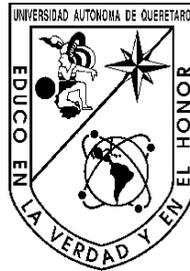


**UNIVERSIDAD AUTÓNOMA DE QUERÉTARO
FACULTAD DE INFORMÁTICA**



**Comparación de algoritmos de desenvolvimiento de fase para la
reconstrucción de imágenes 3d, usando patrones estructurados de
franjas en el Método de Perfilometría de Fourier.**

TESIS

QUE PARA OBTENER EL GRADO DE:

MAESTRO EN INGENIERÍA DE SOFTWARE DISTRIBUÍDO

PRESENTA:

**I.C. ALEJANDRO SERAFÍN
GONZÁLEZ OLVERA**

DIRECTOR DE TESIS:

DR. JOSÉ CARLOS PEDRAZA ORTEGA



Universidad Autónoma de Querétaro
Facultad de Informática
Maestría en Ingeniería de Software Distribuido

Comparación de algoritmos de desenvolvimiento de fase para la reconstrucción de imágenes 3D, usando patrones estructurados de franjas en el Método de Perfiometría de Fourier.

Tesis

Que como parte de los requisitos para obtener el Grado de Maestro en Ingeniería de Software Distribuido

Presenta:

I. C. Alejandro Serafín González Olvera

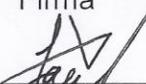
Dirigido por:

Dr. Jesús Carlos Pedraza Ortega

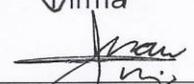
Dr. Jesús Carlos Pedraza Ortega
Presidente


Firma

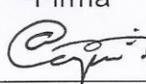
Dr. Saúl Tovar Arriaga
Secretario


Firma

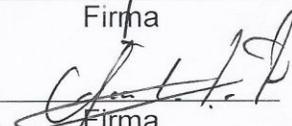
Dr. Juan Manuel Ramos Arreguín
Vocal


Firma

Dr. Marco Antonio Aceves Fernández
Suplente


Firma

Dr. Efrén Gorrostieta Hurtado
Suplente


Firma


M. C. Ruth Angélica Rico Hernández
Directora de la Facultad


Dr. Irineo Torres Pacheco
Director de Investigación y Posgrado

RESUMEN

El método de análisis de patrones estructurados de franjas para la reconstrucción de imágenes de objetos en modelos 3D dentro de la memoria de una computadora, ha sido ampliamente usado desde hace más de treinta años. Debido a que el análisis de patrones de franjas resulta un excelente método en el estudio y la reconstrucción de objetos de forma pasiva no invasiva, se han propuesto un gran número de soluciones que intentan resolver los problemas inherentes al desenvolvimiento de fase, considerado como uno de los problemas más difíciles de resolver en matemáticas e ingeniería. La gran cantidad de investigaciones y propuestas en este campo, ha resultado en una todavía más grande cantidad de soluciones formuladas en algoritmos que buscan mejorar sus características más deseables, su alta efectividad y velocidad de procesamiento. Son muchas las áreas de conocimiento que se benefician de estas técnicas, entre otras, la Medicina, la Astronomía, la Microscopía o la Geología, en aplicaciones específicas como en la Resonancia Magnética (MRI), el Radar y el Sonar de Apertura Sintética (SAR). Con el crecimiento exponencial de la tecnología, particularmente de la tecnología computacional, se pueden llevar a cabo algunos de los algoritmos creados hace años que no pudieron ser implementados debido a limitaciones tecnológicas.

En este trabajo de investigación, se comparan los dos métodos de desenvolvimiento de fase: el método global y el método local, usando objetos virtuales a los que se proyecta un patrón estructurado de franjas, las imágenes resultantes de esta proyección, son analizadas por medio de la *Perfilometría* de Fourier, que consiste básicamente en recuperar la fase envuelta de la señal usando algoritmos de desenvolvimiento de fase en un proceso computacional llamado análisis de patrones de franjas. La transformada inversa de Fourier, es usada para llevar a cabo el desenvolvimiento de la fase, lo que nos llevará a obtener la reconstrucción tridimensional del objeto de estudio. Finalmente se muestran los resultados obtenidos.

(Palabras Clave: imágenes 3D, *perfilometría*, desenvolvimiento, patrones de franjas).

SUMMARY

The structured fringe pattern analysis method to reconstruct images of objects in 3D models within the memory of a computer, has been widely used for over thirty years. Due to the analysis of fringe patterns is an excellent method, in the study and reconstruction of objects in a passive non invasive way, have been proposed a large number of solutions that try to solve the problems inherent in the phase unwrapping, considered one of the most difficult problems to solve in mathematics and engineering. The large amount of research and proposals in this field, has been resulted in a larger amount of solutions still formulated in algorithms that seek to improve their most desirable characteristics, its high efficiency and processing speed. There are many areas of knowledge that benefit from these techniques, among medicine, astronomy, microscopy or geology, in specific applications such as Magnetic Resonance (MIR), radar and Synthetic Aperture Sonar (SAS). With the exponential growth of technology, particularly computer technology, it can develop some of the algorithms created many years before and they could not be implemented due to technological limitations.

In this research, we compare the two methods of phase unwrapping: the global method and local method, using virtual objects to which a structured fringe pattern is projected, the resulting images from this projection, are analyzed by means of Fourier Profilometry, which basically consists in recover the wrapped phase signal using phase unwrapping algorithms in a computational process called fringe pattern analysis. The inverse Fourier transform, is used to carryout the phase unwrapping, which will lead us to obtain the three dimensional reconstruction of the object of study. At last, results obtained are shown.

(Key words: 3D images, profilometry, unwrapping, fringe patterns)

I. Dedicatorias y agradecimientos.

Dedicatorias:

Este trabajo es para Rosalba mi compañera y para mis hijos Alejandro y Jazmín (o Jazmín y Alejandro), con todo mi cariño...

Agradecimientos:

Agradezco a mis asesores de tesis, los doctores Carlos Pedraza y Efrén Gorrostieta, gracias por su paciencia.

A mis maestros, todos, agradezco que se preocuparan por adquirir conocimientos y después se ocuparan en compartirlos.

A mis compañeros, todos.

A la Universidad Autónoma de Querétaro y a sus autoridades por su trabajo a favor de la formación pública. Al espíritu universitario, le agradezco por ser fuente de inspiración y progreso.

I.	Dedicatorias y agradecimientos.	I
II.	Índice general	II
III	Índice de tablas	IV
IV.	Índice de figuras	V
1.	Introducción	1
1.2	Resumen.	3
1.3	Justificación.	3
1.4	Objetivos.	5
1.4.1	Objetivo general	5
1.4.2	Objetivos parciales o específicos.	5
1.5	Definición del problema.	6
1.6	Alcances y limitaciones.	8
1.6.1	Alcances.	8
1.6.2	Limitaciones.	8
2.	Antecedentes	9
2.1	Procesamiento digital de imágenes.	9
2.2	Técnicas de procesamiento de imágenes.	12
2.3	Desarrollo de fase.	16
2.4	Estado del arte en el desarrollo de fase para la reconstrucción de imágenes 3D.	39
2.5	Usos de las técnicas de desarrollo de fase.	41
3.	Algoritmos de desarrollo de fase usados en la reconstrucción de imágenes 3D.	43
3.1	Algoritmo de ruta dependiente.	46
3.2	Algoritmo de compensación residual.	48
3.3	Algoritmo de ruta guiada por la calidad.	51
4.	Implementación de los algoritmos de desarrollo de fase y resultados experimentales.	54
4.1	Análisis	54
4.1.1	Tiempo de ejecución.	55
4.1.2	Aspecto visual.	56

5.	Conclusiones y trabajo futuro.	60
5.1	Perspectivas futuras.	61
6.	Referencias bibliográficas.	63
7.	Anexos.	67
	Anexo 1 Códigos.	67
	Anexo 2 Glosario.	72

III. Índice de tablas.

Tabla 2.1	Algoritmo para el proceso de análisis de franjas y desenvolvimiento de fase.	24
Tabla 2.2	Cálculo de la segunda diferencia en una imagen.	35
Tabla 2.3	Ejemplo del desarrollo numérico del algoritmo.	37
Tabla 3.1	Algoritmo para localizar los residuos positivos o negativos en una ruta o brecha.	50
Tabla 3.2	Algoritmo de ruta guiada por la calidad para el desenvolvimiento de fase.	52
Tabla 4.1	Tiempo de ejecución de los algoritmos local y global y la diferencia de tiempo.	55

IV. Índice de figuras

Figura 1.1	Proceso de proyección del patrón de franjas.	2
Figura 2.1	Modelos de color RGB y CMY.	11
Figura 2.2	Geometría de ejes ópticos cruzados.	13
Figura 2.3	Geometría de ejes ópticos paralelos.	15
Figura 2.4	Flujo del desenvolvimiento de fase	18
Figura 2.5	Relación de fase y altura en la PTF.	19
Figura 2.6	Diferentes ordenes de difracción de un patrón fijo de franjas.	21
Figura 2.7	Función $\phi(x, y)$.	21
Figura 2.8	Secuencia de análisis de un objeto para obtener su altura.	23
Figura 2.9	Vista desde arriba y en perspectiva de un objeto que se intenta reconstruir.	25
Figura 2.10	Patrón de franjas proyectado sobre el plano de referencia y en el objeto.	26
Figura 2.11	Frecuencia espacial f_0 del patrón de franjas sobre una imagen.	26
Figura 2.12	Órdenes de difracción y señal de fase.	28
Figura 2.13	Medida en pixeles entre los órdenes de difracción y medida del primer orden.	29
Figura 2.14	Distancia de la frecuencia central a f_0 , medida en pixeles.	29
Figura 2.15	Lugar en donde se crea la ventana y la señal de fase a través de la ventana.	30
Figura 2.16	Rango en donde se encuentra f_0 igual al número de franjas medido en pixeles.	30
Figura 2.17	Fase del patrón de franjas sobre el plano y sobre el objeto.	31
Figura 2.18	Fase envuelta del patrón de franjas y del objeto.	31
Figura 2.19	El objeto original y el objeto reconstruido en perspectiva.	32
Figura 2.20	Vista frontal del objeto original y la reconstrucción.	33
Figura 2.21	Vista superior del objeto original y la reconstrucción.	33
Figura 2.22	Presentación gráfica del error.	34
Figura 2.23	Imagen original, diferencia del error y la imagen original.	34
Figura 2.24	Importancia de la técnica de desenvolvimiento de fase en diferentes aplicaciones.	42
Figura 3.1	Una posible ruta de solución al desenvolvimiento de fase.	46
Figura 3.2	Representación de residuos positivos, negativos y posibles configuraciones.	48
Figura 3.3	Una carga residual y la forma de obtener su carga.	49

1. Introducción.

El método de análisis de patrones de franjas ha sido ampliamente usado en pruebas no destructivas, no invasivas en sistemas para la reconstrucción de imágenes en tercera dimensión, algunas de sus características deseables, son su alta efectividad y velocidad de procesamiento.

Básicamente, la reconstrucción de objetos en 3D, en el método de análisis de patrones de franjas consiste en aplicar un patrón de franjas con una frecuencia espacial conocida, proyectándolo sobre una superficie plana y posteriormente, sobre el objeto del cual se quiere realizar la reconstrucción en 3D.

La escena es tomada por una cámara con sensores de luz de tipo *CCD (Coupled Charge Diode)* y las imágenes obtenidas se almacenarán en la memoria de una computadora. Estas imágenes serán procesadas digitalmente para el análisis de los patrones de franjas aplicando algún algoritmo de desdoblamiento de fase. El método más conocido y usado para el análisis de patrones de franjas, es el análisis de patrones de franjas mediante la transformada de Fourier.

En las imágenes capturadas, encontramos un patrón de franjas de la forma...

$$g(x, y) = a(x, y) + b(x, y) \cos [2\pi f_0 X + \varphi(x, y)] \quad \text{Ecuación 1.1}$$

... para la imagen del patrón de franjas proyectado sobre el objeto...

$$g_0(x, y) = a(x, y) + b(x, y) \cos [2\pi f_0 X + \varphi_0(x, y)] \quad \text{Ecuación 1.2}$$

La técnica convencional ha sido extraer la información de la fase generando un mapa del contorno de franjas sobre la distribución de la fase [Mitsuo Takeda, Hideki Ina, Seiji Kobayashi, 1982], donde $g(x, y)$ y $g_0(x, y)$ representan las intensidades de iluminación en las imágenes en el punto (x, y) . $a(x, y)$ representan la iluminación de fondo y $b(x, y)$, el contraste entre las franjas claras y oscuras proyectadas, f_0 es la frecuencia portadora

espacial, $\phi(x, y)$ y $\phi_0(x, y)$ son las fases correspondientes al patrón de franjas distorsionado.

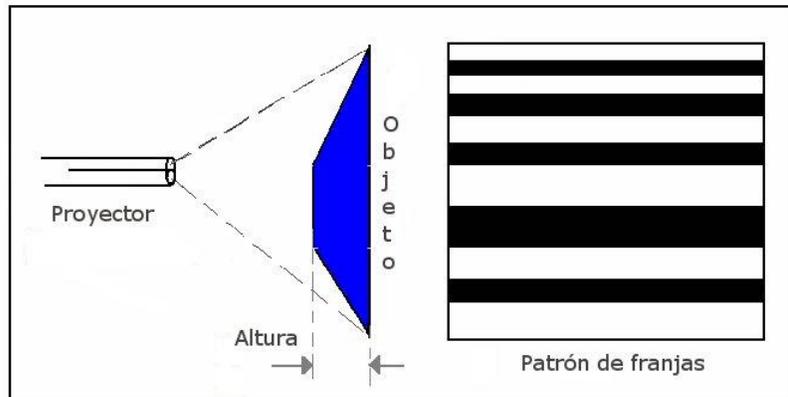


Figura 1.1. Proceso de proyección del patrón de franjas [Abdul-Rahman Hussein, 2007].

Además, es necesario puntualizar que la fase $\phi(x, y)$ contiene, la información deseada y que $a(x, y)$ y $b(x, y)$, se convierten en variaciones de irradiación no deseadas. En la mayoría de los casos $\phi(x, y)$, $a(x, y)$ y $b(x, y)$, varían ligeramente comparados con la frecuencia portadora espacial f_0 .

El ángulo $\phi(x, y)$ es la fase alzada porque la superficie del objeto termina el ángulo de proyección, lo que se expresa en estos términos.

$$\phi(x, y) = \phi_0(x, y) + \phi_z(x, y) \quad \text{Ecuación 1.3}$$

Donde $\phi_0(x, y)$ es la fase causada por el ángulo de proyección de franjas, correspondiente al plano de referencia y $\phi_z(x, y)$, es la fase causada por la distribución de la altura del objeto.

Las áreas de conocimiento donde se aplica el tratamiento de imágenes digitalizadas son muchas, entre otras, la Medicina, la Astronomía, o la Microscopía. Además existen diversos modelos para la formación de imágenes y la utilización de estos depende de lo que se desea estudiar, por ejemplo las imágenes acústicas; en donde se utilizan sonidos tanto en frecuencias bajas (usados en Geología), como de ultrasonido comúnmente usado en medicina [Pedraza J, Rodríguez W, Gorrostieta E, 2007].

Ejemplos de aplicaciones para el uso de patrones de franjas en el procesamiento de imágenes son la resonancia magnética (MRI), el radar y el sonar de apertura sintética (SAR), la óptica adaptativa, el análisis de patrones de franjas por *interferometría*, la tomografía y la espectroscopia, la síntesis de imágenes en radio astronomía y el procesamiento de imágenes sísmicas. Estos sistemas utilizan modernos y maduros algoritmos para extraer la señal de fase [Abdul-Rahman Hussein, 2007].

1.2 Resumen.

En esta tesis se comparan los dos métodos de los algoritmos de desenvolvimiento de fase: los algoritmos del método global y los algoritmos del método local. Esta comparación se realiza analizando un objeto virtual del cual se conocen su forma y sus medidas dimensionales. Para este análisis, se proyecta un patrón de franjas sobre la superficie del objeto virtual y se hace una toma fotográfica digitalizada que la muestra en dos dimensiones (*2D*), posteriormente, por medio de un procesamiento de esta imagen, se podrá reconstruir su perfil en tercera dimensión y será mostrarlo en la pantalla de una computadora, posteriormente, se comparan los resultados obtenidos con los algoritmos seleccionados.

Se observa la clasificación de los algoritmos y se documenta la comparación objetiva del desempeño de estos métodos, sus cualidades, problemas y potencialidades relativas. También, se presenta un panorama general acerca de los métodos y técnicas para la adquisición y procesamiento digital de imágenes, de los algoritmos de ruta dependiente, algoritmos de compensación residual y algoritmos de ruta guiada por la calidad, además del estado del arte del desenvolvimiento de fase para la reconstrucción de imágenes en tercera dimensión.

1.3 Justificación.

Usar patrones de franjas en el análisis de imágenes, resulta un excelente método en la reconstrucción de imágenes *3D*. Sus algoritmos son además, importantes auxiliares en el procesamiento de señales.

De entre las áreas de conocimiento anteriormente citadas, existen muchas aplicaciones del procesamiento digital de imágenes en campos como la ingeniería, la industria y las ciencias en general. Parte de su utilidad y aprovechamiento depende de la calidad de la imagen adquirida, de su correcto análisis para la adecuada extracción de su señal de fase y de los mejores resultados durante el proceso de desenvolvimiento de esa señal de fase por medio de algoritmos apropiados.

El desenvolvimiento de fase es considerado como uno de los problemas más difíciles de resolver en matemáticas e ingeniería, la gran cantidad de investigaciones en este campo, ha resultado en una todavía más grande cantidad de soluciones propuestas.

Estas soluciones están englobadas en una clasificación determinada por dos grandes grupos o métodos de desenvolvimiento de fase, en que se distinguen los algoritmos inherentes a la solución de este problema. Estos son: los algoritmos espaciales y los temporales.

Estos grupos se dividen a su vez, en dos categorías como algoritmos globales y algoritmos locales, entre los que se puede encontrar diversas propuestas de solución, adecuadas o pensadas para resolver problemas inherentes a aplicaciones específicas y definidas por necesidades particulares dentro de la visión artificial y del procesamiento de imágenes.

La relevancia de este trabajo, se encuentra en la comparación de algoritmos de cada uno de los grupos principales, con el objeto de trazar una línea para identificar a cada método por las características de sus algoritmos, los usos que puede tener y las aplicaciones en que deben ser usados para obtener los mejores resultados.

Esta comparación, es para obtener datos precisos acerca del desempeño de los algoritmos de cada uno de los grupos principales y crear un marco de referencia para futuras investigaciones. El análisis y la comparación de los distintos grupos usados en el desenvolvimiento de fase por *perfilometría*, permite determinar el mejor uso de cada algoritmo en aplicaciones específicas.

1.4 Objetivos.

Entre los objetivos de esta investigación, se encuentra, ofrecer un panorama general del proceso de reconstrucción de objetos en tercera dimensión, a partir de sus imágenes fotográficas en dos dimensiones mediante un método pasivo no invasivo ni destructivo conocido como análisis de patrones de franjas por *perfilometría* de Fourier y hacer una comparación entre algoritmos de cada uno de los grandes grupos que usan el análisis de patrones de franjas y el desenvolvimiento de fase: los algoritmos locales y los algoritmos globales.

1.4.1 Objetivo general

Se mostrará un estudio comparativo entre los algoritmos de seguimiento de ruta más usados en el desenvolvimiento de fase para la reconstrucción de imágenes 3D.

1.4.2 Objetivos parciales o específicos:

- 1) Realizar una investigación acerca de los algoritmos de desenvolvimiento de fase por seguimiento de ruta que utilizan patrones de franjas para la reconstrucción de imágenes 3D.
 - a) Técnicas de procesamiento de imágenes.
 - b) Antecedentes de la técnica de desenvolvimiento de fase en la reconstrucción de imágenes 3D.
 - c) Localizar el “estado del arte” de los algoritmos de seguimiento de ruta en la reconstrucción de imágenes 3D.
- 2) Seleccionar los métodos de entre los algoritmos de seguimiento de ruta.
- 3) Implementar cada uno de los algoritmos de seguimiento de ruta seleccionados.

- 4) Realizar una comparación del desempeño de los algoritmos seleccionados.
 - a) Comparación por su aspecto visual, Tiempo de ejecución y consumo de recursos.
 - b) Comparación estadística de los algoritmos probados.

1.5 Definición del problema.

El desenvolvimiento de fase, es el paso final y el más difícil en el proceso de la extracción de la fase de una imagen digital, su éxito o fracaso puede tener un efecto masivo en el desempeño de todo un sistema.

Muchos algoritmos de desenvolvimiento de fase se han desarrollado para la reconstrucción de objetos en *3D*, utilizando diversos métodos para resolver los problemas inherentes al desenvolvimiento de fase, pero los propuestos hasta ahora, presentan cada uno de ellos claras particularidades.

Los métodos conocidos de evaluación de franjas y recuperación de fase como la transformada de *Fourier* o la *Wavelet*, no pueden asegurar el procesamiento exitoso de los complicados patrones de franjas, debido entre otras razones, a la limitada resolución espectral cuando se procesan patrones de franjas con un número escaso de estas, lo que provoca ruido en la reconstrucción debido a los picos en de alta frecuencia que se asocian a este problema y que se acentúan usando métodos de desdoblamiento anteriores o con la inestabilidad de los cálculos en métodos posteriores [D. C. Ghiglia, L. A. Romero, 1994], [D. C. Ghiglia, M. Pritt, 1998].

Los algoritmos de desenvolvimiento de fase, están clasificados en dos grupos según la metodología que siguen para resolver los problemas inherentes al desenvolvimiento de la fase, se les conoce como locales y globales, ambos hacen el recorrido y análisis de la imagen siguiendo una ruta de forma predeterminada.

Estas clasificaciones son adecuadas y pertinentes, delimitan con claridad y exactitud los tipos de algoritmos por sus métodos y dan una idea de cómo y en donde pueden ser

utilizados. Para definir a los grupos, se han llevado a cabo gran cantidad de experimentos y se han documentado propiamente, pero hasta ahora se han realizado pocas comparaciones entre algoritmos de cada grupo que muestren sus cualidades, problemas y potencialidades relativas.

Para hacer esta comparación, es necesario realizar una prueba de desempeño en condiciones iguales, esto es: analizar la misma imagen con el mismo patrón de franjas, equipo de cómputo e iguales condiciones y factores que puedan intervenir en el desarrollo de la prueba. Una vez obtenidos los resultados de las pruebas aplicadas, se hará un análisis comparativo para distinguir el desempeño de los algoritmos. Esta comparación será dividida en cuatro aspectos:

Tiempo de ejecución. Se hará una medición del tiempo que tarda cada uno de los algoritmos seleccionados en procesar la imagen y entrega como resultado una reconstrucción en tres dimensiones del objeto real representado en la imagen con patrón de franjas. El tiempo de ejecución será comparado con el obtenido por cada algoritmo seleccionado.

Aspecto visual. Se hará una inspección visual de la reconstrucción de la imagen en tres dimensiones realizada por cada uno de los algoritmos seleccionados. Esta revisión es puramente apreciativa; es para observar y determinar de forma visual, el parecido de la reconstrucción en tres dimensiones con el objeto real y compararla con los resultados obtenidos por los demás algoritmos seleccionados.

Estimación del error. Se hará una estimación de error producido durante el proceso de análisis de franjas.

Realizadas las comparaciones, se registrarán de forma clara, entendible y consistente y se hará una clasificación de algoritmos por su desempeño, y se harán recomendaciones para su uso, planteando las perspectivas futuras de las observaciones realizadas.

1.6 Alcances y limitaciones.

Los alcances y limitaciones de este trabajo, se circunscriben dentro del tema del mismo de la siguiente manera.

1.6.1 Alcances.

Desde la descripción breve de los procesos que intervienen en la reconstrucción de un objeto tridimensional a partir de una imagen, la adquisición y el tratamiento de imágenes, el análisis de patrones de franjas, el desenvolvimiento de fase, y la reconstrucción de objetos en tercera dimensión dentro de la memoria de una computadora, hasta la comparación de los dos tipos de algoritmos de desenvolvimiento de fase, locales y globales más representativos y usados en la reconstrucción final de los objetos.

1.6.2 Limitaciones.

La variedad y extensión de los temas relacionados con el desenvolvimiento de fase y reconstrucción de objetos en tres dimensiones, que a su vez se encuentran inmersos en dos temas más grandes, que son el campo de la visión artificial y el tratamiento de imágenes, ameritan un análisis cuya profundidad no puede ser abordada en este trabajo.

2. Antecedentes.

El procesamiento digital de imágenes aparece recientemente en la historia de la computación, debido a que para digitalizar una imagen, primero se debía desarrollar la tecnología computacional que cubriera los requisitos necesarios para la tarea [Paz Gutiérrez, Jenaro Carlos, 2006].

Esta capacidad se presentó de forma gradual con el avance de la tecnología, en el *hardware* característico de las computadoras, como procesadores, cantidad de memoria, tarjetas y memoria de video, dispositivos para la adquisición de las imágenes, su digitalización por medio de escáneres o tarjetas de captura, cámaras de video y fotográficas, hasta su almacenamiento en memorias y discos duros, además del *software* para su interpretación, representación y manipulación en sistemas operativos, programas, protocolos, formatos y otros.

Las técnicas para el procesamiento y edición digital de imágenes, han alcanzado un alto grado de sofisticación, el cual es transparente para la mayoría de los usuarios, aficionados o expertos en *software* especializado en la creación, manipulación, modificación, edición y retoque de imágenes digitales.

Las más avanzadas de estas técnicas se manifiestan en algoritmos. Las herramientas usadas en el tratamiento y procesamiento digital de imágenes son matemáticas.

2.1 Procesamiento digital de imágenes.

Para realizar el procesamiento de una imagen, es necesario que al adquirir una, esta sea digitalizada, es decir, debe ser convertida a un lenguaje que los algoritmos usados en su manipulación puedan procesar.

La adquisición de una imagen digital, involucra a un procedimiento de manipulación de la luz o de alguna otra forma de radiación que es emitida o reflejada por los objetos que se pretende capturar. Es necesario el uso de equipos sensibles a la luz para su captura y de equipo de cómputo para su almacenamiento y procesamiento en formatos digitales

nativos o digitalizadas. Una vez adquirida la imagen, puede ser tratada o modificada con fines de estudio técnico y científico.

Una imagen digital es una función $f(x, y)$, que ha sido *discretizada* en coordenadas espaciales y en la intensidad de su luminosidad [González, R. C, Woods, P, 2002]. Puede ser considerada como una matriz cuyos índices de renglón y columna, identifican un punto (un lugar en el espacio bidimensional en los ejes X , y Y de la superficie de la imagen) y cada uno se relaciona con un valor del nivel de profundidad de color en ese punto. Los elementos de estos arreglos digitales son llamados elementos de imagen o píxeles (*pixels*). A esta función se le conoce comúnmente como mapa de *bits*.

Para representar una imagen digitalmente con la mayor gama de colores posible, se necesitan del orden de ocho *bits* para cada uno de los colores usados en fotografía digital, según el modelo de color RGB o *Red, Green and Blue* (rojo, verde y azul), de 0 a 255 tonalidades para cada color (256 incluyendo el cero), que al ser combinadas, de la manera que se hace al mezclar colores, da una gama con 16, 777,216 posibles colores para cada punto. A este método se le conoce como profundidad de color, en este caso, la profundidad de color es de 24 *bits*.

Matemáticamente puede interpretarse a un color como un vector en el espacio tridimensional que describen el Rojo, Verde y Azul. Aparte de RGB [Paz Gutiérrez, Jenaro Carlos, 2006]. Existen otros formatos para describir la gama de colores en la pantalla de un monitor de computadora o del televisor, como el espacio o formato de color CMYK (Cian, Magenta Amarillo y Negro), los modelos HLS de Tinte, Luminancia y Saturación (*Hue, Lightness, Saturation*) y HSV para Tinte Saturación y Valor (*hue, saturation, value*), el modelo $Y'UV$ de Luminancia y *Crominancia*, el modelo YCC de color por componentes (Y, C1,C2), o el CIE XYZ y CIE $L^*a^*b^*$ (CIELAB), modelo cromático usado normalmente para describir todos los colores que puede percibir el ojo humano, y otros modelos más.

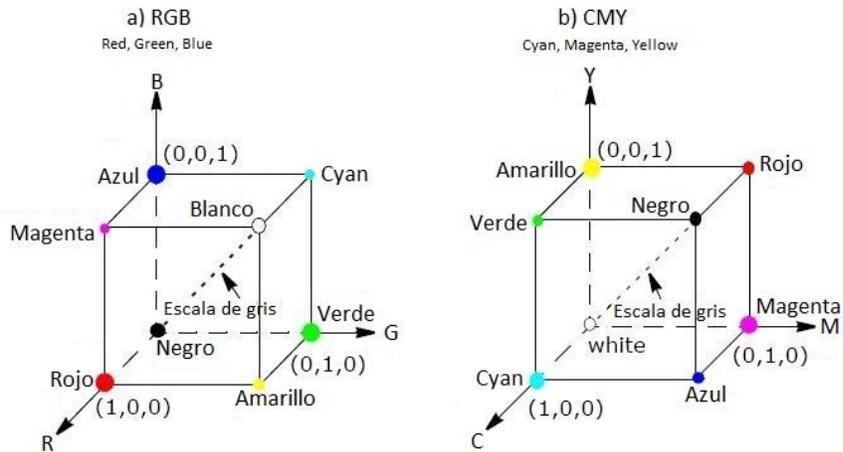


Figura 2.1 Modelos de color a) RGB y b) CMY

Una imagen monocromática se representa con la gama de un solo color, particularmente los tonos de gris que se obtiene tomando valores iguales de cada color básico para toda la escala de gris. De acuerdo al criterio anterior, esta representación se refiere a una función de intensidad de luz en un punto de sus dos dimensiones $f(x, y)$, donde “ x ” y “ y ” indican coordenadas espaciales y el valor de f en cualquier punto (x, y) , es proporcional a la luminosidad o nivel del color en tonos de gris, desde el negro puro al blanco puro (en bits, de 0 a 255), de la imagen en cada punto (*pixel*).

Son además, muchos los formatos con que se define a las imágenes capturadas, los más comunes son BMP o mapa de *bits* (*Bitmap*), TIFF (*Tagged Image File Format*), GIF (*Graphic Interchange Format*), IFF (*Interchange File Format*), JPG (*Joint photographic Group*), PNG (*Portable Network Graphics*), PSD (*Photo Shop Document*).

Estos formatos son interpretados por diferentes programas de edición de imágenes en la memoria de la computadora, y desplegados en la pantalla del monitor. Cuando se tiene la imagen deseada en estas circunstancias, es posible realizar su procesamiento digital. La técnica más utilizada para ello, es la implementación de un filtrado sobre la imagen en el dominio de la frecuencia, técnica derivada del procesamiento digital de las señales.

En la mayoría de los casos, el procesamiento de imágenes por procesamiento de señales, es monocromático y bidimensional (en $2D$) [Esteve Taboada, Juan, 2002]. El contenido de la información que se puede almacenar o transmitir en una imagen está limitado por la extensión temporal de la longitud de onda, el producto espacio ancho de banda que está relacionado con el número máximo de puntos independientes que se puede registrar en una imagen de tamaño y resolución finita.

El propósito de este procesamiento, es la obtención de la información de la fase oculta en la imagen, de forma eficiente y automática, [Pedraza Ortega J.C., Canchola Magdaleno S.L., Gorrostieta Hurtado E], esta información nos permitirá la reconstrucción de una imagen en tercera dimensión a partir de una imagen en dos dimensiones.

2.2 Técnicas de procesamiento de imágenes.

El procesamiento de imágenes, engloba a cualquier tipo de manipulación que se pueda hacer sobre una imagen para modificarla, en la actualidad existen muchas aplicaciones de *software* que permiten el procesamiento digital de imágenes, mucho de éste, utiliza técnicas o algoritmos que son bien conocidos por la comunidad que trabaja con imágenes.

El tratamiento digital de imágenes pertenece más al campo de la electrónica en donde su finalidad además de la manipulación y modificación, es extraer toda la información que pueda ser útil para un propósito específico. Las técnicas de procesamiento de imágenes pueden ser puramente ópticas, puramente digitales y con una combinación de ambas pueden ser híbridas [Esteve Taboada, Juan, 2002].

En la *perfilometría* por medio de la transformada de Fourier, la imagen del objeto con el patrón de franjas proyectadas en su superficie, es puesta directamente en la memoria de una computadora para ser procesada. Se han propuesto dos diferentes geometrías ópticas: La geometría de ejes ópticos cruzados y la geometría de ejes ópticos paralelos [Takeda Mitsuo., Mutoh Kazuhiro, 1983]. En la geometría de ejes ópticos cruzados, los ejes ópticos de un proyector y una cámara permanecen en el mismo plano y ambas intersecan un punto cerca del centro del objeto. En la geometría de ejes ópticos

paralelos, los ejes ópticos de un proyector y una cámara permanecen en el mismo plano y son paralelos, no se cruzan. Estas dos opciones de geometría óptica, pueden ser usadas en la *perfilometría* por medio de la transformada de Fourier.

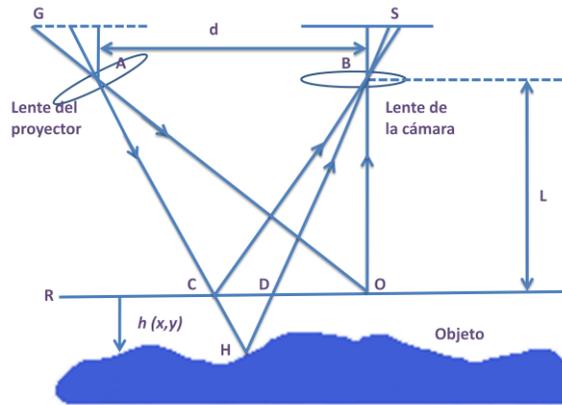


Figura 2.2 Geometría de ejes ópticos cruzados [Esteve Taboada, Juan, 2002]

En la figura se muestra una geometría de ejes ópticos cruzados, el eje óptico de los lentes del proyector A , cruzan el otro eje óptico de los lentes de la cámara B justo en el punto O en un plano de referencia denominado R , el cual es un plano ficticio normal a B y a S como se muestra en la relación L , que servirá como referencia desde la cual se mide la altura $h(x, y)$ del objeto. El mallaado G , tiene su línea normal al plano de la figura y su imagen conjugada está formada por los lentes del proyector a través del punto O ; A denota el centro de la pupila de los lentes del proyector. Los lentes de la cámara, con los centros de la pupila en B , referencia a la imagen al plano sensor S . A y B están localizados a la misma distancia l_0 del plano R [Takeda Mitsuo., Mutoh Kazuhiro, 1983].

Cuando el objeto es un plano delgado y uniforme en R , por ejemplo $h(x, y)=0$, y si A está en el infinito, la imagen del enrejado proyectada en la superficie del objeto y observada a través del punto C , es un patrón de enrejado de franjas regular, el cual puede ser expresado por una expansión de la serie de Fourier:

$$g_T(X, Y) = \sum_{n=-\infty}^{\infty} A_n \exp(2\pi i n f_0 x) \quad \text{Ecuación 2.1}$$

...en donde...

$$f_0 \frac{1}{p_0} = \frac{\cos\theta}{p}$$

Ecuación 2.2

... es la frecuencia fundamental de la imagen de franjas observada.

En el eje del plano en R , Si A está a una distancia finita, en el plano del sensor de la imagen, se formará un patrón de franjas deformado incrementada en su intensidad, incluso para $h(x, y)=0$. Un rayo principal a través de un punto de la imagen conjugada A , golpea el plano de referencia R en el punto B , describimos la imagen de franjas deformada para $h(x, y)=0$ como:

$$g_0(x, y) = \sum_{n=-\infty}^{\infty} A_n \exp\{2\pi i n f_0 [x + s_0(x)]\}$$

Ecuación 2.3

En donde, $S_0(x) = \overline{BC}$ es una función de X . Expresamos la ecuación como una señal espacialmente modulada en su fase...

$$g_0(x, y) = \sum_{n=-\infty}^{\infty} A_n \exp\{i[2\pi n f_0 x + n\phi_0(x)]\}$$

Ecuación 2.4

... en donde...

$$\phi_0(x) = 2\pi f_0 s_0(x) = 2\pi f_0 \overline{BC}$$

Ecuación 2.5

... en esta geometría, la imagen de la malla esta deformada y modulada en su fase incluso para $h(x, y)=0$, Esto es una gran restricción en la aplicación de la geometría de ejes ópticos cruzados [Takeda Mitsuo., Mutoh Kazuhiro, 1983].

En un objeto cualquiera con $h(x, y)$ variable, el rayo principal de A golpea la superficie del objeto en el punto H , el punto H es también el punto D en el plano R cuando es observado a través de B , por lo tanto, la imagen de mallado deformada para un objeto cualquiera es dada por ...

$$g_0(x, y) = r(x, y) \cdot \sum_{n=-\infty}^{\infty} A_n e^{2\pi i n f_0 [x+s(x,y)]}$$

Ecuación 2.6

... 0...

$$g_0(x, y) = r(x, y) \cdot \sum_{n=-\infty}^{\infty} A_n e^{i2\pi n f_0 [x+n\phi(x,y)]} \quad \text{Ecuación 2.7}$$

... en donde...

$$\varphi(x, y) = 2\pi f_0 s(x, y) = 2\pi f_0 \overline{BD} \quad \text{Ecuación 2.8}$$

... y, $r(x, y)$ es una distribución no uniforme de reflectividad en la superficie del objeto.

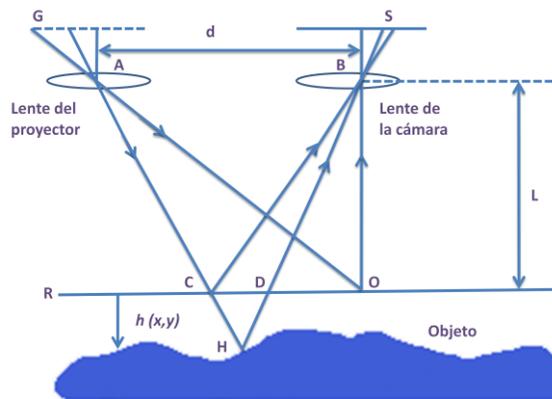


Figura 2.3 Geometría de ejes ópticos paralelos [Esteve Taboada, Juan, 2002].

En la geometría de ejes ópticos paralelos que muestra la imagen, el eje óptico A de los lentes del proyector y los lentes de la cámara B son paralelos y normales al plano de referencia R . La imagen conjugada del enrejado G se forma en el plano R , entonces las ecuaciones (2.5) y (2.8) se convierten en...

$$\varphi_0(x) = 2\pi f_0 s_0(x) = 2\pi f_0 \overline{BC} = 0 \quad \text{Ecuación 2.9}$$

$$\varphi(x, y) = 2\pi f_0 s(x, y) = 2\pi f_0 \overline{CD} \quad \text{Ecuación 2.10}$$

Por lo que la imagen del mallado o patrón de franjas proyectado incluso para $h(x, y)=0$, sea regular sin importar la posición del proyector.

2.3 Desenvolvimiento de fase.

El desenvolvimiento de fase puede ser simplemente definido como el proceso para resolver el problema de ambigüedad causado por el hecho de que la fase absoluta está típicamente envuelta en el intervalo de π y $-\pi$. [Abdul-Rahman Hussein, 2007]. El desenvolvimiento de fase, es la técnica usada para remover las discontinuidades de 2π , que están embebidas dentro del mapa de fase usando algoritmos de desenvolvimiento de fase en un proceso computacional, donde una superficie ϕ , es reconstruida de lo que llamamos su forma envuelta ψ . En ausencia de ruido, la función $\psi(x)$, es igual a $\phi(x) + 2\pi k(x)$, en donde $k(x)$ es una función entera tal que $-\pi < \psi \leq \pi$

Sin embargo, los valores reales de fase no pueden ser extraídos directamente de la señal física, porque estos algoritmos de fase modifican la señal, dejando sus valores en un rango de $\pm \pi$ radianes. Lo más que estos métodos pueden obtener, es la fase envuelta.

Si la fase está limitada a valores de π y $-\pi$, al exceder estos valores (\pm), “saltaría” de π , a $-\pi$, y viceversa. Por cada salto negativo de 2π , sería necesario agrega π , y por cada salto positivo, habría que restar 2π .

$$\begin{array}{l} \text{Si... } x = [\pi \quad -2.7925 \quad -2.4434 \quad -2.0943] \\ \dots \text{ entonces... } Y(x) = [\pi \quad 3.4909 \quad 3.8397 \quad 4.1887]. \end{array}$$

... en donde $Y(x)$ es la función de desenvolvimiento de fase.

Este proceso intenta dar una estimación de la función de fase real (ϕ), a partir de una función (ψ) en donde la relación $\phi - \psi$ no es lineal, además, las aplicaciones *interferométricas* o las de triangulación, procesan la información tomada del mundo real, por lo que es inevitable el ruido en la fase que está siendo medida, lo que hace difícil y complicado el procedimiento.

Otro problema es el reconocimiento de las envolturas genuinas o verdaderas que son producidas por la función arco tangente y los saltos en la fase, causados por las discontinuidades del objeto.

El algoritmo básico de desenvolvimiento de fase, que es en una dimensión ($1D$), inicia con el primer punto de la línea que se compara con el segundo, si la diferencia (salto), es mayor o igual a 2π , se suma o resta 2π a todos los puntos a partir del segundo, dependiendo del signo, también llamado “carga” en la primera diferencia y se repite hasta terminar con esa línea. Si no encuentra diferencias, continúa comparando hasta encontrar una diferencia y realiza el desenvolvimiento.

Desgraciadamente, el algoritmo de desenvolvimiento de fase en $1D$, no es suficiente o ni siquiera correcto si se trabaja por ejemplo, con mapas de fase, por lo que se hace necesario encontrar un método o algoritmo que nos permita trabajar en dos dimensiones ($2D$), lo que complica todo el proceso si lo que deseamos desenvolver viene en dos dimensiones y además agregamos a toda esta complejidad, el ruido en la información de la imagen y la calidad de la misma.

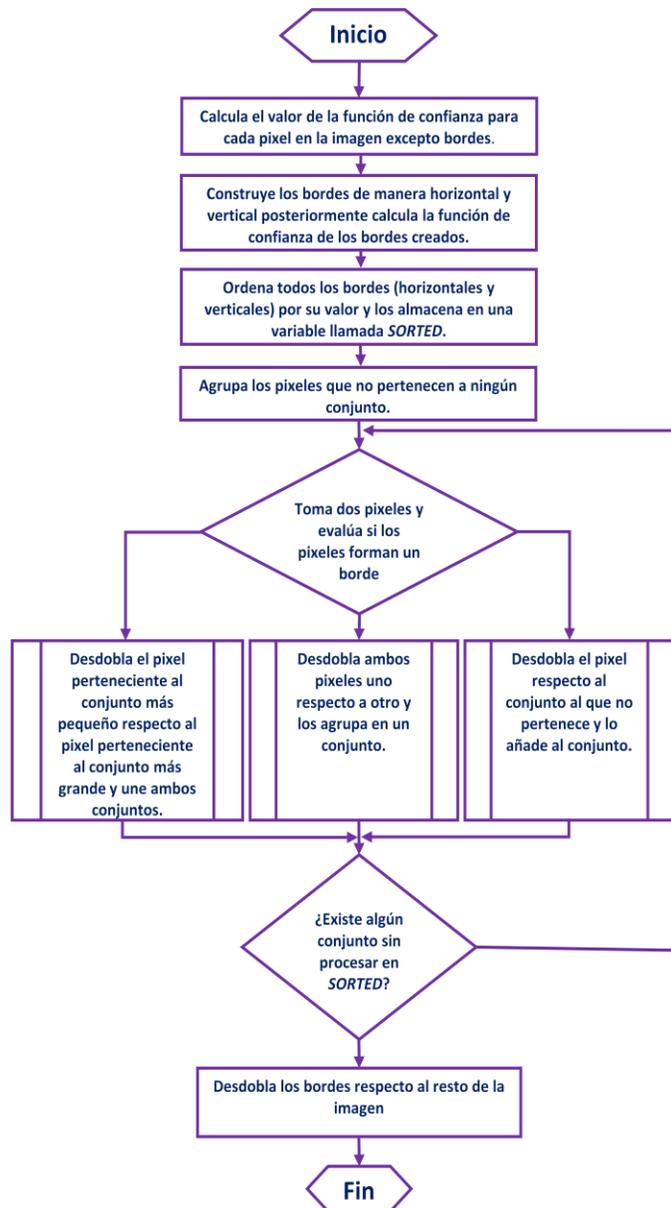


Figura 2.4 Flujo del desenvolvimiento de fase

La transformada de Fourier, es uno de los métodos más populares en el análisis de patrones de franjas aplicados sobre la imagen tratada, en donde se debe extraer de la forma más adecuada la primera armónica de un espectro de Fourier. El método de la transformada de Fourier opera globalmente con poca localización espacial. La primera armónica es a menudo enturbiada por el ruido local producido por los puntos discontinuos en la imagen.

El método de *Perfilometría* de la transformada de Fourier (FTP), es uno de los algoritmos más importantes dentro del conjunto de técnicas que existen para la reconstrucción de imágenes en 3D. Este método pasivo basado en la medición de la fase, fue desarrollado a principios de los años 80 del siglo pasado por Takeda Mitsuo y Mutoh Kazuhiro. Existen variantes al método propuesto por Takeda, los cuales involucran el gradiente del mapa de fase y desarrollan variables múltiples [Pedraza J, Rodríguez W, Gorrostieta E.].

Se determina un plano ficticio R que sirve de referencia para medir la altura $h(x, y)$ del objeto. Las líneas de la red G son perpendiculares al plano de la figura y la lente del proyector forma su imagen conjugada en el plano R . La lente de la cámara forma una imagen de este plano de referencia en el plano del sensor S .

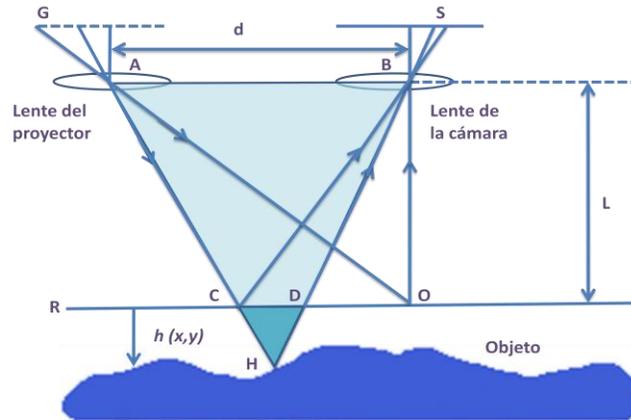


Figura 2.5 Relación de fase y altura en la PTF [Esteve Taboada, Juan, 2002]

Cuando el objeto es uniforme, o sea $h(x, y) = 0$, la red proyectada sobre el objeto grabada por la cámara es un patrón de franjas sin distorsión, que se expresa en serie de Fourier...

$$g_0(x, y) = \sum_{n=-\infty}^{\infty} A_n e^{i2\pi f_0 x} \quad \text{Ecuación 2.11}$$

f_0 es la frecuencia fundamental del patrón de franjas. El eje X es la línea que representa el plano R en la figura 2.3, el eje Y es perpendicular a éste. Para un objeto con variaciones de altura descritas por la función $h(x, y)$, el patrón de franjas estará

distorsionado y contendrá la información 3D del objeto. El patrón de franjas sobre en el plano de fondo del objeto será regular (R). Este patrón de franjas se expresa como

$$g(x, y) = r(x, y) \sum_{n=-\infty}^{\infty} A_n e^{[i2\pi n f_0(x+s(x,y))]} \quad \text{Ecuación 2.12}$$

Se entiende que en los dos patrones que se reflejan, $r(x,y)$ es la distribución de la reflectividad de la superficie del objeto (distorsionada), $r(x,y)$ es cero fuera del objeto (sin distorsión) y $s(x, y)=CD$ en la figura 3.3.

La ecuación 2 se puede expresar como...

$$g(x, y) = r(x, y) \sum_{n=-\infty}^{\infty} A_n e^{[i2\pi n f_0(x+s\phi(x,y))]} \quad \text{Ecuación 2.13}$$

El patrón de franjas distorsionado puede ser interpretado en términos de su orden de difracción, con frecuencia espacial nf_0 , modulados en fase a través de $n \phi(x,y)$ y con una modulación en amplitud dada por $Anr(x, y)$.

Esto es...

$$g(x, y) = r(x, y) \sum_{n=-\infty}^{\infty} q_n(x,y) e^{[i2\pi n f_0 x]} \quad \text{Ecuación 2.14}$$

Utilizando un algoritmo digital, el siguiente paso es calcular la transformada de Fourier unidimensional para la variable x , dejando a la variable y fija

$$G(u, y) = \int_{-\infty}^{\infty} g(x, y) e^{-i2\pi u x} dx = F_{1-D}[r(x, y)] \sum_{n=-\infty}^{\infty} Q_n(u - n f_0, y) \quad \text{Ecuación 2.15}$$

$G(u, y)$ y $Q_n(u, y)$ son las transformadas de Fourier unidimensionales de $g(x, y)$ y $q_n(x, y)$ respectivamente, calculadas con respecto a la variable x , y tratando a la variable y como un parámetro fijo. $r(x, y)$ y $\phi(x, y)$ varían muy lentamente en comparación con la frecuencia f_0 del patrón de franjas. Todos los términos $Q_n(u - n f_0, y)$ están separados unos de otros por la frecuencia portadora f_0 , tal y como se muestra en la figura 3.4.

Seleccionando el término con $n=1$, y calculando la transformada de Fourier inversa se obtiene la función compleja

$$g'(x, y) = r(x, y)q_1(x, y)e^{i2\pi f_0 x} = A_1 r(x, y)e^{i(2\pi f_0 x + \phi(x, y))} \quad \text{Ecuación 2.16}$$

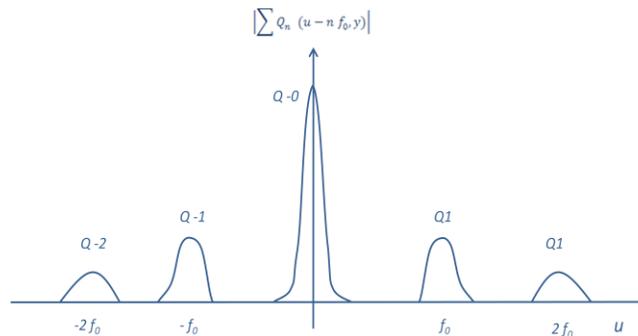


Figura 2.6 Diferentes órdenes de difracción de un patrón fijo de franjas distorsionado para un valor de y fijo [Esteve Taboada, Juan, 2002].

En la figura 2.5, la frecuencia f_0 de cualquiera de los lados del patrón de franjas distorsionado ($Q1$ o $Q-1$), es la que nos interesa. Aislando la fase de esta función compleja y eliminando el factor de fase lineal que viene dado por la función $(2\pi f_0 x)$, podemos obtener a través de la función $\phi(x, y)$ la información 3D del objeto considerado. Para la eliminación del factor de fase lineal podemos considerar el origen de coordenadas sobre el primer orden de difracción al hacer la transformada de Fourier inversa.

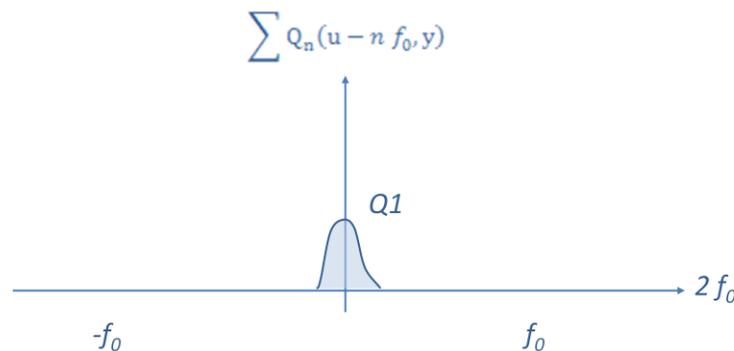


Figura 2.7 Función $\phi(x, y)$, aislando la fase eliminamos el factor de fase lineal y centrando $Q1$ para encontrar la información de la altura $h(x, y)$ [Esteve Taboada, Juan, 2002].

$$g''(x, y) = A_1 r(x, y) e^{[i\phi(x, y)]}$$

Ecuación 2.17

Debido a que la fase $\phi(x, y)$ contiene la información de la altura $h(x, y)$ del objeto, se puede considerar que se ha codificado el objeto 3D en la función compleja que será dada en la ecuación 2.19.

El cálculo de la función de fase $\phi(x, y)$, se realiza digitalmente, los valores principales de una fase calculada digitalmente estén dentro del intervalo $[-\pi, \pi]$, la función de fase $\phi(x, y)$ resultante está limitada dentro de este rango, por lo que presenta discontinuidades con saltos de 2π para variaciones mayores de 2π . Estas discontinuidades se pueden corregir sumando o restando 2π dependiendo de si el salto de fase va de π a $-\pi$, o al revés.

En la función de fase $\phi(x, y)$ obtenida, se puede encontrar la distribución de altura $h(x, y)$ del objeto 3D. Tomando en cuenta la equivalencia entre los triángulos ΔAHB y ΔCHD de la figura 2.3, se puede decir que...

$$CD = \frac{-dh(x, y)}{L-h(x, y)}$$

Ecuación 2.18

Y, por lo tanto, tenemos...

$$\phi(x, y) = \frac{-2\pi f_0 dh(x, y)}{L-h(x, y)}$$

Ecuación 2.19

...donde se puede ver que la función de fase $\phi(x, y)$ contiene la información 3D del objeto que se está considerando. La tarea ahora es extraer la fase ϕ de la intensidad de distribución en el patrón de franjas.

Ahora la tarea es extraer la fase ϕ de la intensidad de distribución del patrón de franjas usando técnicas de análisis de franjas. Estas técnicas producen lo que se conoce como desenvolvimiento de fase ψ , en lugar de la fase ϕ . Consecuentemente, se requieren algoritmos de desenvolvimiento de fase para reducir la fase ϕ desde la fase envuelta ψ ,

finalmente, la diferencia calculada de ϕ será convertida en la altura del modelo [M. Arevalillo Herráez, D. R. Burton, M. J. Lalor and M. A. Gdeisat, 2002].

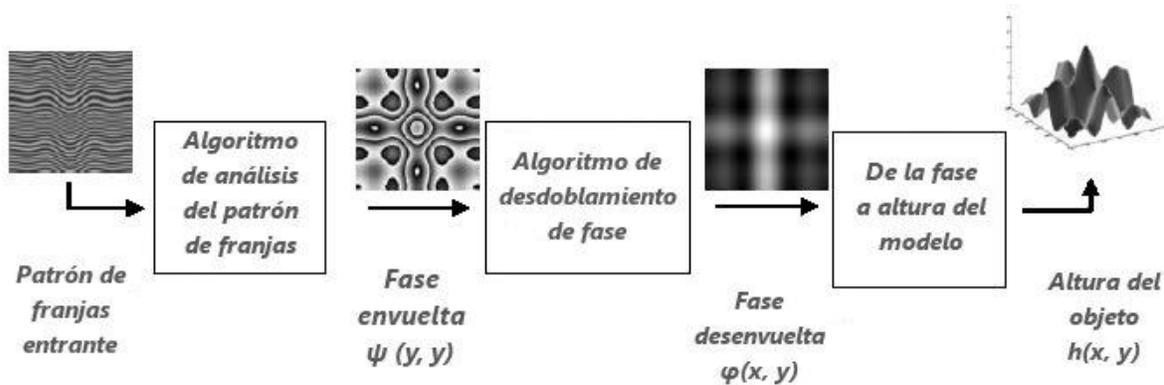


Figura 2.8 secuencia de análisis de un objeto para obtener su altura [Abdul-Rahman Hussein 2007].

En la secuencia de la figura 2.8, se muestra de manera sintética desde la adquisición de la imagen, hasta obtener la altura del objeto. Técnicas similares, donde la información está codificada en la fase envuelta, requieren el uso de algoritmos de desdoblamiento de fase.

Esta es una de las áreas más activas de investigación en el procesamiento digital de imágenes en los últimos veinte años. A pesar de ello, las modernas técnicas de desdoblamiento de fase producen solo estimaciones de la solución correcta. En ocasiones, los complejos algoritmos no pueden resolver todas las envolturas de fase en una imagen y algunos saltos de fase 2π persisten en el mapa de fase ya desdoblado en la mayoría de las imágenes tomadas de aplicaciones prácticas.

Para profundizar el esquema de la figura 2.7, es preciso describir paso a paso, el algoritmo de desdoblamiento de fase y mostrar los resultados obtenidos en cada uno de los pasos del procedimiento, de modo que nos permita observar con mayor precisión, lo que sucede en cada una de las etapas que recorre una imagen desde su adquisición, hasta la reconstrucción del objeto en tres dimensiones la presentación gráfica del modelo 3D.

1. Obtención de una imagen con patrón de franjas.
Se usa una imagen digital de 512x512 pixeles y 32 franjas como patrón.
 - 1.1 Lee la imagen **(a=imread)**;
Convierte a escala de gris **(gray(256))**;
Y a doble precisión **a=im2double(a)**;
2. Se abre una ventana.
 - 2.2 Obtiene el tamaño de la imagen **[ax ay]=size(a)**;
 - 2.3 Llena con "ceros" desde la línea 1, hasta la 512 de forma horizontal para eliminar toda la frecuencia **fondo=zeros(ax, ay)**;
3. Transformada rápida de Fourier
 - 3.1 En la imagen **fft_al=fft(a(1,:))**;
 - 3.2 Y en el fondo **fft_fondo=fft(fondo(1,:))**;
 - 3.1.1 Genera la fase del patrón de franjas **mesh(fftshift(abs(fft_al)))**;
4. Abre la ventana
 - 4.1 Obtiene el tamaño de la imagen **[ax ay]=size(a)**;
 - 4.2 Llena de ceros la imagen obtenida **vent_unos=zeros(ax, ay)**;
 - 4.2.1 Abre la ventana en la parte de f_0 **vent_unos(:, 192:256)=1**;
 - 4.2.2 Se filtra la señal que corresponde a f_0 .
 - 4.2.3 En la imagen **sfiltrada=(fftshift(fft_al).*vent_unos)**;
 - 4.2.4 En el fondo **sfiltradaFondo=(fftshift(fft_fondo).*vent_unos)**;
5. Genera la fase del patrón de franjas
 - 5.1 Sobre el objeto **sfiltrada=(fftshift(fft_al).*vent_unos)**;
 - 5.1.1 Obtiene la fase envuelta del objeto **mesh((abs(sfiltrada)))**;
 - 5.1.2 Sobre el plano **sfiltradaFondo=(fftshift(fft_fondo).*vent_unos)**;
6. Obtiene la fase envuelta del fondo **mesh((abs(sfiltradaFondo)))**;
7. Genera la fase envuelta
 - 7.1 Sobre del objeto filtra la señal **f2(i,:)=(ifft(sfiltrada(i,:)))**;
 - 7.2 Sobre del patrón filtra la señal **f3(i,:)=(ifft(sfiltradaFondo(i,:)))**;
8. Desdoblamiento de fase **fin2=unwrap(unwrap(angle dif'))**;

Tabla 2.1 Algoritmo para el proceso de análisis de franjas y desenvolvimiento de fase.

Como podemos observar, la reconstrucción de objetos en su forma 3D a partir de imágenes digitalizadas en el método de análisis de patrones de franjas, comienza con una imagen digitalizada de una superficie plana y el objeto sobre los cuales se proyecta un patrón de franjas con una frecuencia espacial conocida, por medio de un arreglo que involucra a una superficie plana, una cámara, el objeto de estudio, un proyector y la proyección de un patrón de franjas, además de iluminación adecuada.

Para adquirir una imagen que nos sea útil, se crea una escena con el objeto puesto sobre una superficie plana se proyecta un patrón de franjas sobre el objeto y la superficie. Se toma una imagen de esta escena con la cámara. La imagen digitalizada del objeto con el patrón de franjas proyectado, es almacenada en la memoria de una computadora en la que se realizará el proceso de reconstrucción del objeto.

Como ejemplo, en las figuras siguientes podemos ver un patrón de 32 franjas proyectado sobre un plano de referencia que forma una imagen de 512 x 512 píxeles, después, el mismo patrón es proyectado sobre el objeto en el plano de referencia del mismo tamaño.

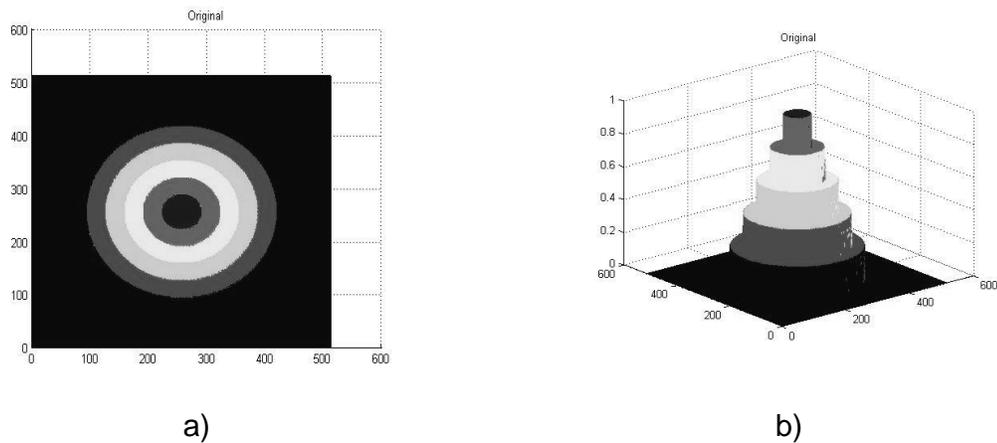


Figura 2.9 En la imagen a) pirámide circular de 5 niveles vista desde arriba. En la imagen b) Pirámide circular de 5 niveles vista en perspectiva.

De manera muy discreta, en la imagen capturada y digitalizada se observa perfilado el objeto de interés entre el enrejado del patrón de franjas, que es la pirámide circular de cinco niveles de la figura 1.

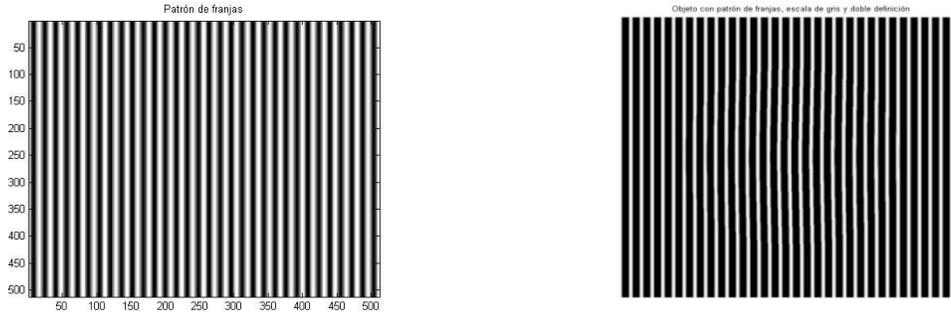


Figura 2.10 En la derecha, el patrón de franjas proyectado sobre el plano de referencia y en la izquierda, proyectado sobre el objeto puesto sobre el plano de referencia.

Cuando se tiene la imagen deseada en estas circunstancias, es posible realizar su procesamiento digital. La técnica más utilizada para ello, es la implementación de un filtrado sobre la imagen en el dominio de la frecuencia, técnica derivada del procesamiento digital de las señales.

Como sabemos, la imagen digitalizada del patrón de franjas, se representa de la forma como se muestran en las ecuaciones 1.1 y 1.2:

La información de la fase genera un mapa del contorno de franjas sobre la distribución de la fase, donde $g(x, y)$ y $g_0(x, y)$ representan las intensidades de las imágenes en el punto (x, y) . $a(x, y)$ representan la iluminación de fondo y $b(x, y)$, el contraste entre las franjas claras y oscuras proyectadas, f_0 es la frecuencia portadora espacial, y $\phi(x, y)$ y $\phi_0(x, y)$ son las fases correspondientes al patrón de franjas distorsionado.

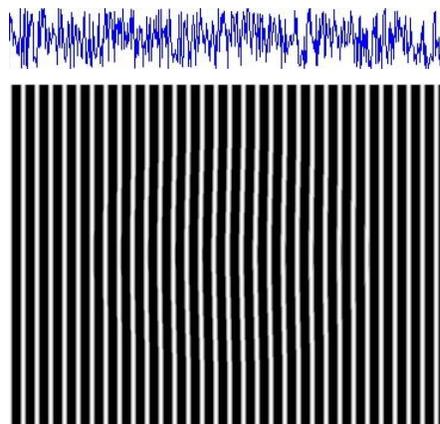


Figura 2.11 Frecuencia espacial f_0 del patrón de franjas sobre una imagen.

La frecuencia espacial es provista por el patrón de franjas proyectado sobre una superficie, esta frecuencia espacial puede variar debido al número de franjas proyectadas por unidad de área, o sea, sobre la superficie y el objeto de interés. El resultado es un patrón sinusoidal cuya calidad depende directamente de la resolución de la imagen capturada, de esta manera aplicando la transformada rápida de Fourier podemos obtener el mapa de la fase, generando un mapa del contorno de franjas sobre la distribución de la fase.

El método para el análisis de patrones de franjas más usado es el análisis de patrones de franjas mediante la transformada de Fourier, en el que se intenta extraer la primera señal armónica dentro del espectro de Fourier obtenido de una imagen digitalizada a partir de la medición de la fase [Takeda Mitsuo., Mutoh Kazuhiro, 1983], a este método se le conoce también como "*perfilometría* por medio de la transformada de Fourier".

Considerando que la fase absoluta se encuentra envuelta dentro del mapa de fase, entre las discontinuidades producidas por el ruido o interferencia provocados por las frecuencias altas, es necesario recuperar la fase absoluta utilizando lo que conocemos como un "algoritmo de desenvolvimiento de fase" que nos permita, en un proceso computacional, eliminar estas discontinuidades para reconstruir la superficie del objeto de estudio en sus tres dimensiones conservando sus características de forma y volumen.

Un algoritmo de desenvolvimiento de fase simple, puede generar una gran cantidad de errores porque la fase envuelta en el patrón de franjas, tiene estas discontinuidades que típicamente se presentan como saltos de 2π en la frecuencia. Una imagen con estas características no puede ser utilizada hasta que no se retiran estas discontinuidades. Al proceso de retirar las discontinuidades de 2π se le denomina desenvolvimiento de fase. Cuando el número de franjas es muy pequeño, las frecuencias obtenidas por medio de la transformada de Fourier ($f_0, 2f_0, 3f_0$, etc.) se pueden traslapar debido a que el número de franjas es proporcional a la distancia de separación de las frecuencias. Por lo anterior, se propone utilizar para una imagen de 512 x 512 pixeles, un número de 32 franjas, como se mencionó anteriormente.

Entonces, la fase $\phi(x, y)$ contiene la información de la altura del objeto y $a(x, y)$ y $b(x, y)$ son variaciones de radiación no deseadas. $\phi(x, y)$ es la fase producida por el ángulo de proyección correspondiente al plano de referencia durante la captura de la imagen. Debido a que la fase $\phi(x, y)$ contiene la información de la altura $h(x, y)$ del objeto, el cálculo de la función de fase $\phi(x, y)$, se realiza digitalmente, los valores principales de una fase calculada digitalmente estén dentro del intervalo $[-\pi, \pi]$, la función de fase $\phi(x, y)$ resultante, está limitada dentro de este rango, por lo que presenta las discontinuidades antes mencionadas con saltos de 2π para variaciones mayores de 2π . Estas discontinuidades se pueden corregir sumando o restando 2π dependiendo de si el salto de fase va de π a $-\pi$ o en sentido inverso.

El desenvolvimiento de fase es el último y más difícil paso en el proceso de la extracción de la fase de una imagen digital, consiste en encontrar discontinuidades de magnitud cercana a 2π , y eliminarlas agregando o quitando 2π de acuerdo al cambio de signo en la fase. Este proceso puede ser una forma de resolver el problema de ambigüedad causado por los intervalos en que se encuentra envuelta la fase de la frecuencia fundamental.

En el análisis de patrones de franjas por *perfilometría* de la transformada de Fourier, el patrón de franjas distorsionado puede ser interpretado en términos de su orden de difracción, con frecuencia espacial nf_0 , en la fase a través de $n\phi(x, y)$ y con una modulación en amplitud. Después de generar la señal de la modulación de fase aplicando la transformada de Fourier, podemos observar que obtuvimos los órdenes de difracción en las frecuencias obtenidas. Estos órdenes se presentan como $-3f_0, -2f_0, -f_0, 0, f_0, 2f_0, 3f_0$, etc.

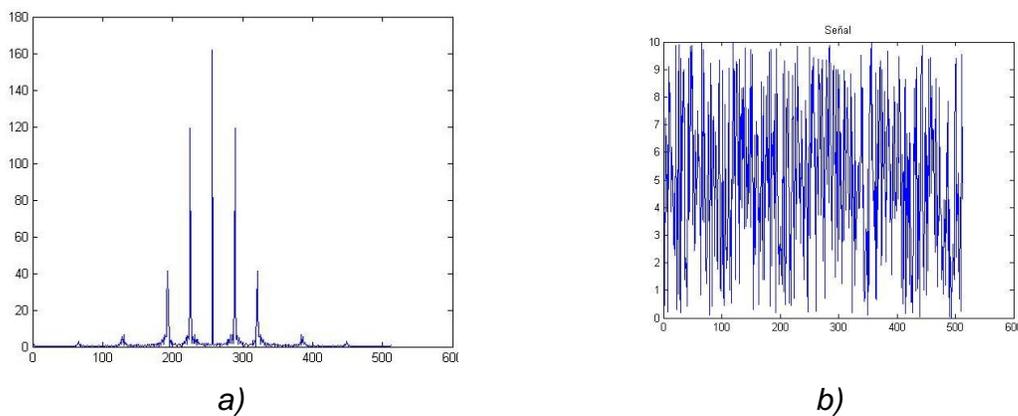


Figura 2.12 a) órdenes de difracción. b) Señal de fase.

De esta señal solo vamos a analizar una parte, justamente f_0 , por lo que es necesario aislar esta parte e ignorar el resto. Para tomar solo una porción de toda la señal, primero creamos un vector que nulifique la señal, multiplicando el valor de cada uno de los pixeles por cero, para que en la imagen los valores de la señal sean iguales a cero, esto eliminará toda la señal en todos los puntos de la frecuencia.

Para obtener solo la parte de la señal que deseamos, se abre una "ventana" sobre la señal nulificada, multiplicando, ahora por 1, solo los pixeles donde se encuentra la parte de la señal que nos interesa, en este caso (una imagen de 512 x 512 pixeles, con un patrón de 32 franjas), solo tomaremos lo pixeles que correspondan al primer orden.

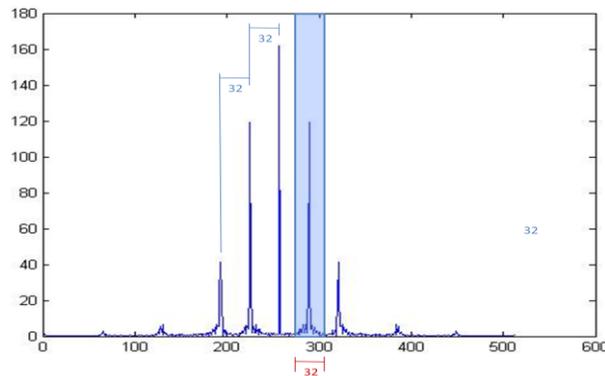


Figura 2.13 Medida en pixeles entre los órdenes de difracción y medida del primer orden.

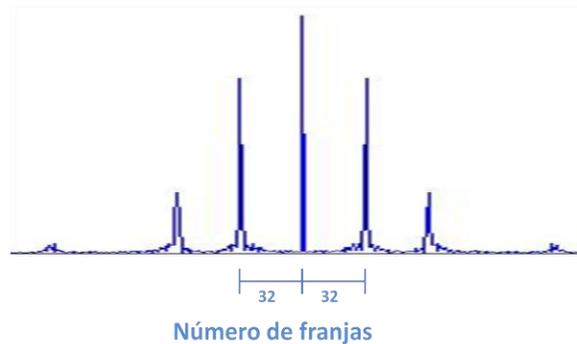


Figura 2.14 Distancia de la frecuencia central a f_0 , medida en pixeles.

El tamaño de la ventana en pixeles, debe ser igual al número de franjas (32 franjas), siempre y cuando la separación en número de pixeles entre la frecuencia central y f_0 (la frecuencia fundamental) sea igual a 32. En la figura 7, tenemos de forma gráfica la "ventana" en donde podemos ver solo la parte que nos interesa de la señal.

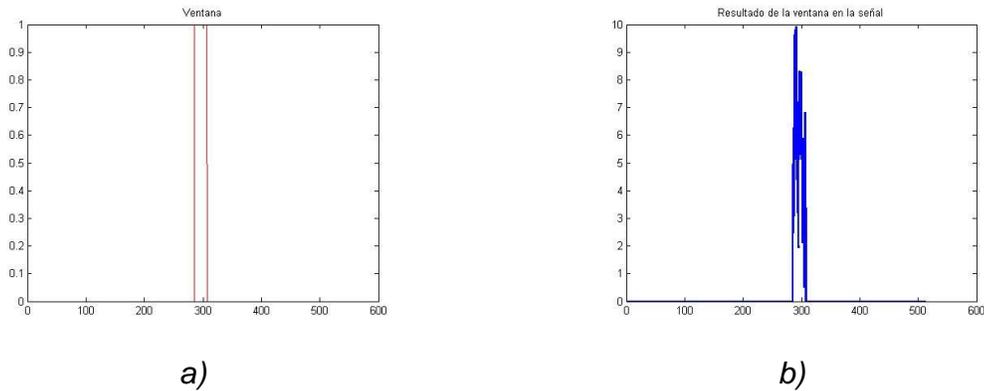


Figura 2.15 En a) lugar en donde se crea la ventana y b), parte de la señal de fase a través de la ventana.

Debemos observar también que el número de franjas debe ser igual a la frecuencia fundamental (f_0), en el caso que nos ocupa, tenemos un patrón de 32 franjas, por lo tanto tenemos una frecuencia de 32 Hz, que se corresponde en tamaño, con el número de pixeles escogido para la ventana. Este número de pixeles se corresponde también con la distancia en pixeles en que se encuentran los diferentes órdenes de difracción.

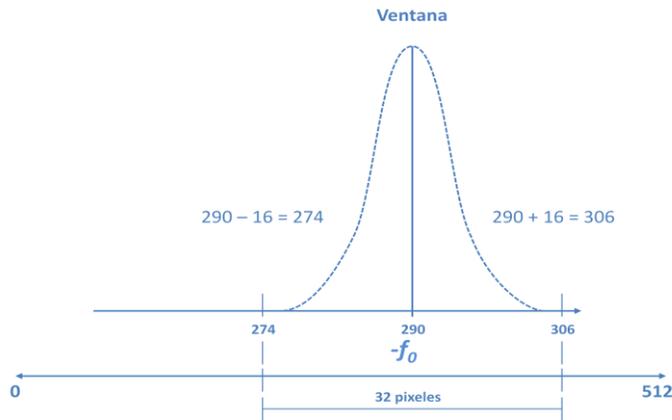


Figura 2.16 Se observa que f_0 se encuentra en un rango igual al número de franjas medido en pixeles.

En la figura 5 podemos observar los órdenes de difracción obtenidos de la modulación de la fase, entre los cuales se encuentra f_0 que es la frecuencia fundamental. De forma aún más clara, en la figura 6, vemos que la distancia entre estos órdenes, medida en pixeles corresponde al número de franjas escogido como patrón de proyección. En la figura 8, vemos como encontrar en f_0 , el lugar exacto para posicionar la ventana y su

medida, que puede ser de 32 píxeles, lo cual es a su vez, un múltiplo de del tamaño total de la imagen (512 x 512) medido también en píxeles.

En un análisis de imágenes por *perfilometría* usando la transformada de Fourier, obtenemos el espectro Fourier de la imagen. En las imágenes que se muestran, se puede observar el espectro de Fourier de la imagen del patrón de franjas sobre el plano de referencia y la imagen del patrón de franjas sobre el objeto.

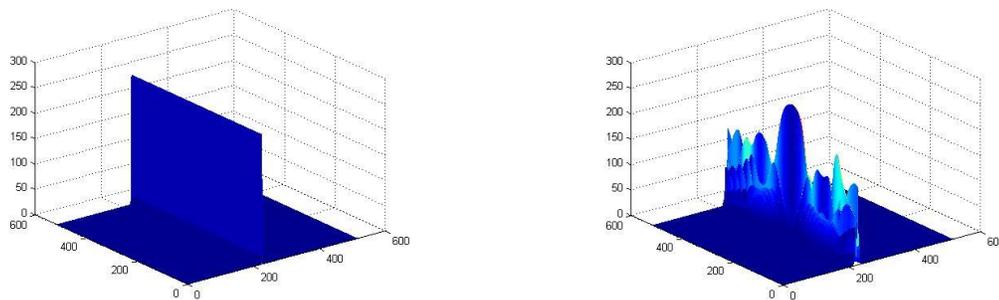


Figura 2.17 En a) la fase del patrón de franjas sobre el plano, en b) la fase del patrón de franjas sobre el objeto.

En la figura a), podemos observar el mapa de profundidad envuelto de la fase de la imagen digitalizada que se obtuvo del patrón de franjas aplicado al plano de referencia. En la figura b), observamos el mismo mapa de profundidad envuelto de la imagen digitalizada del patrón de franjas aplicado al objeto sobre el plano de referencia.

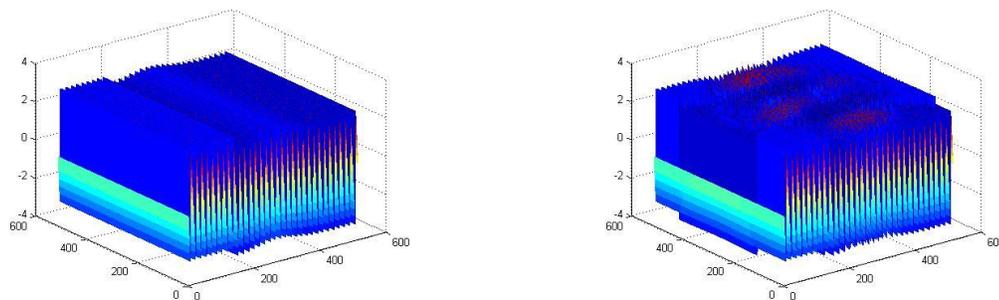


Figura 2.18 a) fase envuelta del patrón de franjas. b) fase envuelta del objeto.

Ahora empezamos el proceso de desenvolvimiento de la fase, que como hemos dicho antes, consiste en eliminar las discontinuidades que se presentan en la fase cuando estas variaciones tienen saltos mayores a 2π . Cambiaremos estos saltos dándoles un valor de 1, de modo que podamos obtener un perfil de distribución de la fase que nos permita encontrar, de forma más precisa, la altura del objeto de estudio.

Para eliminar estas discontinuidades, aplicamos un algoritmo de desenvolvimiento de fase, lo que permitirá la reconstrucción de la forma del objeto en 3D encontrando su altura en la diferencia de fase en la imagen, sin las variaciones que provocan las altas frecuencias debido a los cambios de fase más pequeños que f_0 . La transformada inversa de Fourier, es el método usado para llevar a cabo el desenvolvimiento de la fase en un algoritmo de desenvolvimiento de fase, lo que nos llevará finalmente, a obtener la reconstrucción tridimensional del objeto de estudio.

Podemos observar en las figuras obtenidas de la reconstrucción, efectos de la distorsión causada por las frecuencias altas en

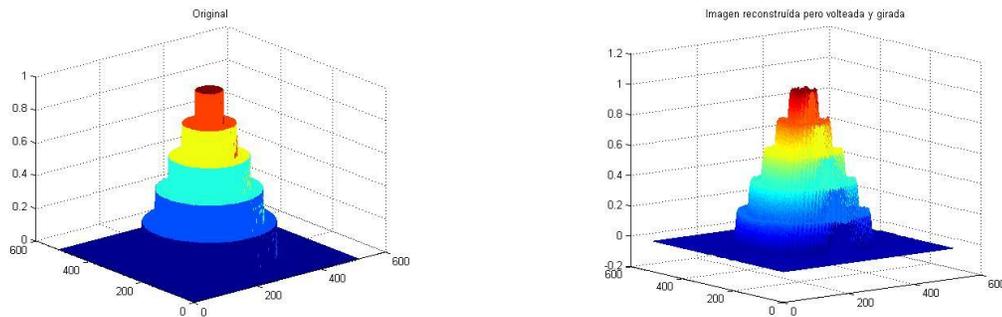


Figura 2.19 En la izquierda, el objeto original usado como modelo para la reconstrucción visto en perspectiva, en la derecha, del objeto reconstruido también en perspectiva.

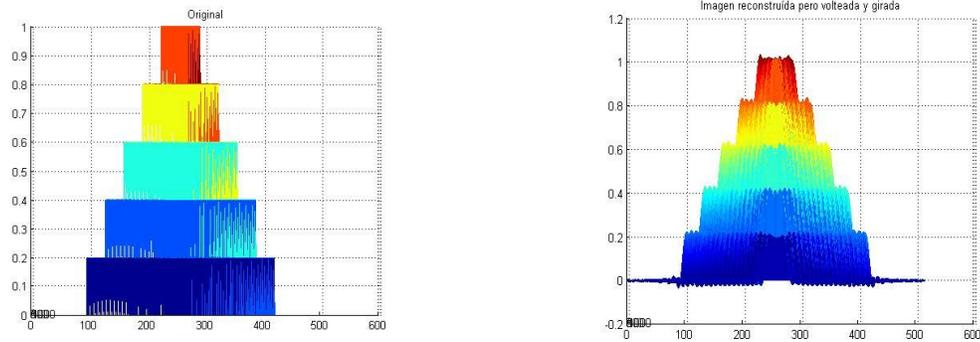


Figura 2.20 Otra vista, a la izquierda vista frontal del objeto original y la reconstrucción a la derecha.

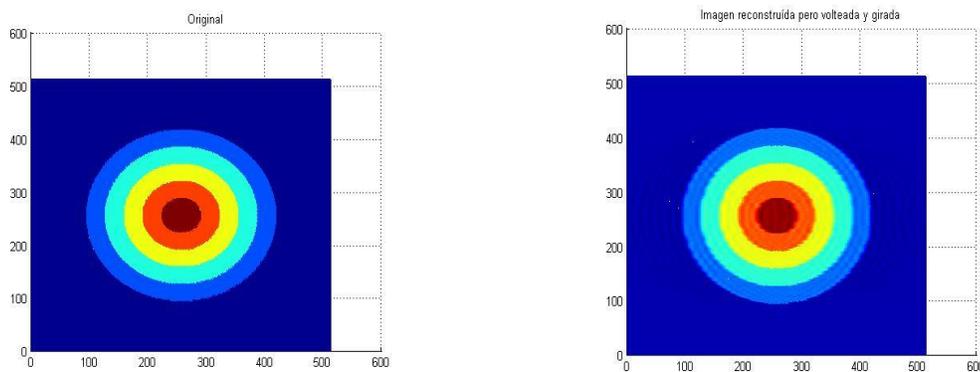


Figura 2.21 Vista superior del objeto original a la izquierda y a la derecha la reconstrucción.

Aunque la reconstrucción fue realizada, podemos observar que existen diferencias entre la imagen original y la imagen reconstruida (figuras 2.18, 2.19 y 2.20). El resultado, aunque semejante y satisfactorio, sigue sin ser del todo preciso debido a que estas técnicas de desenvolvimiento de fase, ofrecen solo aproximaciones a la solución correcta o completa del problema, debido a que en muchos casos los algoritmos no pueden resolver todas las envolturas de fase encontradas en el análisis, lo que significa que todavía existen algunos saltos de fase (2π), sin resolver, estos saltos de fase corresponden a señales de muy alta frecuencia ($n*2\pi$), donde n es un número grande), por lo que estos saltos de alta frecuencia permanecen sin desenvolverse en el mapa de fase y llegan a producir una reconstrucción de no muy buena calidad.

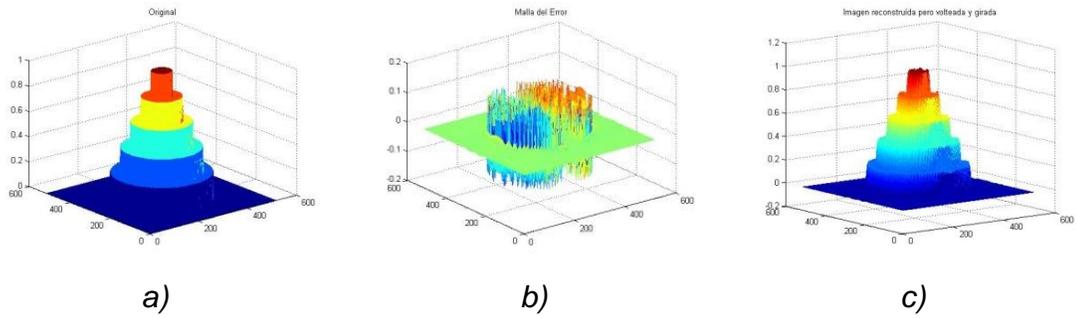


Figura 2.22 Presentación gráfica del error. En la figura a), se muestra la imagen original, la figura b), muestra gráficamente el error en la reconstrucción de la imagen y en la figura c), la imagen resultado de la reconstrucción.

En los siguientes ejemplos de la reconstrucción de objetos en 3D, podemos ver el resultado de la resta del error contra la imagen original, que nos produce un acercamiento más claro para observar la diferencia entre la imagen original y la reconstrucción.

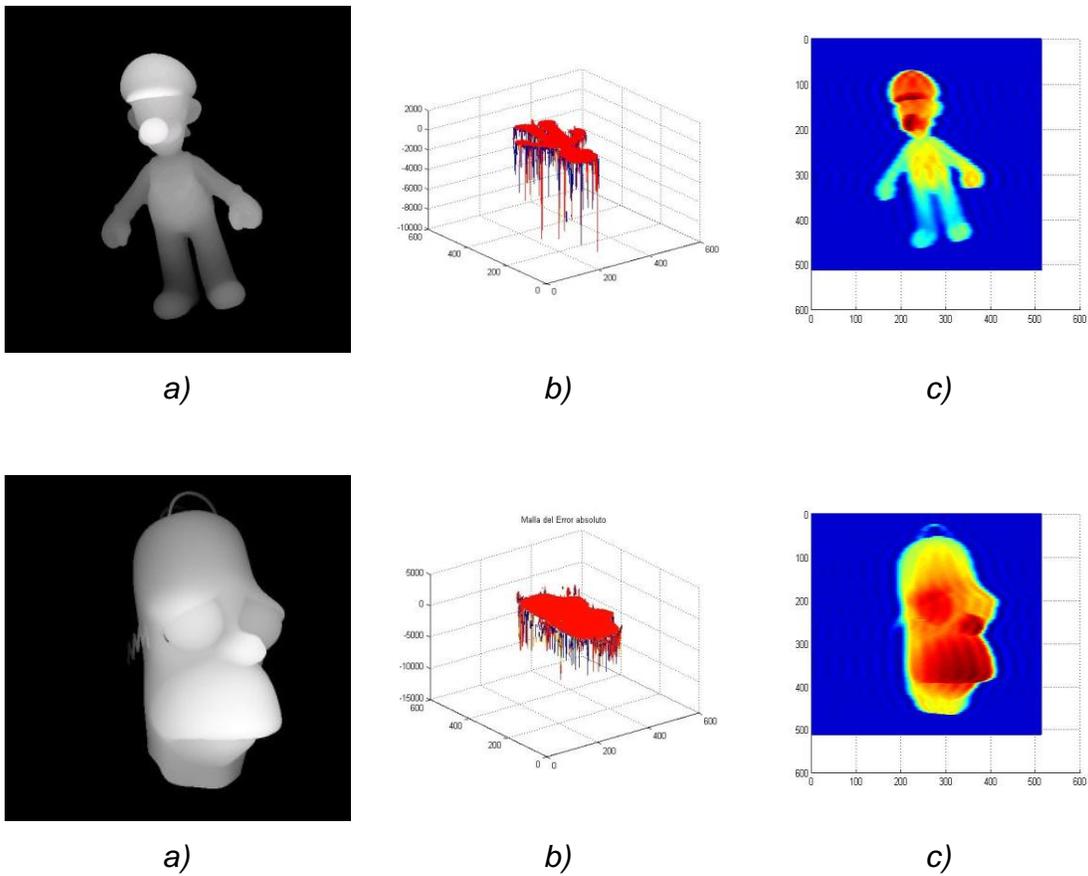


Figura 2.23 Las figuras a), muestran la imagen original, las figuras b), muestran la diferencia del error y la imagen original y en las figuras c), la imagen resultado de la reconstrucción.

Otros factores que influyen en la falta de precisión en la reconstrucción de objetos por este método, son las discontinuidades en la medida de los objetos recuperados o digitalizados en 3D, la posible violación al teorema de Shannon (la función del ancho de banda disponible y el rango de señal a ruido), debida al bajo muestreo en mapas de fase envuelta reales y su variedad de formas, perfiles y densidades de ruido que pueden ser encontradas en mapas de envoltura real, contribuye a hacer del desenvolvimiento de fase, un problema extremadamente complejo.

Pero sin duda, uno de los retos más difíciles para una mejor reconstrucción de objetos en 3D es el reconstruir señales con un cambio abrupto muy pronunciado (problema que plantea el fenómeno de Gibbs en la serie de Fourier), por lo que es recomendable que los objetos que se desea reconstruir tengan un decaimiento suave en su forma.

El algoritmo de desenvolvimiento de fase tiene dos principales consideraciones: la elección de la función confianza y el diseño del recorrido del desenvolvimiento [M. Arevalillo Herráez, D. R. Burton, M. J. Lalor and M. A. Gdeisat, 2002]. Para determinar una función de confianza adecuada en un algoritmo de desenvolvimiento de fase generalmente se utiliza una función de gradiente y la diferencia entre un pixel y sus pixeles vecinos o solo la diferencia. [Takeda Mitsuo., Mutoh Kazuhiro, 1983], [A. Asundi, Z. Wensen, 1998]. Para calcular la función de segunda diferencia entre los pixeles se obtiene de la siguiente manera.

En la figura 2.7, Los pixeles $(i, j-1)$, $(i, j+1)$, $(i-1, j)$ y $(i+1, j)$, son los pixeles vecinos ortogonales e $(i-1, j-1)$, $(i+1, j-1)$, $(i-1, j+1)$ y $(i+1, j+1)$ son llamados pixeles vecinos diagonales [Marklund, Olov, M. Huntley, Jonathan, Cusack, Rhodri, 2007].

$(i-1, j-1)$	$(i, j-1)$	$(i+1, j-1)$
$(i-1, j)$	(i, j)	$(i+1, j)$
$(i-1, j+1)$	$(i, j+1)$	$(i+1, j+1)$

Tabla 2.2 Cálculo de la segunda diferencia en una imagen [Marklund, Olov, M. Huntley, Jonathan, Cusack, Rhodri, 2007]

La segunda diferencia D de un pixel puede ser calculada con la ecuación

$$D(i, j) = [H^2(i, j) + V^2(i, j) + D_1^2(i, j) + D_2^2(i, j)]^{1/2} \quad \text{Ecuación 2.28}$$

... donde...

$$H(i, j) = \gamma[\varphi(i - 1, j) - \varphi(i, j)] - \gamma[\varphi(i, j) - \varphi(i + 1, j)]$$

$$V(i, j) = \gamma[\varphi(i, j - 1) - \varphi(i, j)] - \gamma[\varphi(i, j) - \varphi(i, j + 1)]$$

$$D_1(i, j) = \gamma[\varphi(i - 1, j - 1) - \varphi(i, j)] - \gamma[\varphi(i, j) - \varphi(i + 1, j + 1)]$$

$$D_2(i, j) = \gamma[\varphi(i - 1, j + 1) - \varphi(i, j)] - \gamma[\varphi(i, j) - \varphi(i + 1, j - 1)] \quad \text{Ecuación 2.29}$$

Donde $\gamma(\cdot)$ es una operación de desenvolvimiento simple donde se remueven los saltos de 2π no deseados entre pixeles consecutivos. La segunda diferencia es calculada para todos los pixeles de la imagen excepto para los bordes. La función de confianza es definida entonces como

$$R = \frac{1}{D} \quad \text{Ecuación 2.30}$$

Para el recorrido de la ruta del desenvolvimiento, un borde es una intersección entre dos pixeles ortogonales es decir pixeles conectados horizontal o verticalmente, En la tabla 2.2, (a) muestra las funciones de confianza para cada pixel [A. Asundi, Z. Wensen, 1998]. Los bordes son clasificados en bordes horizontales iluminados en color verde y verticales iluminados en color rojo como se muestra en la figura (b). Un recorrido de desenvolvimiento no solo puede ser definido por la función de confianza de cada pixel de manera individual, se utilizan los bordes con mayor valor en su función de confianza. Las figuras (c) - (j) ilustran el funcionamiento del algoritmo.

En el ejemplo, (a) muestra la función de confianza en una imagen cuyos píxeles se muestran en la figura. Cada dos píxeles vecinos ortogonales puede producir un borde como se muestra en la figura (b). La confianza de un borde se define como la suma de la fiabilidad de los dos píxeles del borde que se conecta como se ilustra. Se aprecian los bordes en color y los bordes verticales, que son de color rojo.

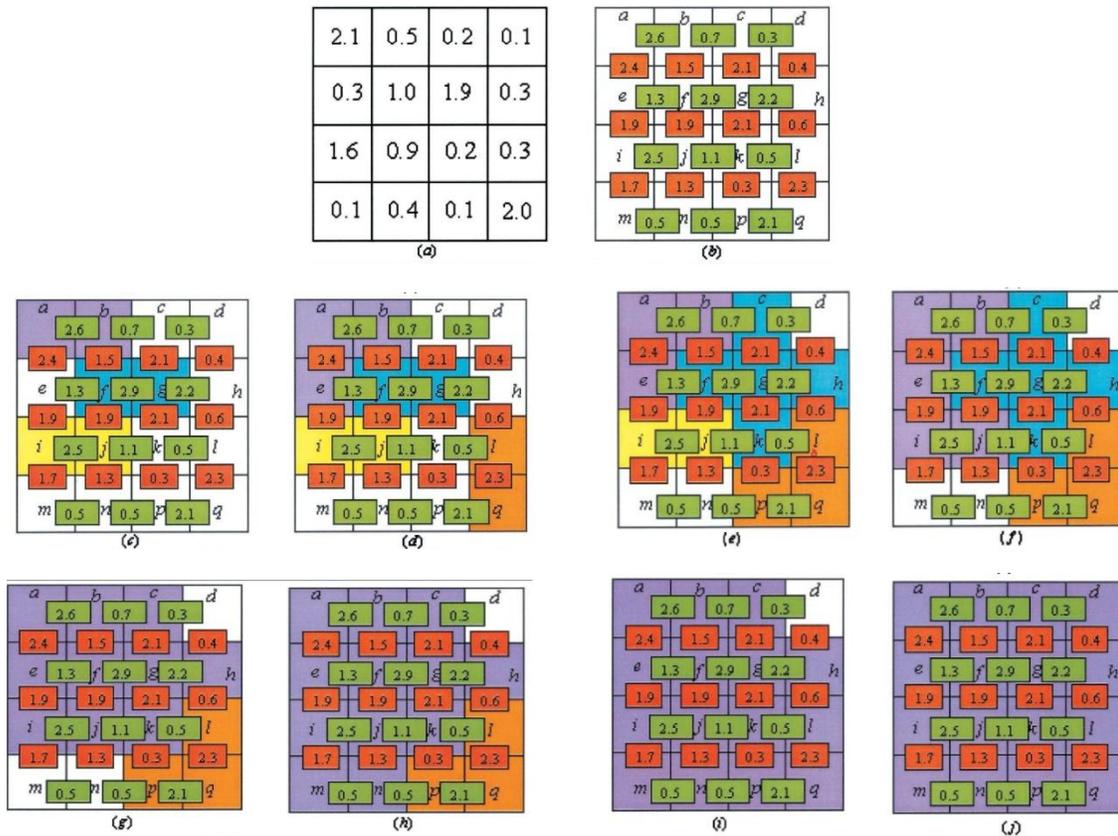


Tabla 2.3 Ejemplo del desarrollo numérico del algoritmo [M. Arevalillo Herráez, D. R. Burton, M. J. Lalor and M. A. Gdeisat, 2002].

La ruta de desenvolvimiento no puede ser definida en relación con la fiabilidad de los píxeles, se define por el valor de la fiabilidad de los bordes. La definición de la ruta de desenvolvimiento es relativamente simple: los bordes con una mayor fiabilidad se desenvuelven primero. Los bordes son almacenados en una matriz y ordenados por su valor de fiabilidad. Los bordes con una mayor fiabilidad se resuelven en primer lugar. Las figuras (c) y (j), ilustran el principio del algoritmo.

Inicialmente se considera que ninguno de los píxeles pertenece a algún grupo. En la figura (c), los píxeles f y g se desarrollieron primero con respecto a los otros, porque el borde que los une tiene el valor más alto de fiabilidad. Por lo tanto, son píxeles pertenecen al mismo grupo, y para ilustrarlo son del mismo color. Entonces la operación de desarrolliendo de los píxeles a y b se lleva a cabo y estos dos píxeles constituyen un segundo grupo. El desarrolliendo de píxeles i y j indican la constitución de un tercer grupo.

Las figuras (c) y (d) muestran que el borde que conecta píxeles a y e , tiene el cuarto valor más alto y los dos píxeles que hacen el borde deben desarrollerse. Pero ya existe un píxel desarrollado con respecto al píxel b y ambos pertenecen al mismo grupo. Para desarrollar el píxel e , se consideran los múltiplos necesarios para ser agregados o restados para desarrollar píxel e . Se considera que los píxeles a , b y e pertenecen al mismo grupo y se les da el mismo color que se muestra en la figura (d).

Las figuras (e) y (f) ilustran el desarrolliendo de dos grupos con respecto a los otros. El primer grupo consta de los píxeles de a , b y e , mientras que el segundo grupo contiene los píxeles i y j . Ambos grupos están conectados por un borde vertical, cuyo valor de fiabilidad es mayor o igual a los valores de fiabilidad de los bordes de otros en estado normal. Ambos grupos deben desarrollerse con respecto a los otros. El segundo grupo contiene un menor número de píxeles, de modo que se calcula la diferencia los múltiplos de 2π entre uno de los píxeles en el primer grupo y uno de los píxeles en el segundo grupo. El valor de los múltiplos 2π se añade o se resta de todos los píxeles en el grupo que contiene el menor número de píxeles. Los cinco píxeles se unieron y ahora se consideran que pertenecen al mismo grupo coma se muestra por su color en la figura (g).

El éxito o fracaso de este proceso puede tener un efecto masivo sobre el desempeño de un sistema completo. Una imagen típica puede tener miles de envolturas de fase individuales, alguna pueden ser genuinas y otras falsas, causadas por la presencia de ruido o pueden ser causadas por el propio algoritmo de extracción. El proceso de diferenciar entre lo falso y lo verdadero, es extremadamente difícil, lo que agrega

complejidad al problema de desenvolvimiento de fase, además del hecho de que el error en este proceso es acumulativo.

Debido a que la imagen es procesada pixel por pixel secuencialmente, si una envoltura de fase genuina se pierde entre dos pixeles vecinos, debido al ruido, o aparece una falsa envoltura en el mapa de fase provoca un error en los dos pixeles desenvueltos. Este tipo de error se propaga a través de toda la imagen, lo que puede causar la inutilidad de toda la imagen. Esta naturaleza acumulativa de del proceso de desenvolvimiento de fase, impone fuerte requerimientos a los algoritmos diseñados para realizar esa tarea.

Las discontinuidades en la medida del los objetos recuperados o digitalizados en *3D*, la posible violación al teorema de Shannon (la función del ancho de banda disponible y el rango de señal a ruido) [Xianyu Su, Wenjing Chen, 2003], debida al bajo muestreo en mapas de fase envuelta reales y su variedad de formas, perfiles y densidades de ruido que pueden ser encontradas en mapas de envoltura real, contribuye a hacer del desenvolvimiento de fase, un problema extremadamente complejo.

2.4 Estado del arte en el desenvolvimiento de fase para la reconstrucción de imágenes 3D.

El desenvolvimiento de fase se ha convertido en un problema clásico en el procesamiento de señales [Lei Ying, Zhi-Pei Liang, David C. Munson, Jr., Fellow Ralf Koetter and Brendan J. Frey, 2006]. Son muchas las soluciones propuestas a este problema por medio de algoritmos cuyos autores proponen soluciones ingeniosas y novedosas. La mayoría se enfoca en tres procesos clave, la construcción de un modelo para la función de fase verdadera, un algoritmo para la recuperación de esta fase obtenida del patrón de franjas del modelo y el algoritmo que permite el mejor desenvolvimiento de esta fase.

Encontrar los valores de la fase desenvuelta, bajo un modelo de fase determinado, lleva implícita la solución de un problema de optimización, que por supuesto, no es trivial. Los métodos basados en la transformada de Fourier, como los conocidos como métodos de seguimiento de ruta

Hemos visto como, desde la condición de Itoh [José M. Bioucas Dias, Gonçalo Valadao], se propone ya un algoritmo para resolver los problemas inherentes al desenvolvimiento de fase, en esta condición descansan la mayoría de los algoritmos y técnicas de desenvolvimiento de fase que emplean el concepto de seguimiento de ruta ampliamente usado en la reconstrucción de modelos estáticos. Los trabajos de Takeda Mitsuo y Mutoh Kazuhiro, dieron forma a los algoritmos de estado del arte.

Los métodos basados en la transformada de Fourier, se dividen en algoritmos globales, de región y de seguimiento de ruta

Los Algoritmos globales formulan el desenvolvimiento de fase en términos de una función global, todos los algoritmos de esta clase son robustos pero computacionalmente intensos, pues consumen muchos recursos. Desenvuelven el mapa de la fase integrando los gradientes de la fase sobre una ruta que conecta todos los pixeles en el mapa de fase envuelta [Abdul-Rahman Hussein, 2007].

Los Algoritmos de región, dividen la imagen en regiones [Itoh, Kazuyoshi, 1982]. Cada región es procesada individualmente y después la región procesada es “desenvuelta” con respecto a las demás para formar regiones más grandes hasta que toda la imagen es procesada. Pueden ser a su vez, clasificados en dos grupos: los basados en mosaicos y los basados en regiones. Los algoritmos de región proveen un balance entre el desempeño robusto e intensidad computacional.

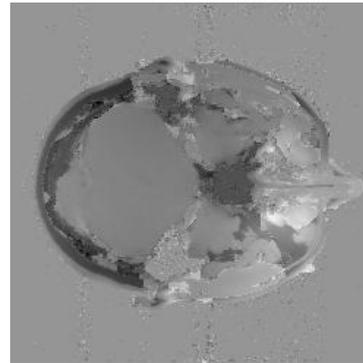
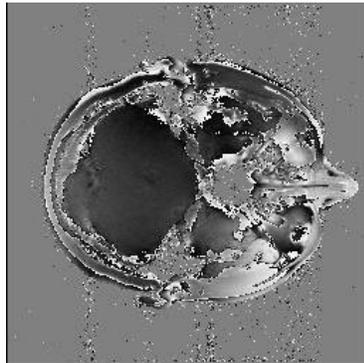
Los Algoritmos de seguimiento de ruta pueden ser sub-clasificados a su vez en tres grupos: métodos dependientes de la ruta, métodos de compensación residual y métodos de ruta guiados por la calidad [Xianyu Su, Wenjing Chen, 2003]. Aplican directamente el concepto introducido por Itoh en su algoritmo, la integración discreta de la fase a lo largo de la ruta.

2.5 Usos de las técnicas de desenvolvimiento de fase.

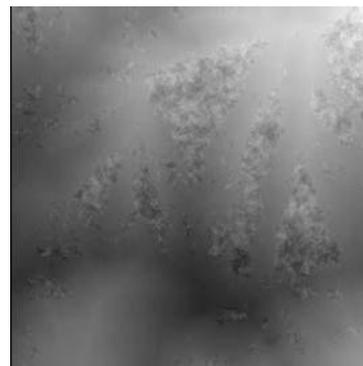
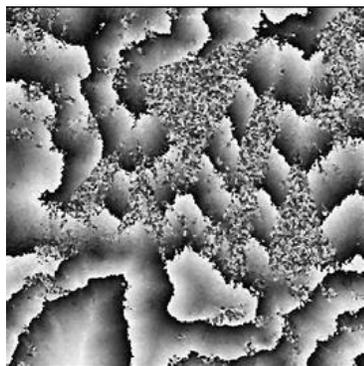
Las áreas donde se aplica el tratamiento de imágenes digitales son muchas, la Medicina, Astronomía, Microscopía, los ámbitos militares y en la industria entre otras tantas beneficiarias del procesamiento digital de imágenes. Además existen diversos modelos para la formación de imágenes y la utilización de estos depende de lo que se desea estudiar, como por ejemplo las imágenes acústicas; en donde se utilizan sonidos tanto en frecuencias bajas (usados en Geología), como de ultrasonido comúnmente usado en medicina.

Ejemplos de aplicaciones para el uso de imágenes digitales procesadas, son la resonancia magnética (MRI), el radar y el sonar de apertura sintética (SAR), la óptica adaptativa, el análisis de patrones de franjas, la *interferometría*, la tomografía y la espectroscopia, la síntesis de imágenes en radio astronomía y el procesamiento de imágenes sísmicas. Estos sistemas utilizan modernos y maduros algoritmos para extraer la señal de fase [Abdul-Rahman Hussein, 2007].

1



2



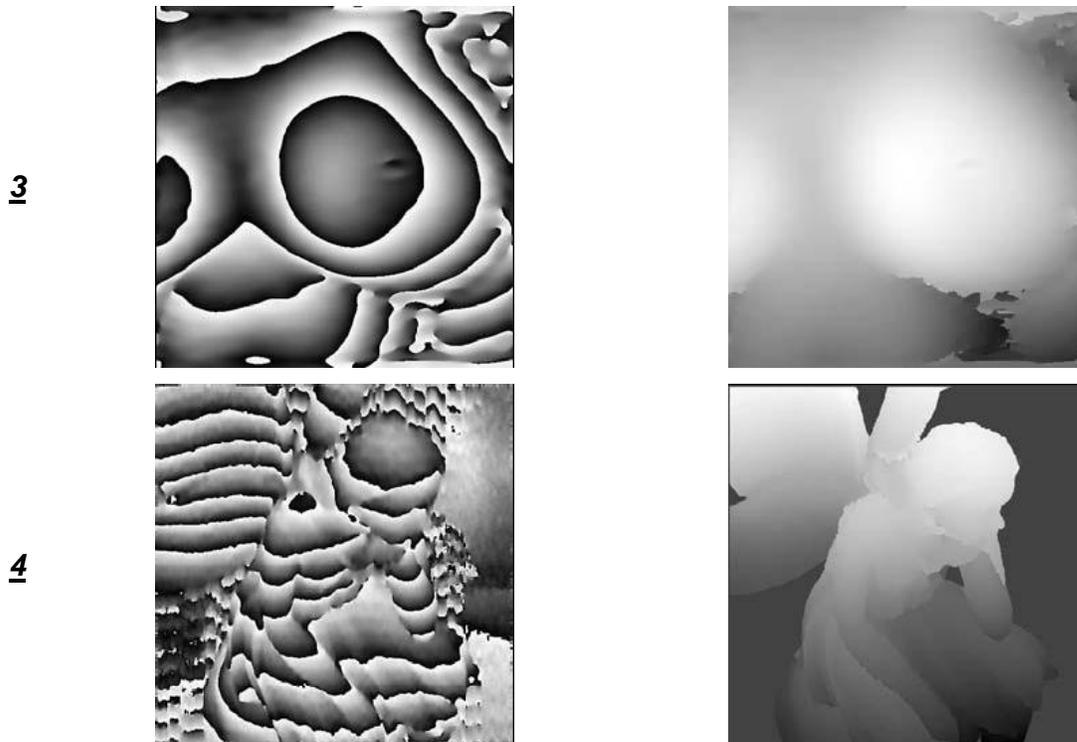


Figura 2.24 Importancia de la técnica de desenvolvimiento de fase en diferentes aplicaciones. Comparación de las imágenes antes de la reconstrucción (columna izquierda) y después (columna derecha).
 Línea 1: Cabeza humana de imagen de MRI, (Ghiglia and Pritt, 1998).
 Línea 2: Parte de la tierra obtenida en imagen de SAR (Ghiglia and Pritt, 1998).
 Línea 3: Torso humano bajo tratamiento de radioterapia.
 Línea 4: Estatua artística de un hada [Abdul-Rahman Hussein, 2007].

Algunos ejemplos de aplicaciones, son la medición de objetos dinámicos utilizando la proyección de patrones de franjas, por ejemplo, en la adquisición de imágenes por resonancia magnética, radares de apertura magnética, tomografías, espectroscopia, análisis de patrones de franjes etc.

Estos algoritmos producen una gran variedad de resultados, los cuales no garantizan la solución correcta al problema de desenvolvimiento de fase, en lugar de eso, ofrecen solo estimaciones a la solución correcta debido a que en muchos casos, estos complejos algoritmos no pueden resolver todas las envolturas de fase encontradas en el análisis de la imagen, por lo que siguen existiendo saltos de fase (2π), en el mapa de fase desenvuelto de la mayoría de las imágenes analizadas en aplicaciones prácticas.

Generalmente, los algoritmos más robustos, requieren de largos periodos de procesamiento, e incluso de más poder de procesamiento en la medida en que se incrementa el ruido que corrompe el mapa de fase envuelta. Por otro lado, los algoritmos rápidos, generalmente no son robustos y no pueden ser usados en aplicaciones reales.

3. Algoritmos de desenvolvimiento de fase usados en la reconstrucción de imágenes 3D.

Para resolver los problemas inherentes al desenvolvimiento de fase, se han propuesto varios métodos de algoritmos que plantean soluciones sofisticadas aplicables en diversos casos, son los algoritmos de seguimiento de ruta, y de estos los que nos ocupan en esta tesis, son tres tipos conocidos como Algoritmos de Ruta Dependiente, Algoritmos de Compensación Residual y Algoritmos de Ruta Guiada por la Calidad.

El mapa de fase absoluta y continua es adquirido por los métodos de desenvolvimiento de fase a partir de la fase envuelta original. Como hemos explicado anteriormente, durante este proceso surgen discontinuidades provocadas por los saltos en los intervalos de $-\pi$ a π , los que en imágenes que no fueron muestreadas con frecuencia espacial suficientemente alta o cuando la imagen tratada tiene imperfecciones o distorsiones desde su captura, pueden generar errores en la desenvoltura de la fase [Mitsuo Takeda, Hideki Ina, Seiji Kobayashi, 1982].

La solución para recuperar el valor absoluto de la fase envuelta entre pixeles vecinos, se obtiene asumiendo que ésta es menor a π y permanece de esta manera a lo largo de una ruta de integración, esta condición se conoce como la condición Itoh, [José M. Bioucas Dias, Gonçalo Valadão], propuesta por Kazuyoshi Itoh, usada en la mayoría de las técnicas de desenvolvimiento de fase [Itoh, Kazuyoshi, 1982].

Podemos definir al operador de desenvolvimiento $W(\cdot)$ de la siguiente manera...

$$\begin{aligned}
 W: \mathbb{R} &\mapsto]-\pi \ \pi] && \text{Ecuación 3.1} \\
 \phi &\rightarrow \phi - 2\pi k
 \end{aligned}$$

... donde $k \in \mathbb{Z}$, por lo que $(\phi - 2\pi k) \in]-\pi \pi]$. El operador W , envuelve cualquier fase ϕ en el intervalo $]-\pi \pi]$, el cual está centrado en cero en el intervalo principal, que es, en esencia, el operador módulo 2π , sustrayendo 2π radianes [Itoh, Kazuyoshi, 1982]. Si consideramos una secuencia de pixeles que son vecinos $\{\phi_n\}$ dentro de la fase absoluta de una imagen, con una secuencia de diferencias lineales como...

$$\Delta \phi_n = \phi_n - \phi_{n-1} \quad \text{Ecuación 3.2}$$

En este caso, ϕ denota una fase absoluta desenvuelta y el operador Δ , la contendrá independientemente de si es absoluta o está envuelta. La condición Itoh puede expresarse como...

$$|\Delta \phi_n| \leq \pi \quad \text{Ecuación 3.3}$$

... agregando la secuencia propuesta en la ecuación anterior...

$$\sum_{n=1}^m \Delta \phi_n = \phi_m - \phi_0 \quad \text{Ecuación 3.4}$$

... de modo que los valores de las secuencias intermedias se cancelan uno con otros. De la definición del operador de desenvolvimiento de la ecuación 3.1, obtenemos...

$$\Delta W(\phi_n) = \phi_n - \phi_{n-1} - 2\pi (k_n - k_{n-1}) \quad \text{Ecuación 3.5}$$

... en donde $k_n, k_{n-1} \in \mathbb{Z}$. Si consideramos nuevamente la secuencia de la ecuación 3.2, tenemos que...

$$\underbrace{W[\Delta W(\phi_n)]}_a = \underbrace{\Delta \phi_n - 2\pi (k_n - k_{n-1}) - 2\pi k}_b \quad \text{Ecuación 3.6}$$

... en donde $k_n, k_{n-1}, k \in \mathbb{Z}$, $2\pi k$ es el múltiplo de 2π para atraer a "a" al intervalo principal. Como por definición $|a| \leq \pi$, necesitamos que "b = 0", entonces...

$$W[\Delta W(\phi_n)] = \Delta\phi_n \quad \text{Ecuación 3.7}$$

... para finalmente introducir la ecuación anterior en la ecuación 3.4...

$$\phi_m = \sum_{n=1}^m W[\Delta W(\phi_n)] + \phi_0 \quad \text{Ecuación 3.8}$$

... y así obtenemos el método o condición de Itoh, un procedimiento que desenvuelve la fase de cualquier pixel (ϕ_m) del cual se conoce su fase absoluta (ϕ_0), a partir de los valores de fase envueltos a lo largo de una ruta ($k_n - k_{n-1}$), que encadena pixeles vecinos. Siguiendo esta ruta, es posible desenvolver correctamente la totalidad de los pixeles de una imagen [José M. Bioucas Dias, Gonçalo Valadão].

No obstante, la condición Itoh, puede ser violada si la superficie de fase verdadera presenta discontinuidades o si el mapa de fase contiene ruido, es este caso, las diferentes rutas de integración pueden conducir a diferentes valores cuando la diferencia en la fase entre pixeles vecinos es mayor a π radianes, provocando inconsistencias en el desenvolvimiento de fase [Abdul-Rahman, Hussein S., Gdeisat, Munther A., Burton, David R, et All, 2007].

El término “inconsistencia”, es también conocido como “residuo” y se calcula siguiendo el sentido inverso o contrario en ciclos cerrados. Los ciclos cerrados se establecen cuando el desenvolvimiento de fase recorre una ruta y regresa a un pixel ya desenvuelto. Considerando un ciclo cerrado entre dos rutas paralelas, la suma de sus residuos es un buen criterio para la detección de inconsistencias [Dongliang Zheng, Feipeng Da, 2011].

Aplicar esquemas de integración de líneas sobre la imagen cuya fase se encuentra envuelta, es la forma como los algoritmos de seguimiento de ruta, intentan resolver estas discontinuidades.

En la literatura existen diferentes definiciones y clasificaciones de algoritmos de desenvolvimiento de fase. Muchos de ellos se basan en el tipo de aplicación específica, dependiendo del área de estudio. Después de haber llevado a cabo una revisión

bibliográfica exhaustiva, y considerando el área de reconstrucción tridimensional de objetos cotidianos, se encontró que se utilizan principalmente tres algoritmos de desenvolvimiento de fase, los cuales serán el objeto de estudio de este trabajo, y se presentan a continuación.

3.1 Algoritmos de ruta dependiente

Hemos visto como la condición de Itoh, provee inmediatamente un método de desenvolvimiento de fase que utiliza el concepto de seguimiento de ruta para resolver las discontinuidades de la fase, pero resultaría irreal esperar que sea aplicable a todos los casos debido a la riqueza y variedad que presenta la topografía típica del mapa de fase y por lo tanto, también de sus discontinuidades [Abdul-Rahman, Hussein S., Gdeisat, Munther A., Burton, David R, et All, 2007].

Todos los métodos de seguimiento de ruta, aplican el concepto de integración de fase discreta a lo largo de una ruta, por medio del algoritmo del vecino más cercano, es un algoritmo de búsqueda heurística [Xianyu Su, Wenjing Chen, 2003]. A partir de un pixel inicial del cual se conoce su fase absoluta, se puede computar la fase absoluta de cualquier otro pixel, sin restringir el encadenamiento de rutas en cualquier punto o pixel [Itoh, Kazuyoshi, 1982], [Abdul-Rahman, Hussein S., Gdeisat, Munther A., Burton, David R, et All, 2007]. También se le conoce como algoritmos de corte de ramas. integración de la fase discreta, no necesariamente es dependiente de la ruta que sigue el algoritmo de desdoblamiento, la dependencia de la ruta se da cuando el algoritmo escoge de entre los pixeles vecinos a cuáles de ellos desenvolverá primero y en qué orden.

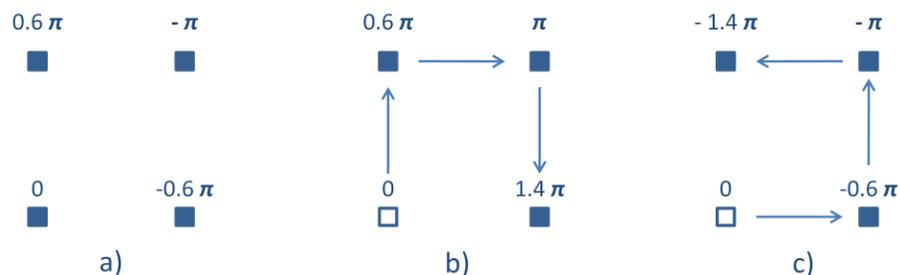


Figura 3.1 a) 1 imagen envuelta, b) una posible ruta de solución, c) otra posible ruta de solución invirtiendo el sentido. El pixel inicial es el 0 [Itoh, Kazuyoshi, 1982].

Este fenómeno de dependencia de las rutas en el desenvolvimiento de fase 2D, está restringido a algunas regiones de la fase de la imagen. Por definición, la dependencia de integración de rutas se prueba entre rutas paralelas, revirtiendo la dirección de una de ellas. Llamamos rutas paralelas a dos o más rutas que comparten los pixeles inicial y final [José M. Bioucas Dias, Gonçalo Valadao], existe dependencia de la ruta solo si la cantidad de rutas no es, eventualmente, igual a cero [Itoh, Kazuyoshi, 1982].

Siguiendo las rutas trazadas por los algoritmos de ruta dependiente, podemos prevenir, también, la propagación de ruido que producen las inconsistencias o residuos a través de los diversos puntos o regiones de la imagen, proporcionando una mejor y más correcta desenvolvura de la fase [Xianyu Su, Wenjing Chen, 2003].

Una estrategia para superar la dependencia de rutas en los algoritmos de seguimiento de ruta, es conectar los residuos de carga opuesta entre sí, por medio de líneas entre las cuales no se permite el cruzamiento de las rutas de integración, de manera que la carga de los residuos que no se encuentran integrados en ninguna ruta, sea igual a cero [Abdul-Rahman, Hussein S., Gdeisat, Munther A., Burton, David R, et All, 2007]. Esto claramente rompería el esquema del algoritmo de ruta dependiente.

Existen diversos tipos de algoritmos de ruta dependiente o corte de ramas. Entre los que se encuentran variantes de algoritmos de norma mínima, que tratan de encontrar la solución a la fase absoluta de la imagen, para lo cual, la diferencia de los pixeles (con orientación horizontal y vertical), entre la fase absoluta y la fase envuelta, es minimizada.

El grado de disimilitud entre las dos fases puede ser procesada como el peso total de los bordes que fueron removidos. En teoría de grafos, a esto se le llama "corte". Básicamente, un grafo $G = (V, E)$, puede ser dividido en dos grupos A, B , $A \cup B = V$, $A \cap B = \emptyset$, simplemente removiendo los ejes que conectan ambas partes.

$$cut(A, B) = \sum_{u \in A, v \in B} W(u, v)$$

Ecuación 3.9

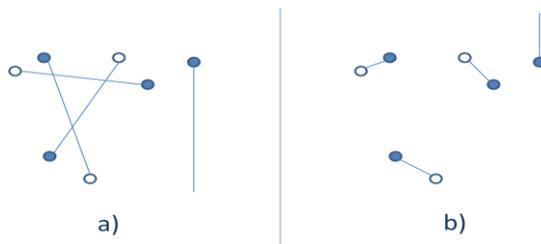


Figura 3.2 Representación de residuos positivos (puntos negros), y residuos negativos (puntos blancos) a), y sus posibles configuraciones de corte de ramas (Branch cut) b).

La definición en un algoritmo de los métodos de corte de ramas (*Branch cut*) se define de la manera siguiente:

Los métodos de ruta dependiente han sido desarrollados para usarse en mapas de fase envuelta con grandes inconsistencias por ruido en su fase envuelta.

3.2 Algoritmos de compensación residual.

Otro método para corregir los errores de desenvolvimiento causados por las discontinuidades en el mapa de fase, son los algoritmos llamados de compensación residual, los cuales son efectivos en mapas de fase con ruidos de distorsión (*moiré*) en las franjas proyectadas. Para esto, el algoritmo coloca una franja o brecha mediante un corte (*branch cuts*), dando una ruta de desenvolvimiento que no permite rebasar estas brechas, previniendo la propagación de ruido generado por los puntos de regiones corruptas. Esta ruta conduce por un único y correcto desenvolvimiento de fase, encontrando las discontinuidades en los mapas de fase envueltos. El concepto de corte de brechas y balance de cargas residuales son ampliamente usadas en diversos algoritmos de seguimiento de ruta para el desenvolvimiento de fase en 2D [Gonçalo Ramiro, Valadão Matias, 2006].

Los pixeles que contienen discontinuidades, llamados también residuos, pueden ser negativos o positivos. Para un correcto desenvolvimiento de fase, la ruta de desenvolvimiento debe encerrar un número igual de residuos positivos y negativos. Una vez que los pixeles sin distorsión han sido desenvueltos, se procede a desenvolver los

pixeles dentro de las franjas. El número de pixeles dentro de las franjas de corte, deberá ser menor al de los pixeles fuera de ellas.

D. Ghiglia y otros autores [1987], fueron los primeros en reportar el fenómeno de la dependencia de la ruta en el desenvolvimiento de fase en $2D$, y observaron que las inconsistencias están restringidas a algunas regiones en la fase de la imagen.

Para identificar con detalle y precisión la localización de las inconsistencias en la imagen, se sigue la estrategia de integrar la fase en la ruta dentro de cada ciclo elemental de 2×2 pixeles (la ruta cerrada más corta). Cuando en esta integración, la suma no es cero, evidentemente señala una inconsistencia precisamente en ese ciclo [S Karout, M A Gdeisat, D R Burton, M J Lalor, 2007]. Sabemos que, la dependencia en la integración entre dos rutas paralelas se puede probar revirtiendo la dirección de una de ellas para después integrarla a lo largo de la ruta del bucle resultante.

Los residuos son calculados usando ciclos cerrados recorridos en sentido de las manecillas del reloj, el valor absoluto de su suma es siempre un múltiplo de 2π y su signo define a la carga residual como positiva o negativa. Cualquier ruta cerrada puede estar dada por una composición de todos los ciclos elementales que encierra.

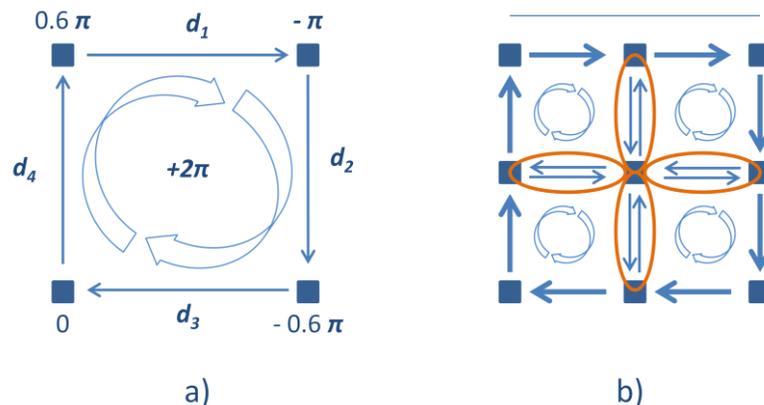


Figura 3.3 a) Una carga residual positiva (inferior izquierda), sumando siempre múltiplos de 2π en ciclos positivos (en el sentido de las manecillas del reloj). b) forma de obtener la carga residual mostrada en la composición de un ciclo cerrado (en la elipse) entre pixeles vecinos en rutas paralelas mostrando ciclos cerrados [Abdul-Rahman, Hussein S., Gdeisat, Munther A., Burton, David R, et All, 2007].

Mientras un residuo localizado en un pixel implica la existencia de una inconsistencia, otro pixel con inconsistencia de carga contraria puede cancelar la inconsistencia de ambos.

1. Calcula el gradiente de fase en las direcciones x y y .
2. Identifica residuos positivos y negativos en el mapa de fase envuelta.
3. Empieza aleatoriamente desde un residuo en el mapa de fase.
 - 3.1 Identifica el residuo localizando pixeles consecutivos con valores de gradiente altos o bajos en los mapas de fase de gradiente x y y respectivamente.
 - 3.2 Sigue la ruta residual en ambos lados del residuo hasta encontrar un residuo balanceado de polaridad opuesta o hasta que el vector residual alcance a un cero. Si encuentra un vector cero:
 - 3.2.1 Identifica el vector residual más cercano de otro residuo e incluye pixeles del más alto al más bajo gradiente localizado entre los dos vectores.
 - 3.2.2 Iterativamente localiza una ruta residual más cercana de otro residuo cercano a la último residuo hasta encontrar un residuo balanceado de polaridad opuesta de la del residuo inicial.
 - 3.3 Corta todas las ramas de todos los pixeles de ese par de residuos.
4. Repite el mismo procedimiento hasta que todos los residuos son balanceados por cortes de rama creados por este método.

Tabla 3.1 Algoritmo para localizar los residuos positivos o negativos en una ruta o brecha creando cortes de rama para balancear los residuos [S Karout, M A Gdeisat, D R Burton, M J Lalor, 2007].

En la compensación residual, un mapa de fase desenvuelto, se define por tener todos los residuos en el mapa de fase neutralizados o separados y compensados por segmentos de brechas, lo que no quiere decir que el conjunto de brechas con sus residuos localizados y desenvueltos, resulten en la correcta estimación de la fase para su apropiado desenvolvimiento [S Karout, M A Gdeisat, D R Burton, M J Lalor, 2007].

3.3 Algoritmos de ruta guiada por la calidad.

Estos algoritmos emplean una clase de mapas de calidad para guiar las rutas de integración. Debido a que los residuos tienden a estar localizados en regiones de la imagen con poca calidad de definición, usan la información adicional que típicamente proporcionan los mapas de correlación para evitar residuos originados fuera de las rutas de dependencia identificando explícitamente estos residuos [Zhang, Li. Liao, Mingsheng. Zhang, Zuxun. Zhang Yong, 1999]. También constituyen un intento de superar la mala identificación de las brechas de corte, lo que por sí mismo representa un problema difícil de superar.

La mayoría de estas técnicas consisten en dejar que los valores de calidad definan el orden en que se efectuará el desenvolvimiento de fase, dando prioridad a los píxeles con una alta calidad de fase. La concepción de los algoritmos que utilizan mapas de calidad para delinear los cortes de rama, consiste en emplear los mapas de calidad, de forma que al definir y retener los cortes de brecha, se garantice la independencia de la integración de la ruta de desenvolvimiento de fase.

Las imágenes obtenidas de la fase envuelta tienen, además de significativas discontinuidades, altas variaciones en los valores de fase, por esto son necesarios en algunos casos, los mapas de calidad para guiar el proceso de desenvolvimiento de fase. Por ejemplo, en una imagen de un mapa de fase en la que el valor de la fase se incrementa linealmente en saltos de magnitud más allá del intervalo de 2π (envolturas), las líneas de las curvas que se observan gráficamente, tienen saltos de magnitud que marcan esta transición de 0 a 2π , estas discontinuidades causan el ruido y los problemas de exactitud en la reconstrucción final de la imagen. Básicamente, los mapas de calidad son arreglos de valores que definen o evalúan la calidad o bondad de cada píxel [Suprijanto, F. M. Vos, H. M. Vrooman, A. M. Vossepoel].

Como sabemos, la relación entre la fase envuelta $\varphi_0(x, y)$, y la fase desenvuelta $\varphi(x, y)$ puede ser establecida como:

$$\varphi_0(x, y) = W [\varphi(x, y)] = \text{mod}\{\varphi(x, y), 2\pi\}, 0 \leq \varphi_0(x, y) < 2\pi$$

Ecuación 3.10

En donde W , es el operador de envoltura (envuelve el valor de su argumento en el rango de $-\pi$ hasta π). como se observa, el desenvolvimiento de fase, se esfuerza en retener los valores continuos de la fase desenvuelta [Heping Zhong, Jinsong Tang, Sen Zhang, and Ming Chen, 2011]. De acuerdo con la descripción de Itoh [Itoh, Kazuyoshi, 1982], el problema del desenvolvimiento de fase puede ser resuelto integrando los valores envueltos de las diferencias de la fase envuelta, pero en la práctica, sabemos que el proceso de desenvolvimiento de fase se dificulta por la presencia de ruido residual.

Los algoritmos de ruta guiada por la calidad, son esencialmente algoritmos de llenado como en una pila de datos, en la cual el orden de llenado es determinado por la calidad del mapa, la lista generada en la pila de datos, proporcionará la ruta más adecuada para el desenvolvimiento de todo el arreglo.

- | |
|---|
| <ol style="list-style-type: none"> 1. Genera el mapa de calidad de un pixel . <ol style="list-style-type: none"> 1.1 Encuentra el pixel con el valor de calidad más alto. 1.2 Toma su valor envuelto y también su valor desenvuelto. 1.3 Introduce estos valores en una lista. 2. Desenvuelve a todos los vecinos de ese pixel. <ol style="list-style-type: none"> 2.1 Agrega a los pixeles vecinos desenvueltos a la lista. <ol style="list-style-type: none"> 2.1.1 Encuentra el pixel con el valor de calidad más alto en la lista. 2.1.2 Remueve el pixel encontrado con el valor más alto y regresa al paso 2. 3. Termina cuando todos los pixeles de esa lista han sido desenvueltos y regresa al paso 1. |
|---|

Tabla 3.2 Algoritmo de ruta guiada por la calidad para el desenvolvimiento de fase [Heping Zhong, Jinsong Tang, Sen Zhang, and Ming Chen, 2011].

La calidad de un mapa, depende de la calidad de cada uno de sus pixeles y su relación con la de sus vecinos, estos, deben tener valores continuos por debajo de 2π . En un

mapa de calidad, el valor de gradiente más grande significa menor calidad [Heping Zhong, Jinsong Tang, Sen Zhang, and Ming Chen, 2011].

En el algoritmo convencional de ruta guiada por la calidad, el orden de desenvolvimiento de la fase es determinada solo por el valor de la calidad. Las zonas del mapa con discontinuidades el error puede verse propagado afectando de forma determinante el resultado del desenvolvimiento de fase de todo el mapa y por lo tanto, de la imagen resultante.

4. Implementación de los algoritmos de desenvolvimiento de fase y resultados experimentales.

La implementación de los algoritmos de desenvolvimiento de fase de cada uno de los métodos, como ya se explicó anteriormente, se realiza en el entorno del programa Matlab, que es un entorno interactivo con un poderoso lenguaje de alto nivel muy usado para cálculo numérico, visualización y graficado, además de programación en todos los ámbitos donde se requiera procesamiento de datos matemáticos.

Las pruebas fueron realizadas a cinco imágenes, a las que se aplicó todo el procedimiento descrito anteriormente, a grandes trazos, la obtención de la imagen en la memoria de la computadora, la proyección de un patrón de franjas, la obtención de la fase envuelta, el desenvolvimiento de la fase y la reconstrucción final del modelo en tres dimensiones. Se usó la misma computadora y recursos para procesar a todas las imágenes, un computadora HP Pavillion DV4, con un procesador AMD Athlon Dual-Core, M320 a 2.10 Giga Hertz, seis Giga Bites de memoria de acceso aleatorio y tarjeta gráfica Vision AMD M880G with ATI Mobility Radeon HD 4200 con 3022 Mega Bites de memoria compartidos. Sistema operativo de 64 Bits Windows 7 Ultimate, Service Pack 1.

4.1 Análisis

Se espera que un mejor algoritmo de desenvolvimiento de fase, sea el que permita lograr la mejor reconstrucción en 3D del objeto de estudio, la más parecida al objeto real; esto no es siempre así. A partir de la utilidad que se destine a las imágenes reconstruidas, serán deseables distintas características. Por ejemplo, alguna aplicaciones para sonar precisan mayor velocidad de procesamiento y más rápida entrega de los resultados. algunas reconstrucciones con fines médicos, requieren mayor precisión o más detalle, y de este modo, podemos imaginar porque se han desarrollado y se siguen desarrollando muchas más propuestas de algoritmos y soluciones al problema de el desenvolvimiento de fase y reconstrucción en tercera dimensión.

Los algoritmos usados en este trabajo de investigación, tenían como finalidad mostrar el desempeño típico de cada método en la obtención de la fase de la imagen cargada, realizar el tipo de seguimiento de ruta para encontrar discontinuidades y resolverlas

desenvolviendo la fase en los pixeles señalados y realizar la reconstrucción. A continuación se muestran los resultados, divididos en los tiempos de ejecución y aspecto visual.

Los tiempos de ejecución, representan la cantidad de tiempo que tarda cada uno de los algoritmos en realiza la misma tarea con las distintas imágenes. El aspecto visual, muestra de forma gráfica el aspecto que tiene cada una de las reconstrucciones en tres dimensiones de los objetos de estudio con cada uno de los métodos usados en este trabajo.

4.1.1 Tiempo de ejecución.

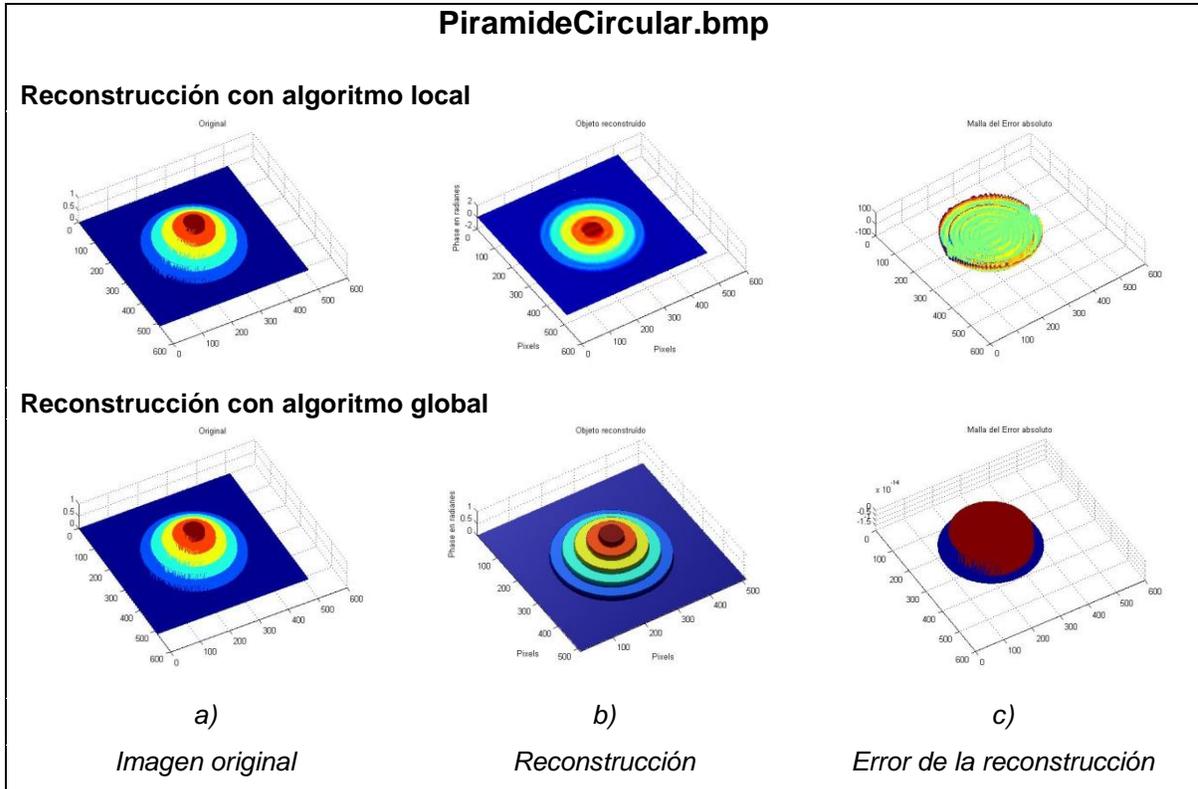
En la tabla 4.1, podemos observar los resultados en tiempo de ejecución que cada algoritmo tardó en realizar todo el procedimiento en cada una de las imágenes, desde la adquisición de la imagen, hasta la reconstrucción del objeto de estudio en un modelo en tres dimensiones. Se aprecia que, para estos casos y usando los algoritmos descritos, el algoritmo global, obtiene los mejores tiempos de ejecución, lo que no necesariamente significa la mejor reconstrucción o la más adecuada para nuestro trabajo y perspectiva.

Figura	Número de pixeles	Algoritmo local	Algoritmo global	Diferencia de tiempo
PiramideCircular.bmp	512	4.964872	3.403429	1.561452
Homero512.jpg	512	5.032898	3.805465	1.227433
Luigi512.jpg	512	5.123592	3.686146	3.686146
OsoTed.jpg	512	5.132744	3.613178	1.519566
SistemaSolar512.jpg	512	5.091677	3.970545	1.121132

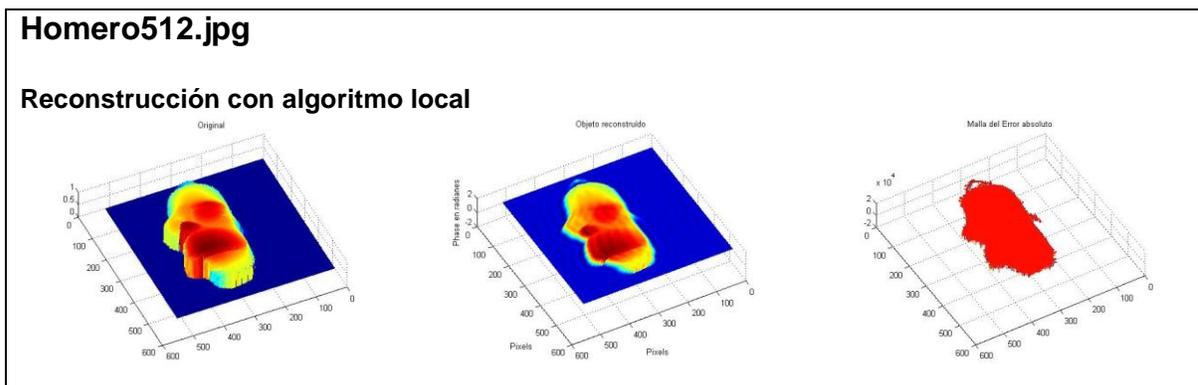
Tabla 4.1 Tiempo de ejecución de los algoritmos local y global y la diferencia de tiempo

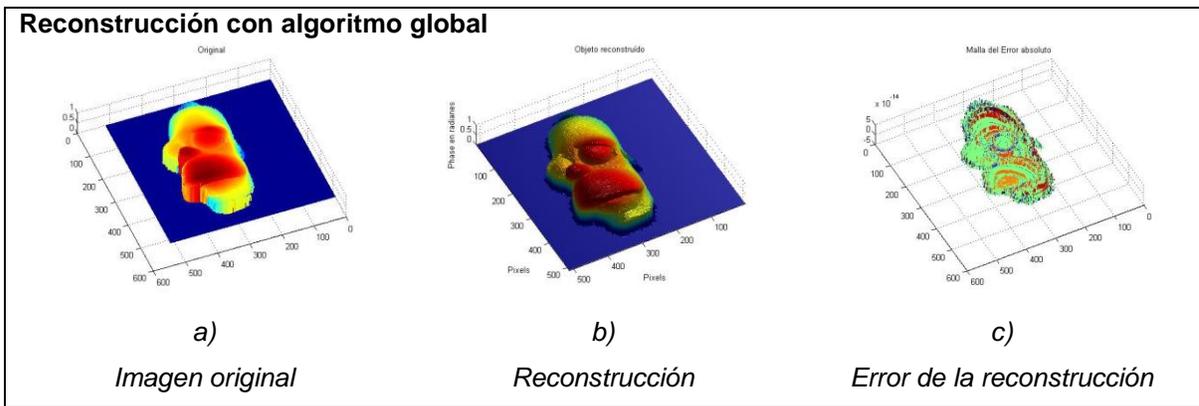
4.1.2 Aspecto visual.

Con las imágenes de los resultados obtenidos, comparamos visualmente las reconstrucciones de los distintos objetos y observamos las diferencias entre las imágenes reconstruidas y los originales y gráficamente también observamos el error en la reconstrucción, que se define de la diferencia entre ambas imágenes. Las diferencias están a la vista.

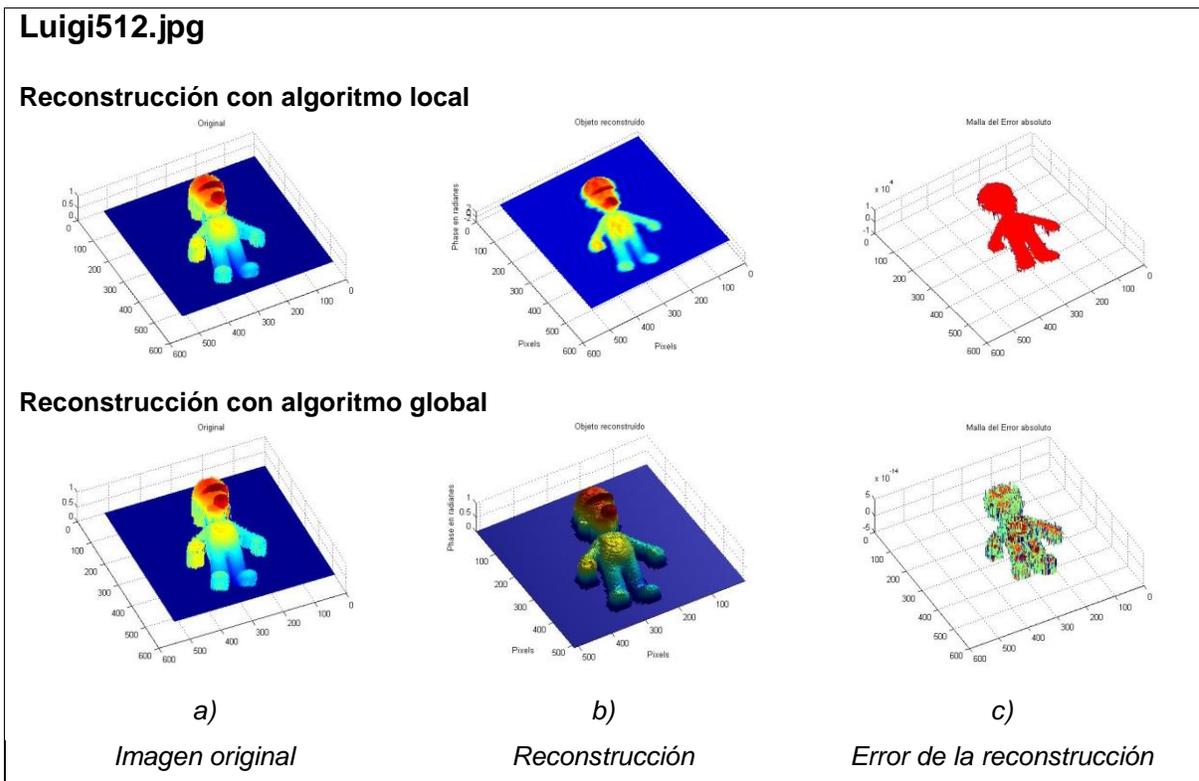


En las imágenes de arriba (PiramideCircular.bmp) vemos que el algoritmo global, favorece la reconstrucción de superficies planas y los bordes con ángulos rectos.





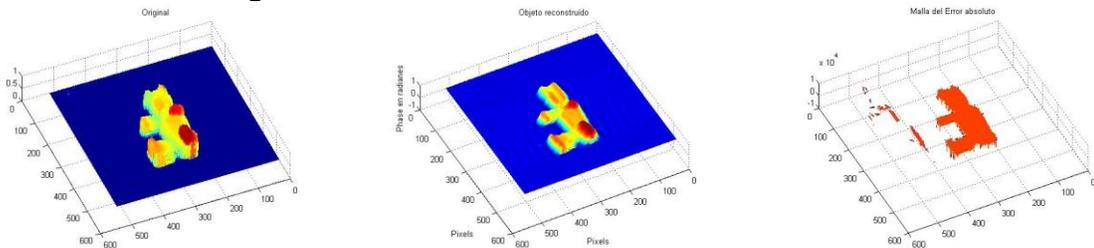
En las imágenes (Homero512.jpg), observamos que el algoritmo global, favorece la reconstrucción de superficies curvas con ángulos suaves. Es destacable la forma en que cada algoritmo interpreta a la imagen adquirida y que aquí se menciona como "imagen original".



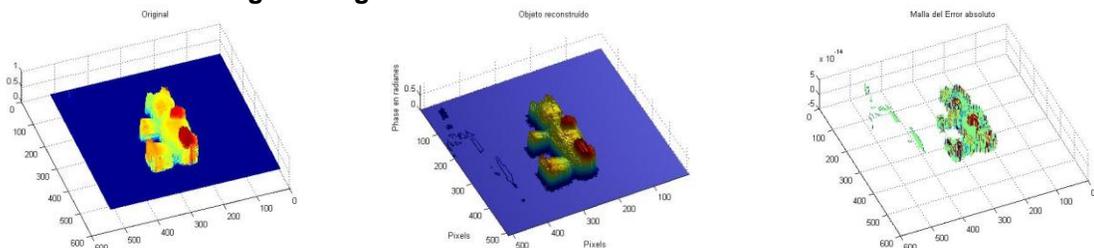
En las imágenes de la reconstrucción de arriba (Luigi512.jpg), el error de la reconstrucción con el método local, está restringida a la base de la imagen, mientras que en el método global los puntos de error cubren un área más significativa.

OsoTed.jpg

Reconstrucción con algoritmo local



Reconstrucción con algoritmo global



a)

Imagen original

b)

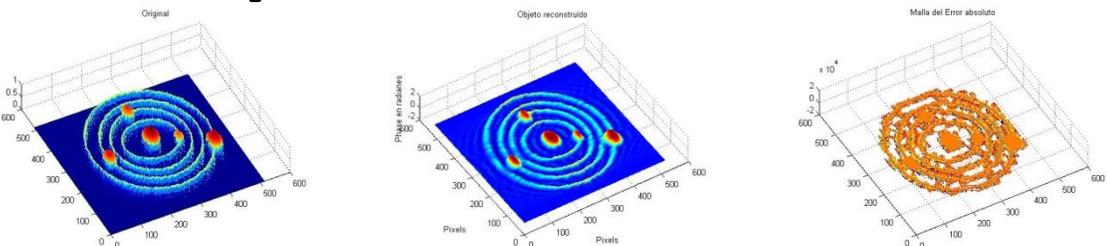
Reconstrucción

c)

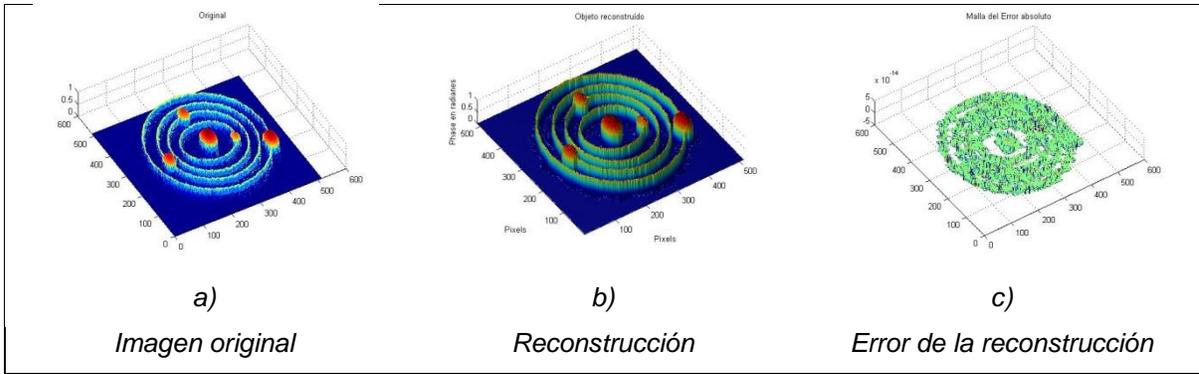
Error de la reconstrucción

SistemaSolar512.jpg

Reconstrucción con algoritmo local



Reconstrucción con algoritmo global



Las dos imágenes restantes (OsoTed.jpg y SistemaSolar512.jpg), son ejemplos de lo comentado en las anteriores reconstrucciones.

5. Conclusiones y trabajo futuro.

Los resultados obtenidos a partir de las reconstrucciones hecha con los dos métodos de desenvolvimiento local y global, muestran diferencias entre cada uno de ellos. Mientras que el algoritmo local consume más tiempo en entregar la imagen del objeto reconstruido, parece ser más efectivo para resolver la reconstrucción de objetos con superficies más suaves y bordes menos pronunciados o en ángulos menos rectos. En cambio, el algoritmos global consume menos tiempo en reconstruir al objeto en la imagen, y es muy preciso en superficies planas y ángulos recto o pronunciados, pero la vista de las superficies que presenta aparenta ser más áspera.

La reconstrucción de objetos en *3D*, como ya se ha dicho, es uno de los problemas más difíciles de resolver en ingeniería y ciencias, este trabajo es solo una pequeña aportación a la investigación de los temas involucrados en la reconstrucción de objetos en *3D* por el método de *perfilometría* con la transformada de Fourier.

Básicamente la metodología empleada se enfoca en obtener la fase envuelta desde una imagen digitaliza del objeto de estudio, procesar la fase envuelta y encontrar la altura del objeto para hacer su reconstrucción a partir del mallado obtenido.

Esta metodología involucra a varios procedimientos y técnicas que empiezan con la adquisición de las imágenes con el patrón de franjas, el análisis de este patrón, *perfilometría* por la transformada rápida de Fourier, obtención y desdoblamiento de la fase y la reconstrucción del objeto de estudio. Consiste básicamente en el procesamiento de la fase envuelta producida por el patrón de franjas y posteriormente, el desenvolvimiento de las discontinuidades en la fase en las frecuencias altas producidas por el ruido en la frecuencia fundamental de la señal [8].

Este método solo puede reconstruir una parte del objeto a la vez (la parte frontal a la cámara y al proyector), por lo que una reconstrucción completa de un objeto, requiere de diversas tomas en diferentes ángulos y posiciones. Unir los resultados de

cada una de estas reconstrucciones para tener un solo resultado en tres dimensiones y comprobar la efectividad de los algoritmos usados puede ser una nueva línea de investigación.

Es necesario insistir en la importancia que tiene el hecho de que estos procedimientos de análisis de patrones de franjas, desdoblamiento de fase y reconstrucción final, se realizan sin contacto físico con el objeto. Esta característica lo convierte en un excelente auxiliar en la reconstrucción de objetos de gran valor, como piezas históricas, u objetos posiblemente irrecuperables y que además pueden ser de casi cualquier material que tenga la solidez suficiente para reflejar la luz de la proyección del patrón de franjas. Otros métodos de reconstrucción, son particularmente sensibles e inestables o incluso inservibles con materiales metálicos.

La calidad de la imagen obtenida, depende fuertemente de la resolución con la que se digitaliza, pero esta resolución en *pixeles*, está íntimamente relacionada con número de franjas que se proyectarán sobre el objeto. Existen otros factores importantes que destacar relacionados con la obtención de las imágenes, por ejemplo, el tipo de luz usado en la iluminación de la escena (en grados K). El contraste entre franjas blancas y negras puede ser determinante en la obtención de la fase.

5.1 Perspectivas futuras.

Los campos para la aplicación de estas metodologías son muchos y pueden ser aún más las áreas de la ciencia, la ingeniería, las artes y otras, que pueden beneficiarse de su uso, esto depende de la eficacia y maduración que alcancen en el futuro.

Cada uno de los procesos que intervienen en el desarrollo de algoritmos de desenvolvimiento de fase para la reconstrucción de imágenes 3D, usando patrones estructurados de franjas en el Método de *Perfiometría* de Fourier, ha sido extensamente estudiado y explicado por un gran número de investigadores de todos

los niveles, se han hecho también, muchas propuestas de solución a cada uno de los procesos que intervienen en este problema en una gran cantidad de publicaciones, reportes técnicos y científicos, además de congresos y otros foros.

Aún así, cada proceso que marca los pasos hacia la reconstrucción de imágenes tridimensionales, a partir del método de *perfilometría* de Fourier, es un área con un gran potencial de investigación y desarrollo. desde la metodología y equipos para adquirir las imágenes. El diseño y proyección del patrón de franjas que involucra el uso de distintas fuentes luminosas, tipos de luz, frecuencias etc. Los algoritmos de desenvolvimiento de fase y reconstrucción de imágenes. Los equipos de cómputo usados para el proceso. Todo lo anterior con idea en la mente de cuál es la finalidad para la reconstrucción de imágenes tridimensionales y que características son deseables para cada aplicación.

6. Referencias bibliográficas.

A. Asundi, Z. Wensen; “*Fast phase-unwrapping algorithm based on a gray-scale mask and flood fill*”. Proceedings of APPLIED OPTICS, Optical Society Of America, vol. 37, No. 23. 1998.

Abdul-Rahman Hussein. “*Three-Dimensional Fourier Fringe Analysis and Phase Unwrapping*”. A thesis submitted in partial fulfilment of the requirements of Liverpool John Moores University for the degree of Doctor of Philosophy. General Engineering Research Institute (GERI), Liverpool John Moores University. pp 6-14, 2007.

Abdul-Rahman, Hussein S., Gdeisat, Munther A., Burton, David R., Lalor, Michael J., Lilley, Francis and Moore, Christopher J. (2007) “*Fast and robust three-dimensional best path phase unwrapping algorithm*”. Applied Optics, 46 (26). pp. 6623-6635. ISSN 0003-935.

D. C. Ghiglia, L. A. Romero. “*Robust two-dimensional weighted and unweighted phase unwrapping that uses fast transforms and iterative methods*” J. Opt. Soc. Am. A 11, 107–117, 1994.

D. C. Ghiglia, M. Pritt. “*Two-dimensional phase unwrapping theory, algorithms and software*” John Wiley & Sons, 1998.

D. Ghiglia, G. Mastin, and L. Romero. “*Cellular automata method for phase unwrapping*”. Journal of the Optical Society of America, 4(1):267–280, 1987.

Dongliang Zheng, Feipeng Da. “*A novel algorithm for branch cut phase unwrapping*”. Optics and Lasers in Engineering. Institute of Automation, Southeast University, Nanjing 210096, China. January 2011.

Esteve Taboada, Juan. "Procesamiento de imágenes por técnicas de multiplexado. Aplicación al reconocimiento de objetos tridimensionales" Tesis doctoral. Universitat de Valencia, Servei de Publicacions, Valencia España, 2002. ISBN 84-370-5560-1.

Gonçalo Ramiro, Valadão Matias. "*Radar interferometry: 2d phase unwrapping via graph cuts*". Dissertação para obtenção do Grau de Mestre. Universidade Técnica de Lisboa, Instituto Superior Técnico. Julho de 2006.

González, R. C, Woods, P. "*Digital Image Processing*", Addison Wesley, 2002.

Heping Zhong, Jinsong Tang, Sen Zhang, and Ming Chen. "An Improved Quality-Guided Phase-Unwrapping Algorithm Based on Priority Queue". IEEE Geoscience and Remote Sensing Letters, VOL. 8, NO. 2, march 2011.

Itoh, Kazuyoshi. "*Analysis of the phase unwrapping algorithm*". Applied Optics, Vol. 21, Issue 14, pp. 2470-2470 (1982).

José M. Bioucas Dias, Gonçalo Valadão. "Phase Unwrapping via Graph Cuts". Instituto de Telecomunicações - Instituto Superior Técnico. Av. Rovisco Pais, 1049-001 Lisboa, Portugal.

Lei Ying, Member, IEEE, Zhi-Pei Liang, Fellow, IEEE, David C. Munson, Jr., Fellow, IEEE, Ralf Koetter, Member, IEEE, and Brendan J. Frey, Member, IEEE. "*Unwrapping of MR Phase Images Using a Markov Random Field Model*". IEEE Transactions on medical imaging, Vol. 25, No. 1, January 2006.

M. Arevalillo Herráez, D. R. Burton, M. J. Lalor and M. A. Gdeisat, "A Fast two-dimensional phase unwrapping algorithm based on sorting by reliability following a non-continuous path" Applied Optics, Vol. 41, No. 35, pp 7437-7444, 2002.

Marklund, Olov, M. Huntley, Jonathan, Cusack, Rhodri. "*Robust unwrapping algorithm for three-dimensional phase volumes of arbitrary shape containing knotted*

phase singularity loops". Society of Photo-Optical Instrumentation Engineers. DOI: 10.1117/1.2771652. Sep. 7, 2007.

Mitsuo Takeda, Hideki Ina, Seiji Kobayashi. *"Fourier-transform method of fringe-pattern analysis for computer-based topography and interferometry"*. University of Electrocommunications, 1-5-1, Chofugaoka, Chofu, Tokyo, 1982, Japan.

Paz Gutiérrez, Jenaro Carlos. "Textos Universitarios" / Serie Docencia. Instituto de Ingeniería y Tecnología, Universidad Autónoma de Ciudad Juárez. Ciudad Juárez, Chihuahua. Serie Docencia II. indd 1 5/8/06. México. 2006.

Pedraza J, Rodríguez W, Gorrostieta E. *"Image Processing for 3D Reconstruction Using a Modified Fourier Transform Profilometry Method"*. MICAI 2007, LNAI 4827, pp. 705-712, 2007.

Pedraza Ortega J.C., Canchola Magdaleno S.L., Gorrostieta Hurtado E., Aceves Fernandez M.A., Ramos Arreguin, J.M., Delgado Rosas. *"Three-Dimensional Reconstruction System based on a Segmentation Algorithms and a Modified Fourier Transform Profilometry"*. M CIDIT, Facultad de Informática, Universidad Autónoma de Querétaro. Querétaro, México.

S. Karout, M A Gdeisat, D R Burton, M J Lalor. *"Comparison between the residue-vector map and existing quality maps for phase unwrapping"* Liverpool John Moores University General Engineering Research Institute, Byrom Street, Liverpool, L3 3AF, UK. Journal of Physics: Conference Series 76 (2007) 012029 doi:10.1088/1742-6596/76/1/012029.

Suprijanto, F. M. Vos, H. M. Vrooman, A. M. Vossepoel. "Quality maps to support phase ungraphing". Pattern Recognition Group. Departament of Applied Physics, Lorentzweg 1. Delft University of Technology, 2628 CJ Delft The Netherlands.

Biomedical Imaging Group Rotterdam. Department of Radiology, Erasmus University Medical Center. P.O. Box. 1738, 3000, Rotterdam, The Netherlands.

Takeda Mitsuo., Mutoh Kazuhiro. "*Fourier transform profilometry for the automatic measurement of 3-D object shapes*" Appl. Opt. 22, 3977-3982, 1983.

Xianyu Su, Wenjing Chen, "*Reliability-guided phase unwrapping algorithm: a review*" Opto-Electronics Department, Sichuan University, Chengdu, 610064, China. November 2003.

Zhang, Li. Liao, Mingsheng. Zhang, Zuxun. Zhang Yong. "*Map and the Coherence Coefficient Residue Guided Least Square Phase Unwrapping Algorithm*". Geo-spatial Information Science. Vol.2,No.1,p.55-62 Dec. 1999.

7. Anexos

7.1 Anexo 1. Códigos.

Código para obtener las imágenes de patrones de franjas, las fases creadas por los patrones, los objetos reconstruidos, la mayas de error y los tiempos de ejecución.

```
clear all;
close all;
clc;

tic

S=256;
x=-S+1:1:S;
[X, Y] = meshgrid(x,x);
aa = size(X);

a=imread('piramide_circular.bmp');
shape=im2double(a);

fo = 1./16;
Beta=1;
fringes_base = cos(2.*pi.*fo.*X);
fringes_object = cos(2.*pi.*fo.*X + Beta*shape);

figure(1);
mesh(shape);
title('Original');
figure(2);
colormap(gray(256));
imagesc(fringes_base);
title('Patrón de franjas');
figure(3);
colormap(gray(256));
imagesc(fringes_object);

a=fringes_object;
a=im2double(a);      % Convierte a doble
imshow(a);          % Muestra la imagen
[ax ay]=size(a);    % Obtiene el tamaño de ax y ay

fondo=zeros(ax, ay); % Llenar de ceros
for i=1:ax % poner unos en ax
    fondo(i,:)=a(5,:);
end

fftw('planner','patient');

fft_al=fft(a(1,:));
fft_fondo=fft(fondo(1,:));
% Aplica la fft a la primera línea "a1"
for m=2:ay
    a1=a(m,:);
    a_calc=fft(a1);
    fft_al=[fft_al;a_calc];

    %fondo
    afondo=fondo(m,:);
```

```

afondo_calc=fft(afondo);
fft_fondo=[fft_fondo;afondo_calc];

end

figure;
mesh(fftshift(abs(fft_a1)));
title('Fase del patrón de franjas sobre el objeto');

% Obtener el tamaño de la imagen original
% para aplicar el filtro
[ax,ay]=size(a);
vent=zeros(ax,ay);

% Se aplica el filtro en la posición
% 209 a 241, donde se encuentra una de las
% frecuencias fo
figure()
plot(fftshift(abs(fft_a1(255,:))));
vent_unos= zeros(ax,ay);
vent_unos(:, 209:241)=1;

% se filtra la señal en fo
sfiltrada=(fftshift(fft_a1).*vent_unos);
% se filtra la señal en fo
sfiltradaFondo=(fftshift(fft_fondo).*vent_unos);

% se muestran las señales filtradas
figure;
mesh((abs(sfiltrada)));
title('Fase del patrón de franjas sobre el objeto');

figure();
mesh((abs(sfiltradaFondo)));
title('Fase del patrón de franjas sobre el plano');

% se regresa el shift aplicado a la señal
sfiltrada=ifftshift(sfiltrada);
% se regresa el shift aplicado a la señal
sfiltradaFondo=ifftshift(sfiltradaFondo);

f2=zeros(ax,ay);
f3=zeros(ax,ay);

for i=1:ay
    f2(i,:)=(ifft(sfiltrada(i,:)));
    f3(i,:)=(ifft(sfiltradaFondo(i,:)));
end

%Fases obtenidas de la transformada inversa
figure, mesh(angle(f2));
title('Fase envuelta del patrón del objeto');
figure, mesh(angle(f3));
title('Fase envuelta del patrón de franjas');
xlabel('Pixels'), ylabel('Pixels'), zlabel('Phase en radianes')

%diferencia de fases

```

```

angle_dif=angle(f3)-angle(f2);

%Desdoblamiento de fase
fin2=unwrap(unwrap(angle_dif));
figure, mesh(fin2);

toc;

% Espejo de la imagen y rotación

flip = flipdim(fin2 ,2);
flip3 = imrotate(flip , 90,'nearest','loose');
figure, mesh(flip3);

title('Objeto reconstruido');
xlabel('Pixels'), ylabel('Pixels'), zlabel('Phase en radianes')

malla_error=shape-flip3;
figure,mesh(malla_error);
title('Malla del Error');

error=(malla_error./shape)*100;
figure,mesh(error);
title('Malla del Error absoluto');

```

Código para desenvolvimiento de fase y reconstrucción 3D en el método local.

```

clear all;
close all;
clc;

tic

S=256;
x=-S+1:1:S;
[X, Y] = meshgrid(x,x);
aa = size(X);

a=imread('piramide_circular.bmp');
shape=im2double(a);

fo = 1./16;
Beta=1;
fringes_base = cos(2.*pi.*fo.*X);
fringes_object = cos(2.*pi.*fo.*X + Beta*shape);

figure(1);
mesh(shape);
title('Original');
figure(2);
colormap(gray(256));
imagesc(fringes_base);
title('Patrón de franjas');
figure(3);
colormap(gray(256));

```

```

imagesc(fringes_object);

a=fringes_object;
a=im2double(a);
[ax ay]=size(a);

fondo=zeros(ax, ay);
for i=1:ax
    fondo(i,:)=a(5,:);
end

fftw('planner','patient');

fft_a1=fft(a(1,:));
fft_fondo=fft(fondo(1,:));

for m=2:ay
    a1=a(m,:);
    a_calc=fft(a1);
    fft_a1=[fft_a1;a_calc];

    afondo=fondo(m,:);
    afondo_calc=fft(afondo);
    fft_fondo=[fft_fondo;afondo_calc];

end

[ax,ay]=size(a);
vent=zeros(ax,ay);

figure()
plot(fftshift(abs(fft_a1(255,:)))));
vent_unos= zeros(ax,ay);
vent_unos(:, 209:241)=1;

sfiltrada=(fftshift(fft_a1).*vent_unos);
sfiltradaFondo=(fftshift(fft_fondo).*vent_unos);

sfiltrada=ifftshift(sfiltrada);
sfiltradaFondo=ifftshift(sfiltradaFondo);

f2=zeros(ax,ay);
f3=zeros(ax,ay);

for i=1:ay
    f2(i,:)=(ifft(sfiltrada(i,:)));
    f3(i,:)=(ifft(sfiltradaFondo(i,:)));
end

angle_dif=angle(f3)-angle(f2);

fin2=unwrap(unwrap(angle_dif'));
figure, mesh(fin2);

toc;

flip = flipdim(fin2 ,2);
flip3 = imrotate(flip , 90,'nearest','loose');
figure, mesh(flip3);

```

```

title('Objeto reconstruido');
xlabel('Pixels'), ylabel('Pixels'), zlabel('Phase en radianes')

malla_error=shape-flip3;
figure,mesh(malla_error);
title('Malla del Error');

error=(malla_error./shape)*100;
figure,mesh(error);
title('Malla del Error absoluto');

```

Código para desenvolvimiento de fase y reconstrucción 3D en el método global.

```

clear all;
close all;
clc;

tic

S=256;
x=-S+1:1:S;
[X, Y] = meshgrid(x,x);
aa = size(X);

a=imread('piramide_circular.bmp');
shape=im2double(a);

fo = 1./16;
Beta=1;
fringes_base = cos(2.*pi.*fo.*X);
fringes_object = cos(2.*pi.*fo.*X + Beta*shape);

figure(1);
mesh(shape);
title('Original');
figure(2);
colormap(gray(256));
imagesc(fringes_base);
title('Patrón de franjas');
figure(3);
colormap(gray(256));
imagesc(fringes_object);

figure, colormap(gray(256)), imagesc(shape)
title('Imagen de fase continua como un arreglo de intensidad visual')
xlabel('Pixels'), ylabel('Pixels')

figure
surf(shape, 'FaceColor', 'interp', 'EdgeColor', 'none', 'FaceLighting', 'phong')
view(-30,30), camlight left, axis tight
title('Fase envuelta del patrón de franjas')
xlabel('Pixels'), ylabel('Pixels'), zlabel('Phase en radianes')

shape_wrapped = atan2(sin(shape), cos(shape));
image2_unwrapped = shape_wrapped;

```

```

figure
surf(image2_unwrapped,'FaceColor','interp','EdgeColor','none',
'FaceLighting','phong')
view(-30,30), camlight left, axis tight
title('Objeto reconstruido')
xlabel('Pixels'), ylabel('Pixels'), zlabel('Phase en radianes')

toc

malla_error=image2_unwrapped-shape;
figure, mesh(malla_error);
title('Malla del Error');

error=(malla_error./shape)*100;
figure, mesh(error);
title('Malla del Error absoluto');

```

7.2 Anexo 2. Glosario.

3D:

Tres dimensiones. En tratamiento y manipulación de imágenes, se refiere a las imágenes que visualmente aparentan tres dimensiones

Desenvolvimiento de fase:

Proceso que consiste básicamente en recuperar la fase envuelta de la señal usando algoritmos en un proceso computacional llamado análisis de patrones de franjas.

Reconstrucción de imágenes:

Técnica para recuperar la información de la altura de un objeto contenida en el mapa de fase de una señal sinusoidal.

Patrón de franjas:

Sombra de una rejilla proyectada sobre de un objeto y la superficie, de la cual se conoce el número de franjas que proyecta con la finalidad de producir un patrón sinusoidal.

Tratamiento de imágenes:

Toda manipulación que modifique las características de una imagen real o digital.

Perfilometría:

Conocimiento de un objeto en sus características de forma y medidas dimensionales

Matlab:

Entorno interactivo con un poderoso lenguaje de alto nivel muy usado para cálculo numérico, visualización y graficado, además de programación

Moiré:

En imágenes, es un efecto no deseado que resulta de patrones intercalados de luz y sombra alternadas, provocando ruido e interferencia.