagenquemada(x,y1+p)=ondafinal(p); %- and ondader
    end

%
%     y2 = max(y); %max
%     y2 = y2+1; %+
%     for p=1:distanciarest/2
%     imagenquemada(x,y2-p)=ondader(p); %- and ondader
%     end

    end
end
%imtool(imagenquemada)
end

function [senthreepoints, submatrixizq]=
submatrizizq(x,y,imagenquemada,rmin, rmax)
y = min(y);
y = y-1;
n=1;
indexlogic=[];

while sum(indexlogic==1)<3
matrix = imagenquemada(x,y-n:y);
matrix = double(transpose(matrix));

    for index = 1:numel(matrix)-1
    element = matrix(index);
    element2 = matrix(index+1);
    if element == element2

        yizq = y-numel(matrix)+index;
        yder = y-numel(matrix)+index+1;

        matrixasum1 = imagenquemada(rmin:rmax,yizq);
        matrixasum2 = imagenquemada(rmin:rmax,yder);

```

```
sumizq = sum(matrixasum1);
sunder = sum(matrixasum2);

    if sumizq > sunder
    imagenquemada(x,yizq) = imagenquemada(x,yizq)+1;
    else
    imagenquemada(x,yder) = imagenquemada(x,yder)+1;
    end

end
end

matrix = imagenquemada(x,y-n:y);
matrix = double(transpose(matrix));

    diffmatrix = diff(matrix);
    signmatrix = sign(diffmatrix);
    circmatrix = circshift(signmatrix,1);
    pointmatrix = signmatrix.*circmatrix;
    ind = find(pointmatrix(2:end)<0);
    ind = ind+1;
    indexlogic = logical(ind);
    n=n+1;
end
senthreepoints = ind;
submatrixizq = matrix;
%imtool(imagenquemada);
end

function [y]= senoidalizq(A,offset,k, x, offsetx,distinpaint )
e = k*(x)+ offsetx;
n = 0:distinpaint-1;
y =(A*cos(k*n+e))+ offset;
%figure
%plot (n,y)
end

function [y]= diffbothwaves(Ader,Aizq, offsetder, offsetizq, kder, kizq,
xder, xizq, offsetxder, offsetxizq, distinpaint)
%amplitud
A = Aizq;
Astep = (Ader - Aizq)/(distinpaint);
%k
k = kizq;
kstep = (kder-kizq)/(distinpaint);

%e
%x = xizq;
%xstep = (xder - xizq)/distinpaint;

%fslope = offsetxizq;
```

```
%fslopestep = (offsetxder - offsetxizq)/distinpaint;

e = kizq*xizq + offsetxizq;
e2 = mod(kder*xder + kder*xder*distinpaint + offsetxder,2*pi);
estep = abs((e2-e))/(distinpaint);

%offset
offset = offsetizq;
offsetstep = (offsetder - offsetizq)/(distinpaint);

for n = 0:distinpaint-1
    Astepacu=Astep*(n);
    kstepacu=kstep*(n);
    estepacu=estep*(n);
    %xstepacu=xstep*(n);
    %fslopestepacu=fslopestep*(n);
    offsetstepacu=offsetstep*(n);
    y(n+1)
    = ((A+Astepacu)*cos((k+kstepacu)*n+(e+estepacu)))+(offset+offsetstepacu);
end
%e = k*x+offsetx;
%n = 0:distinpaint-1;
%y = (A*cos(k*n+e))+ offset;
%figure
%plot (n,y)
end
```

