

INTRODUCCION

La visión robótica hoy en día es uno de los temas que más ha llamado la atención alrededor del mundo, dado que es una aplicación con mayor interés industrial, en lo que abarca lo que es visión, se da la importancia a la óptica que es una ciencia que estudia la luz, como se observara más adelante en el desarrollo del proyecto, uno de los temas importantes para el desarrollo del trabajo es la luz estructura, ya que es un método para emplear la visión robótica usando la técnica de Moiré (conocido también como *moaré*) esta técnica ayuda en el procesamiento y análisis de imágenes, donde se aplicara proyección de franjas lo que son comúnmente conocidas como rejillas, para que después se puedan modular y desenvolver la imagen, ya que esta es la base para el proceso de este proyecto.

Además verán el desarrollo matemático de la implementación de la técnica de Moiré ya que es muy importante para su aplicación en este trabajo, podrán observar el resultado obtenido en cada método que se implementó y los detalles de este mismo, ya que para el comienzo de este trabajo se empezó bajo una programación en el software MathCad para poder ver las franjas desde una ecuación matemática dándole los respectivos valores para poder observar la imagen generada por esta misma, y observar algunos resultados obtenidos en la modulación de imágenes generadas de la misma programación y otras extraídas desde la PC que se trabajó, como también el proceso de cómo se lleva a cabo la captura de imágenes que son el posicionamiento de la cámara CCD y el proyector.

Y para concluir con el trabajo, se llega al último paso que es el desenvolvimiento de fase ya que esta es una de las partes más difíciles de todo el proceso a elaborar ya que analice los resultados si son aceptables en el desenvolvimiento y esta se trabajara con la ayuda de Matlab, ya que el algoritmo es más fácil trabajarlo con él.

JUSTIFICACIÓN

Con el propósito de tener un algoritmo adecuado para obtener imágenes en 3D se planteó este proyecto de investigación con el fin de poder usarlo para fines prácticos en la cual permita el uso más fácil de lo que es el proceso de proyección de franjas por el método de Fourier, ya que esto lleva a usar un software propio y no uno con el cual genere un gran costo de compra del software y licencia, para ello se elaboró este trabajo con la certeza de reducir costos y el uso fácil del mismo. El algoritmo está elaborado por medio de métodos matemáticos con el cual se trabajó teóricamente pero basándose en el uso de lenguaje de programación matemático.

OBJETIVOS

OBJETIVO GENERAL

Diseñar el algoritmo para la reconstrucción de imágenes en 3D para el estudio de objetos por el método de luz estructurada.

OBJETIVO ESPECIFICO

Para llevar a cabo el diseño del algoritmo se tomaron en cuenta los siguientes puntos:

- Se generan franjas por medio de una programación en MathCad.
- Se trabaja con figuras geométricas para tomar la fase envuelta.
- Sacamos la fase desenvuelta para obtener una Imagen en 3D.

Capítulo 1.- MARCO TEÓRICO

1.1.- *Visión Robótica Usando Luz Estructurada*

Para poder comenzar con el desarrollo de este trabajo de tesis se debe saber primeramente que es la Visión Robótica y que es la Luz Estructurada son dos cosas muy distintas pero tan importantes para este trabajo a realizar.

La *Visión* su significado común de esta, es la capacidad de poder interpretar el entorno gracias a los rayos de la luz que alcanza el ojo. La *Robótica* es una rama de la tecnología que es una ciencia de investigación, estudio o tecnología de los robots, la robótica combina diversas disciplinas como son: la mecánica, electrónica, informática, inteligencia artificial e ingeniería de control entre otras. Posteriormente se puede definir a la *Visión Robótica* como el fin de extraer propiedades del entorno a partir de un conjunto de imágenes, ejemplos; para la detección de imágenes digitales, y el análisis y procesamiento de imagen.

¿Qué es la *Luz*? Es una onda electromagnética compuesta por fotones (partículas energizadas), cuya frecuencia y energía determinan la longitud de onda de un color que puede ser percibido por el ojo humano. Ahora al hablar de la palabra *Estructurada* la referimos hacia una persona o cosa que cuenta con una estructura y que por lo tanto tiene firmeza o donde apoyarse, ya se ha algo simbólico como concreto. Entonces se define a la *Luz Estructurada* como el estudio de las deformaciones que sufre un patrón de luz al ser contacto con cualquier objeto, ya que se necesita concentrar en un punto la luz en este caso sobre el objeto a estudiar o analizar. El tipo de luz que se emplea sobre los objetos por lo regular son luz de laser o proyectores de luz especiales, ya que el propósito de esto es de que la luz no se propague o extienda, sino que se enfoque en un solo punto por ello no se puede utilizar cualquier tipo de luz ya que hay que tomar en cuenta que otros tipos de fuente de luz están compuesto por ondas de diferentes frecuencias y como había mencionado antes provoca que el haz de luz se difumine sobre el entorno.

Una vez que se tiene el tipo de luz a emplear, será necesario elegir un patrón adecuado. Para comenzar a trabajar, utilizare la ecuación del libro *Optical Metrology* [1], del cual base para el estudio de las proyección de franjas, que en este caso como ya había mencionado desde el principio de este trabajo, es de llevar acabo la técnica de Moiré (moaré), para ello hay que saber primero que es la técnica de moiré o que significa Moiré.

1.2.- ¿QUE ES MOIRE?

La palabra *Moiré* no es el nombre de una persona, de hecho es una palabra francesa que tiene como referencia a un “tipo particular de textil en seda y que posee una apariencia ondulante o fluctuante, gracias a los patrones de interferencia formados por la estructura misma del tejido”. En Óptica se refiere a un patrón producido entre dos rejillas de espaciamiento aproximadamente igual. Este efecto se puede observar en las cosas cotidianas, como en la preselección de una imagen de medio tono, o con una camisa de rayas visto en televisión. El uso de *moiré* para las pruebas de sensibilidad reducida fue presentado por Lord Rayleigh en 1874. [2].

Lord Rayleigh miró el Moiré entre dos rejillas idénticas para determinar su calidad a pesar de que cada red individual no se puede resolver con un microscopio.

La proyección de franjas supone proyectar un patrón de franjas o rejillas sobre un objeto y verlo desde una dirección diferente. El primer uso de la proyección de franjas para determinar la topografía de la superficie fue presentado por Rowe y Welford en 1967.

Es una Técnica apropiada para contornear objetos que son demasiado gruesos para ser medidos con interferómetros estándar. La proyección de franjas se relaciona con la triangulación óptica con un único punto de luz y la luz seccionamiento en una sola línea que se proyecta sobre un objeto y poder verlo en una sola dirección diferente para determinar el contorno de la superficie. [3].

En la figura 1.1 se observa un patrón de franjas el cual como se observa las franjas tiene la misma distancia de separación una con respecto a la otra, si el ángulo entre las rejillas se incrementa, la separación entre las dos franjas brillantes y oscura disminuye. Y si las franjas no son idénticas (franjas brillantes y oscuras) la función al cual se aplicara no servirá para el proceso del estudio o análisis del objeto.

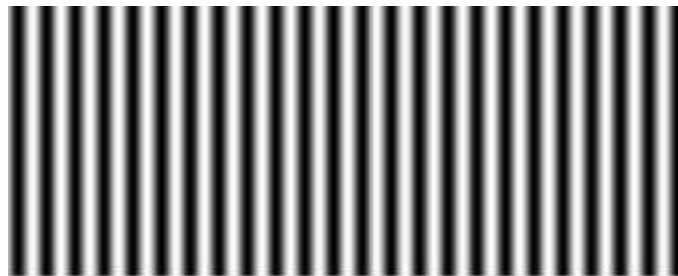


Fig. 1.1. Patrón de franjas generado en MathCad.

1.3.- PROYECCION DE FRANJAS

En la actualidad el método de reconstrucción por franjas consta de un sistema de proyección, un sistema de observación y un plano de referencia, donde se focaliza el sistema de observación, ubicado en la coordenada $Z_0 = 0$ del sistema coordenado (X_0, Y_0, Z_0) como se puede observar en la figura 1.2. Cuando se proyectan franjas sobre el plano de referencia, las franjas se ven paralelas entre sí.

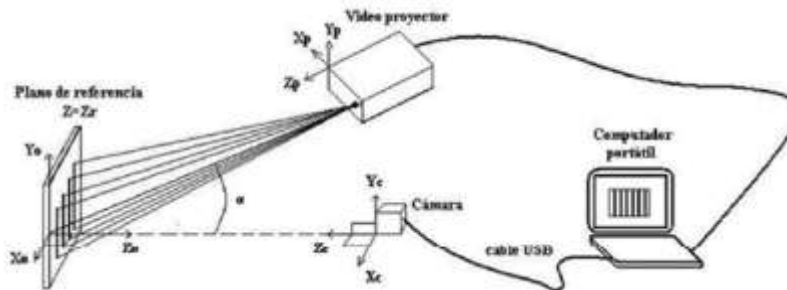


Fig. 1.2. Sistema de proyección por franjas.

Un patrón de franjas es sinusoidal, es proyectado sobre un plano de referencia y posteriormente sobre el objeto de estudio; ambas imágenes son almacenadas en la memoria de una computadora. La valoración de las diferencias de estos registros constituye el núcleo de la técnica de proyección de franjas, y en la actualidad se realiza por medios completamente digital lo que le brinda gran versatilidad y velocidad de desempeño.

Si las franjas proyectadas son cosenoidales, el desplazamiento del patrón de franjas se refleja en un cambio en el argumento de la función coseno, es decir, produce un cambio en fase que llamaremos $\psi(x)$. Este cambio de fase se puede relacionar con la distancia entre el plano de referencia y la superficie de interés del objeto z . El valor de z para cada uno de valores de x se puede hallar por medio de la evaluación de la fase $\psi(x)$ del sistema de franjas deformado por el punto x de interés como se observa en la ecuación (1.1).

$$Z = \frac{d_x}{2\pi \tan \theta} \psi(x) \quad (1.1)$$

De acuerdo con la ecuación anterior el mapa de alturas z para las distintas posiciones del objeto está completamente determinado si se conoce el valor de la fase $\psi(x)$ y esta se puede obtener por medio de diversas metodologías: los métodos de la transformada de Fourier o cualquiera de las técnicas de corrimiento de fase que figuran las más eficientes y ampliamente utilizadas. Una vez que se tiene la fase $\psi(x)$, se obtiene la altura z para cada x y de esta forma se reproduce la superficie del objeto en estudio.

1.4.- TRANSFORMADA DE FOURIER

La transformada de Fourier convierte coordenadas espaciales en frecuencias. Cualquier curva o superficie se puede expresar como la suma de un cierto número (quizá infinito) de curvas del seno y del coseno. En el dominio de Fourier (llamado el dominio de la frecuencia) la imagen se representa con los parámetros de las funciones del seno y del coseno. La transformada de Fourier es el mecanismo matemático para moverse dentro y fuera del dominio de la frecuencia.

El dominio de la frecuencia se llama así porque los dos parámetros de una curva del seno son la amplitud y la frecuencia. El hecho de que una imagen se pueda convertir en una representación del dominio de la frecuencia implica que la imagen puede contener información de alta frecuencia o de baja frecuencia. Si el nivel gris de una cierta porción de la imagen cambia lentamente a través de las columnas, entonces sería representado en el dominio de la frecuencia como función del seno o del coseno teniendo una frecuencia baja. Una cierta cosa que cambia rápidamente, por ejemplo un borde, tendrá componentes de alta frecuencia.

Es por lo tanto posible construir los filtros que quitarán o realzarán las frecuencias en la imagen, y éstas tendrán a veces un efecto reconstituyente.

De hecho, el ruido consiste principalmente en información de alta frecuencia, y así el filtrado de alta frecuencia debe tener un efecto de reducción del nivel de ruidos. Desafortunadamente también tiene un efecto de la reducción del borde.

La transformada de Fourier divide una imagen (en una dimensión, una señal) en un conjunto de componentes del seno y del coseno. Es importante mantener estas componentes separadas, y así un vector de la forma (coseno, seno) es utilizado en cada punto del dominio de la frecuencia de la imagen; es decir, los valores de los píxeles en el dominio de la frecuencia de la imagen son vectores componentes. Una manera conveniente de representarlos es mediante números complejos.[6].

Cada número complejo tiene una parte real y una parte imaginaria, se puede interpretar tal cual como un vector. Un número complejo típico podría ser escrito como se muestra en la ecuación 1.2.

$$z = (x, jy) = x + jy \quad (1.2)$$

Donde j es imaginaria. El exponencial de un imaginario se puede representar como la suma de un seno y un coseno, que es exactamente como se muestra en la ecuación 1.3.

$$e^{j\theta} = \cos\theta + j\sin\theta \quad (1.3)$$

Esta forma polar es la que se usará a partir de ahora, pero es importante recordar que realmente es la suma del seno y del coseno.

En una dimensión, la transformada de Fourier de $f(x)$ de una función continua es representada en la ecuación 1.4.

$$F(w) = \int_{t=0}^{\infty} f(t)e^{-j\omega t} dt \quad (1.4)$$

Si se ha muestreado la función de modo que ahora es discreta, la integral se convierte en la suma de los puntos muestreados como en la ecuación 1.5.

$$F(w) = \sum_{k=0}^{N-1} f(k)e^{-\frac{2\pi j\omega k}{N}} \quad (1.5)$$

Si la función $f(k)$ es una curva muestreada del seno, entonces la transformada de Fourier $F(w)$ debe producir un solo punto que muestra los parámetros de la curva.

1.5.- TRANSFORMADA INVERSA DE FOURIER

La inversa de la transformada de Fourier deshará la transformación; cuando se da la transformada de Fourier de un sistema de datos, lo contrario de la transformada reconstruirá los datos originales. La función logarítmica y la función de exponencial tienen una relación similar, donde una deshace la otra.

La fórmula para el cálculo de la transformada discreta de Fourier se muestra en la ecuación 1.6.

$$f(k) = \frac{1}{N} \sum_{w=0}^{N-1} F(w) e^{\frac{2\pi jwk}{N}} \quad (1.6)$$

Que se diferencia de la transformada directa en el signo del exponente.

Ahora puede ser el momento de precisar que el factor constante $1/n$ es algo flexible. Algunos lo aplican a la transformada directa, otros lo parten entre la directa y la transformada inversa y otros lo aplican solamente a la transformada inversa. Desde luego, el algoritmo rápido puede aplicarse a la transformada inversa, así un sistema de datos puede ser transformado tal cual fácilmente en cualquier dirección. En las rutinas de la librería de `fftlb` en lenguaje `C`, la función inversa del 1D *FFT* es la misma que la función directa de *FFT*; hay un parámetro que especificará una dirección directa o inversa.

La Transformada de Fourier de una imagen f es calcular primero la transformada de Fourier de cada fila, dando una imagen f' . Entonces se calcula la transformada de Fourier de cada columna de f' , dando F , la transformada de la imagen. Esto permite que utilice los métodos de cálculo de la *FFT* de una dimensión discutidos ya para calcular la transformada de 2 dimensiones.

Además del uso de la transformada de Fourier para permitir la filtración de ciertas frecuencias, hay otras propiedades útiles que llegan a ser absolutamente importantes al discutir sobre datos de dos dimensiones. Lo más importante desde el punto de vista del procesamiento de una imagen es que la convolución es mucho más fácil de hacer en el dominio de la frecuencia que en el dominio espacial. En verdad, una convolución es simplemente el producto de un elemento por un elemento de la transformada de Fourier de las dos imágenes que son convolucionadas. [7].

1.6.- METODO DE LA TRANSFORMADA DE FOURIER

La imagen de una rejilla deformada por una ecuación puede ser interpretada como múltiples señales moduladas en fase $\phi(x,y)$, en una amplitud $r(x,y)$ y con frecuencia espacial portadora. Basado en que la fase posee información acerca de la forma tridimensional a medir, el problema es como separar $\phi(x,y)$ del resto de la información.

Se usara la notación utilizada en diferentes literaturas, que es una representación en forma exponencial y no en forma de expansión de series de Fourier. El propósito de usar esta notación, es el de dar una explicación más clara en los procedimientos del método de la Transformada de Fourier, para ello se muestra la ecuación 1.7.

$$g(x,y) = a(x,y) + b(x,y)\cos[2\pi f_0 + \phi(x,y)] \quad (1.7)$$

En donde para respetar las variables propuestas por los textos, $a(x,y)$ es la distribución no uniforme de la reflectividad sobre la superficie del objeto. $b(x,y)$, $\phi(x,y)$ y f_0 , la intensidad, la fase y la frecuencia espacial de la malla (patrón de líneas deformadas) respectivamente. Ahora convirtiendo exponencialmente se tiene la ecuación 1.8.

$$g(x,y) = a(x,y) + c(x,y)e^{2\pi f_0 x} + c(x,y)e^{-2\pi f_0 x} \quad (1.8)$$

Usando el algoritmo de *FFT* se calcula la transformada unidimensional de Fourier de la ecuación anterior para cada variable de x únicamente, con y manteniéndose fija obtenemos la ecuación 1.9.

$$G(x,y) = A(f,y) + C(f + f_0,y) + C(f - f_0,y) \quad (1.9)$$

En donde las letras mayúsculas representan el espectro de Fourier y f representa la frecuencia espacial en la dirección de x y y siendo tratada como una variable fija. Si asumo que $a(x,y)$, $b(x,y)$ y $\phi(x,y)$ tienen frecuencias mucho menores que f_0 , entonces la ecuación anterior se observa en la figura 1.3.

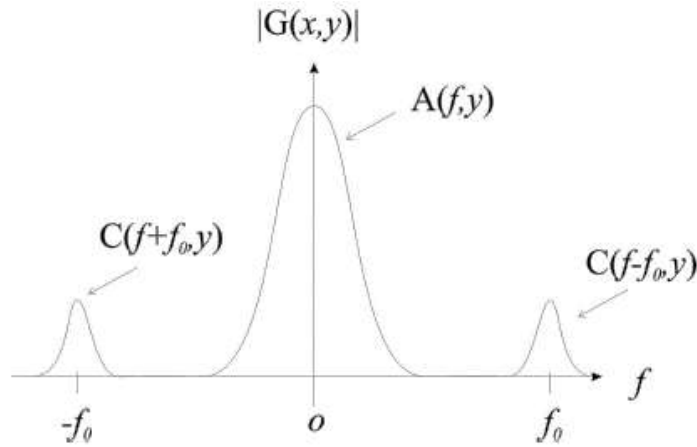


Figura 1.3. Espectro espacial de Fourier para un valor de y fijo.

Para poder separar el índice de reflectividad $a(x,y)$ se necesita pasar la señal por un filtro pasa-bandas el cual debe de estar centrado en f_0 . Para realizar lo anterior es posible utilizar un filtro FIR, FII o un filtro ideal (mascara) ya que la información no se esta procesando en tiempo real.

La componente que se va a obtener filtrándola de las demás es la primera componente armónica. Después de dicho filtrado el espectro espacial de Fourier de la señal queda como se muestra en la figura 1.4.

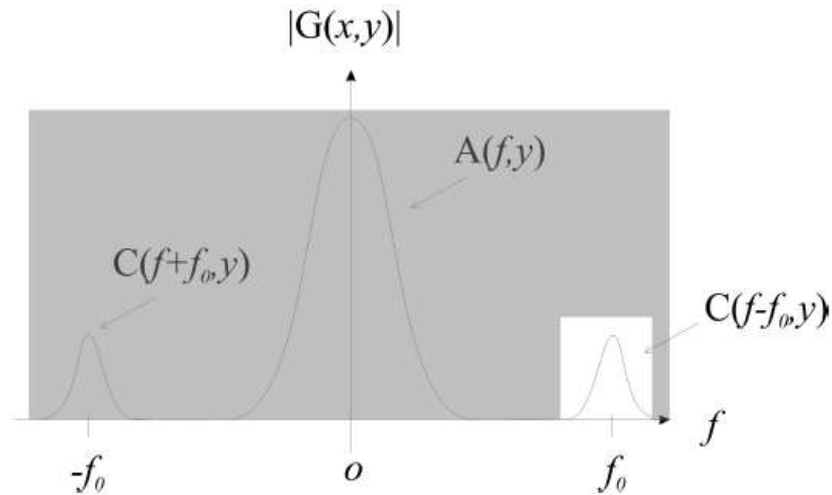


Figura 1.4. Espectro espacial de Fourier la cual contiene la información de fase.

Después de filtrar la señal, esta sea trasladada al origen para que $C(f + f_0, y)$ se convierta en $C(f, y)$ esto se logra a partir de una sencilla operación basada en un teorema de la transformada de Fourier discreta como se observa en la ecuación 1.10.

$$e^{i\omega_0 n} x[n] \leftrightarrow X(e^{i(\omega-\omega_0)}) \quad (1.10)$$

Entonces para hacer el dicho corrimiento se tiene que obtener primero la FFT inversa de la señal, una vez mas, esta transformada se realiza en una dimensión, es decir, avanza en x mientras que en y se mantiene como un parámetro constante. Después de lo anterior, se hace la multiplicación por el exponencial para poder enviar la señal al origen en el espectro espacial de Fourier mostrada en la ecuación 1.11.

$$\hat{g}(x, y) = \hat{g}'(x, y)e^{i\omega_0 n} = \frac{1}{2}b(x, y)e^{i\phi(x, y)} \quad (1.11)$$

En donde $\hat{g}(x, y)$ es la transformada inversa de Fourier de la información ya filtrada.

En el caso de los ejes ópticos cruzados, se realizan los mismos pasos para la imagen que solo contiene el fondo (plano de referencia) y así se obtiene la ecuación 1.12.

$$\hat{g}_0(x, y) = \hat{g}'_0(x, y)e^{i\omega_0 n} = \frac{1}{2}b(x, y)e^{i\phi(x, y)} \quad (1.12)$$

La información concerniente a las franjas, pero de esta información la única que interesa es la fase ya que contiene los datos de la profundidad del objeto. Al igual que en otros pasos, se proponen métodos para la separación n , que en realidad son equivalentes, uno de los métodos es el de por logaritmos complejos y el otro por arco-tangente. Ambos proveen resultados idénticos y además son igual de sencillos de implementar.

1.7.- DESENVOLVIMIENTO DE FASE

El desenvolvimiento de fase es un proceso computacional en donde una superficie llamada ϕ es reconstruida de su llamada forma envuelta ψ . En la ausencia de ruido $\psi(x)$ es igual $\phi(x) + 2\pi k(x)$, en donde $k(x)$ es una función entera tal que $-\pi < \psi \leq \pi$. El problema del desenvolvimiento de fase ha ganado considerablemente la atención en los últimos años. Es una parte esencial de muchas aplicaciones de procesamiento de señales coherentes como por ejemplo Radar Interferómetro de Apertura Sintética (ISAR siglas en inglés "*Interferometric Synthetic Aperture Radar*"), Interferometría Óptica y Perfilometría por Transformada de Fourier. Sin embargo, los valores reales de fase no pueden ser extraídos directamente de la señal física debido a que dichos algoritmos modifican la señal de forma en que sus valores se encuentran en un rango de $\pm\pi$ radianes. Lo único que se puede obtener de esos métodos es la fase envuelta.[8].

Como se mencionó anteriormente, la fase se encuentra limitada a valores entre $-\pi$ y π , pero cuando esta exceda el valor, la fase brincara a $-\pi$ y π . Por cada salto negativo de 2π se necesitara añadir π , mientras que por cada salto positivo se necesitaran restar otros 2π .

El proceso de desenvolvimiento intenta proveer una estimación de la función de fase real ϕ dada la función ψ , algo que se convierte en un problema muy difícil [6]. Para empezar, la relación entre ϕ y ψ es no lineal. Además, las aplicaciones interferométricas o las de triangulación activa procesan información del mundo real, por lo tanto, el ruido en la fase medida es inevitable y eso complica aún más la ya complicada tarea. Además otro de los problemas que se presentan es el de poder distinguir entre envolturas de fase genuinas producidas por la función arco-tangente y saltos en la fase causados por discontinuidades del objeto.

De una manera más general, el algoritmo básico de desenvolvimiento de fase 1D es como sigue: Se inicia con el primer punto, y se compara con el segundo, si la diferencia entre el primero es igual o mayor a 2π entonces se le suma o se le resta dependiendo del signo de la primera diferencia 2π a todos los puntos a partir del segundo, y se repite dicha operación hasta terminar con la línea de información. Si no existe salto se continúa hasta encontrar un salto y se realiza el desenvolvimiento. [5].

En el caso de estar trabajando con mapa de fase, el desenvolvimiento en 1D no es suficiente, y en la mayoría de las veces es incorrecto, así que se tiene que encontrar algún método que trabaje en 2D. Dado el desarrollo de aplicaciones que trabajan con mapas de fase como lo son algunos microscopios electrónicos y los sistemas de radar de apertura sintética, se han creado muchos algoritmos bidimensionales de desenvolvimientos de fase tanto complejos como simples.

Como se ha comentado anteriormente sobre lo que es el desenvolvimiento de fase este también tiene aplicaciones militares, en medicina e industriales entre otras, sobre el procesamiento digital de imágenes, que como parte de su operación fundamental depende de la extracción de una señal de fase de la imagen de entrada.

Desenvolver fases es una de las tareas más activas de la investigación en procesamiento digital de imágenes y ha sido ampliamente investigado. Como bien se decía hay muchos algoritmos propuestos para el desenvolvimiento de fase pero pueden producir una variedad de resultados diferentes y no una perfecta solución que garantice un buen resultado, que en algunos casos puede no existir la solución.

Capítulo 2.- CARACTERIZACIÓN DEL AREA DE TRABAJO

2.1 Donde se Realizó el Proyecto

El *Centro de Investigaciones en Óptica* A.C. se encuentra en: Loma del Bosque # 115, Col. Lomas del Campestre de la ciudad de León Guanajuato. Es un centro público cuyo propósito es desarrollar investigación básica y aplicada en el campo de la óptica, que contribuya a los esfuerzos de la comunidad científica para impulsar las fronteras del conocimiento y su aplicación en el sector productivo y social del país, además de la formación de recursos humanos de alto nivel, el desarrollo y la transferencia de tecnología, y la contribución de una cultura científica y tecnológica en la sociedad mexicana.

En el centro hay dos grandes divisiones de investigación que es la Óptica y la Fotonica. En la División Óptica (DIVO), se encuentra las áreas de investigaciones de pruebas ópticas no destructivas, holografía y materiales fotosensibles, Interferometría óptica, visión computacional e inteligencia artificial, óptica médica, instrumentación, infrarrojo entre otras. En la División Fotonica (DIVF) se encuentran las siguientes áreas de investigación: comunicación óptica, óptica integrada, óptica ultrarrápida, espectroscopia, dispositivos de fibra óptica, sensores de fibra óptica entre otras. En la División Óptica es donde está la base de mi proyecto de investigación en el área de Interferometría óptica en la cual trabaje para el estudio y análisis de las proyecciones de franjas en el laboratorio de Interferometría óptica.

En el laboratorio de Interferometría es un espacio dedicado a procesar imágenes, desarrollo de software y sistemas de visión computacional (prototipos tecnológicos que involucran óptica, mecánica, electrónica y computación).

Las actividades en el laboratorio son respaldadas por personal científico y técnico de alto nivel de conocimiento y experiencia en procesamiento de imágenes, estas involucran diferentes campos del conocimiento (óptica, matemáticas, mecánica, electrónica y computación). Las actividades comunes desarrolladas en el laboratorio son: desarrollo de software, técnicas y prototipos de reconstrucción tridimensional, reconocimiento de patrones, metrología óptica, visión computacional, desarrollo y empleo de algoritmos de inteligencia artificial, redes neuronales, algoritmos genéticos, procesamiento digital de imágenes, criptografía visual, morfología matemática entre otras.

El laboratorio está equipado con equipo especializado para la captura y procesamiento de imágenes (computadoras, cámaras CCD, tarjetas digitalizadoras, accesorios ópticos, fuentes de iluminación entre otros.)

Capítulo 3.- DESARROLLO DEL ALGORITMO

3.1 GENERAR Y MODULAR FRANJAS

A continuación se muestra en el siguiente diagrama de flujo las etapas en las cuales se trabajaron durante todo el proceso de la realización del proyecto, donde se observa que el primer paso es como generar franjas esta se realiza por medio de una ecuación matemática. Teniendo las franjas capturadas en una imagen, se utiliza el método de Fourier. El método de Fourier genera como resultado la fase envuelta de la imagen. Después se hacen pruebas con figuras geométricas para analizar los resultados obtenidos si son favorables. Una vez conociendo que el resultado de la fase envuelta, se diseña el algoritmo para obtener la fase desenvuelta de la imagen y tener como resultado una imagen en 3D, como se muestra en el siguiente figura 3.

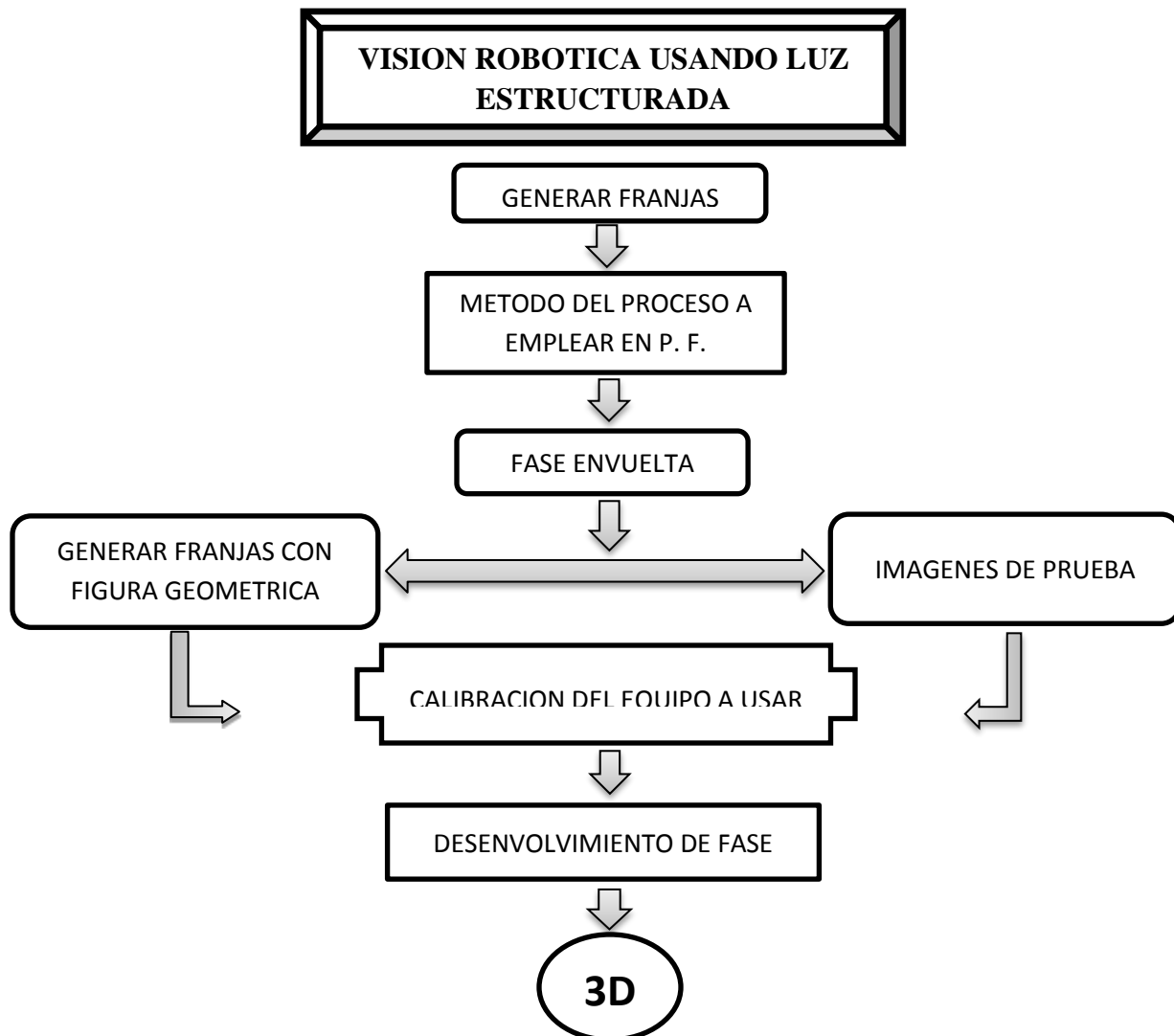


Figura 3. Diagrama de flujo del método de franjas.

Ahora se emplea el desarrollo del trabajo, como ya se explicó desde un principio se comenzó a trabajar con la ecuación 3.1 sobre la *técnica de Moiré* [1], para genera franjas para el uso de reconstrucción de imágenes en 3D.

$$t_1 = a \left\{ 1 + \cos \left[2\pi \frac{x}{d_x} + \psi_1(x) \right] \right\} \quad (3.1)$$

Donde a es la amplitud y $\psi(x)$ es la fase, donde esta ecuación representa unas franjas tipo cosenoidales, en la cual al programarlo y proporcionales los valores apropiados esta genera una imagen con franjas.

Para poder programar la ecuación 3.1 se le proporciona un valor para la amplitud a la imagen a generar, se sabe que al obtener las franjas esta la genera en negro (0) y blanco (255) en valor binario. Entonces como se mostrara la imagen en un plano x - y le proporcione valores para genere la imagen a cierta amplitud o rango, en el cual se programó un ciclo para ir incrementando hasta llegar a su valor máximo:

$$\begin{aligned} x &:= 0..255 \\ y &:= 0..255 \end{aligned}$$

De esta forma se dice que tanto x y y tendrán valores en incremento de 0 hasta 255 que será la amplitud máxima. Ahora se programa la ecuación 3.1 con sus respectivos valores como se muestra a continuación:

$$t_{x,y} := 128 + 127 \cdot \cos \left(\frac{40\pi y}{255} + r2_{x,y} \right)$$

Donde $r2 =$ es la fase $\psi(x, y)$, y 128 es el valor dado de la amplitud y y es el sentido de las franjas que en este caso empezara sobre el eje de las y . Teniendo el primer paso del algoritmo para generar franjas, se imprime en pantalla el resultado, pero a la fase se le asigna un valor de cero. De esta forma queda la programación y genera la imagen de la figura 3.2.

$$x := 0..255 \qquad y := 0..255$$

$$r2_{x,y} := 0$$

$$t_{x,y} := 128 + 127 \cdot \cos \left(40\pi \frac{y}{255} + r2_{x,y} \right)$$

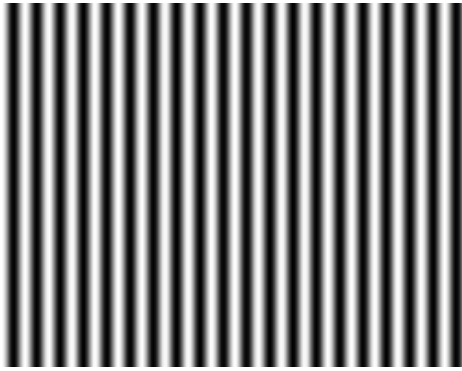


Figura 3.2. Franjas generadas desde MathCad.

Como se puede observar las franjas blancas y negras son las ondas cosenoidales vista desde la parte superior de la onda, como a continuación se explica:

A la representación de la oscilación cosenoidal en la forma de patrón de franjas, de la ecuación 3.1, se le considerara como una representación binaria. Ya que en vez de tomar todos lo valores posibles de $-A$ y A (amplitud), solo se considera los valores de uno o cero.

Es decir que el patrón geométrico de la figura 3.3 b) puede considerarse como una visualización simplificada de la onda de la figura 3.3 a).

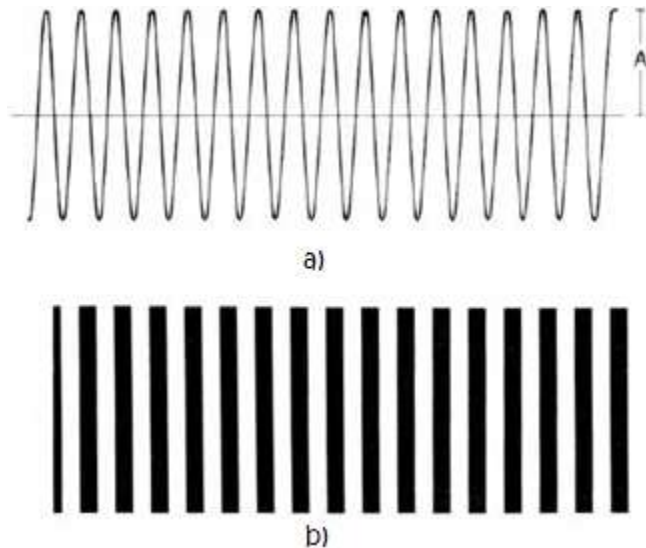


Figura 3.3. a) Onda cosenoidal, b) patrón geométrico asociado.

El valor de 40 se obtiene de multiplicar el 20 por 2π ya que el valor de 20 es el número de franjas a generar en la imagen, como se observar en la figura 3.2.

Ahora se le extrae la fase envuelta a la figura, para ello se le asignó un valor a la fase.

Como utilice una amplitud de 255 entre los ejes x y y adquirí el valor de 127.5 que es la mitad de 255 para generar una ondulación a la figura, entonces la programación es la siguiente:

$$\begin{aligned}x &:= 0..255 & y &:= 0..255 \\ r^2_{x,y} &:= (x - 127.5)^2 + (y - 127.5)^2 \\ t_{x,y} &:= 128 + 127 \cdot \cos \left[\frac{40\pi y}{255} + 0.001(r^2_{x,y}) \right]\end{aligned}$$

Para poder ver las franjas onduladas por la fase la multiplique la fase por un valor pequeño y de esta forma se podrá apreciar las franjas como se muestra en la figura 3.4.

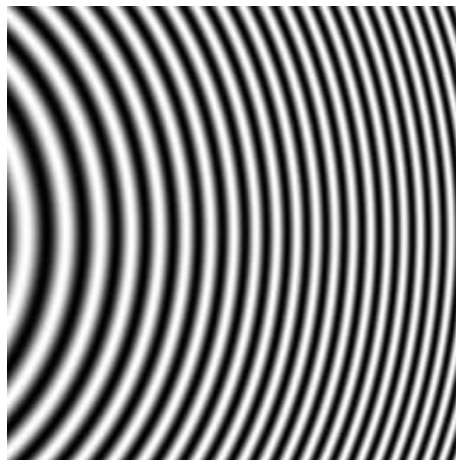


Figura 3.4. Franjas onduladas con su respectiva fase.

Como se observa en la figura al multiplicar por un valor pequeño a la fase que en este caso es $r^2(x,y)$ esta genera las franjas rectas a pesar de ir disminuyendo demasiado el valor esta llegara en un punto en el cual ya no pueda observar la diferencia de la rectitud de las franjas y si se aumenta el valor por el cual está siendo multiplicado la fase esta genera el resultado de la imagen muy pequeña en la cual no se podrá distinguir las franjas debido a las interferencias que se generara por el valor que es muy grande que sería arriba de uno.

Ahora como ya se generó franjas de la programación, el siguiente paso es de pasar la imagen a su fase envuelta o corrimiento de fase, para ello se forma una esfera con la cual se modificaran varios aspectos de las franjas y eliminar de la imagen la información no a utilizar. El resultado de esta programación se observa en la figura 3.5.

$$x := 0..255 \qquad y := 0..255$$

$$r_{x,y} := \sqrt{(x - 127.5)^2 + (y - 127.5)^2}$$

$$\text{Mask}_{x,y} := \begin{cases} 1 & \text{if } r_{x,y} < 120 \\ 0 & \text{otherwise} \end{cases}$$

$$\psi_{x,y} := 0.002(r_{x,y})^2 \cdot \text{Mask}_{x,y}$$

$$t_{x,y} := \begin{cases} (128 + 127 \cdot \cos(0.5\pi y + \psi_{x,y})) & \text{if } r_{x,y} < 120 \\ 0 & \text{otherwise} \end{cases}$$

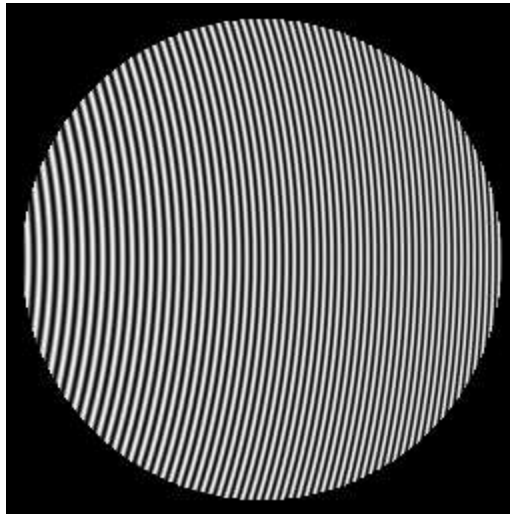


Figura 3.5. Esfera con franjas y con su máscara (fondo negro).

Donde $r(x,y)$ es el valor de la posición de la esfera, y $\psi(x,y)$ es la fase donde multiplica por $\text{Mask}(x,y)$, esta es la que va a generar la esfera y que se pueda observar claramente las franjas siempre y cuando esta condición se cumpla cuando sea menor que 120. Y posteriormente está dentro de un comparador, que si se llega a cumplir la condición lo que esta fuera de la esfera lo hace cero y este valor es el que representa toda la zona oscura.

Con ello se forma la esfera con su máscara y ahora trabajo para obtener su fase, para obtener la fase envuelta de la imagen hay que extraer su transformada de Fourier para ello se programa lo siguiente:

$$I_{x,y} := t_{x,y} \cdot \exp[i \cdot \pi \cdot (x + y)] \cdot \exp(i \cdot 0.5 \pi \cdot y)$$

$$FI := \text{cfft}(I)$$

$$\text{PowFI}_{x,y} := \begin{cases} 63 & \text{if } \sqrt{|FI_{x,y}|} > 63 \\ \sqrt{|FI_{x,y}|} & \text{otherwise} \end{cases}$$

Y el resultado de este algoritmo se puede observar en la figura 3.6.

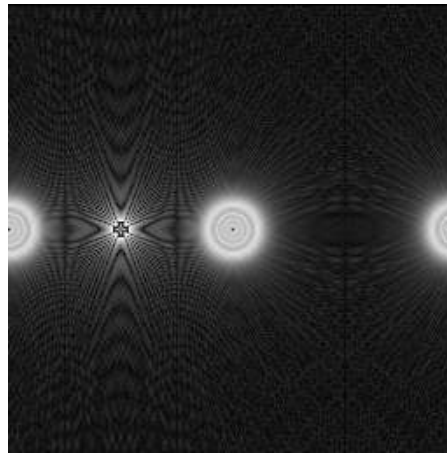


Figura 3.6. Transformada de Fourier de la esfera.

En esta parte lo que se hace es obtener la transformada de Fourier de la figura 3.5 pero antes paso esa ecuación a exponencial llamada *carrier* que es la parte fundamental y de esta forma sacar su transformada de Fourier con el código *cfft(I)* donde la variable *FI* que llame Fourier de I. Una vez que se obtiene la transformada de Fourier se puede saber la potencia de *FI* donde se le asigna un comparador para que se cumplan algunas condiciones necesarias para mejorar el resultado.

Como se observa en la figura 3.6, es el resultado de la transformada de Fourier y la parte importante es la fase y usare un filtro para desechar lo que está alrededor, por lo consiguiente se hizo el filtro para eliminar lo que no va a servir, para ello se programa lo siguiente:

$$\text{Fil}_{x,y} := \begin{cases} 1 & \text{if } r_{x,y} < 40 \\ 0 & \text{otherwise} \end{cases}$$

$$\text{FI}_{x,y} := \text{FI}_{x,y} \cdot \text{Fil}_{x,y}$$

$$\text{PowFI}_{x,y} := \begin{cases} 63 & \text{if } \sqrt{|\text{FI}_{x,y}|} > 63 \\ \sqrt{|\text{FI}_{x,y}|} & \text{otherwise} \end{cases}$$

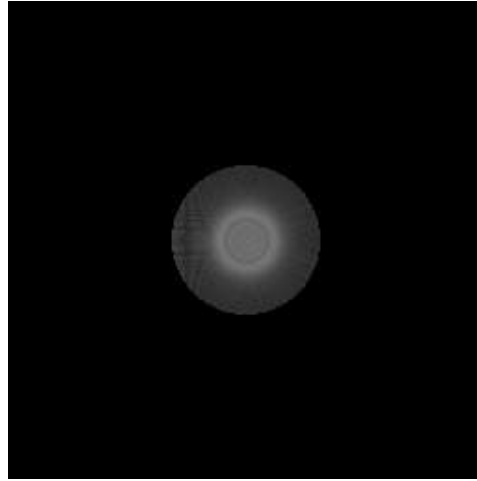


Figura 3.7. Filtro de la Transformada de Fourier.

Como se observa en la figura 3.7, el filtro elimina la parte que no va a servir, dejando nada más la esfera, en lo cual en la programación de *Fil* donde le asignó un comparador en la que dice que $r(x, y)$ es mayor que 40 el valor de *Fil* será de uno, y si no se cumple esta condición esta será cero. Luego se multiplica por la Fourier de *I* que es la transformada de Fourier y se guarda dentro de la misma variable, y después obtengo la potencia de esta misma, como se realizó anteriormente.

Para obtener la fase envuelta de la esfera, hay que tener la transformada inversa de Fourier y guardarla en una variable llamada *Ic* como se muestra a continuación:

$$Ic := \text{icfft}(FI)$$

$$\text{Phase}_{x,y} := -\arg\left[Ic_{x,y} \cdot \exp[i \cdot \pi \cdot (x + y)]\right] \cdot \text{Mask}_{x,y}$$

Como se observa en el código anterior que es la transformada inversa de Fourier se obtiene su argumento que es la multiplicación por *Mask*, que es lo que está entre corchetes dentro de los rangos de $-\pi$ y π . En otras palabras el argumento es el valor de θ cuando lo que está entre corchetes se describe como:

$$\left| \left[I_{c_{x,y}} \cdot \exp[i \cdot \pi \cdot (x + y)] \right] \right| \cdot e^{i\theta}$$

Y obtengo como resultado la figura 3.8 donde se puede observar la fase envuelta de la esfera que se generada desde la misma programación.

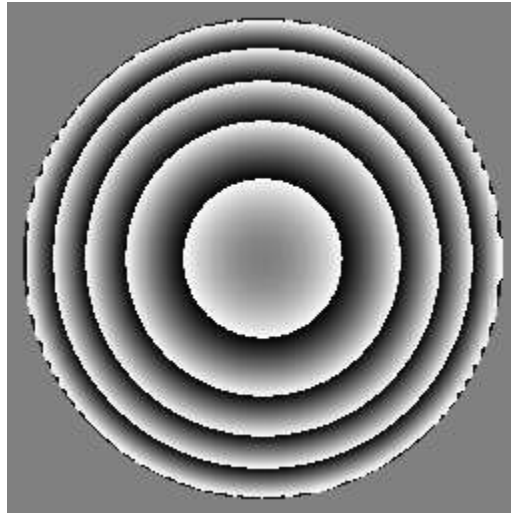


Figura 3.8. Fase envuelta de la esfera.

Como se muestra en la figura el resultado de la fase envuelta de la esfera esta dividida en 4 esferas de diferentes tamaños, que esto se realiza al multiplicar 40 por *Phase* para que muestra 4 esfera, en otras palabras seria cuantas capas quiero tener de la esfera dentro de la misma pero de diferente tamaño y a eso le sume 128 para que genere ese color gris que esta entre 0-255.

El trabajo realizado hasta el momento es la parte fundamental para entender lo que es la proyección de franjas por medio de ecuaciones matemáticas, con la ayuda del software MathCad, los resultados obtenidos muestran que en cierta forma la comprensión necesaria de lo que son las franjas y la fase envuelta (corrimiento de fase), pero a continuación se realizó todo el proceso para llevar acabo la fase envuelta de imágenes ya con franjas de las cuales serán extraídas desde el disco local de una PC y almacenada en una variable tipo matriz. Para ello se detallara el proceso más adelante de lo que se realizó en esa parte del trabajo.

Capítulo 4.- PROCESO DE FASE ENVUELTA

4.1 primeras pruebas de imágenes extraídas de la pc

A partir de este tema se trabajó con imágenes de las cuales ya han sido tomadas con el sistema de proyección de franjas como se muestra en la figura 1.2, estas están almacenadas desde una carpeta local de la PC para ser llamado con facilidad.

La programación que se lleva a cabo para realizar este proceso, es prácticamente la misma que del tema anterior pero ahora, usare un comando para llamar la imagen desde la carpeta almacenada y guardarla en una variable tipo matriz para que esta se pueda llevar a cabo el proceso de su fase envuelta.

En MathCad el comando que se utilizó se llama *READBMP* (“*Dirección del archivo*”) en el cual dentro del paréntesis entre comillas va la dirección donde se encuentra almacenada la imagen a procesar y esta tiene que tener el formato de imagen (.bmp). Una vez hecho esto se guarda en una variable para poder llamar con más facilidad las veces que se han necesarias. En la figura 4.1 se observa la imagen que se llama desde la PC, tomando en cuenta la dirección donde esta guardada la imagen con la que se trabaja en la PC:

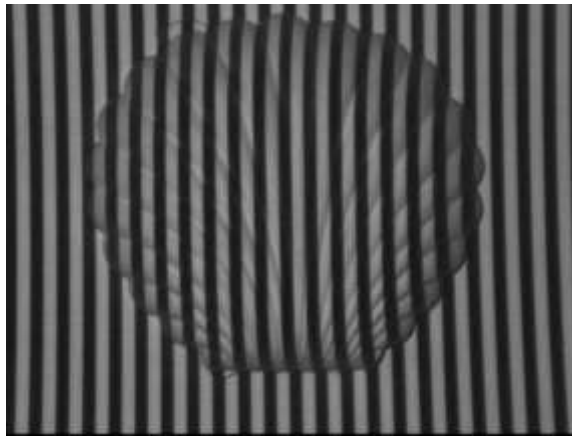


Figura 4.1. Imagen de una concha, sus dimensiones son de 640 x 480 pixeles.

El código para llamar la imagen y guardarla en una variable llamada “*Concha*” queda de la siguiente manera:

```
Concha := READBMP("C:\direccion de la ubicación de la imagen\concha1.bmp")
```


Como se muestra en la figura, se sabe que la imagen tiene una dimensión de 640 en el eje de las x y 480 en el eje de las y , entonces asigne valores dentro de los ejes correspondientes en la programación, solo que en el software para que no ocupe mucho espacio se minimizo sus dimensiones a la imagen y la programación es la siguiente:

$$x := 0..319 \quad y := 0..239 \quad r_{x,y} := \sqrt{(x - 159.5)^2 + (y - 119.5)^2}$$

Ya que como bien se sabe que el valor en este caso empieza de 0-319 que es la mitad de 640 y de 0-239 que es de 480, se guarda en sus respectivas variables y se realiza el mismo proceso de la fase envuelta de la figura de la esfera, sabiendo de que esta imagen ya tiene sus propias franjas como se muestra en la figura 4.2 y además esta reducida de sus dimensiones original de la figura 6.1, lo que se hará es de guardarla en una variable y mandar a imprimir pero con un tamaño pequeño para que no muestre la imagen en su tamaño original y así no ocupe mucho espacio en el software, y guardarla en $t(x, y)$, como se observa a continuación:

$$t_{x,y} := \text{Concha}_{y,x}$$

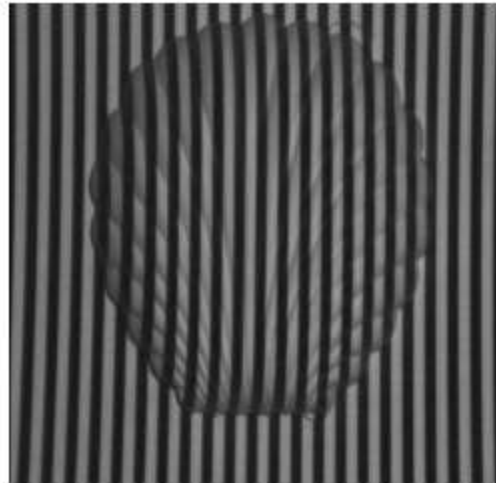


Figura 4.2. Imagen de la concha reducida de sus dimensiones originales.

Teniendo la imagen a un tamaño considerable, se logra saber cuál es su ω_0 para ello hay que saber cuántas franjas son las que están proyectadas en la imagen de la figura 6.2. Ahora hay que calcular la frecuencia con la ecuación 4.1.

$$\omega_0 = N^\circ F \frac{(2\pi)}{N^\circ P} \quad (4.1)$$

Donde:

Nº F = número de franjas proyectadas en la imagen.

Nº P = número de pixeles de la imagen sobre el eje de las x.

Conociendo la formula se obtienen los valores de la ecuación 4.1 dados por las propiedades de la imagen y queda de la siguiente manera:

$$\omega_0 := 2 \cdot 23 \cdot \frac{2 \cdot \pi}{640}$$

Como se observa en la misma imagen son 23 franjas que tiene la figura 4.2 proyectadas sobre si y 640 pixeles en x, pero el número de franjas está siendo multiplicada por un valor de 2, es para ampliar un poco la frecuencia ya que como se sabe la imagen se tuvo que dividir sus dimensiones.

Teniendo la frecuencia continúa la programación, esta parte es la que tenía anteriormente de la forma exponencial que guardo en la variable $I(x, y)$ solo que en esta ecuación le sustituyo ha ω_0 como se muestra de la siguiente manera:

$$I_{x,y} := t_{x,y} \cdot \exp[i \cdot \pi \cdot (x + y)] \cdot \exp[i \cdot (\omega_0 \cdot x + 0.0005y)]$$

En este caso se observa que es diferente a la esfera anterior ya que aquí hay que tomar en cuenta a ω_0 en x y el valor que multiplica a y es para poder observar el resultado final de la imagen como se verá más adelante. Ahora lo que continúa es prácticamente lo mismo de la esfera ya que la parte donde cambia es la exponencial y lo que sigue es lo que tenía anteriormente:

$$FI := \text{cff}(I)$$

$$PowFI_{x,y} := \begin{cases} 63 & \text{if } \sqrt{|FI_{x,y}|} > 63 \\ \sqrt{|FI_{x,y}|} & \text{otherwise} \end{cases}$$

$$Fil_{x,y} := \begin{cases} 1 & \text{if } r_{x,y} < 20 \\ 0 & \text{otherwise} \end{cases}$$

$$FI_{x,y} := FI_{x,y} \cdot Fil_{x,y}$$

Se obtiene la transformada de Fourier de $I(x,y)$ y se guarda en la variable FI , obteniendo FI puedo saber su potencia con $PowFI$, además se utiliza el mismo filtro para esta misma imagen, y se multiplica el filtro que es el atenuador de frecuencias con la transformada de Fourier en $FI(x,y)$, y muestra la imagen del resultado de $PowFI$ y $PowFI + Fil$ como se observa en las figuras 4.3.

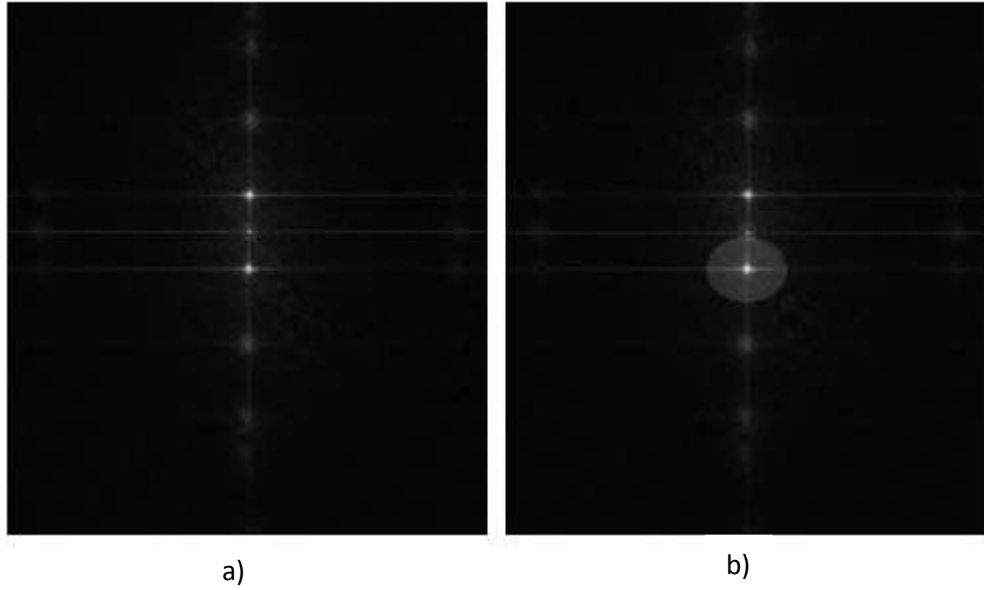


Figura 4.3. a) Potencia de FI de la concha, b) Filtro de PowFI.

Obteniendo estas imágenes se observa que son similares al resultado de la esfera y que el resultado es correcto y para la parte final de esto, se utiliza la transformada inversa de Fourier de la siguiente forma:

$$I_c := \text{icfft}(FI)$$

$$\text{Phase}_{x,y} := -\arg\left[I_{c,x,y} \cdot \exp[i \cdot \pi \cdot (x + y)]\right]$$

Y el resultado de la programación es la siguiente imagen que se muestra en la figura 4.4.

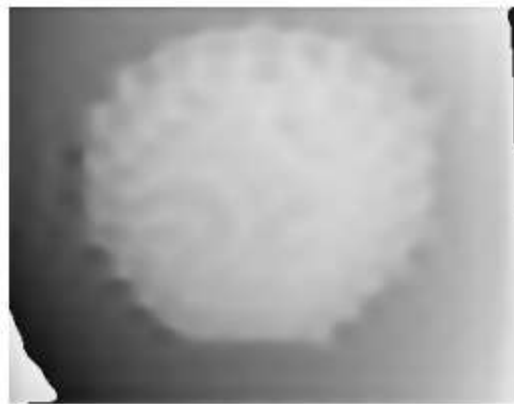


Figura 4.4. Fase envuelta de la Concha.

Como se observa la figura, se muestra la imagen que está casi limpia y con alguna borrosidad pero es por los valores de las ecuaciones que se le tiene que estar modificando para tener una imagen posiblemente mejor ya que existe el problema de ruido que es un factor que afecta a la imagen durante el filtrado.

Teniendo este algoritmo realizado hasta el momento se lleva a cabo el mismo proceso para otras imágenes como se observa en la figura 4.5 y 4.6 el proceso es el mismo para obtener su fase envuelta y su resultado del procedimiento es favorable como el de la concha y se muestran a continuación.

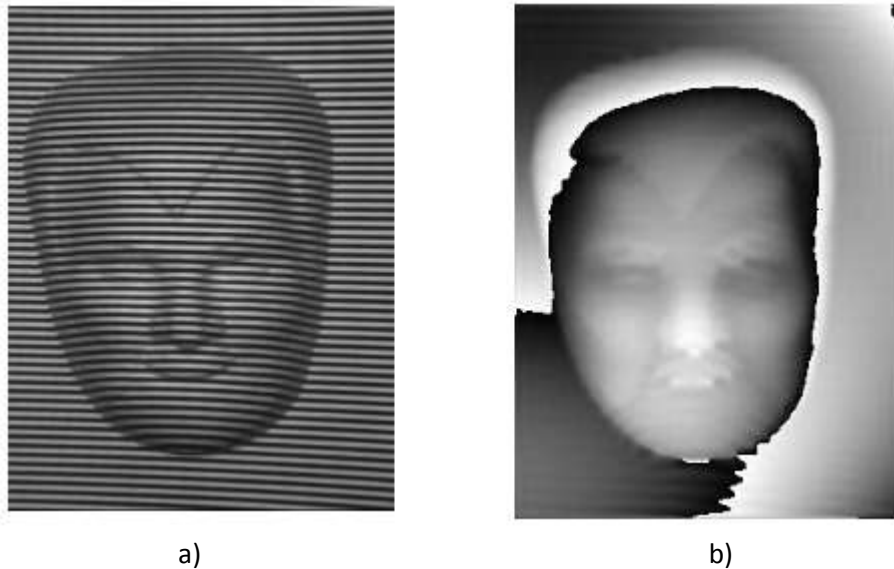


Figura 4.5. a) Imagen de la primera mascara, b) fase envuelta de la imagen.

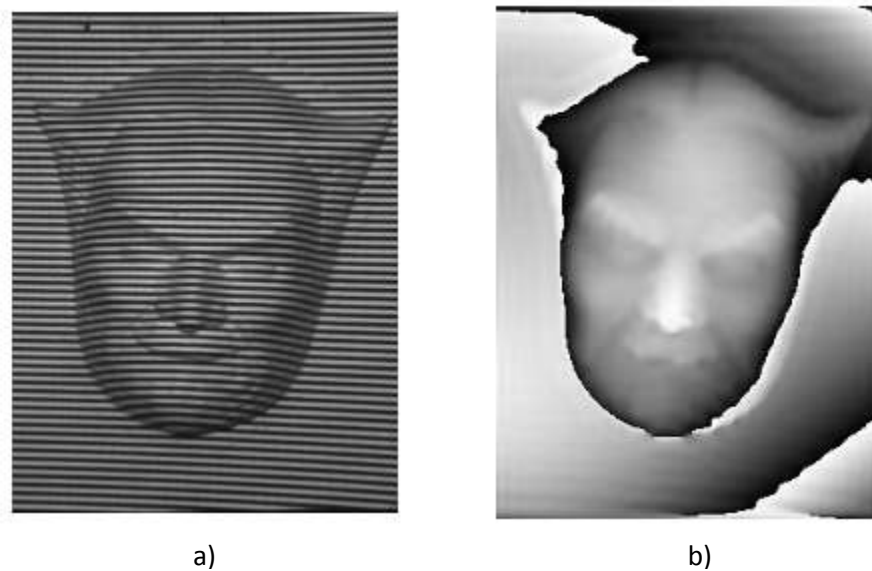


Figura 4.6. a) Imagen de la segunda mascara, b) fase envuelta de la imagen.

El resultado obtenido de las figuras 4.4, 4.5 y 4.6 en su fase envuelta son prácticamente los mismos, los problemas durante el procesamiento de la imagen es el ruido cuyo nivel de intensidad es muy diferentes entre las 3 imágenes (concha y las dos mascararas) y cuyo origen puede estar tanto en el proceso de adquisición de la imagen como en el de transmisión, también involucra lo que es detectar los posibles bordes en las imágenes ya que esta detecta los pixeles donde se produce un cambio brusco en la función de intensidad, y reducir la cantidad de variaciones de intensidad entre pixeles.

Capítulo 5.- METODO DE PROYECCION DE FRANJAS

5.1 Equipo de trabajo

Hasta ahora se ha estado realizando la fase envuelta del procesamiento de las imágenes extraídas desde una PC, en este tema se empieza a trabajar con el proceso de tomas de imágenes de objetos realizando el sistema de proyección de franjas.

Para ello el material a utilizar es el siguiente:

- 1.- Proyector.
- 2.- Cámara CCD.
- 3.- PC de escritorio.
- 4.- Laptop.
- 5.- Objeto.

Teniendo el material se comienza a colocar el objeto sobre una base de fondo blanco mate, para proyectar franjas sobre esta, y se coloca la cámara CCD y el proyector a una distancia de metro y medio de la base del objeto en lo cual desde la laptop se conectó el proyector para proyectar las franjas y la cámara CCD se conecta desde la computadora de escritorio para tomar las imágenes de la franjas proyectadas hacia el objeto.

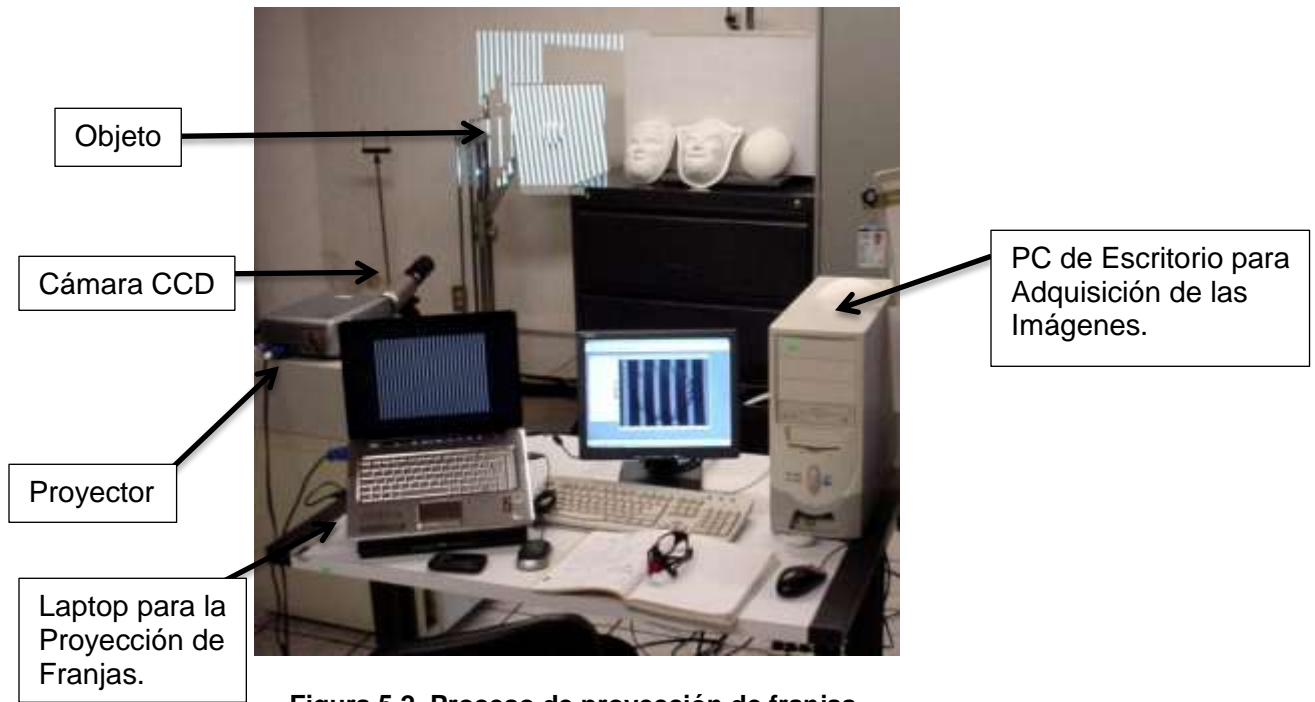
El proceso de calibración de la cámara CCD y el proyector se trabajó con mucho cuidado debido a que el proyector debe estar proyectando claramente hacia el objeto ya que la lente del proyector se calibro hasta conseguir una proyección deseada y de la misma manera se realiza con la cámara CCD en lo cual se buscó una posición en la cual esta estuviera al mismo nivel del proyector como se muestra en la figura 5.1, para tomar imágenes desde el mismo punto que proyecta la cámara y tomando en cuenta que la cámara capte todas las franjas y evitar tomar sombras, ya que eso genera un gran error al pasarlo a su fase envuelta de la imagen.



Figura 5.1. Calibración de la Cámara CCD y el Proyector.

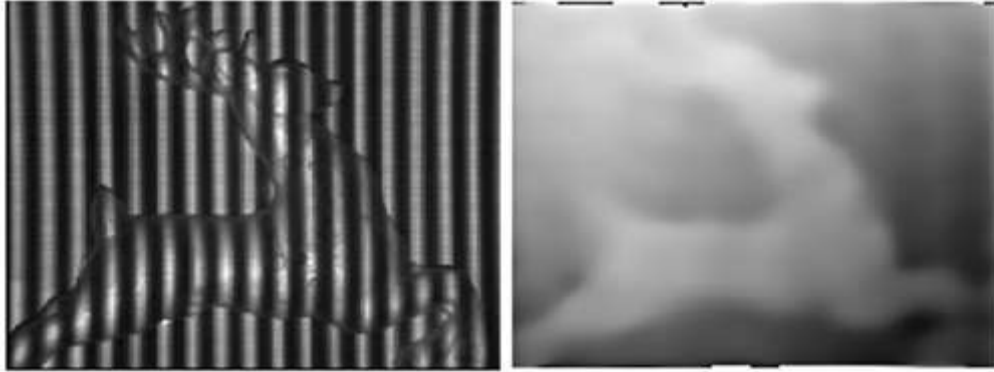
Uno de los problemas que se tuvo con la Cámara CCD es que al observar lo que estaba captando la cámara desde la computadora esta genera una ondulación en las franjas y se trabajó en la toma para eliminar esa ondulación de la imagen, teniendo calibrado tanto la

cámara como el proyector se empezó hacer las tomas de un objeto, esto sirve para conocer los detalles que se lleva en el proceso de la toma de imágenes de la proyección de franjas, la figura 5.2, se muestra el proceso en el cual se llevó acabo la proyección de franjas.[9].



Como se observa en la figura, de esa manera se calibra el proyector con la cámara CCD y la captura de la proyección de franjas hacia el objeto desde la PC de escritorio. Durante la captura de imágenes se estuvieron haciendo pruebas con 10, 20, 40, 60, 80, 100 franjas para ver el resultado de las fases envueltas de las imágenes y al tener los resultados pude observar que la imagen que se obtuvo una mejor fase fue con la de 100 franjas.

Aunque cabe mencionar que al proyectar las 100 franjas no significa que la imagen que grabo la cámara CCD tiene las 100 franjas, sino que esta capto sobre el objeto de las 100 franjas solo 17 de ellas, esto quiere decir que en el proceso de la fase envuelta se le dio el valor de 17 por las franjas que tiene la imagen como se muestra en la figura 5.3 a) y en el inciso b) se observa la fase envuelta.



a)

b)

Figura 5.3. a) Imagen tomada del reno proyectando 100 franjas, b) Fase envuelta de la imagen a).

Al ver los resultados se analizó y se llegó a una conclusión que para tener una mejor fase envuelta de las imágenes se tiene que proyectar más franjas y para ello se dedujo lo siguiente: las dimensiones de las imágenes son de 640x480 pixeles, entonces se optó por proyectar en una franja 4 pixeles entonces se dividió $640/4= 160$ esto quiere decir que se proyectaran 160 franjas en 640 pixeles, entonces se volvió a generar las franjas para proyectarlas, una vez teniendo las franjas como se muestra en la figura 5.4, se procede a tomar las imágenes del objeto.

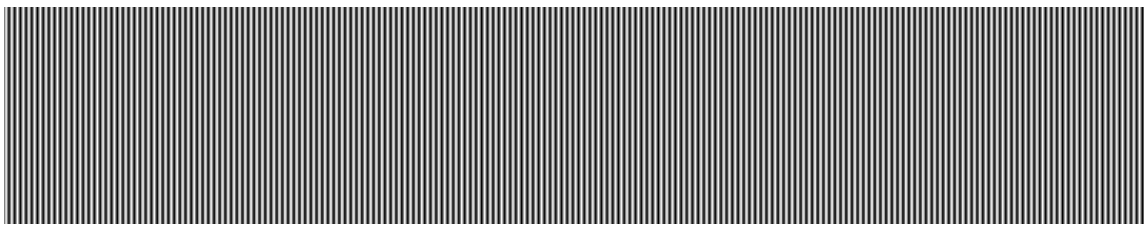


Figura 5.4. 160 franjas proyectadas para el proceso de captura de imágenes.

Debido a la alta frecuencia de la figura se observa que genera el efecto Moiré y que debido a ello genera otras franjas por encima de las mismas. Al proyectar las franjas se elimina el efecto Moiré y se comienza a tomar imágenes del objeto con las 160 franjas proyectadas, aunque la cámara no pueda captar las 160 de las cuales los resultados se muestran en el inciso de la figura 5.5.

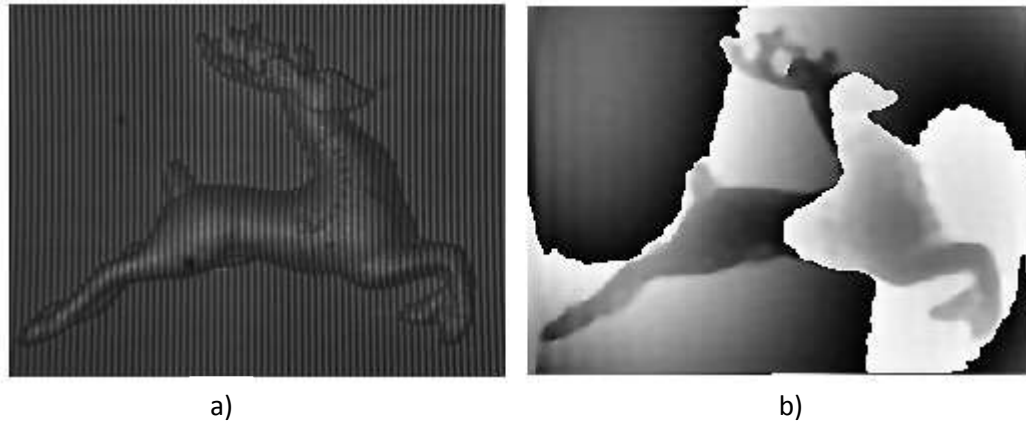


Figura 5.5. Imagen del reno a) utilizando 160 franjas, b) fase envuelta de la imagen.

Como se puede observar en la figura 5.5 b) el resultado es aún mejor que la figura 5.3 b) ya que como se proyectaron más franjas ahora se tiene un mejor resultado en el proceso de la fase envuelta de la imagen, al tener este resultado se comenzó a trabajar en la programación para desenvolver la imagen y poder obtenerla en 3D ya que esta es la parte más complicada de proceso.

Capítulo 6.- METODO DESENVOLVIMIENTO DE FASE

6.1 resumen

Hasta ahora se ha trabajado y hablado de lo que es la captura y modulación (fase envuelta) de una imagen, pero a partir de este tema se hablara de cómo realizar el desenvolvimiento de fase de las imágenes envueltas que genero la modulación del programa que estado trabajando hasta el momento, esta es la última parte de proyecto de investigación y la cual es una de las partes más difíciles ya que se complica cuando la información que se desea desenvolver pueda tener ruido o la calidad misma que puede hacer que esta sea más difícil de desenvolver o ambas cosas dentro de una misma imagen.

El resultado a obtener del desenvolvimiento de fase no siempre es bueno, pero al final de cuentas debe de ser aceptable. Para ello primero se tuvo que saber de lo que se trata el desenvolvimiento de fase.

Para el desarrollo del algoritmo el cual desenvuelva fases de las imágenes envueltas de los resultados anteriores, y se analizó detalladamente el proceso el cual se implementó para llevarlo a cabo, y también ver cuáles fueron los resultados obtenidos mediante el algoritmo desarrollado.

El desenvolvimiento de fase es la parte final del proyecto y como se había comentado anteriormente es la parte más difícil, hay que tener en cuenta que este proceso primero se llevó a cabo con imágenes generadas del mismo software para ver el resultado de este mismo, y hacer las debidas correcciones necesarias que surgieron en su momento, y para ello el algoritmo trabajado se realizó con MATLAB ya que es una herramienta muy útil para este proceso.

A continuación se hablara del desarrollo del algoritmo, para tener un mejor entendimiento del proceso elaborado y para eso se colocó un diagrama para ver los pasos en los cuales se llevó acabo y posteriormente la descripción misma del algoritmo.

6.2.- ALGORITMO DESENVOLVIMIENTO DE FASE

Para el desarrollo de este algoritmo se analizó cada parte a implementar y con la ayuda del siguiente diagrama que se muestra puede resolver el algoritmo para el desenvolvimiento de fase de imágenes.

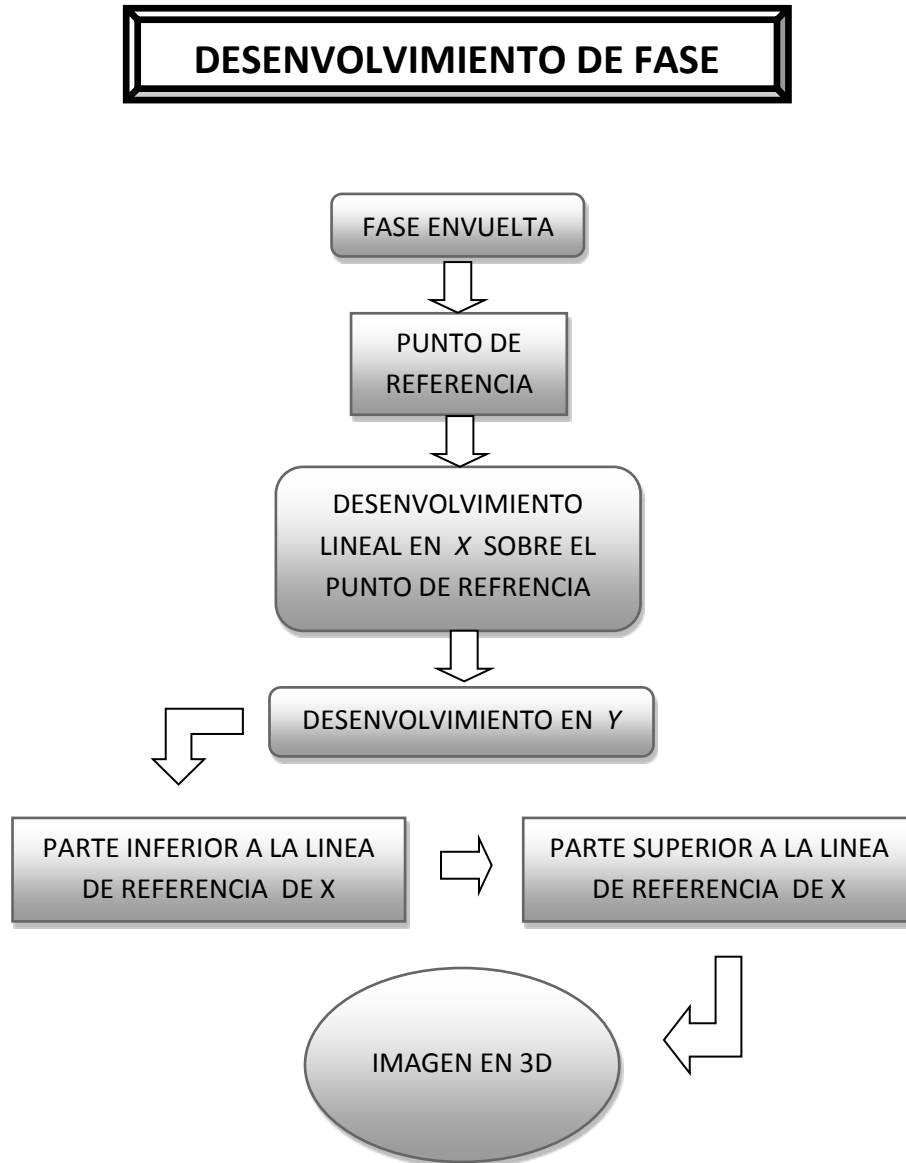


Figura 6. Diagrama de desenvolvimiento de fase.

Para este proceso se utilizó MATLAB para realizar el desenvolvimiento, así que la programación está basada en este software. Para comenzar a desenvolver una imagen envuelta primero hay que ver cuáles son las dimensiones de dicha imagen y guardar el valor de los respectivos ejes x y y , para ello se guarda en la variable EX y EY .

Después se definió un punto dentro de la imagen que este entre las dimensiones de la misma, para que inicie a partir de ese punto a desenvolver, para ello las variables serán x y y .

Ahora se creó una variable donde guardo el proceso de desenvolvimiento por partes ya que para este proceso primero desenvuelvo un punto en el eje de las x y a partir del punto desenvolverá a la derecha y después hacia la izquierda y una vez teniendo el primer punto desenvuelto por ambos lados se procederá a desenvolver hacia arriba y abajo en el eje de las y .

Entonces se guarda la imagen en la variable A , también se crea otra variable donde se ira guardando el resultado obtenido en cada proceso desenvuelto, a esta le llame *suma* y se le da el valor de cero y a esta misma se guarda en $B(y, x)$ donde se está evaluando la función.

Para desenvolver la primera imagen de prueba, se empieza con una imagen sencilla como se muestra en la figura 6.1 que diseñe con el mismo software. Las dimensiones de la imagen son de 512x512 pixeles, y el punto de inicio para desenvolver será en 128. La programación se muestra de la siguiente manera:

```
EX=512;  
EY=512;  
  
y = 128;  
x = 128;  
  
suma = 0;  
A=image1;  
  
B=zeros(EX);  
B(y,x)= suma;
```

De esta forma queda la primera parte de la programación, el comando *zeros(n)* sirve para describir una matriz de $n \times n$ elementos. Ya que la figura 6.1 sus dimensiones son iguales.

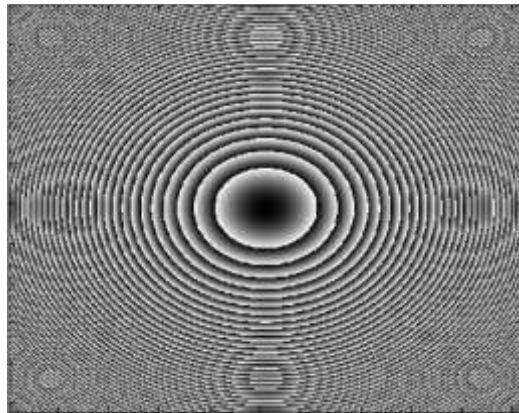


Figura 6.1. Imagen de prueba para el desenvolvimiento de fase.

Como se observar la figura la imagen se llama *image1* y esta se guarda en la variable *A*, teniendo esto se procede a empezar a desenvolver el primer punto de la siguiente manera:

```
for x=128: EX-1
    suma = suma + atan( tan( A(y,x+1) - A(y,x)));
    B(y,x+1)= suma;
end
```

En el primer ciclo *for* se inicia en *x* y se termina hasta *EX-1* que en este caso seria 512-1=511, como se sabe que *x* contiene un valor de referencia donde se comienza a desenvolver en punto, y con la función *tangente* y *arco-tangente* de la variable *A* que es donde se encuentra la imagen guardada y se lleva cabo esa operación y el resultado la guarda en *suma*, y esta misma se guarda en la función *B* donde indica que se desenvuelve hacia la derecha. En el siguiente ciclo se realiza lo mismo pero el desenvolvimiento inicia del punto *x* hacia la izquierda, como se muestra a continuación el algoritmo:

```
suma = 0;
for x=128:-1:2
    suma = suma - atan( tan( A(y,x) - A(y,x-1)));
    B(y,x-1)= suma;
end
```

Como se observar en los dos ciclos anteriores que es el mismo proceso pero la diferencia es de que uno inicia en *x+1* y el otro en *x-1* lo que significa que uno va desenvuolviendo hacia la derecha y el otro hacia la izquierda. Teniendo el primer punto sobre el eje de las *x* desenvuelto o la línea desenvuelta sobre el punto de referencia, se comienza el desenvolvimiento en el eje de las *y*, para ello el desenvolvimiento tanto hacia arriba como hacia abajo se programa dentro de un mismo ciclo cada una con su respectivo ciclo *for*. Como se muestra a continuación:

```
for x=1: EX
    suma = B(128,x);
    for y=128: EY-1
        suma = suma + atan( tan( A(y+1,x) - A(y,x)));
        B(y+1,x)= suma;
    end
```

```

suma = B(128,x);
for y=128:-1:2

    suma = suma - atan( tan( A(y,x) - A(y-1,x)));
B(y-1,x)= suma;
end
end

figure, colormap(gray(256)), imagesc(B)

```

Como se observa en la programación, el desenvolvimiento en y está dentro de un mismo ciclo así que, $suma$ se iguala a B que es donde guardo los resultados obtenidos anteriormente del desenvolvimiento, y le digo que punto en el eje y donde comenzara a desenvolver. Después comienza el siguiente ciclo donde desenvolverá hacia arriba en y y posteriormente lo guarda en $suma$ y está a la vez en B . Para comenzar en la última parte del desenvolvimiento se le indica nuevamente los valores de $suma$, y empieza a ejecutar el último ciclo for , y esta desenvolverá hacia abajo sobre el ejes de las y , y el resultado lo guarda en $suma$ y al mismo tiempo en B y se terminan los ciclos for del desenvolvimiento en y .

Una vez terminado la programación para el desenvolvimiento se imprime en pantalla el resultado como se puede observar en la figura 6.2 en donde se aprecia la esfera en 3D.

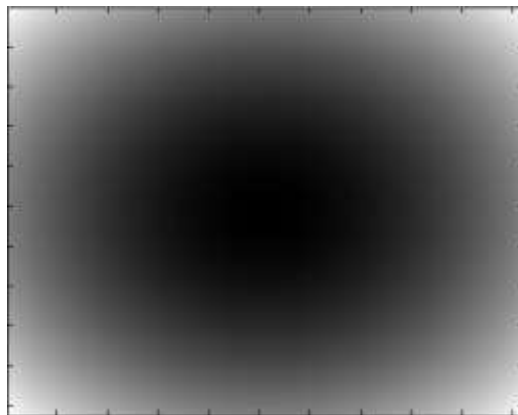


Figura 6.2. Imagen del desenvolvimiento de fase

Como se observar en la figura 6.2, la esfera ya está desenvuelta, se puede ver el resultado que es aceptable para esta primera imagen. El resultado es favorable ya que se obtuvo una muy buena fase envuelta y un desenvolvimiento de fase correcto, ya que estas son imágenes sintéticas, ahora se aplicara este mismo algoritmo para otras imágenes capturadas físicamente, pero usando una esfera de la misma forma que se

generó en la figura anterior y poder observar el resultado obtenido de esta y comparar los resultados de cada imagen.

En la siguiente figura se podrá observar las etapas del proceso que se realizó para obtener la imagen resultante en 3D, en el inciso a) de la figura 6.3, se observa la esfera en la cual se le proyecta franjas con el sistema de proyección utilizada y que el objeto (esfera) está sobre la base de referencia de la proyección, de esa imagen capturada se procede a pasarla a su fase envuelta con el algoritmo realizado anteriormente y se obtiene la figura 6.3 b), y teniendo la fase envuelta se logra desenvolver con el algoritmo creado desde MATLAB y se puede observar el resultado obtenido en la figura 6.3 c).

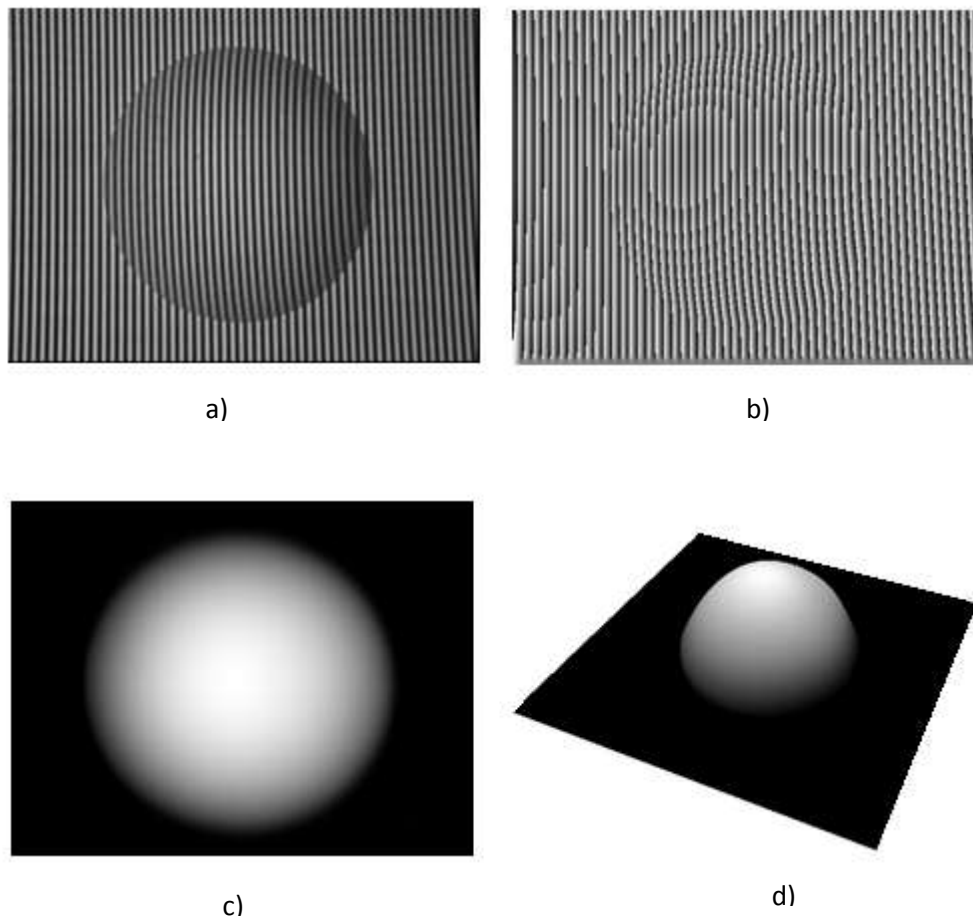


Figura 6.3. Esfera, a) Imagen capturada, b) fase envuelta, c) Desenvolvimiento de fase de la imagen b), d) vista de la esfera en 3D desde otro ángulo.

Como se observa en la figura 6.3 c), el resultado es aceptable y comparándolo con el de la figura 6.2, se dice que tiene una gran similitud, y que el proceso de desenvolvimiento es correcto para la figura 6.2 y para la figura 6.3.

También se muestra otros ejemplos realizado con otros objetos, el de una mascarar de las que se tenía en su fase envuelta anteriormente. Como la figura 4.5 b) y 4.6 b) que ya están en su fase envueltas se puede sacar su desenvolvimiento de fase y el resultado que se obtuvo de esas imágenes se muestra en la figura 6.4 y 6.5.

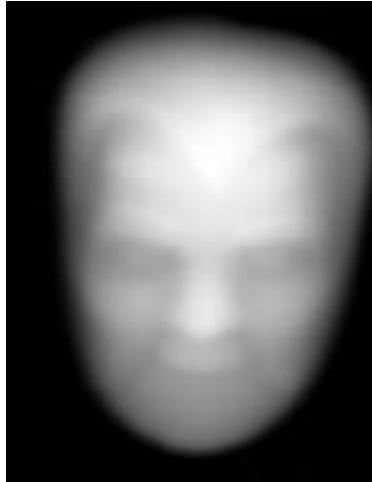


Figura 6.4. Desenvolvimiento de fase (imagen en 3D).



Figura 6.5 Desenvolvimiento de fase (Imagen en 3D).

Al igual que las figuras 4.4, 6.4 y 6.5 se pueden observar desde otra vista, como se muestra en las figuras 6.6, 6.7 y 6.8.



Figura 6.6. Imagen de la Concha vista en 3D.



Figura 6.7. Imagen de la Máscara vista en 3D.



Figura 6.8. Imagen de la Máscara vista en 3D.

Como se ha observado durante el desarrollo de este trabajo la Luz Estructurada es una técnica muy apropiada para el estudio o análisis de objetos y al saber que los resultados en cada parte del proceso de la programación son aceptables, finalizo el trabajo realizado hasta el momento, ahora solo queda darle uso al algoritmo generado.

CONCLUSIÓN

Se puede decir que el trabajo de las proyección de franjas son muy importantes para la construcción de objeto y para su análisis propio del objeto, ya que me ayudan a procesar la imagen adecuadamente, su uso de los arreglos experimentales nos permite realizar de manera práctica las ecuaciones que ayudan a comprobar el experimento realizado de las proyección de franjas que genere con el software, y la imagen que se obtiene representa el mapa de fase envuelta para un objeto y teniendo un algoritmo apropiado para el desenvolvimiento de fase para obtener el resultado final del trabajo realizado.

También se observa la presencia de ruido en las imágenes adquiridas, mismas que se trató de eliminar mediante un filtro con un radio menor a la longitud de onda de las franjas utilizada en la proyección.

Con este proceso se puede llevar a cabo muchas otras aplicaciones en las cuales es muy útil y su bajo costo instrumental lo hacen una técnica de investigación muy apropiada para este tipo de uso. De manera general se puede mencionar que la técnica de proyección de franjas con luz blanca resulta una técnica de fácil manejo, ya que otras técnicas utilizan iluminación láser, lo que con lleva a una fácil desalineación del sistema. Existen muchas áreas en las que se puedan llevar a cabo el proceso de proyección de franjas, ahora solo queda el aplicarlas.

REFERENCIAS BIBLIOGRAFICAS

- [1] Kjell J. Gasvik, “*Optical Metrology*”, (Spectra Vision AS, Trondheim, Norway).
- [2] Kjell J. Gasvik, “*Moiré Technique by means of Digital image Processing*”.
- [3] Luz estructurada,
<http://www.elai.upm.es:8009/spain/Investiga/GCII/areas/luz/luz%20estructurada.htm>
- [4] Jorge Ojeda, Eliezer Jara, Jorge Ibarra, “*Franjas de Moiré*”, INAOE (Instituto Nacional de Astrofísica, Óptica y Electrónica)
- [5] Carlos Hinojosa Espinosa, Alfonso Serrano Heredia, “*Método Óptico Digital para la Reconstrucción Tridimensional de Objetos*”
- [6] Dennis C. Ghiglia, Mark D. Pritt, “*Two-Dimensional Phase Unwrapping: theory, algorithms, and software*”.
- [7] Liverpool John Moores University, “*Unwrapping Phase*”,
<http://www.ljmu.ac.uk/GERI/90202.htm>.
- [8] *Interferometry: Phase Unwrapping*, <http://earth.esa.int/landtraining07/D1LB4-Rocca.pdf>
- [9] Restauración de imágenes digitales,
<http://alojamientos.us.es/qtocoma/pid/pid10/Restauracion.htm>