



UNIVERSIDAD MICHOACANA DE SAN
NICOLÁS DE HIDALGO

FACULTAD DE CIENCIAS FÍSICO MATEMÁTICAS
“MAT. LUIS MANUEL RIVERA GUTIÉRREZ ”

**Digitalización de superficies e
impresión 3D**

T E S I S

QUE PARA OBTENER EL GRADO DE:
Licenciado En Ciencias Físico Matemáticas

PRESENTA:

Luis Fernando Cervantes Barriga

DIRECTOR DE TESIS:
DR. MAURICIO ORTIZ GUTIÉRREZ



MORELIA MICH.

OCTUBRE DE 2019

Dedicado con cariño y aprecio a mi familia.

Agradecimientos

“De los mayores pecados que comete el hombre la soberbia es el mayor dicen algunos, pero el desagrado es mayor, digo yo.”

Don Quijote de la Mancha.

Agradezco a mis padres Lupita y Javier por su cariño y afecto de toda la vida y sobre todo por su infinita paciencia y apoyo durante mi estancia en la facultad; a mis hermanos Mariela y Esteban que de una u otra manera contribuyeron a mi curiosidad y toma de decisión en mi camino del conocimiento. A mi asesor el Dr. Mauricio Oritz Gutierrez por toda la paciencia y tiempo dedicado, por la diligencia y esmero al resolver mis preguntas y orientar mis pasos hasta la conclusión de este proyecto.

A mis colegas y amigos del laboratorio a quienes tengo en especial consideración en mi memoria como auténticos amigos tras las infinitas tardes que pase a su lado y sin cuya habilidad técnica y alto grado de capacitación y humanismo este trabajo no hubiera sido posible.

A mis amigos de la facultad que siempre estuvieron desde el comienzo, por aquellas tardes tratando de resolver los innumerables y divertidos problemas, estudiando, comparando resultados, la realimentación y convivencia, amigos que fueron ejemplo en el transcurso del camino universitario, que contribuyeron en mi formación tanto académica como humanista, y no menos importante a mis amigos ajenos a la facultad, con los que siempre me apoye y fueron un gran pilar en esta meta alcanzada.

Por último quiero agradecer a Dios por darme la oportunidad de haber vivido para ver concluida una meta más en el largo camino de la vida.

Abstract

In this thesis it is proposed to use structured light projection on the surface of three-dimensional objects to reproduce them digitally and print them with the help of a 3D printer. This will be carried out using the spatial phase wrapping technique through the application of the Fourier Transform method, presenting an algorithm developed in MATLAB with the proposal to develop this phase and reproduce the topography of the study objects three-dimensionally, later with Software help will rebuild our object in STL format and finally our object will be printed in plastic (PLA). In this technique it can be applied to different types of objects and in this thesis it is applied to unicel objects.

Keywords: Structured light; wrapped phase; unwrapped phase; Fourier transform, sampling theorem.

Resumen

En esta tesis se propone utilizar la proyección de luz estructurada sobre la superficie de objetos tridimensionales para reproducirlos digitalmente e imprimirlos con ayuda de una impresora 3D. Esto se llevará a cabo empleando la técnica espacial de envolvimiento de fase mediante la aplicación del método de la Transformada de Fourier, se presentará un algoritmo desarrollado en MATLAB con la finalidad de desarrollar dicha fase y reproducir tridimensionalmente la topografía de los objetos de estudio, posteriormente con ayuda de software se reconstruirá nuestro objeto en formato STL y finalmente se imprimirá en plástico (PLA) nuestro objeto. En esta técnica se puede aplicar a diferentes tipos de objetos y en esta tesis se aplica a objetos de unicel.

Palabras clave: Luz estructurada; fase envuelta; fase desenvuelta; transformada de Fourier, teorema de muestreo.

Índice general

1. Introducción	1
1.1. Antecedentes	1
1.2. Objetivo General	3
1.3. Objetivos Específicos	3
2. Teoría de Fourier	5
2.1. Series de Fourier	5
2.2. Transformada de Fourier	6
2.3. Teorema de muestreo	10
3. Proyección de luz estructurada	13
3.1. Método espacial para medición de fase por transformada de Fourier	13
4. Desarrollo de Fase	17
4.1. Técnica de Desarrollo de Fase	18
4.2. Análisis y ejemplos sobre el desarrollo de fase	20
4.3. Errores en el desarrollo de fase	24
5. Impresión en 3D	27
5.1. Funcionamiento de una impresora 3D	28
5.2. FreeCAD un programa dedicado a la creación y modificación de objetos	31
5.3. Repetier Host comunicación entre la impresora y la computadora	35

5.4. Partes de una Impresora 3D de tecnología FDM tipo cartesia- na y calibración	37
5.5. Impresión de imágenes en 3D	41
6. Diseño experimental y Resultados	45
6.1. Arreglo experimental	45
6.2. Reproducción digital de superficies en 3D	50
6.3. Materialización del objeto con ayuda de una impresora 3D . .	55
7. Conclusiones	59
A. Códigos de MATLAB	61
A.1. Código para el desenvolvimiento de fase	61
A.2. Código para la generación de rejillas	62
A.3. Código para generar el filtro para el orden 1	63
Referencias	65

Capítulo 1

Introducción

1.1. Antecedentes

Los métodos de medición ópticos sin contacto, son técnicas realmente efectivas que presentan diversas aplicaciones. La metrología óptica es una técnica con la cual se pueden tomar medidas más pequeñas que por otros medios, esta aplicación permite analizar propiedades físicas con alta precisión, la cual tiene que ver con la medición y monitoreo de parámetros físicos usando métodos de no contacto mediante el manejo de luz. Entre los métodos sin contacto están los sistemas ópticos basados en la técnica de proyección de franjas, efecto moiré entre otros [1].

La técnica del efecto moiré tiene su principio físico en la superposición de dos rejillas, donde una de ellas se distorsiona debido a la deformación de la forma del objeto bajo estudio siendo esta proporcional a la cantidad física que se desea medir; la otra rejilla no sufre distorsión alguna y se le conoce como rejilla de referencia, la primera parte para obtener el relieve de una superficie es lograr desenvolver la fase de la rejilla de muestreo sobre el objeto. Un método para llevar a cabo tal fin es empleando fase por pasos en las cuales se toman una serie de fotografías con un cambio de fase específico entre ellas. Dicho método se puede ver aplicado en el trabajo de Nokolaus Karpinsky [2]

Uno de los métodos más efectivos que involucran el uso de patrones de

luz estructurada, los cuales son proyectados sobre la superficie de un objeto formando un patrón de franjas para obtener su reproducción 3D, es por el método de transformada de Fourier descrita por primera vez por Takeda [3] que consiste en un algoritmo para analizar los datos para la reconstrucción de dicha superficie que se basa en el análisis de Fourier de franjas que consiste en un filtrado en el dominio de frecuencias y en la recuperación de la parte imaginaria de la fase. El método tiene ventajas sobre la topografía de Moiré y la interferometría de proyección de franja convencional tanto en precisión como en sensibilidad.

A diferencia de las técnicas de exploración de franjas el método es fácil de aplicar porque no utiliza componentes móviles.

En la Facultad de Cs. Físico Matemáticas de la UMSNH se han realizado trabajos sobre el análisis de superficies por diferentes técnicas uno de estos fue la tesis de licenciatura de Jorge Adrián Arias del Angel [4] que consiste en la proyección de luz estructurada sobre la superficie de objetos tridimensionales para reproducirlos digitalmente empleando la Transformada de Fourier para el desenvolvimiento de fase mediante el desarrollo de un algoritmo en MATLAB con la finalidad de desenvolver dicha fase para reproducir tridimensionalmente la topografía de los objetos de estudio.

La impresión 3D es un tipo de fabricación donde se genera la pieza desde cero, pero se necesita un prototipo digital, hay muchos tipos de formatos 3D sin embargo el formato STL es el más común de la impresión 3D.

Las técnicas de impresión 3D han aumentado su popularidad en los últimos años, para uso tanto doméstico, industrial y educativo entre otros. Se ha popularizado tanto debido, entre otros a la finalización de diversas patentes, sobre todo relacionados con la tecnología de impresión por deposición de material plástico, que son las más utilizadas por las máquinas de ámbito no industrial.

Con el aumento del uso de la impresión 3D se ha desarrollado software gratuitos para la creación o conversión de formatos a STL como es el caso del DICOM (imagen usada en la medicina)[5] o como el caso de BRep (imagen

3D) [6].

Desde el inicio de la impresión 3D en 1976 con la impresión de inyección de tinta se han transformado llegando así en 1984 la primera impresora de materiales. A lo largo de las últimas décadas, han habido una gran variedad de aplicaciones de la tecnología de impresión que se han desarrollado a través de varias industrias, una de las tecnologías más utilizadas en la impresión 3D dentro de la ingeniería como en la medicina es la creación de objetos a partir de un modelo existente (huesos, tuercas, ect.), la importancia de la exactitud de dicho modelo es cada vez mas importante que a impulsado a desarrollar diversos trabajos tanto de escaneo utilizando barrido láser como la creación de mejores programas para procesar y reconstruir de una manera mas eficiente el objeto[7].

En esta tesis se estudia la digitalización de la superficie de un objeto mediante la proyección de luz estructurada con el método de Fourier, después se reconstruye con ayuda de diversos software y se imprime con ayuda de una impresora 3D, este proceso es usualmente llamado ingeniería inversa.

1.2. Objetivo General

Obtener digitalmente la superficie de objetos macroscópicos mediante la proyección de luz estructurada sobre su superficie mediante la teoría de Fourier, para después obtener por medio de diferentes software y con ayuda de la impresora 3D la creación de los objetos obtenidos digitalmente.

1.3. Objetivos Específicos

1. Comprender el análisis de Fourier para envolver la fase.
2. Comprender la teoría de envolvimiento y desenvolvimiento de fase.
3. Reconstruir la superficies de objetos en 3D.

4. Comprender el funcionamiento de una impresora 3D.
5. Transformar una imagen 2D en formato BMP a STL en 3D para la creación de un objeto en la impresora 3D.

Capítulo 2

Teoría de Fourier

2.1. Series de Fourier

Una serie de Fourier puede definirse como el desarrollo o representación de una función en una suma de series armónicas como:

$$g(x) = \frac{a_0}{2} + \sum_{n=1}^{\infty} [a_n \cos(2\pi n f_x x)] + \sum_{n=1}^{\infty} [b_n \sin(2\pi n f_x x)], \quad (2.1.1)$$

donde f_x es la frecuencia, a_n y b_n son amplitudes de cada una de las componentes armónicas y $g(x)$ debe ser periódica e infinitamente extendida. Si la función $g(x)$ es real, estos coeficientes son también reales. Los coeficientes pueden ser calculados integrando la función $g(x)$ en un periodo completo y utilizando propiedades de ortogonalidad se llega a las siguientes ecuaciones

$$\begin{aligned} a_n &= \frac{1}{x_0} \int_{-x_0}^{x_0} g(x) \cos(2\pi n f_x x) dx, \\ b_n &= \frac{1}{x_0} \int_{-x_0}^{x_0} g(x) \sin(2\pi n f_x x) dx, \end{aligned} \quad (2.1.2)$$

donde la x_0 es el periodo de la función. Si la función es simétrica es decir si $g(x) = g(-x)$ entonces solo los coeficientes a_n pueden ser diferentes de cero y si es antisimétrica ($g(x) = -g(-x)$) entonces serán los coeficientes b_n los que pueden ser diferentes de cero. Si la función es asimétrica ambos coeficientes

pueden ser diferentes de cero.

Expresando el seno y coseno en la forma exponencial la ecuación 2.1.1 se puede expresar también como:

$$g(x) = \sum_{-\infty}^{\infty} C_n e^{2\pi i n f_x x}, \quad (2.1.3)$$

los coeficientes pueden ser calculados como:

$$C_n = \int_{-x_0}^{x_0} g(x) \cos(2\pi n f_x x), \quad (2.1.4)$$

En este caso, los coeficientes C_n corresponden a valores positivos de amplitud. En general estos coeficientes son complejos.

2.2. Transformada de Fourier

Si el periodo de la función $g(x)$ incrementa y la separación de las componentes sinusoidales decrece, en el límite cuando el periodo tiende a infinito, el intervalo de frecuencia entre armónicas tiende a cero. Cualquier función no periódica puede ser considerada como una función periódica con un periodo infinito, por lo tanto, una función continua no periódica se puede representar como una serie de Fourier, como consecuencia la suma de la ecuación 2.1.4 se puede convertir en una integral y esto nos lleva al concepto de transformada de Fourier. Sea $g(x)$ una función continua dependiente de la variable real x . La transformada de Fourier de $g(x)$ se representa por $G(f)$ y está definida por:

$$\mathcal{G}\{f\} = \int_{-\infty}^{\infty} g(x) e^{-2\pi i f x} dx. \quad (2.2.1)$$

La Transformada de Fourier $\mathcal{G}(f)$ es el espectro de amplitud de la función $g(x)$ o espectro de Fourier y su amplitud es el módulo al cuadrado de este espectro. También dada $\mathcal{G}(f)$, la función $g(x)$ puede ser obtenida a través de

la Transformada Inversa de Fourier, definida por:

$$g(x) = \int_{-\infty}^{\infty} \mathcal{G}\{f\} e^{2\pi i f x} dx, \quad (2.2.2)$$

donde x es la variable espacial y su dominio se refiere como dominio espacial. Por otro lado, f es una variable de frecuencia, por lo que su dominio es el de frecuencias espaciales o dominio de Fourier.

La transformada de Fourier de dos variables x y y se representará de la siguiente forma:

$$\mathcal{F}\{g(x, y)\} = \int \int_{-\infty}^{\infty} g(x, y) e^{-2i\pi(f_x x + f_y y)} dx dy, \quad (2.2.3)$$

la transformada de Fourier se define como una función compleja de dos valores independientes f_x y f_y las cuales se conocen como frecuencias espaciales. La transformada inversa de Fourier de $\mathcal{G}\{f_x, f_y\}$ se representará como $\mathcal{F}^{-1}\{G\}$ y está definida como:

$$\mathcal{F}^{-1}\{G(f_x, f_y)\} = \int \int_{-\infty}^{\infty} G(f_x, f_y) e^{2i\pi(f_x x + f_y y)} df_x df_y. \quad (2.2.4)$$

Antes de continuar es necesario establecer las “condiciones de existencia”, que son una variedad de conjuntos de condiciones para la existencia de la ecuación 2.2.3 donde entre las más comunes son:

1. La función g debe ser absolutamente integrable sobre el plano infinito (x, y) .
2. La función g debe tener solo un número finito de discontinuidades y un número finito de máximos y mínimos en su dominio.
3. La función g no debe tener infinitas discontinuidades.

A continuación se expondrán algunas de las propiedades matemáticas básicas de la transformada de Fourier. En los siguientes casos se considera que las transformadas de Fourier de las funciones $g(x)$ y $h(x)$ están dadas por $\mathcal{F}(g) = \mathcal{G}(f_x, f_y)$ y $\mathcal{F}(h) = \mathcal{H}(f_x, f_y)$, respectivamente.

Teorema de linealidad: La transformada de una suma de dos o más funciones es igual a la suma de sus transformadas individuales.

$$\mathcal{F}\{\alpha g + \beta h\} = \alpha \mathcal{F}\{g\} + \beta \mathcal{F}\{h\}. \quad (2.2.5)$$

Teorema de escalamiento La amplificación de las coordenadas en el dominio espacial (x, y) resulta en una contracción de las coordenadas en el dominio de las frecuencias (f_x, f_y) , además de un cambio en la amplitud total del espectro.

$$\mathcal{F}\{g(ax, by)\} = \frac{1}{|ab|} \mathcal{G}\left\{\frac{f_x}{a}, \frac{f_y}{b}\right\}. \quad (2.2.6)$$

Teorema de desplazamiento Una traslación de la función en el dominio espacial introduce un cambio de fase lineal en el dominio de frecuencias.

$$\mathcal{F}\{g(x - a, y - b)\} = \mathcal{G}\{f_x, f_y\} e^{-i2\pi(f_x a + f_y b)}. \quad (2.2.7)$$

Teorema de Rayleigh (Teorema de Parseval) La integral del cuadrado de una función es igual a la integral del cuadrado de su transformada, su interpretación física es que la integral en el dominio espacial puede ser interpretada como la energía contenida en un frente de onda, Lo que nos conduce a la idea de que la integral en el dominio de frecuencias puede ser interpretada como una densidad de energía.

$$\int \int_{-\infty}^{\infty} |g(x, y)|^2 dx dy = \int \int_{-\infty}^{\infty} |G(f_x, f_y)|^2 df_x df_y. \quad (2.2.8)$$

Teorema de convolución La transformada de Fourier de una convolución de dos funciones en el dominio espacial es equivalente a multiplicar sus transformadas individuales.

$$\mathcal{F}\left\{\int \int_{-\infty}^{\infty} g(\xi, \eta) h(x - \xi, y - \eta) d\xi d\eta\right\} = \mathcal{G}(f_x, f_y) \mathcal{H}(f_x, f_y). \quad (2.2.9)$$

Teorema de autocorrelación Es un caso especial del Teorema de convo-

lución donde se realiza la convolución entre una función y su conjugado en el dominio espacial

$$\mathcal{F} \left\{ \int \int_{-\infty}^{\infty} g(\xi, \eta) g^*(x - \xi, y - \eta) d\xi d\eta \right\} = |\mathcal{G}(f_x, f_y)|^2. \quad (2.2.10)$$

Teorema integral de Fourier La transformada de Fourier de la transformada inversa de Fourier de una función producen la función original de manera similar la transformada inversa de la transformada de una función, nos regresa la función original, excepto por sus puntos de discontinuidad.

$$\mathcal{F}\mathcal{F}^{-1}\{g(x, y)\} = \mathcal{F}^{-1}\mathcal{F}\{g(x, y)\} = g(x, y). \quad (2.2.11)$$

Estos teoremas no son solo de interés teórico sino que tienen una gran aplicación práctica y serán utilizados con frecuencia ya que pueden ahorrar cantidades enormes de trabajo.

A continuación se mencionarán algunas funciones con su respectiva transformada de Fourier que suelen utilizarse.

1. Función rectángulo:

$$rect(x) = \begin{cases} 1 & |x| < \frac{1}{2} \\ \frac{1}{2} & |x| = \frac{1}{2} \\ 0 & \text{otro caso.} \end{cases} \quad (2.2.12)$$

2. Función seno-cociente

$$sinc(x) = \frac{sen(\pi x)}{\pi x}. \quad (2.2.13)$$

3. Función signo

$$sgn(x) = \begin{cases} 1 & x > 0 \\ 0 & x = 0 \\ -1 & x < 0 \end{cases} \quad (2.2.14)$$

4. Función peine de Dirac

$$\text{comb}(x) = \sum_{-\infty}^{\infty} \delta(x - n). \quad (2.2.15)$$

2.3. Teorema de muestreo

Considere una función real de banda limitada es decir una función o serie de tiempo cuya transformada de Fourier se restringe a un rango finito de frecuencias o longitudes de onda, $g(x)$ cuyo espectro (transformada de Fourier) es $\mathcal{G}(f)$ y su ancho Δf es igual a la frecuencia máxima contenida en dicha función como se muestra en la figura 2.1. Considere la función peine de Dirac que es $(\text{comb})h(x)$ cuyo espectro $\mathcal{H}(f)$ es igual a otro peine de Dirac como se muestra en la figura 2.2.

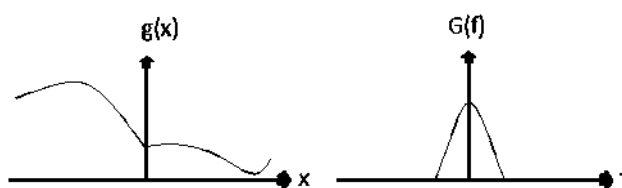


Figura 2.1: Función $g(x)$ y su transformada de Fourier.

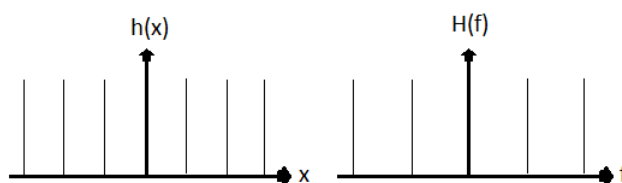


Figura 2.2: Función peine de Dirac $h(x) = \text{comb}(x)$ y su transformada de Fourier.

Para muestrear la función $g(x)$ es necesario multiplicar esta función con la función peine de Dirac se muestra en la figura 2.3. La frecuencia fundamental de la función peine de Dirac $h(x)$ está definida como la frecuencia

de muestreo. Una consecuencia directa del teorema de convolución es que el espectro de esta función muestreada (un producto de las dos funciones) es la convolución de las dos transformadas de Fourier $G(f)$ y $H(f)$ [4].

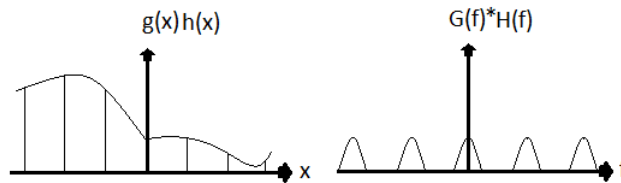


Figura 2.3: Convolución de las dos funciones y su transformada de Fourier.

Si la frecuencia de muestreo de la función $h(x)$ disminuye, los elementos espectrales en la convolución de las funciones $G(f)$ y $H(f)$ se acercan una a la otra. Si estos elementos espectrales no se superponen, la transformada inversa de Fourier recupera la función original. Después de muestreo si los elementos espectrales se superponen total o parcialmente la función original puede que no se recupere del todo. Por lo tanto, los requisitos del teorema de muestreo son violados cuando los elementos espectrales se tocan mutuamente.

El ancho total ($2 \Delta f$) de la base de los elementos espectrales debe ser menor que el doble de la frecuencia máxima (f_{max}) presente en la función a ser muestreada, la frecuencia de separación entre los picos en la Transformada de Fourier de la función Peine de Dirac es igual a la frecuencia de muestreo por lo tanto, la frecuencia de muestreo $f_m = \frac{1}{\Delta x}$ debe ser mayor que la mitad de la frecuencia máxima f_{max} contenida en la señal o función muestreada:

$$f_m = 2f_{max}. \quad (2.3.1)$$

Esta condición es conocida como el Teorema de Muestreo de Whittaker – Shannon y la frecuencia mínima de muestreo es referida a la frecuencia de Nyquist[4].

Capítulo 3

Proyección de luz estructurada

3.1. Método espacial para medición de fase por transformada de Fourier

En los métodos espaciales de medición de fase están basados en la idea de superponer un patrón de franjas en una superficie, dicho patrón esta deformado y representado por la siguiente ecuación:

$$g(x, y) = a(x, y) + b(x, y)\cos[\Phi(x, y) + 2\pi f_0 x], \quad (3.1.1)$$

donde $\Phi(x, y)$ es la fase, f es la frecuencia del patrón de franjas en dirección x , $a(x, y)$ y $b(x, y)$ representan variaciones de irradiancia no deseadas que surgen de la reflexión o transmisión de luz no uniforme por un objeto de prueba se conocen como radiación de fondo y contraste respectivamente. La importancia de conocer la fase $\Phi(x, y)$ se verá a continuación. En la figura(3.1) se muestra un patrón de franjas proyectadas (rejilla) en el plano $-xy$ sobre este se encuentra una superficie S , el patrón de franjas se proyecta a un ángulo Θ_1 mientras es observado por una cámara a un ángulo Θ_2 respecto al eje z respectivamente. Se muestra que las franjas están desplazadas, el punto P_1 se desplaza al punto P_2 sobre la superficie S . El desplazamiento está dado por:

$$u = z(\tan \Theta_1 + \tan \Theta_2), \quad (3.1.2)$$

donde z es la altura entre el plano- xy y el punto P_2 mientras tanto la fase de modulación de la rejilla esta dada por:

$$\Phi(x) = \frac{u}{dx} = \frac{z}{d}(\tan \Theta_1 + \tan \Theta_2) \cos \Theta_1 = \frac{z \sin(\Theta_1 + \Theta_2)}{d \cos \Theta_2} = \frac{z}{d}G, \quad (3.1.3)$$

donde G es el factor geométrico y d el periodo de la rejilla.

Para obtener el valor de la fase se utiliza el método de Transformada de

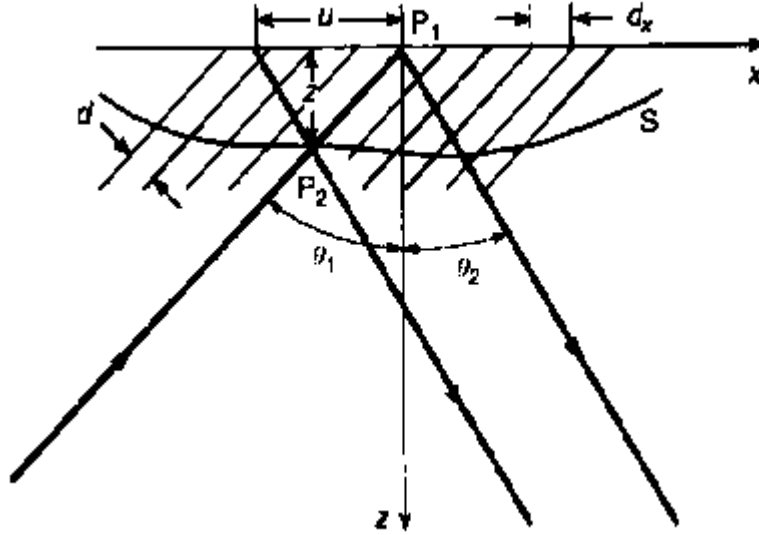


Figura 3.1: Proyección de una rejilla sobre una superficie[1].

Fourier presentada por Takeda [3] siguiendo su método se tiene la siguiente ecuación:

$$g(x, y) = a(x, y) + c(x, y)e^{i2\pi f_0 x} + c^*(x, y)e^{-i2\pi f_0 x}, \quad (3.1.4)$$

con

$$c(x, y) = \frac{1}{2}b(x, y)e^{i\Phi(x, y)}, \quad (3.1.5)$$

donde $*$ denota el complejo conjugado. La transformada de Fourier de la ecuación (3.2) con respecto al eje x es:

$$G(f_x, y) = A(f_x, y) + C(f_x - f_0, y) + C^*(f_x + f_0, y), \quad (3.1.6)$$

donde las letras en mayúsculas denotan el espectro de Fourier y f_x es la frecuencia espacial en dirección x . Esto es posible gracias a que $A(f_x, y)$, $C(f_x - f_0, y)$ y $C^*(f_x + f_0, y)$ están separadas a una distancia f_0 en el espacio de frecuencias como se muestra en la figura 3.2. haciendo uso de la función

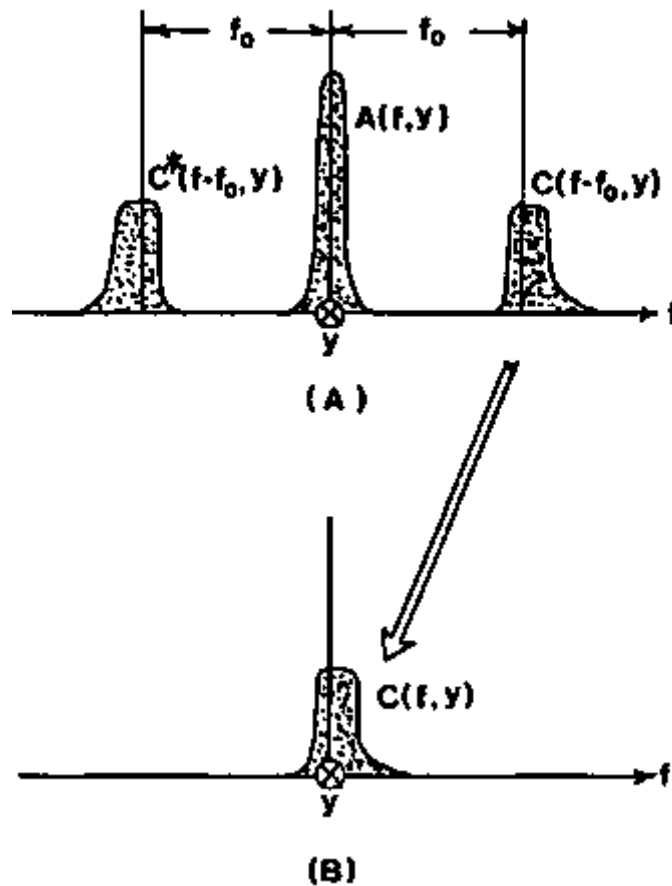


Figura 3.2: Espectro de Fourier separado de un patrón de franjas[3].

Filtro $F(f_x - f_0, y)$ en el espacio de frecuencias la función $C(f - f_0, y)$ es aislada y se traslada f_0 hacía el origen, se obtiene $C(f, y)$ como se muestra

en la figura 3.2(b). Se obtiene la Transformada de Fourier Inversa a $C(f, y)$ y aplicando logaritmo en el dominio espacial se obtiene:

$$\log[c(x, y)] = \log\left[\frac{1}{2}b(x, y)\right] + i\Phi(x, y), \quad (3.1.7)$$

remarcando que $\Phi(x, y)$ se encuentra en la parte imaginaria y $b(x, y)$ en la parte real.

Capítulo 4

Desenvolvimiento de Fase

En el capítulo anterior se vio como a través de la transformada de Fourier podemos aislar y conocer la fase ya que esta se encuentra en la parte imaginaria de la ecuación 3.1.7 también se vio cuando se proyecta un patrón de franjas sobre una superficie que presenta relieve, las líneas del patrón se deforman de acuerdo a la superficie y la altura z se relaciona con la fase por la ecuación 3.1.3 queda claro la importancia de conocer la fase o patrón de franjas proyectadas. Otro método para encontrar la fase es por método IMF (Interferometría de medición de fase) que se puede escribir como

$$I = a + b \cos \Phi, \quad (4.0.1)$$

donde a y b son constantes y se les conoce como iluminación de fondo y contraste respectivamente. De la ecuación 4.01 se obtiene la fase teniendo

$$\Phi = \tan^{-1}\left(\frac{C}{D}\right), \quad (4.0.2)$$

donde C y D son funciones dependientes de la irradiancia. La solución de Φ de los casos mencionados anteriormente encuentran discontinuidades como la función dientes de sierra mostrada en la figura(4.1.a) debido que las funciones arcotangente y coseno no son inyectivas. Las discontinuidades ocurren cada vez que Φ cambia en 2π , si Φ crece la pendiente de la función es positiva y viceversa si decrece la pendiente de la función es negativa. El termino

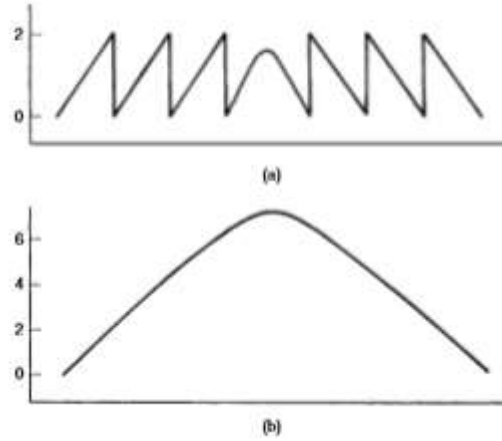


Figura 4.1: a) Fase envuelta en la función dientes de sierra. b) Función de fase continua obtenida por desenvolvimiento de a.

fase desenvuelta viene del método computacional que consiste en integrar el valor 2π a la fase en el caso de la función diente de sierra en la figura(4.1.b) se muestra la función con su fase desenvuelta, la eficiencia del método dependerá del algoritmo utilizado, que tan eficiente sea a la hora de detectar los brincos de fase y si los datos son continuos (sin huecos) o si tiene ruido (información extra no deseada obtenida de manera experimental).

Un método simple de desenvolvimiento de fase consiste en una imagen conformada por un conjunto de datos sin huecos es decir continua en toda la imagen, un defecto que podría complicar este método es una discontinuidad repentina sobre una franja o en el punto donde las franjas se detienen (ruptura de franjas) como en la figura 4.2, esto es particularmente serio si los datos están siendo escaneados línea por línea en un dirección el error de la fase se propagará por el resto de la imagen.

4.1. Técnica de Desenvolvimiento de Fase

Una de las técnicas para el desenvolvimiento de fase consiste en integrar (en unidades de 2π) la fase envuelta Φ a lo largo de todos los datos, a cada

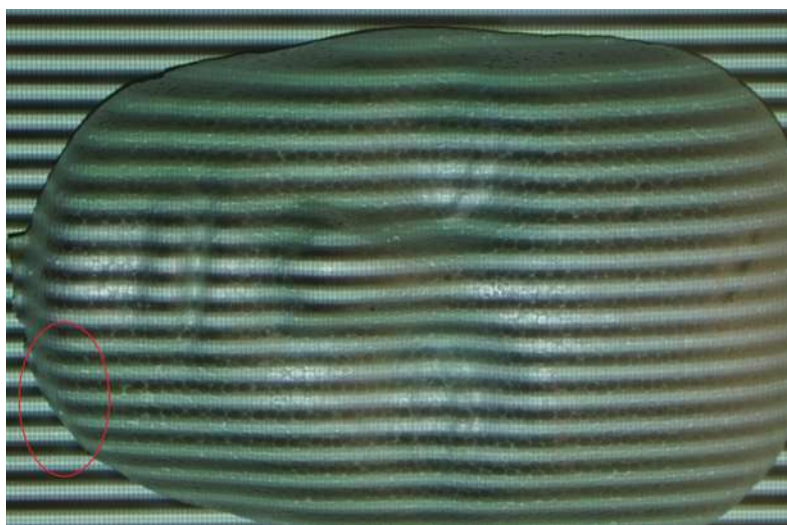


Figura 4.2: Se muestra dentro del círculo rojo discontinuidades repentinas sobre las franjas.

pixel se le calcula la diferencia de el con su anterior es decir

$$\Delta \Phi = \Phi_n - \Phi_{n-1}, \quad (4.1.1)$$

donde n es el número del pixel, si $|\Delta \Phi|$ está entre π y 2π , entonces se suma o resta 2π dependiendo del signo de $\Delta \Phi$, si es positivo se resta 2π y si es negativo se le suma.

El método más simple para desenvolver la fase involucra un escaneo secuencial a través de los datos, línea por línea. Al final de cada línea, la diferencia de fase entre el último pixel y el pixel de la línea debajo es determinada y esta última línea es escaneada en dirección contraria (Figura 4.3).

Una de las aproximaciones más populares que es empleada para evadir la propagación de los errores de fase a través de un arreglo de datos es desenvolver las regiones con “buenos” pixeles en primer lugar. Los “malos” pixeles con alto grado de incertidumbre son desenvueltos enseguida, pero la propagación de errores se reduce a solo pequeñas regiones.

La forma para definir buenos pixeles es examinar el contraste en la rejilla de interferencia (visibilidad) sobre el pixel en consideración. Es importante

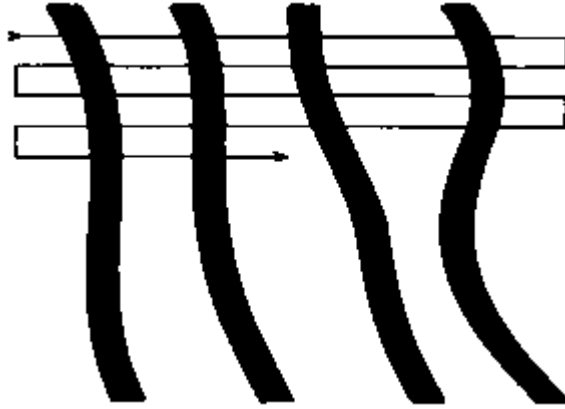


Figura 4.3: Línea por escaneo secuencial de líneas.

con esta técnica que el orden del procesamiento de los píxeles sea tal que el proceso de desenvolvimiento se propague a lo largo conectando trayectorias a través del arreglo.

4.2. Análisis y ejemplos sobre el desenvolvimiento de fase

Se presentara ejemplos del proceso de envolvimiento de fase, desenvolvimiento y tipos de errores.

En la figura 4.4 a) se muestra una fase continua generada por computadora en una dimensión, la cual no contiene envolvimiento de fase (saltos 2π) mientras que en la figura 4.4 b) se muestra la misma fase pero envuelta. Mientras en la figura 4.5 se muestra una imagen de fase continua generada por computadora, en la figura 4.5a) se muestra la imagen pero en arreglo de intensidades, en la Figura 4.5b) se muestra un corte transversal de la superficie.

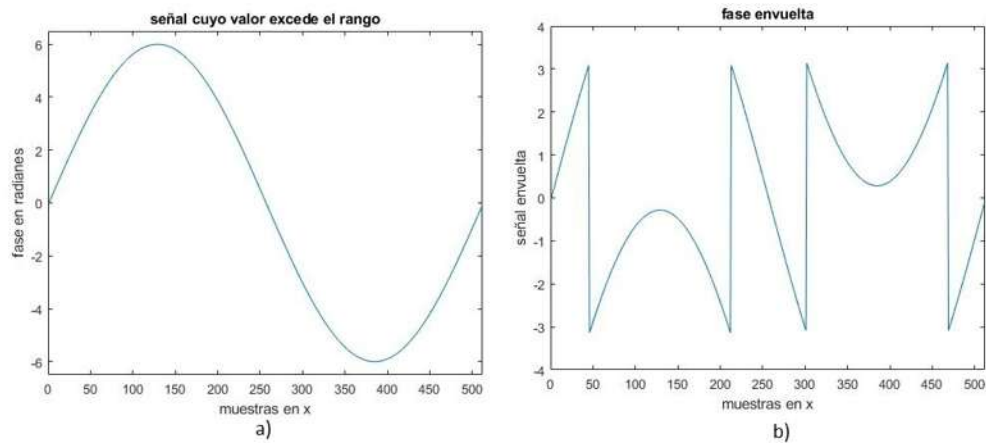


Figura 4.4: a) Señal (sin envolver), b) Señal envuelta.

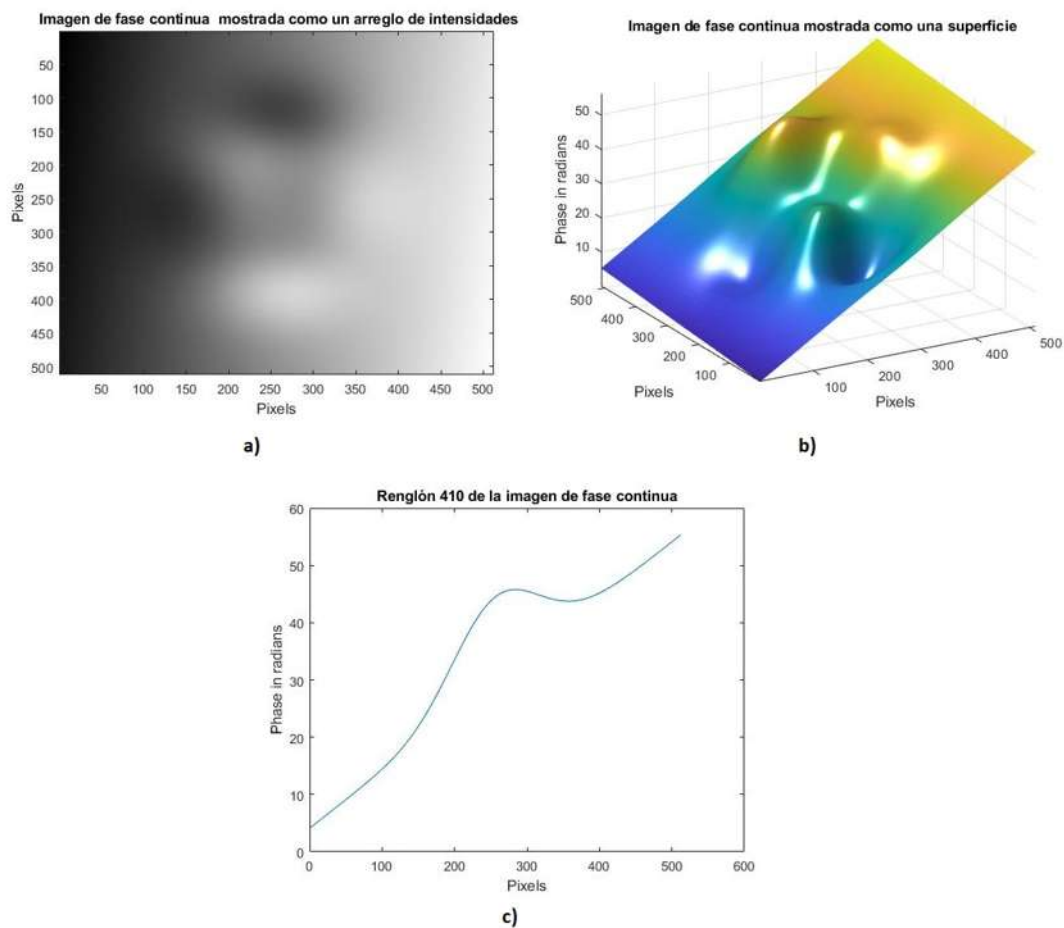


Figura 4.5: a) Imagen de fase continua mostrada como un arreglo de intensidades, b) Imagen de fase continua mostrada como una superficie, c) Renglón 410 de la imagen de fase continua.

A continuación se envuelve la imagen de fase continua dando como resultado las imágenes mostradas en la Figura 4.6, la Figura 4.6a) muestra la fase envuelta en 3D mientras que la figura 4.6b) se tiene la imagen de fase envuelta como curvas de contorno y finalmente en la figura 4.6c) se muestra un frente de onda de dientes de sierra tratándose con líneas de valores de fase, la fase envuelta aparece como saltos múltiples de 2π .

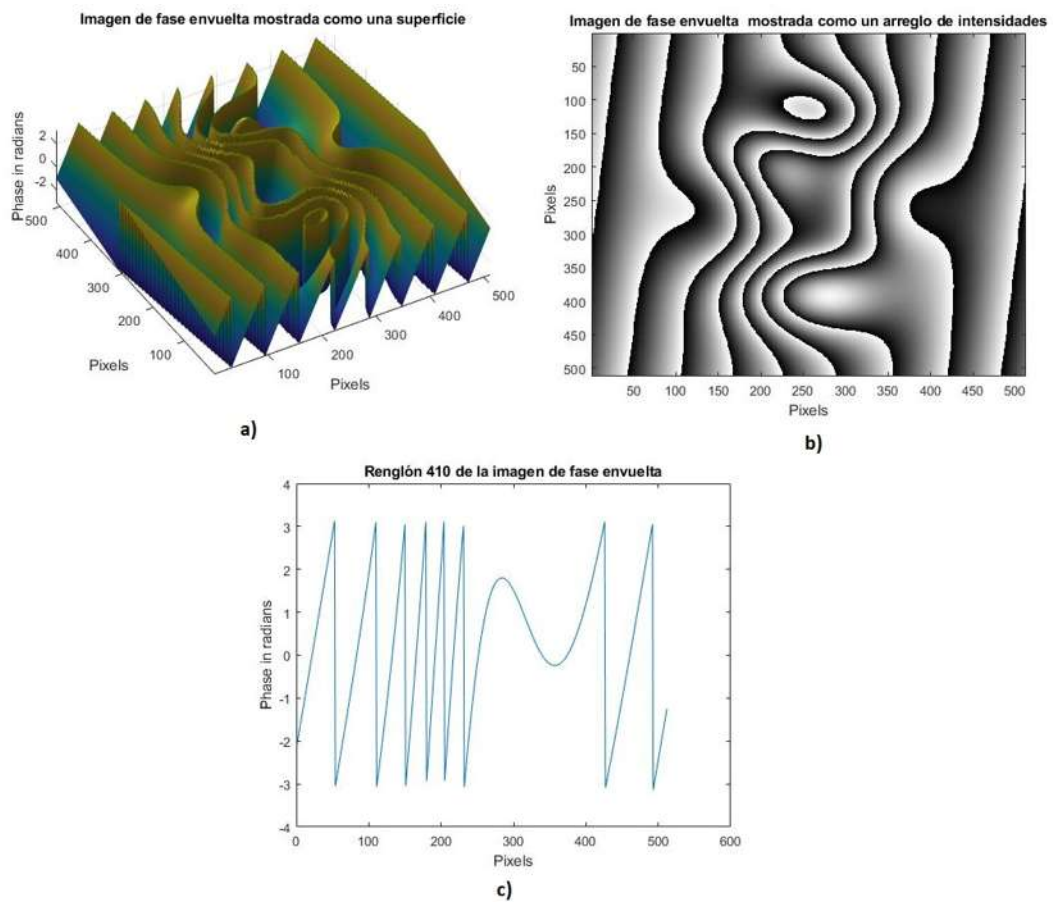


Figura 4.6: a) Imagen de fase envuelta mostrada como una superficie, b) Imagen de fase envuelta mostrada como un arreglo de intensidades, c) Renglón 410 de la imagen de fase envuelta.

Para desenvolver la imagen se puede utilizar el método de desenvolvimiento de fase 2D de Itoh [8] dicho código es utilizado en esta tesis y se muestra

en el Apéndice, existen dos formas de implementar este método la primera que consiste en desenvolver las filas de la imagen envuelta esto produce una imagen intermedia la cual está solo parcialmente desenvuelta entonces solo falta desenvolver la imagen intermedia pero ahora sus columnas como se muestra en la figura 4.7

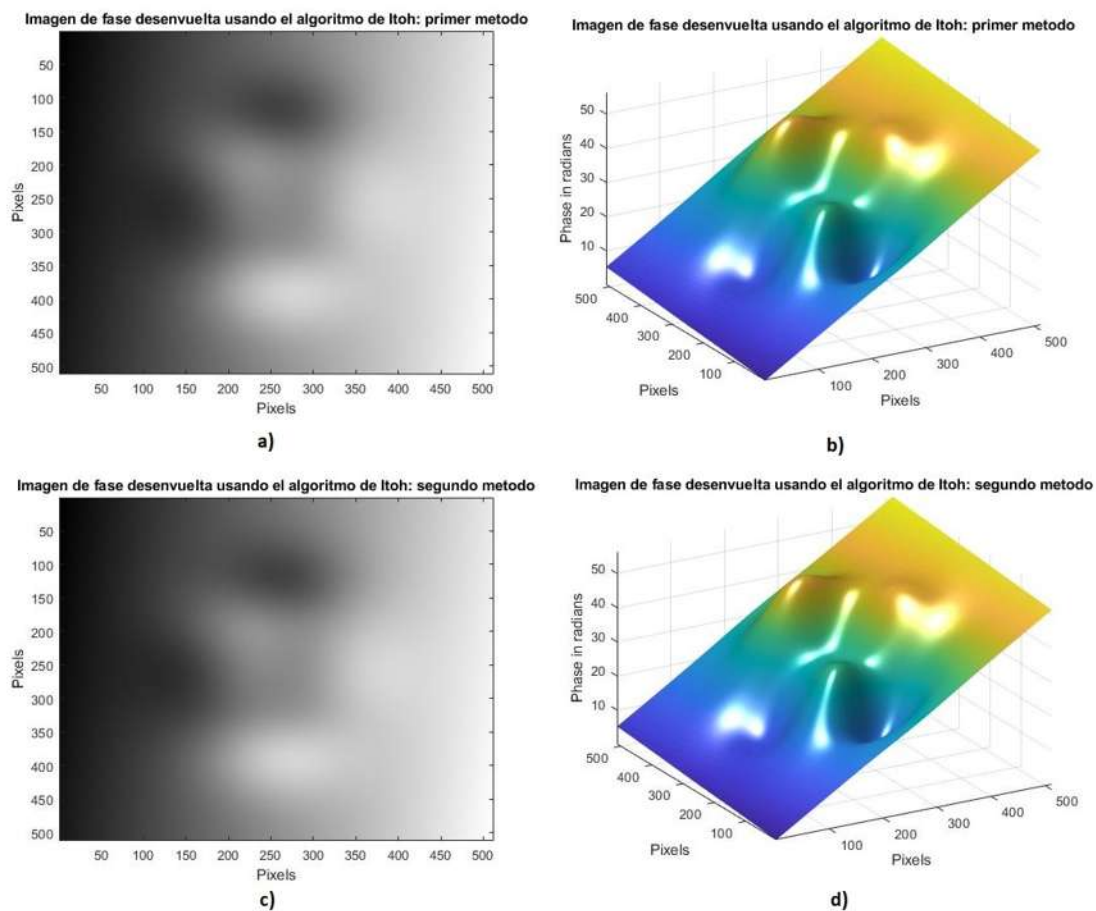


Figura 4.7: Imagen de fase envuelta mostrada como una superficie usando el algoritmo de Itoh: a) y b) Primer método del algoritmo Itoh. c) y d) Segundo método del algoritmo Itoh.

Para una imagen creada en Matlab se muestra que ambos métodos de Itoh producen un buen resultado sin embargo solo funciona en los casos de traba-

jar con imágenes de fase libre de errores. En la mayoría de las aplicaciones las imágenes de fase envueltas contienen errores, en el caso del desenvolvimiento de fase $2D$, existen cuatro fuentes de errores las cuales complican los procesos de desenvolvimiento de fase estas se mencionan continuación:

1. Ruido.
2. Sub – Muestreo.
3. Cambios abruptos de fase.
4. Errores producidos por el algoritmo al extraer la fase.

4.3. Errores en el desenvolvimiento de fase

El ruido puede tener efectos catastróficos en el proceso de desenvolvimiento de fase a la que se le atribuye la identificación falsa de saltos de fase esto consiste porque el algoritmo para detectar la existencia de fase envuelta en una imagen calcula la diferencia entre la intensidad de dos pixeles consecutivos, si la diferencia es mayor que π , entonces el algoritmo considera que en ese punto existe una discontinuidad en la fase esto puede ser una fase envuelta genuina o puede ser una fase envuelta falsa debido a la presencia de ruido. En la figura 4.8 se muestra un ejemplo en una dimensión.

Mientras que el algoritmo se procesa se va acumulando errores durante el proceso de desenvolvimiento de fase, está es la razón por la cual se complica el proceso de desenvolvimiento de imágenes de fase envuelta en $2D$ con ruido.

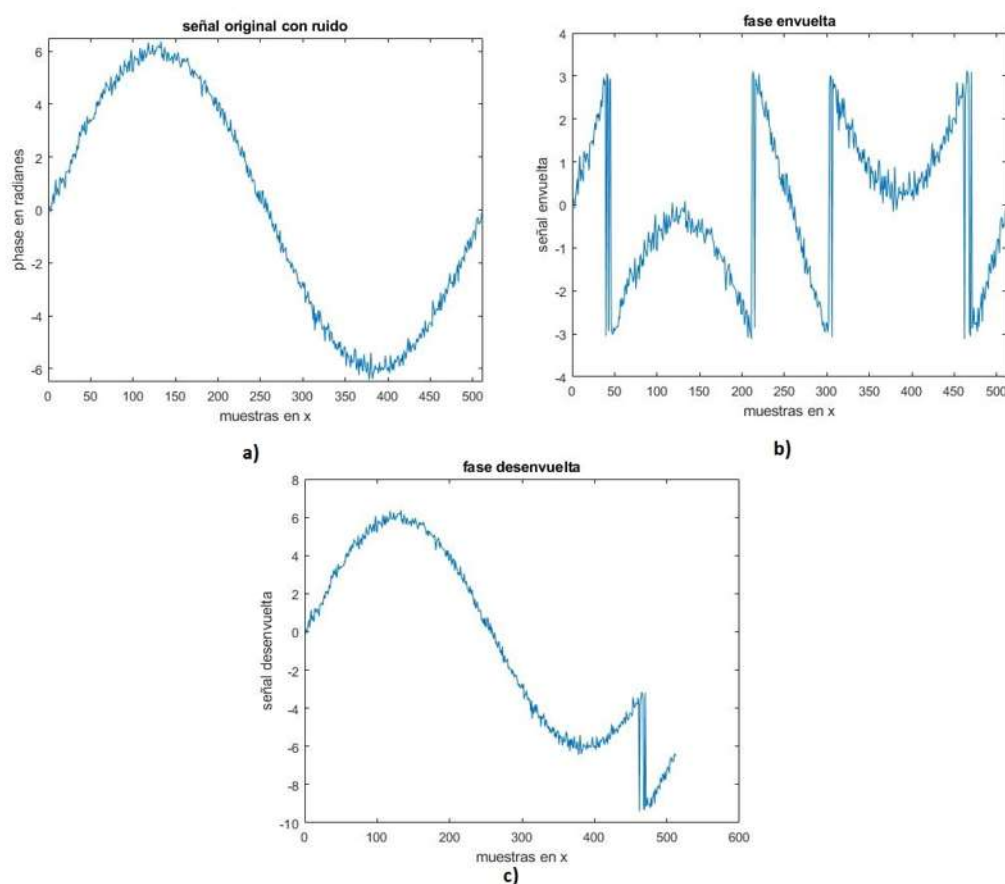


Figura 4.8: a) Señal original con ruido. b) Señal de fase envuelta. c) Señal de fase desenvuelta.

Los errores de sub – muestreo ocurren cuando la diferencia entre dos muestras sucesivas es mayor de $+\pi$ o menor de $-\pi$. Esta gran diferencia entre muestras adyacentes está presente simplemente debido a que la imagen de fase no contiene suficientes muestras y no por la existencia de una fase envuelta verdadera. Esta situación inmediatamente genera una incorrecta fase envuelta[9][10].

El efecto de la discontinuidad en el desenvolvimiento de la fase es otro tipo de error común que consiste cuando el desenvolvimiento de fase de una imagen que contiene cambios repentinos de fase esto ocurre cuando estos cambios de fase son mayores de $+\pi$ o menores de $-\pi$. Tales cambios de fase pueden ser

vistos como envolturas reales, pero en realidad no lo son un ejemplo de esto se muestra en la figura 4.2.

Capítulo 5

Impresión en 3D

La impresión 3D es un proceso de fabricación basado en la idea de convertir un modelo digital en un objeto tridimensional sólido. El inicio de la tecnología 3D se remota en 1976, cuando se inventó la impresora de inyección de tinta. El origen del concepto de impresión 3D se remota a los años 80 cuando el Dr. Kodama del Instituto Municipal de Investigación Industrial de Nagoya ideó una técnica de prototipado rápido utilizando la fabricación de un objeto capa a capa, en 1984 un grupo de investigadores presentó lo que vendría siendo la primera patente para después dejarla olvidada hasta que finalmente Chuck Hull, cofundador de 3D Systems, obtiene lo que vendría siendo la primera patente comercial en 1986 por una técnica llamada “Estereolitografía” la impresión 3D tuvo sus inicios con el proceso de Estereolitografía (SLA) que utiliza como material de impresión una resina líquida solidificada bajo el efecto de la luz UV. Desde este primer avance, nuevas técnicas de fabricación aditiva han surgido a lo largo de los años. Unas de las más usadas debido a su fácil manejo y a un coste de adquisición relativamente bajo son las impresoras 3D mediante extrusión de filamento de plástico, esta técnica esta patentada como FDM (Fused Deposition Modeling) y es la utilizada en esta Tesis.

El interés sobre la tecnología de la impresión 3D a crecido bastante a través de diferentes disciplinas por ejemplo en el 2010 la empresa Kor Ecologic presentó “Urbee”, el primer automóvil en el que su armazón está impreso

completamente en 3D también en este año la empresa Organovo Inc., centra sus investigaciones en la tecnología de bioimpresión, realizando publicaciones sobre vasos sanguíneos bioimpresos, en el 2011 la universidad de Exeter, en conjunto con la Universidad de Brunel desarrollan la impresión de alimento, a partir del 2012 la medicina ha implementado con mayor frecuencia el uso de prótesis impresas en 3D incluso en la área de la construcción se a llegado a utilizar.

5.1. Funcionamiento de una impresora 3D

A pasar de los años se han creado diversas tecnologías de impresoras para la creación de modelos físicos capa por capa, de dichas tecnologías se encuentran:

- Modelado por Deposición Fundida (FDM)
Es la única tecnología de impresión 3D profesional que usa termoplásticos de grado de producción. Las piezas de termoplástico pueden resistir la exposición al calor, químicos, entornos húmedos o secos y tensión mecánica.

- PolyJet
La tecnología PolyJet es un proceso de impresión 3D que inyecta y cura delgadas capas de fopolímero líquido con energía UV, la ventaja de esta tecnología es su resolución de hasta $0,016mm$, crea superficies lisas y una amplia gama de materiales y colores con un costo y un tiempo relativamente bajo.

- Sinterización Selectiva por Láser (SLS)
La impresión se realiza capa por capa, a partir de polvos fusionados, gracias a la temperatura generada por un láser CO2. La SLS permite obtener piezas coloreables y sin límites de forma gracias a la ausencia de soportes, logrando un alto nivel de calidad del producto. Este proceso es

utilizado, no solamente en el área de diseño sino también en la industria automotriz, la aeronáutica y en la ingeniería biomédica.

- Estereolitografía (SLA)

SLA utiliza el principio de fotopolimerización para crear modelos 3D a partir de resinas sensible a los rayos UV. A comparación de las anteriores el SLA requiere un postprocesamiento para finalizar el proceso de fotopolimerización y conseguir la maximiza fuerza del material. La SLA nos da un acabado superficial ligeramente vítreo, pero generalmente es superior a los procesos FDM o SLS (con el mismo espesor de capa), aunque tenemos pocos colores de impresión.

- Sinterización Directa de Metal por Láser (DMLS)

Fusiona el polvo de metal y los materiales de aleación con un láser de alta potencia para producir piezas metálicas robustas. Encontramos generalmente materiales metálicos como el acero, el cobalto-cromo, el aluminio, el titanio o incluso el inconel, esta tecnología es muy utilizada en la industria aeroespacial, automotriz y médica.

- Fusión por Haz de Electrones (EBM)

permite producir piezas que sería muy difícil fabricar de otra forma y con una resistencia equivalente a las piezas fabricadas por los métodos convencionales esta tecnología imprime piezas metálicas, este proceso se basa sobre el principio de cargas eléctricas, por lo cual los materiales utilizados deben ser conductores imperativamente.

la tecnología utilizada en esta tesis es la FDM cabe mencionar que podemos encontrar 4 tipos de formas de impresión: cartesiana, delta, polar y brazo robótico, la utilizada en esta tesis es de tipo cartesiana mostrada e la figura 5.1

A continuación es mencionado los pasos a seguir para la creación de un objeto:

1. Modelado Digital

Se crea el diseño del objeto a imprimir por medio de un software CAD,



Figura 5.1: Impresora 3D de tecnología FDM de tipo cartesiana.

en esta tesis se utiliza freeCAD por ser un programa gratuito y sencillo de usar.

2. Exportación

El objeto previamente diseñado por el software CAD se guarda como un archivo STL (Standard Triangle Language) u OBJ (3D Object File).

3. Importación y procesamiento

Utilizando un software para impresión 3D se abre nuestro archivo STL u OBJ se ajusta y configura para una óptima impresión y se crea un archivo G-Code (lenguaje utilizado en las impresoras 3D para trabajar).

4. Impresión y acabado

El G-Code generado se envía a la impresora 3D para ser impreso, ya terminada la impresión se retiran los excesos de material (dependiendo de la tecnología utilizada).

5.2. FreeCAD un programa dedicado a la creación y modificación de objetos

Como se menciona anteriormente FreeCAD es un programa de software libre para la creación y modificación de objetos CAD. A continuación se menciona el proceso del diseño de un objeto.

Dentro de FreeCAD nos dirigimos a la pestaña "Start" y se selecciona la herramienta que convenga, la finalidad de la tesis no es mencionar la creación de todo tipo de objetos, pero se describe de manera simple la creación de objetos con figuras geométricas y para ello se selecciona la opción *Part* como se muestra en la imagen 5.2.

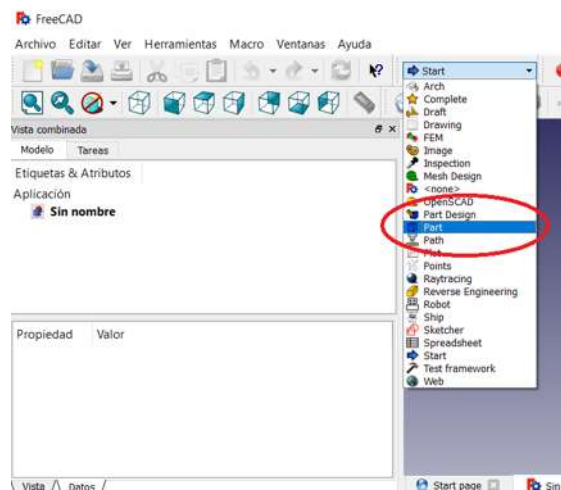


Figura 5.2: Programa FreeCAD: se muestra la pestaña Start para la creación de distintos diseños CAD remarcando la opción Part para la creación a partir de figuras geométricas.

En la figura 5.3 se muestran herramientas del lado izquierdo que permite controlar la dirección de observación del objeto, desde ver el objeto en el espacio tridimensional hasta ver algunos de sus planos, en la misma imagen en la parte central se encuentran herramientas para la creación de figuras geométricas que a su vez permite la construcción de objetos mas complejos.

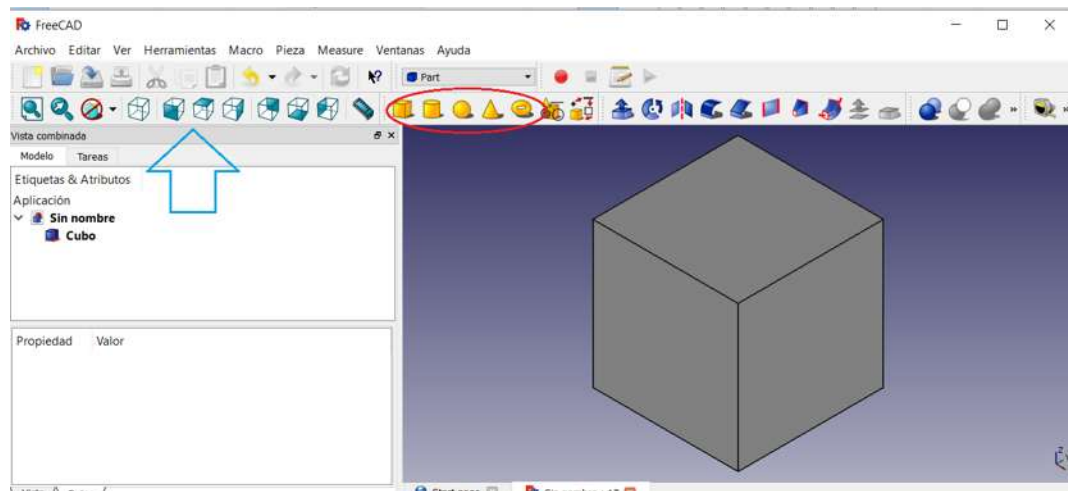


Figura 5.3: Programa FreeCAD: la flecha azul indica las opciones para cambiar la visualización del objeto, del lado derecho se remarca en rojo los objetos tridimensionales.

Al generar una nueva figura esta aparecerá en el origen sin embargo se puede trasladar a cualquier lugar en el espacio como se muestra en la figura 5.4 y también se puede modificar la longitud de todos sus lados, cuando se tiene mas de una objeto se puede unir, puede también generar una figura entre la intersección o incluso hacer un corte entre figuras. Todas estas opciones se encuentran en la parte superior derecha como se muestra en la figura 5.5; en la figura 5.6 se muestra como quedaría un cubo con un corte de otro cubo de las mismas medidas pero trasladado del origen.

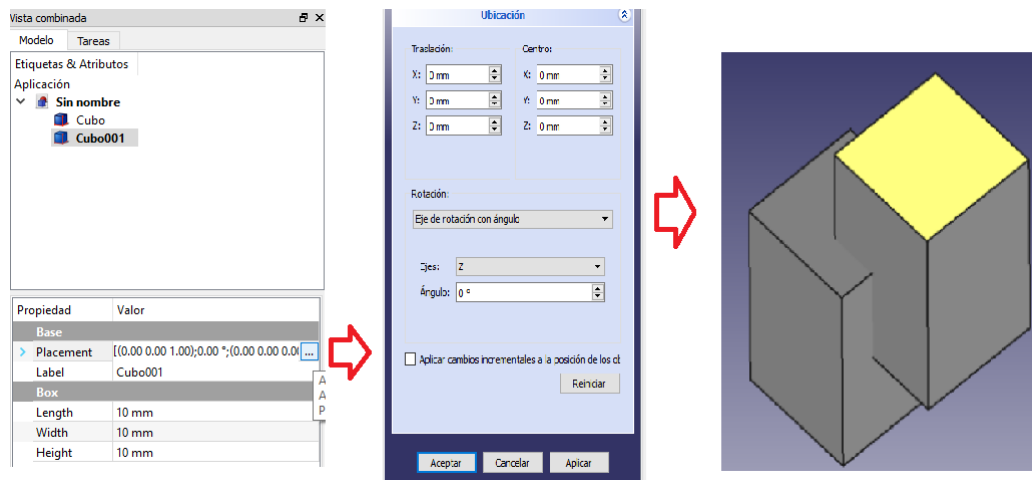


Figura 5.4: Programa FreeCAD: se muestra los pasos a seguir para trasladar un objeto.

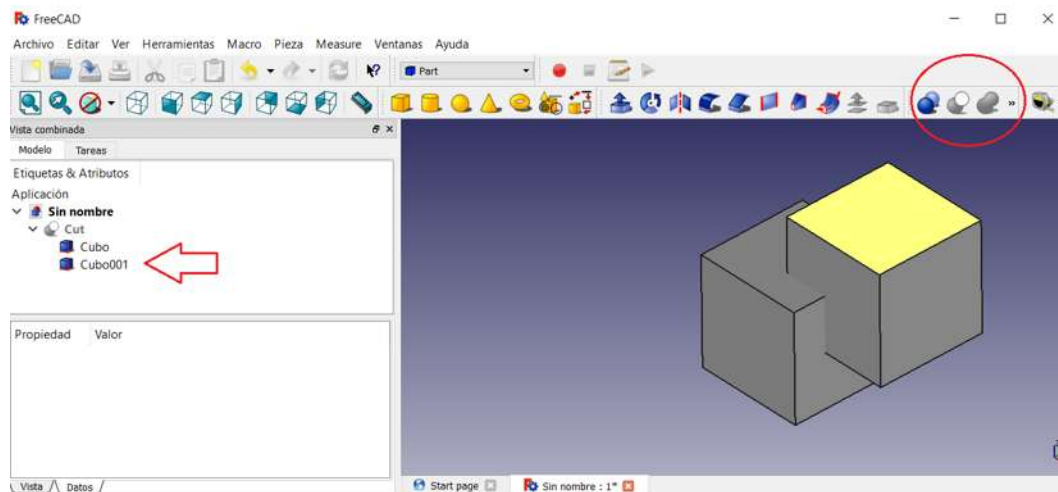


Figura 5.5: Programa FreeCAD: se muestra a la izquierda los objetos llamados cubo y cubo001, para conseguir un corte en forma del cubo001 es necesario seleccionar ambos cubos y hacer click sobre "hacer un corte de dos formas" que es la segunda instrucción marcada por el círculo rojo.

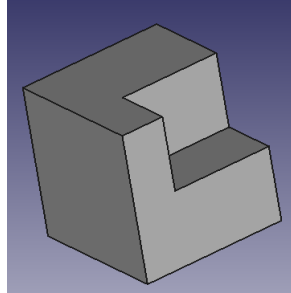


Figura 5.6: Programa FreeCAD: se muestra como quedaría al final de seguir las instrucciones de la figura 5.5.

A continuación se menciona como exportar un archivo STL, primero seleccionamos el objeto que desea ser exportado, y se dirige a archivo, después a exportar y por ultimo se guarda con el nombre deseado así como se muestra en la figura 5.7

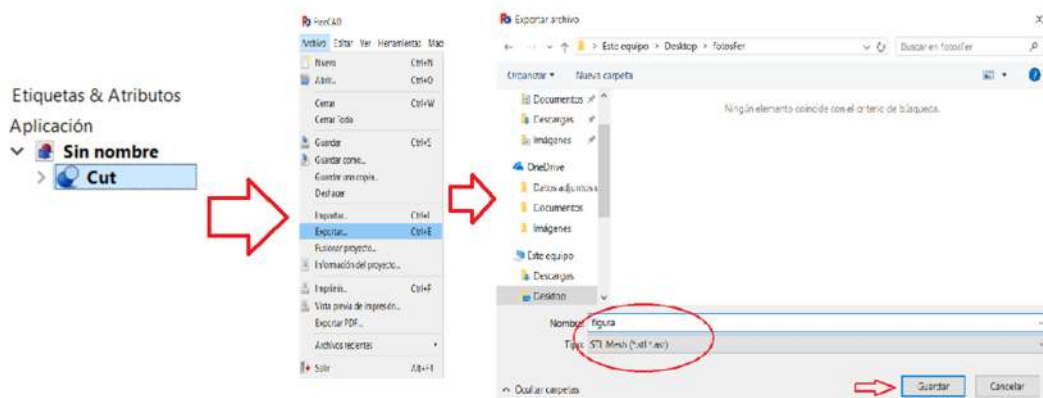


Figura 5.7: Programa FreeCAD: se muestra como exportar un objeto.

Hay ocasiones que al tener un objeto muy complejo o pesado que necesite ser arreglado o modificado, FreeCAD puede no ser tan eficiente y llegué a trabarse, en estos casos se recomienda un programa para reparar archivos 3D, existe una variedad de estos algunos son gratuitos como por ejemplo el programa de reparación archivos 3D de Microsoft *Microsoft 3D Printing* que repara automáticamente el archivo, o también Meshmixer que es un pro-

grama para editar y preparar archivos .stl y .obj.

5.3. Repetier Host comunicación entre la impresora y la computadora

Repetier Host es un software para impresión 3D, con este programa permite la obtención de un archivo G-Code a partir de un STL. Se menciona a continuación como cargar un archivo STL, la conversión a G-Code y la impresión.

Se arrastra nuestro archivo STL desde la carpeta donde se encuentra hasta la pantalla de Repetier Host como se muestra en la figura 5.8, otra manera de cargar un archivo STL es desde la pestaña *archivo* seleccionamos *archivo*, se selecciona el archivo STL que se desea cargar, ya cargado el archivo se puede manipular de tal manera que se puede escalar, rotar, girar de tal manera que quede en optimas condiciones para ser impreso, al ser la impresora cartesiana está misma imprimirá de abajo hacia arriba por lo que podemos tener problemas en áreas voladizas, para estos casos el programa tiene la opción de colocar soportes donde cree que habrá problemas. En otros programas para impresión como *Simplify3D* nos da la opción de elegir donde colocar dichos soportes, el ancho, la altura y la cantidad. A continuación se menciona como manejar la impresora 3D de dos formas: la primera solo se necesita una tarjeta de memoria SD que contenga nuestro objeto G-Code y se imprime directamente en la impresora y la segunda que se explicara mas detalladamente a continuación.

Ya cargado el archivo STL se podrá ver en la parte cuadriculada el objeto, conectando la impresora al programa se podrá controlar tanto movimiento como la temperatura, una vez este calibrada la impresora, en la figura 5.9 se muestra la pestaña *Print Preview* donde se tiene la cantidad que se gasta de filamento, el tiempo estimado, el numero de capas, entre otras características de la impresión. Luego de verificar dicha información se hace click en imprimir para empezar el proceso de impresión.

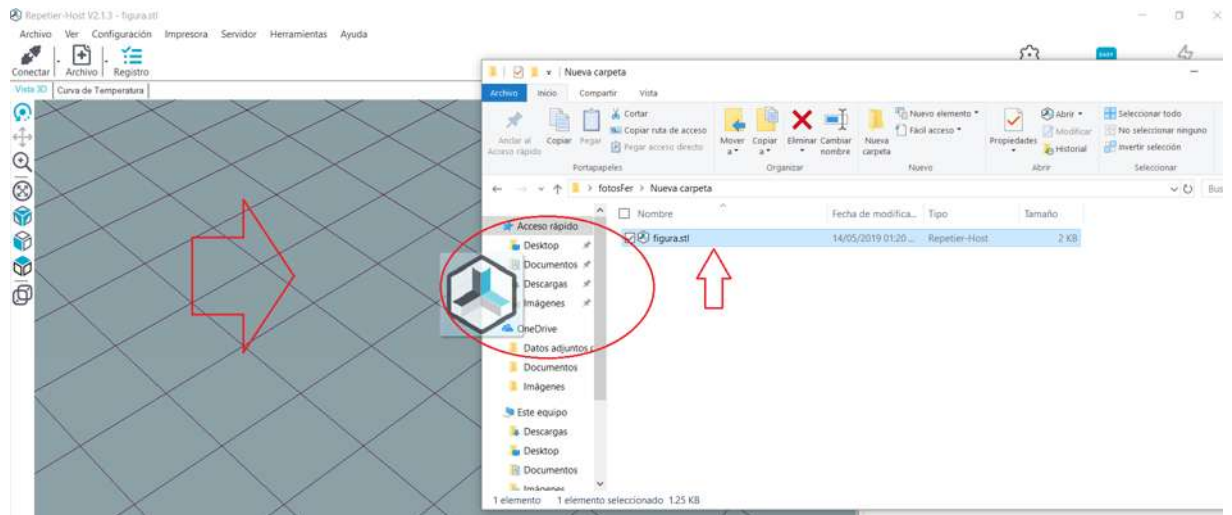


Figura 5.8: Programa Repetier Host: se puede cargar un archivo STL solo arrastrando el archivo a la pantalla cuadrículada de Repetier Host.

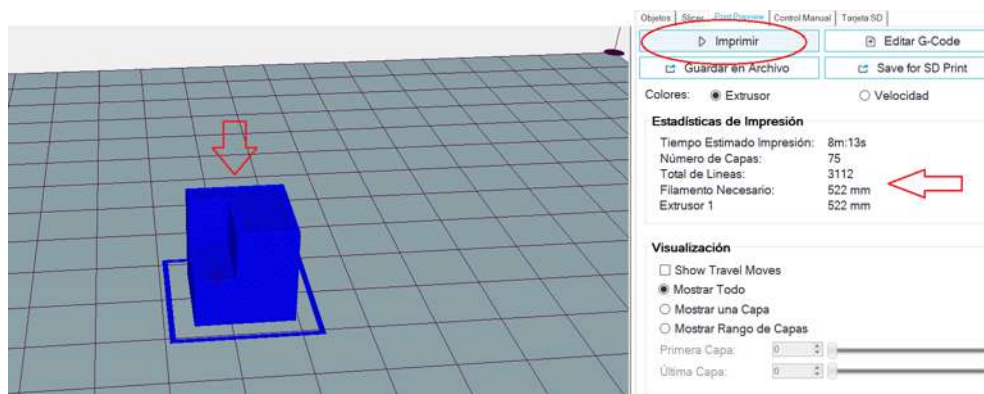


Figura 5.9: Programa Repetier Host: se muestra la pestaña Print Preview donde se tiene la información de la cantidad de filamento que se llevara, el tiempo estimado, el numero de capas entre otras.

5.4. Partes de una Impresora 3D de tecnología FDM tipo cartesiana y calibración

Las partes que tiene una impresora 3D de tecnología FDM tipo cartesiana, se muestra en la figura 5.10, dichas partes son las siguientes:

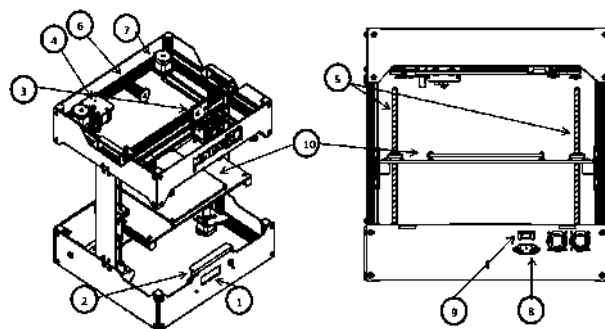


Figura 5.10: Impresora 3D: se muestra una impresora FDM tipo cartesiana de la marca Mexicana MINIFAB3D, tomada del manual de dicha impresora.

1. Pantalla LCD:

Visualiza la temperatura de la cama y del cabezal extruso, la velocidad de impresión, la altura de impresión, porcentaje de lectura del archivo (SD), el tiempo transcurrido. A partir de ella se puede ajustar algunos parámetros. En el caso de la MINIFAB3D se puede operar sin estar imprimiendo, entre estas operaciones está la función para apagar los motores que retira la tensión de los motores dando facilidad de movimiento manual del cabezal de impresión, *llevar al origen* ésta opción lleva el cabezal de impresión al origen, *precalentar PLA o ABS* esta opción sirve para que la máquina precaliente hasta la temperatura de funcionamiento de dicho material, *enfriar* esta opción provoca un corte de la energía con los calentadores y deje de calentar, *mover ejes* esta opción permite mover en dirección x, y, z del extrusor. También se puede modificar los parámetros durante la impresión, otra de las cosas que se pueden ajustar es en el caso de impresión con memoria SD, dando un menú para seleccionar el archivo.

2. Ranura tarjeta SD:
Permite la lectura de una tarjeta SD.
3. Hotend:
El hotend está formado por un disipador de calor, un bloque calefactor, sensor de altura, la boquilla nozzle y un ventilador hotend, el hotend derrite el material termoplástico para después sacarlo por la boquilla nozzle.
4. Extrusor:
El extrusor es la primera parte que toca el filamento, además de que es encargado del control del mismo, en algunas impresoras el extrusor y el hotend están juntos en una sola pieza.
5. Eje z:
Controla por medio de dos motores el movimiento en dirección z el eje vertical, de la cama caliente.
6. Soporte filamento:
Como su nombre lo dice sirve como soporte para el filamento termoplástico.
7. Motores a pasos:
Se encargan de transmitir el movimiento, a través de correas dentadas o varillas roscadas, a las partes móviles de la impresora.
8. Toma de corriente.
9. Botón de encendido.
10. Cama caliente:
Es la que se encarga de soportar y hacer que se adhiera por medio de una temperatura controlada, la pieza a imprimir.

Para imprimir un objeto es importante la calibración de la impresora, una buena calibración implica que el objeto salga de buena calidad. La calibración depende del tipo de impresora, ya que varía entre marcas y tipos de

impresora, se mencionará a continuación los pasos de una de las formas para calibrar una impresora *MINIFAB3D*:

1. Se verifica que la cama se encuentre a la misma altura con respecto al Hotend como se muestra en la figura 5.11, es decir que tanto el lado izquierdo de la cama como el derecho se encuentre a la misma altura, en caso de que esto no sea así primero se apaga los motores luego se ajusta la altura manualmente girando la tuerca que se muestra en la figura 5.12.
2. Se lleva a Home el Hotend, este mismo se acomodara enfrente del lado izquierdo donde se marcará el origen y partiendo de ahí se pasará a calibrar el sensor localizado en el Hotend, este sensor ajustará la boquilla nozzle, entre más abajo se encuentre el sensor la boquilla estará más arriba con respecto a la cama, para que este bien equilibrada, la boquilla debe encontrarse al ras de la cama, de tal manera que al pasar una hoja esta pase con un muy leve forcejeo entre la cama y la boquilla, cabe mencionar que cada vez que se mueva el sensor se lleva a Home el Hotend.
3. Se pasará a calibrar las esquinas de la cama, una vez calibrada la boquilla con el origen, se dirige el Hotend a una de las esquinas una vez ahí se pasará la hoja entre la cama y la boquilla, debe quedar exactamente igual que en el origen, en dado caso de que no lo sea se pasará a mover los tornillos localizados en las esquinas (figura 181) de la cama para hacer ajustes muy finos, se repetirá el proceso para las otras 2 esquinas.
4. Completado el paso 3, pasaremos a escribir el comando G29 (auto-calibración) un con comando en el lenguaje de las impresoras, dicho comando es para que se calibre automáticamente, en caso de calibrar sin la ayuda de un programa como el Repetier Host, deberá cargarse

este comando desde la memoria SD que a su vez estará cargado dentro del G-Code.

5. Una vez completado los los pasos anteriores se llevará a Home el Hotend y estará listo para la impresión.

En caso de que se este imprimiendo desde la SD, solo se tendrá que seguir los pasos del 1-3.

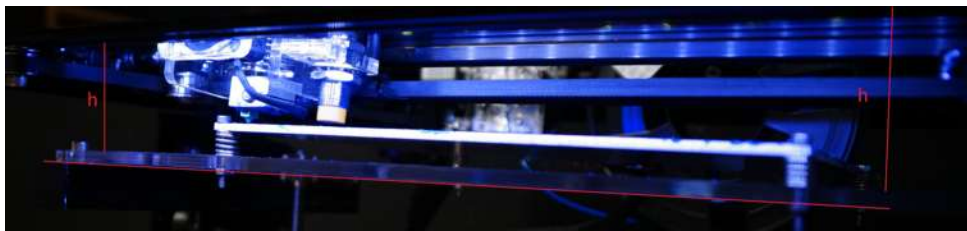


Figura 5.11: Impresora 3D: la distancia entre la cama y el Hotend debe ser la misma tanto del lado derecho como izquierdo, como la cama es la unica con movimiento en z, se tomó de referencia un lugar fijo de la impresora que esta en el mismo plano que el Hotend.



Figura 5.12: Impresora 3D: Se muestra una de las tuercas que mueve la cama en dirección z.

5.5. Impresión de imágenes en 3D

Como se mencionó anteriormente la mayoría de los programas de impresión usan formatos STL (Standard Tessellation Language), los mas básicos (sin color) se caracterizan por definir la geometría de los objetos tridimensionales exclusivamente con caras triangulares y, suelen escribirse en representación ASCII y también en binario para ficheros mas compactos.

Todo sólido STL debe cumplir la regla vértice-a-vértice esto consiste en que cada triángulo debe compartir un vértice con cada uno de sus triángulos adyacentes, (ver figura 5.13a) ss decir, el vértice de un triángulo no puede descansar sobre dos vértices de los triángulos adyacentes como se muestra en la figura 5.13b.

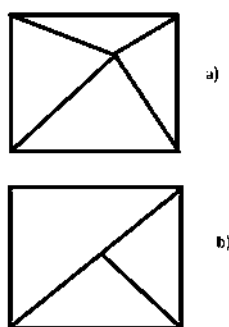


Figura 5.13: Regla vértice-a-vértice: En la imagen *a)* muestra un ejemplo de la regla vértice-a-vértice mientras que en *b)* se muestra la manera incorrecta.

La normal de cada cara es un vector unitario que apunta hacia el exterior del objeto sólido, el propio software el que calcula automáticamente una normal basada en el orden de los vértices del triángulo, los cuales siguen la regla de la mano derecha.

Existen diversos programas que permiten el cambio de formato de imágenes por ejemplo BMP, DICOM, JPG, etc. al formato STL, sin embargo no se explica a detalle este proceso ya que no es la finalidad de ésta tesis, pero si

se menciona brevemente de manera elemental y simplificada en que consiste.

La mayoría de programas que transforman de un tipo de formato a otro tomando mayor interés en los que transforman formato STL trabajan de manera diferente, sin embargo como se mencionó anteriormente el formato STL son imágenes tridimensionales representadas de forma triangular, los programas utilizan diferentes algoritmos entre si para transformar cada píxel, voxel, puntos, etc. en un conjunto de triángulos; esta transformación puede ser automática, manual o ambos para después tener el recuento de triángulos reflejado en el modelo físico final.

El proceso de formación de la imagen 3D hasta la fase de prototipado rápido se puede dividir en tres pasos básicos:

1. Adquisición de datos.
2. Procesamiento de imagen.
3. Fabricación del modelo.

Cada paso es una fuente potencial de errores geométricos y distorsiones en el modelo final. En el paso dos tenemos el procesamiento de la imagen que consiste en la triangulación de dicha imagen.

La triangulación de la imagen es la creación de una malla de triángulos a partir de un conjunto de puntos, para esto hay diferentes algoritmos de triangulación que se usan para hacer teselados de geometrías donde teselado es un patrón de figuras geométricas.

Por mencionar alguno tenemos el algoritmo Ear Clipping que consiste en el simple hecho de que un polígono con al menos 4 vértices sin agujeros tiene como mínimo dos *orejas* (éars), las cuales son triángulos, separados por la arista interna que atravesaría al polígono uniendo dos vértices que no pertenecieran a una misma arista del perímetro. El algoritmo consiste en encontrar una oreja y quitarla del polígono, lo cual daría como resultado otro polígono al que volver a aplicar el algoritmo hasta que sólo quedara un triángulo como se muestra en la figura 5.14.

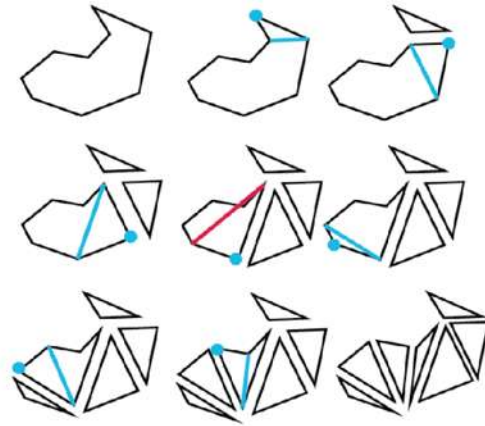


Figura 5.14: Ejemplo del algoritmo Ear Clipping, consiste en la triangulación de un polígono sin huecos. [6]

Existen diferentes algoritmos más complejos para la triangulación para evitar huecos u otros tipos de problemas. Cabe mencionar la existencia de errores por ello es importante antes de la manipulación del objeto final (la imagen 3D con formato STL) la reparación de dicho objeto y para ello existen diversos programas dedicados solo a reparar archivos STL.

Capítulo 6

Diseño experimental y Resultados

En este capítulo se describe el arreglo experimental y los pasos que se realizaron durante el proceso de obtención en 3D del objeto analizado así como sus dificultades. Se utiliza el método de Fourier para la reproducción digital en 3D realizado en MatLab y una impresora MINIFAB para la reconstrucción del objeto entre otros programas que se mencionaran en este capítulo.

6.1. Arreglo experimental

En la Figura 6.1 se muestra una fotografía del arreglo experimental que consiste en un cañón EPSON modelo H553A y una cámara SONY $\alpha 77II$ alineadas sobre el mismo plano, las franjas proyectadas a través del cañón son programadas en MATLAB y consiste en rejillas cosenoidales y binarias cuyo programas se muestra en el apéndice A. La frecuencia de la rejilla empleada para realizar la proyección se rige por la ecuación (2.3.1), sin embargo, debido a la dificultad que presenta la descripción matemática de los objetos reales que se emplean en este trabajo, fue necesario probar con rejillas de diferentes frecuencias hasta obtener aquella que reprodujera de mejor manera la superficie del objeto.



Figura 6.1: Fotografía del arreglo experimental, cuenta que muestra un cañón EPSON modelo H53A y una cámara SONY $\alpha 77II$.

Para el correcto desarrollo de este experimento es altamente recomendable que el mismo sea realizado en un cuarto oscuro o con un mínimo de iluminación posible ya que puede ocasionar ruido al experimento. Para llevar a cabo este trabajo fueron necesarias dos fotografías por objeto. La primera de ellas es aquella donde se proyecta únicamente la rejilla sobre la pantalla blanca, a la que llamaremos imagen de referencia y, aquella donde se proyecta la rejilla sobre el objeto de estudio a la que llamaremos imagen de estudio como se muestra en la figura 6.2. Cabe mencionar que la frecuencia de la rejillas fue de $155 \text{ lin}/m$ en el caso de la semiesfera y $145 \text{ lin}/m$ para la cara.

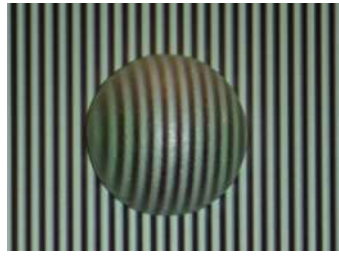


Figura 6.2: Fotografía de la rejilla proyectada sobre el objeto de estudio, con una frecuencia de la rejillas de $155 \text{ lin}/\text{m}$.

Los objetos empleados a los que se digitalizó su superficie se muestran en la Figura 6.3 que consiste en un rostro de unicel de 12.4 cm de ancho por 19 cm de largo con una altura de 7 cm y la mitad de una esfera del mismo material de radio 3.85 cm.



Figura 6.3: Fotografía de los objetos estudiados que consiste en un rostro de unicel con medidas de 12.4 cm de ancho, 19 cm de largo y una semiesfera del mismo material de radio 3.8 cm.

Una vez que las imágenes han sido digitalizadas se tiene que obtener su fase envuelta y posteriormente desenvolverla para obtener el relieve del objeto después se transforma dicha imagen en formato STL para poder convertirlo en G-code e imprimir dicho objeto. Para entender claramente todo el proceso de envoltimiento, desenvolvimiento y obtención de nuestro objeto material se presenta los pasos relevantes a seguir de cada uno de los objetos presentados

en esta tesis.

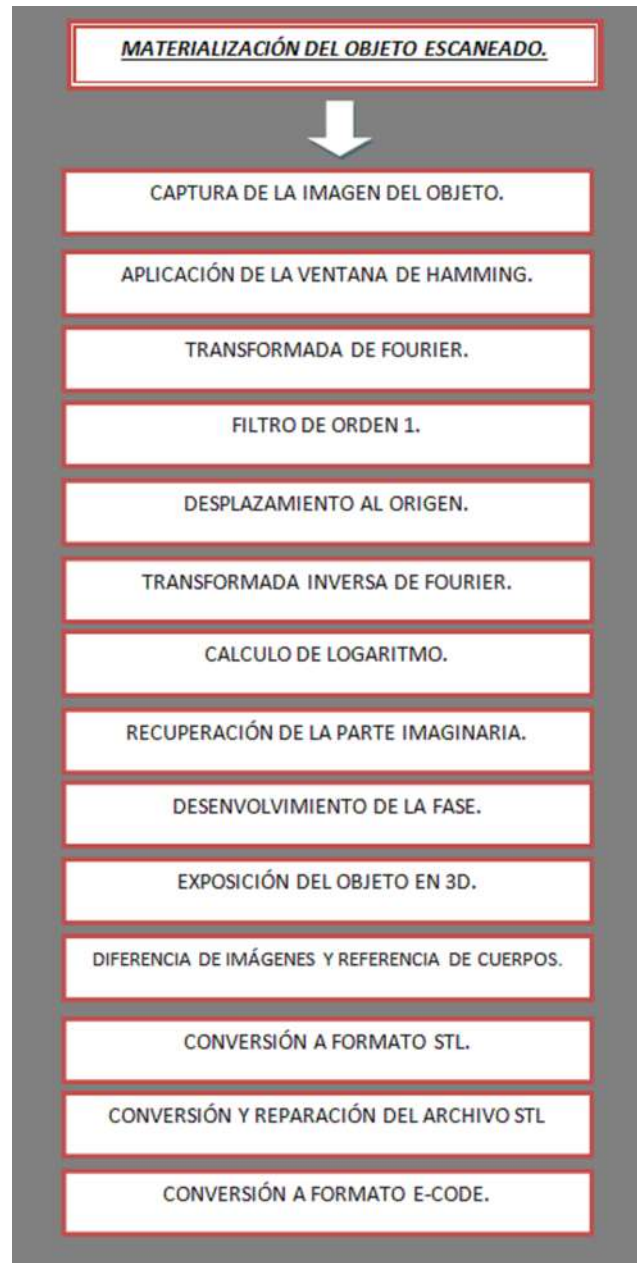


Figura 6.4: Pasos relevantes para escanear, reproducir y obtener nuestro objeto.

A continuación se presenta el algoritmo para superficies en 3D de cada

una de las imágenes que se utilizaron en esta tesis.

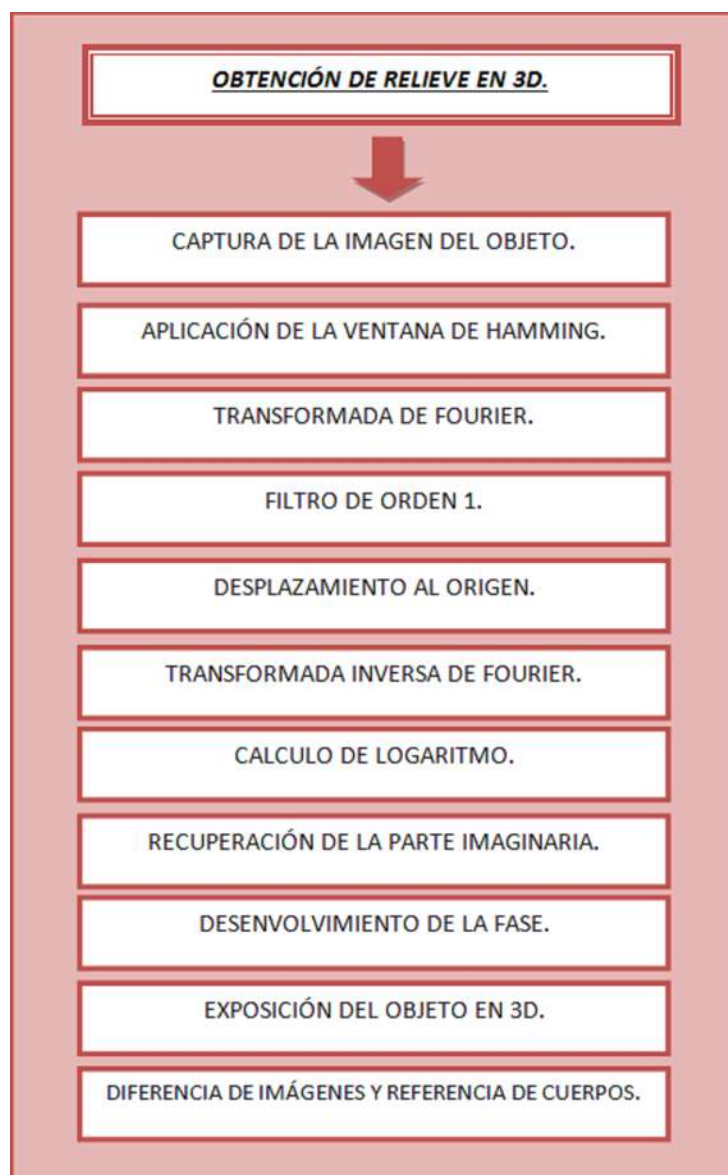


Figura 6.5: Diagrama del algoritmo para la recuperación de la fase envuelta de un objeto.

6.2. Reproducción digital de superficies en 3D

Los objetos utilizados fueron una cara y una semiesfera ambas de unicel, ya que el procedimiento empleado para la reproducción digital en 3D es la misma para ambos objetos se explicará para la semiesfera y para la cara solo se mostrará los resultados.

Una vez proyectada la rejilla binaria se toman las dos fotografías idénticas para la imagen de referencia y la imagen de estudio, después se importan a la computadora con formato JPG, con ayuda de Matlab se recorta las dos imágenes del mismo tamaño en el mismo lugar para ahorrar trabajo de cómputo, de ser una imagen de 3376X6000 pixeles pasa a ser de 835X853 pixeles, además se cambió la imagen a escala de grises con formato BMP esto último puesto que el formato .bmp utiliza una sola matriz para almacenar sus datos y el uso de escala de grises de 8 bits asegura que la matriz contenga en cada uno de sus registros solo el valor de intensidad de luz que le corresponde. [4]

Una vez teniendo nuestras figuras 6.6 y 6.7, y debido a que el procedimiento para determinar la imagen de fase desenvuelta es el mismo tanto para la imagen de referencia como para la imagen de estudio, se presenta solamente los resultados obtenidos por el algoritmo para la imagen de estudio.

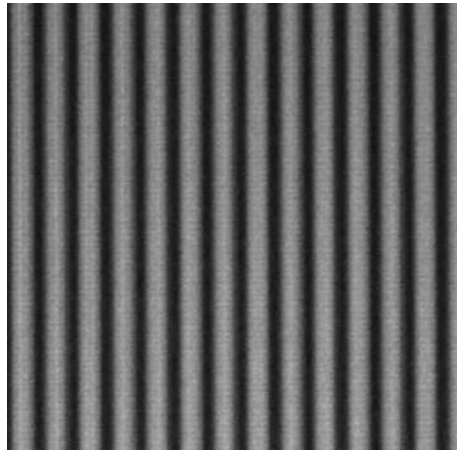


Figura 6.6: Imagen de referencia de 835X853 pixeles.

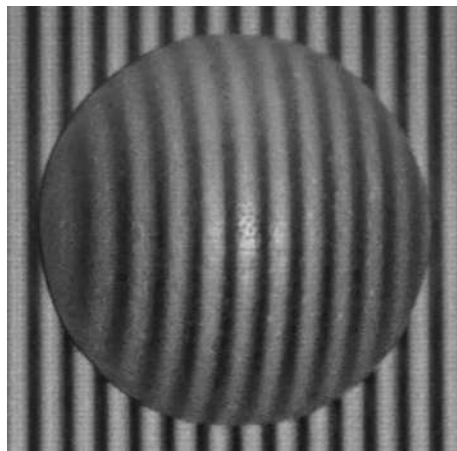


Figura 6.7: Imagen de estudio de 835X853 pixeles con la semiesfera de radio 3.8.

Primero se aplica el filtro de ventana a nuestra imagen para eliminar los efectos de borde, dicho algoritmo se encuentra en el Apéndice A. Después de aplicar el filtro a la imagen se calcula la transformada de Fourier el resultado de este cálculo se muestra en la Figura 6.8.

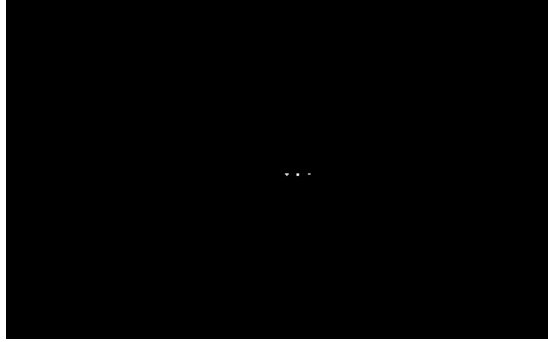


Figura 6.8: Transformada de Fourier de la imagen 6.7.

Para el filtraje del orden 1 se emplea el código que se muestra en el Apéndice A.3, el cual localiza el orden 1 aparte permite determinar el tamaño del filtro, posteriormente se recorre dicho orden al origen de nuestro sistema. Después pasamos a calcular la Transformada Inversa de Fourier (IFT) que se muestra en la figura 6.9 de la cual se recupera únicamente la parte imaginaria.

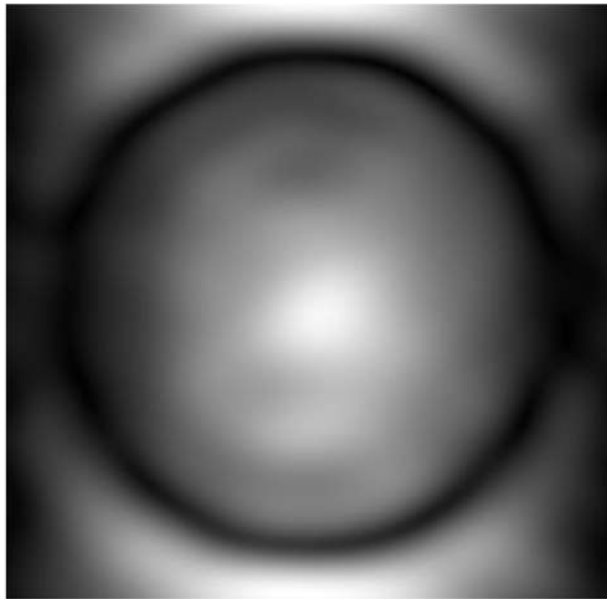


Figura 6.9: Transformada Inversa de Fourier de nuestra imagen 6.8.

Tras el cálculo del logaritmo de la IFT se tiene la figura 6.10 que corres-

ponde a la imagen de fase envuelta del experimento.



Figura 6.10: Parte imaginaria del logaritmo de la imagen de Transformada Inversa de Fourier.

Después se pasa a desenvolver la fase con ayuda de el comando predefinido de Matlab (`unwrap`), el código se muestra en el apéndice A.1. Se muestra nuestra imagen desenvuelta en la figura 6.11, como se puede apreciar en la imagen se encuentra en un plano inclinado, para eliminar dicho plano se determina la diferencia de la imagen de fase desenvuelta de la imagen de estudio y la imagen de referencia.

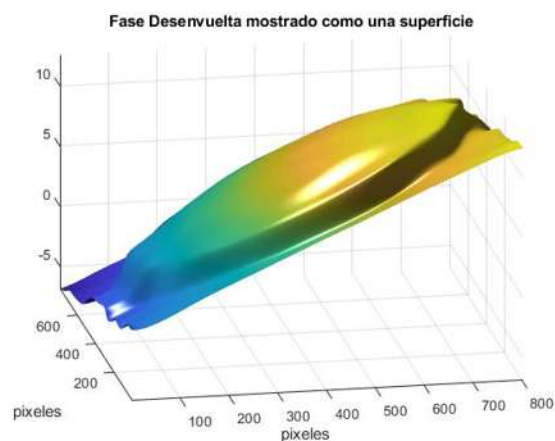


Figura 6.11: Fase desenvuelta mostrado como una superficie.

Entre mas exactas sea las dos imágenes (imagen de estudio e imagen de referencia) mejor será la reconstrucción, por eso se cuenta con la gran ventaja de que sea fija la pantalla, así solo se remueve el objeto de estudio, si las imágenes no son lo mas exactas posible las diferencias en los planos de estas imágenes generarán una deformación en la superficie. Cabe mencionar que puede existir la presencia de ruido en la imagen resultante, esto debido a que el plano de las dos imágenes empleadas no es idéntico por errores en la toma fotográfica.

Se pasa a hacer exactamente lo mismo que se hizo con la imagen de estudio pero con la imagen de referencia, con la misma información (Transformada de Fourier, recorte, filtrado) para poder hacer la diferencia de las imágenes, los resultados se muestran en las figuras 6.12 y 6.13. En la figura 3.14 se presenta el resultado para la cara de unicel.

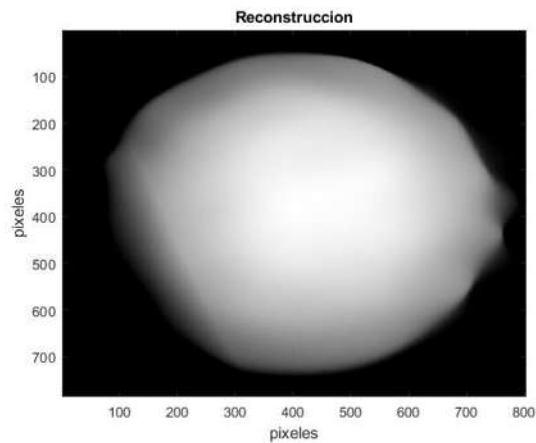


Figura 6.12: Imagen de fase desenvuelta para la semiesfera mostrada como arreglo de intensidades.

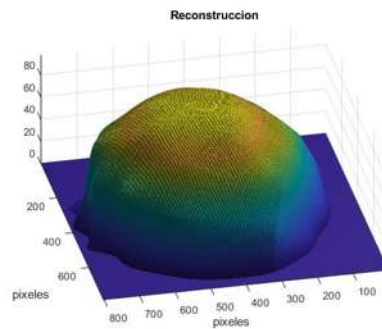


Figura 6.13: Imagen de fase desenvuelta para la semiesfera mostrada en 3D.

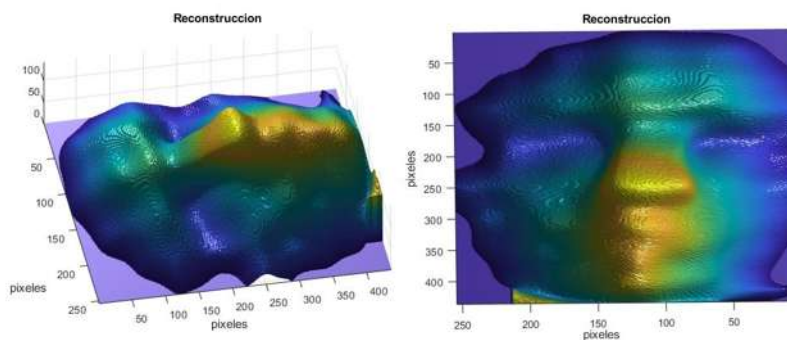


Figura 6.14: Imagen de fase desenvuelta para la cara mostrada en 3D.

6.3. Materialización del objeto con ayuda de una impresora 3D

Se materializa los objetos antes mencionados (semiesfera, cara), utilizando la imagen de fase ya desenvuelta como arreglo de intensidades figura 6.12, guardada con formato BMP y usando el programa gratuito Online (Image to Lithophane) podemos transformar nuestra imagen a una imagen 3D con formato STL, como se muestra en la figura 6.15.

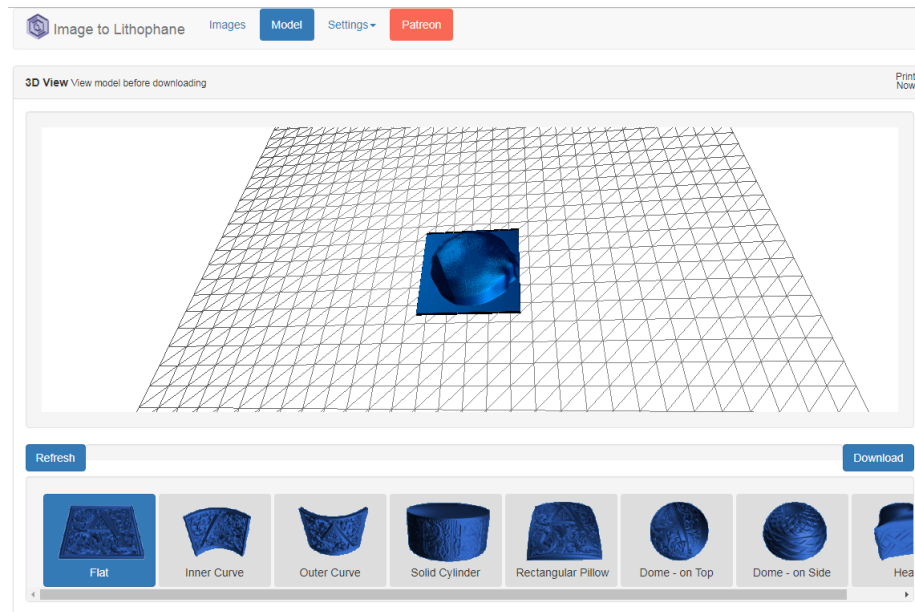


Figura 6.15: Imagen 3D de la semiesfera recién convertida a formato STL.

Después de obtener la imagen en 3D se recomienda arreglar nuestro archivo, ya que en la mayoría de los casos suele tener errores. Se usó el programa gratuito Meshmixer y con ayuda del programa Repetier-Host se logra imprimir el objeto en la impresora 3D, como se muestra en las figuras 6.16-6.19.

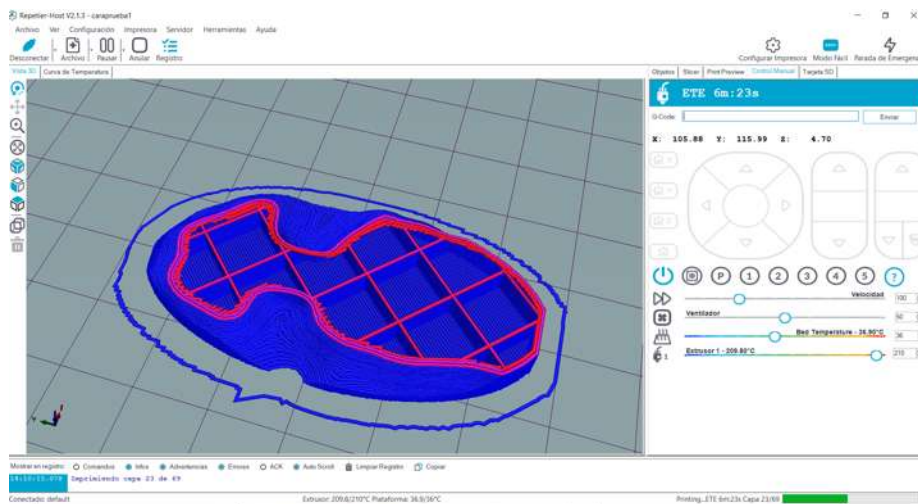


Figura 6.16: Se muestra la imagen del funcionamiento Repetier-Host con la cara.

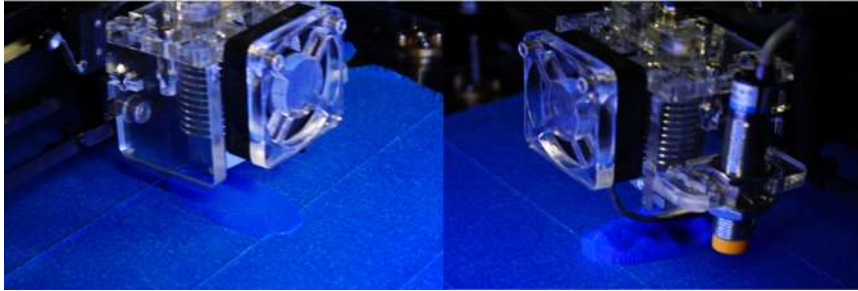


Figura 6.17: Se muestra la imagen del funcionamiento de la impresora MINIFAB con la cara.

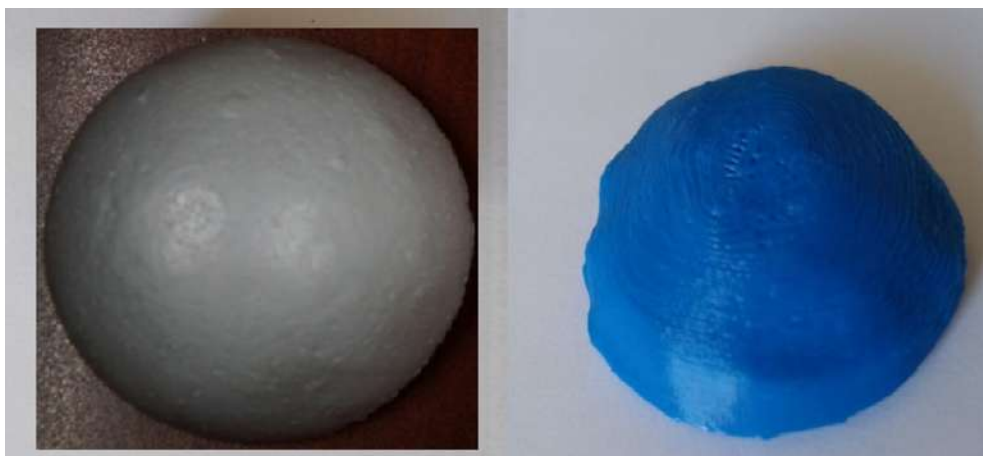


Figura 6.18: Se muestra la fotografía de la semiesfera ya impresa.



Figura 6.19: Se muestra la fotografía de la cara ya impresa.

Los resultados obtenidos son satisfactorios sin embargo, la calidad de la reproducción como se muestra en la figura 6.19 que se pierde detalles del rostro, como la boca entre otros, es debido a los errores tanto a la hora de digitalizar como se muestra en la figura 6.14 como rompimiento de franjas, ruido, etc, como las limitaciones en el equipo, es necesario mayor fuerza de computo, para analizar una imagen de mayor tamaño y así aumentar la frecuencia de la rejillas, así como a la hora de imprimir ya que la calidad de la impresora es baja con una boquilla de 0.4 mm y una resolución de posicionado de 0.05 mm .

Capítulo 7

Conclusiones

En esta tesis se propuso utilizar la proyección de luz estructurada sobre una semiesfera y una cara de unicel, para después ser reproducidos digitalmente e impresos con ayuda de la impresora 3D.

La proyección de luz estructurada se realizó con un cañón EPSON modelo H553A conectado a la computadora, donde se desplegó una rejilla osenoidal diseñado en Matlab. Las frecuencias de las rejillas utilizada fueron de 155 *lin/m* para la semiesfera y de 145 *lin/m* para la cara.

La captura de las imágenes de las rejillas proyectadas sobre los los objetos se realizó con una cámara SONY α 77II. El formato captado por la cámara es JPG y se transforma en blanco y negro para que se pueda utilizar en Matlab.

Para reconstruir la superficie en 3D se empleó el método de Takeda. Se presentó el algoritmo de Itoh que está desarrollado en MATLAB que tuvo la finalidad de desenvolver dicha fase para posteriormente reproducir tridimensionalmente la topografía de los objetos de estudio. Para este fin se diseño un programa en Matlab para tal propósito en el cual se utilizó los comandos FFT e IFT para obtener la transformada inversa de la transformada de Fourier. Además también se utilizó el comando `unwrap` para el desenvolvimiento de la transformada. Posteriormente se guardo nuestra superficie como un arreglo de intensidades como se muestra en la figura 6.12 en formato BMP,

y con ayuda de software se construyó el objeto en formato STL y finalmente se imprimió en ácido poliláctico (PLA) que es un termoplástico.

El arreglo experimental es de fácil montaje, en general está compuesto por pocos elementos y gracias al sistema de cómputo se cuenta con un amplio rango de frecuencia de muestreo y una variedad de herramientas con las cuales manipular las imágenes además el uso de software de impresión es amigable con el usuario y de fácil uso.

Los resultados obtenidos son satisfactorios sin embargo, para mejorarlos se debe crear un programa más robusto que evite errores en el procesamiento de la imagen provocado por ruido, rompimiento de fase, etc. También es importante mencionar que la calidad de la impresora es baja, la resolución de posicionado que tiene es de 0.05 mm con boquilla de 0.4 mm según característica del fabricante, sin embargo, en la práctica no es posible imprimir un objeto con buena precisión, por lo que se recomienda, para trabajos futuros emplear una impresora de mayor calidad.

Las técnicas empleadas son de fácil implementación en el laboratorio y tiene un gran potencial de aplicaciones dentro de la cultura, medicina, ciencia e ingeniería así como en la vida cotidiana.

Apéndice A

Códigos de MATLAB

A.1. Código para el desenvolvimiento de fase

Código para el desenvolvimiento de la fase y visualización del objeto.
Desenvolvimiento de una imagen utilizando el segundo método del algoritmo de Itoh.

```

1%Primero se desenvuelven todas las columnas.
2image2_unwrapped=L;
3for i=1:nn
4image2_unwrapped(i,:)=unwrap(image2_unwrapped(i,:));
5end
6
7%Despues todas las filas
8for i=1:nn
9image2_unwrapped(:,i)=unwrap(image2_unwrapped(:,i));
10end
11
12%visualizando
13R=image2_unwrapped;
14Q=-1.*R(25:nn-25,25:nn-25).*(1);
15figure, colormap(gray(256)), imagesc(Q) %gray 256
16title('Reconstruccion')
17xlabel('pixeles'), ylabel('pixeles')
18figure
19surf(Q,'FaceColor','interp','EdgeColor','none','FaceLighting','
    phong')

```

```

20view(-30,30),camlight left , axis tight
21title('Fase Desenvuelta mostrado como una superficie')
22xlabel('pixeles'), ylabel('pixeles'), %label('fase (rad)')
23figure
24surf(Q,'EdgeColor','none','FaceLighting','gouraud')
25view(-30,30),camlight right , axis tight
26box on
27grid off
28axis on
29colormap([0.5 0.5 0.5])

```

A.2. Código para la generación de rejillas

Los siguiente código permite generar las rejillas necesarias para generar una serie de rejillas de frecuencia variable.

```

1
2%Rejilla binaria
3n=600;
4m=500;
5a=3; %frecuencia del 1-200
6b=0;
7z=zeros(m,n); %m*n
8for j=1:m
9    b=0;
10   for i=1:n
11       b= b + 1;
12       if(b>a)
13           z(j,i)=1;
14       end
15       if (b>2*a)
16           b=0;
17       end
18   end
19end
20figure
21imshow(z)
22

```

```

23 %Rejilla cosenoidal
24 n=600;
25 m=500;
26 db=400; %frecuencia del 100-2,000
27 a=10000;
28 J=zeros(m,n); %m*n
29 for j=1:m
30     s=0;
31     for i=1:n
32         s= s + b;
33         if(s>a)
34             s=0;
35         end
36         J(j,i)=0.5+0.5*cos((s/a)*2*pi);
37     end
38 end
39 figure
40 imshow(J)

```

A.3. Código para generar el filtro para el orden 1

Con este código se crea el filtro de orden 1, permite introducir el tamaño, forma del filtro y la posición del mismo, así como centrar después de aplicarlo a la transformada de Fourier.

```

1 %Filtro
2 %Creacion del filtro elipctico
3 x=10;
4 y=15; % si x=y sera un circulo
5 %se tendra que sacar la infomacion previa de la posicion del
   orden 1 de la TF
6 a=442; %rasladar eje x segun la TF
7 b=420; %rasladar eje y segun la TF
8 sr=0.5; %en caso de un circulo r sera el radio
9 z=zeros(m,n);

```

```
1z1=zeros(m,n);
2k=0;
3for i=1:m
4    w=x*sqrt(r^2-((i-b)/y)^2)+a;
5    w2=-x*sqrt(r^2-((i-b)/y)^2)+a;
6    for j=1:n
7        if j<w
8            z(i,j)=1;
9        end
10       if j>w2
11           z1(i,j)=1;
12       end
13   end
14end
15z2=z.*z1;
16figure
17imshow(z2)
18z2=double(z2);
19y3=double(y3); % y3 es nuestra TF
20z3=z2.*y3;
21figure
22imshow(100*z3)
23
24%centrar la transformada
25M1=z3(:,1:a-n/2+1);
26M2=z3(:,a-n/2+2:n);
27zc=[M2,M1];
28[mm,nn]=size(zc);
29figure
30imshow(100*zc)
```

Referencias

- [1] Kjell J Gåsvik. *Optical metrology*. John Wiley & Sons, 2003.
- [2] Nikolaus Karpinsky and Song Zhang. High-resolution, real-time 3d imaging with fringe analysis. *Journal of Real-Time Image Processing*, 7(1):55–66, 2012.
- [3] Mitsuo Takeda, Hideki Ina, and Seiji Kobayashi. Fourier-transform method of fringe-pattern analysis for computer-based topography and interferometry. *JosA*, 72(1):156–160, 1982.
- [4] Jorge Arieas. Obtención de relieves y reconstrucción de objetos en 3d utilizando proyección de luz estructurada. *Univercidad Michoacana de San Nicolás de Hidalgo*, pages 1–72, 2013.
- [5] Bárbara Aline Gerke, Amanda Lury Yamashita, Eder Alberto Sigua-Rodriguez, Sergio Olate, Lilian Cristina Vessoni Iwaki, and Liogi Iwaki-Filho. Análisis descriptivo y cualitativo de tres software gratuitos usados para la conversión de formato dicom para stl. *International journal of odontostomatology*, 13(1):103–111, 2019.
- [6] Isabel Gracián Torrent. Conversor brep-stl. *Universitat Politècnica de Valencia*, 2015.
- [7] Adrian Bermejo Pulgar et al. Adquisición, tratamiento y validación de modelos 3d mediante luz estructurada. *Univercidad de Valladolid*, 2015.
- [8] Kazuyoshi Itoh. Analysis of the phase unwrapping algorithm. *Applied optics*, 21(14):2470–2470, 1982.

- [9] Munther Gdeisat and Francis Lilley. One-dimensional phase unwrapping problem. *signal*, 4:6, 2011.
- [10] Gdeisat, Munther, and Francis Lilley. two-dimensional phase unwrapping problem. *signal*, 4:6, 2011.