

Caracterización experimental de sistemas ópticos. Reto: Diseño de un escáner 3D

Isaac Alexis López Paredes* and Eduardo Juárez Hernández†

Tecnológico de Monterrey

(Dated: March 19, 2024)

En este informe se presenta la implementación del método de proyección de franjas para la obtención de la topografía de un objeto. Las franjas se proyectaron utilizando dos rejillas superpuestas para producir un patrón de Moaré. Se realizó la recuperación de fase utilizando el método de cambio de fase con 4 pasos. Los perfiles de intensidad se procesaron en el software de **MATLAB®**, en donde se ajustó el contraste y el brillo de las imágenes, se desarrolló la fase utilizando el algoritmo de Constantini y se calculó la altura del objeto. Con ello se logró reconstruir satisfactoriamente la superficie tridimensional de una media esfera. Finalmente, se discuten las limitaciones del método y posibles mejoras.

Keywords: escáner 3D, proyección de franjas, efecto moaré

I. OBJETIVO

El objetivo de este reporte es presentar los resultados del diseño de un escáner 3D. Para ello se aplicó el método de proyección de franjas para la obtención de la topografía de un objeto. El documento está organizado de la siguiente manera: en la sección II se describe la técnica utilizada y el arreglo experimental implementado; en la sección III se presentan los principios físicos y matemáticos relevantes para la obtención de resultados; en la sección V se presenta el análisis de los resultados obtenidos; en la sección VI se muestra el cálculo del error con respecto a los resultados esperados y finalmente se presentan las conclusiones en la sección VII

II. DESCRIPCIÓN

En la técnica elegida para la medición de una superficie 3-D, se proyecta un patrón de franjas regular sobre la superficie del objeto que luego es visto desde otra dirección. Las franjas proyectadas están moduladas por la topografía del objeto. Luego se toma una muestra de la distribución de intensidad del patrón de franjas deformadas, tal como se muestra en el plano, utilizando el sensor CCD de una cámara. Este perfil es procesado para recuperar la distribución de fases resultante. Ahora bien, existen varias técnicas de medición de fase. En este caso, se utilizó el método de cambio de fase que se basa en adquirir un conjunto de n imágenes correspondientes a n cambios de fase conocidos de la rejilla proyectada [3].

En este proyecto se eligió $n = 4$, con lo cual se deben obtener 4 imágenes, cada una con un desfase de $\frac{\pi}{2}$ entre ellas. La recuperación de fase del objeto se debe

comparar después con un plano de referencia. Para ello también se registran 4 imágenes con el perfil de intensidad de las franjas proyectadas sobre el plano de observación sin el objeto. Esta técnica ofrece ventajas como la alta precisión, la capacidad de medición en tiempo real y la no necesidad de contacto con la superficie.

Para este trabajo, la proyección de franjas se obtuvo por medio del *efecto Moaré*. Este efecto se puede obtener con la superposición de dos rejillas primarias, con lo cual se obtienen franjas de Moaré, como se muestra en la figura 1. Estas rejillas, fueron hechas en **MATLAB®** e impresas en acetato.

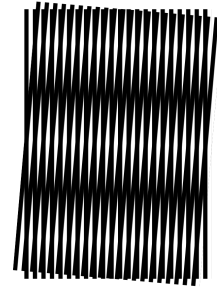


Figure 1: Patrón de Moaré por superposición de dos rejillas primarias

La figura 2 muestra el arreglo óptico utilizado para implementar la técnica descrita anteriormente. Los elementos que componen el arreglo son:

1. Plano de observación. Una superficie plana donde se coloca el objeto de interés
2. Una fuente de luz blanca. Colocada a un ángulo $\alpha = 10^\circ$ con respecto a la normal del plano de observación
3. Una cámara. Colocada a un ángulo $\beta = 5^\circ$ con respecto a la normal del plano de observación

* Correspondence email address: A01747148@tec.mx

† Correspondence email address: A01749005@tec.mx

4. Rejillas binarias superpuestas. Colocadas en un rodillo milimétrico alineado con el ángulo de la fuente de luz blanca.

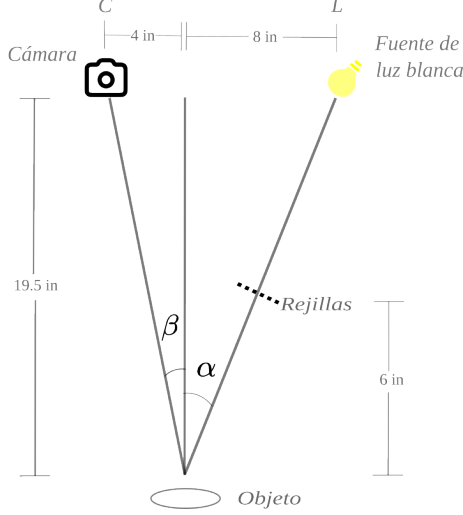


Figure 2: Arreglo óptico para el escáner

III. MARCO TEÓRICO

En primer lugar, es importante mencionar que es necesario obtener un factor que permita escalar los píxeles en las imágenes a centímetros. Esto con el fin de visualizar los resultados en las dimensiones reales del objeto y el plano de observación. Estas dimensiones D_x y D_y se obtienen como:

$$D_x = p_x \cdot \frac{l}{p_{x\text{adjust}}} \quad (1)$$

$$D_y = p_y \cdot \frac{l}{p_{y\text{adjust}}}, \quad (2)$$

donde p_x es la cantidad de píxeles de la imagen en las columnas, p_y es la cantidad de píxeles de la imagen en las filas, l es la longitud real del objeto, $p_{x\text{adjust}}$ es la cantidad de píxeles en las columnas de la imagen ajustada a las dimensiones reales del objeto, y $p_{y\text{adjust}}$ es la cantidad de píxeles en las filas de la imagen ajustada a las dimensiones reales del objeto.

Además, es necesario obtener la fase envuelta ϕ del plano de incidencia con el objeto y sin el objeto. Entonces en cada desplazamiento de $\frac{\pi}{2}$ se registran 4 intensidades, tal que las intensidades registradas son las siguientes [1], [5]:

$$I_1 = I_A + I_B + 2\sqrt{I_A I_B}(\cos(\phi)) \quad (3)$$

$$I_2 = I_A + I_B + 2\sqrt{I_A I_B}(-\sin(\phi)) \quad (4)$$

$$I_3 = I_A + I_B + 2\sqrt{I_A I_B}(-\cos(\phi)) \quad (5)$$

$$I_4 = I_A + I_B + 2\sqrt{I_A I_B}(\sin(\phi)), \quad (6)$$

por lo que al final se obtiene que ϕ es [3], [5]:

$$\phi = \arctan\left(\frac{-(I_2 - I_4)}{I_1 - I_3}\right) \quad (7)$$

Además, para encontrar la altura $z(x, y)$ del objeto es necesario encontrar una relación entre la cantidad de franjas p en un periodo a en una distancia AB . Así como la cantidad de franjas q en el periodo a en una distancia AC como se muestra en la figura 3,

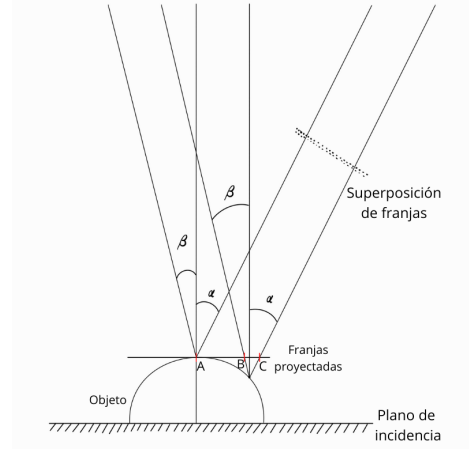


Figure 3: Esquema del arreglo óptico

Estas relaciones se pueden encontrar como:

$$AB = pa \quad (8)$$

$$AC = qa \quad (9)$$

$$\therefore LC = AC - AB = (q - p)a = Na \quad (10)$$

donde N es la cantidad de franjas proyectadas, y LC es la distancia entre la fuente de luz y la cámara (figura 2). Pero LC también es igual a :

$$LC = LP' + P'C, \quad (11)$$

donde LP' es la distancia de la fuente de luz al eje central del objeto, y $P'C$ es la distancia del eje central del objeto a la cámara. Entonces LP' y $P'C$ siguen las relaciones:

$$\tan(\alpha) = \frac{LP'}{z} \rightarrow LP' = z \tan(\alpha) \quad (12)$$

$$\tan(\beta) = \frac{P'C}{z} \rightarrow P'C = z \tan(\beta). \quad (13)$$

Igualando las ecuaciones 10 y 11 se obtiene que:

$$Na = z(\tan \alpha + \tan \beta) \quad (14)$$

$$\therefore z(x, y) = \frac{Na}{(\tan(\alpha) + \tan(\beta))} \quad (15)$$

Ahora bien, como N es la diferencia de franjas entre $AC - AB$, este se puede reescribir como la diferencia de fase Φ de las franjas en el plano con el objeto y sin el objeto[4], dividido entre 2π (debido a que la diferencia de fase fue hecha para un ciclo completo). En resumen, la topografía del objeto $z(x, y)$ se puede calcular como:

$$z(x, y) = \frac{\Phi}{2\pi} \frac{a}{(\tan \alpha + \tan \beta)}. \quad (16)$$

IV. PROCEDIMIENTO

A. Arreglo óptico y captura de intensidades

Lo primero que se realizó fue montar el arreglo óptico tal como se describió en la sección II. Una vez hecho esto, lo siguiente fue realizar la caracterización del corrimiento de fase. Para esto se movieron las rejillas superpuestas con el rodillo milimétrico. Este instrumento tiene una graduación en la que cada vuelta al rodillo (50 marcadores), las franjas se desplazan medio milímetro. Primero, se colocó un punto de referencia en un máximo de intensidad y comenzamos a desplazar las rejillas hasta llegar de nuevo a otro máximo de intensidad. Lo que se registró es que entre máximo y máximo habían 5 vueltas las cuales son equivalentes a 2.5 mm o 250 marcas en el rodillo. Este resultado lo dividimos entre 4 para obtener cuantas marcas tendremos que avanzar para registrar cada una de las intensidades, obteniendo como resultado 62.5 marcas.

Una vez hecho esto, se corroboró que al dar una vuelta de 2π el punto de referencia regrese de nuevo a encontrarse en un máximo de intensidad. Cuando se aseguró esto, se comenzó en ese mismo punto a registrar las cuatro intensidades de las franjas en el plano de referencia. Posteriormente, se colocó ahora el objeto a analizar en el plano de incidencia de tal manera que las franjas incidieran sobre el objeto. En este caso se utilizó una media esfera de unicel de 1.5 cm de radio. De manera similar, se registró cada intensidad con la misma relación de 4 pasos.

Finalmente, se colocó una regla de manera horizontal y vertical para tener mediciones del objeto real y obtener un factor de escalamiento entre los píxeles de la imagen y las medidas reales en cm. Con esto, también es posible medir la frecuencia de franjas por centímetro y su período, el cual es indispensable para obtener la superficie del objeto.

B. Rutina computacional

Una vez tomadas las fotografías de las intensidades con la media esfera y sin la media esfera, lo siguiente fue realizar una rutina computacional para recuperar la fase y consecuentemente recrear en 3 dimensiones el objeto. Para ello se utilizó el software de **MATLAB®**.

En el anexo se muestra un diagrama de flujo del procedimiento general que se siguió. Lo primero que se realizó, fue cargar las imágenes y recortarlas para solamente procesar el área de interés. Posteriormente se probaron las funciones de **MATLAB®** disponibles para mejorar el contraste y el brillo de la imagen. Estas son *imadjust*, *histeq*, *adapthisteq* e *imlocalbrighten*.

Posteriormente se aplicaron las ecuaciones 1 y 2 para obtener las dimensiones reales. Después se encontró la fase envuelta a partir de la ecuación 7. En seguida se desarrolló la fase aplicando el algoritmo de *Costantini phase unwrapping*, implementado por Bruno Luong [2]. Una vez obtenidas la fase desenvuelta del objeto y del plano de referencia, se calcula la diferencia de fase ϕ como:

$$\phi = \phi_{\text{obj}} - \phi_{\text{ref}} \quad (17)$$

donde ϕ_{obj} es la fase desenvuelta del objeto y ϕ_{ref} es la fase desenvuelta del plano de referencia. Con esto se puede obtener la altura, utilizando la ecuación 16. A esta altura se le aplicó un filtro gaussiano utilizando la función de **MATLAB®** *imgaussfilt*.

Posteriormente se aplicó un máscara para quedarse solamente con la esfera reconstruida. Esto se hizo estimando el centro de la esfera, y multiplicando la altura por 1 en la región delimitada por un círculo de radio $r = 1.5$ y centrada en las coordenadas estimadas, y por 0 en la región fuera del círculo. Esto permitió eliminar el ruido, y ciertos picos en la altura que se obtenían por brillos o sombras en las imágenes fuera de la esfera. A esta superficie con máscara se le calculó el máximo para encontrar la altura de la esfera reconstruida. Con este radio se volvió a aplicar una máscara, pero con un radio igual a la altura obtenida. Finalmente, se visualizan los resultados, los cuales se describen en la siguiente sección.

V. ANÁLISIS DE RESULTADOS

Después de cargar las imágenes y recortarlas para analizar el área de interés, estas se guardaron en un archivo **MAT**. En la figura 4 se muestran dichas fotos que sirvieron como entrada para el algoritmo. Entre cada una, las franjas tienen un desfase de $\frac{\pi}{2}$, como se mencionó anteriormente.

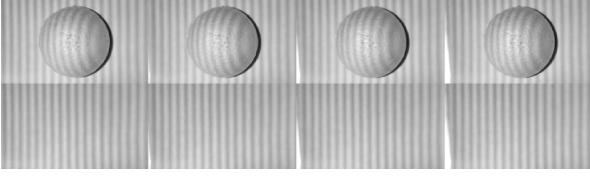


Figure 4: Fotos capturadas con el corrimiento de fase

En la figura 5 se muestran las distintas opciones que se probaron para mejorar el contraste de las fotos y obtener un mejor resultado. La primera imagen es la foto original, y las siguientes tres muestran el resultado de aplicar las funciones integradas de MATLAB: *imadjust*, *histeq* y *adapthisteq* respectivamente.



Figure 5: Comparación de técnicas de MATLAB para mejora de contraste

De estas tres funciones disponibles, se seleccionó la función *imadjust*, con un parámetro $\gamma=0.8$, el cual especifica la forma de la curva que describe la relación entre los valores de la intensidad de entrada y la intensidad de salida. Además, se aplicó la función *imlocalbrighten*, con un brillo de 0.2. En la figura 6 se muestra el resultado después de aplicar ambas funciones. Esto se realizó para las 8 imágenes de entrada.



Figure 6: Mejora de contraste y brillo

Posteriormente se calculó la escala en centímetros en el eje transversal, con el fin de tener una referencia adecuada de las dimensiones del objeto. En la figura 7 se muestra el objeto con la escala real en centímetros.

Después se realizó el envoltimiento y desenrollamiento de fase. En la figura 9 y 8 se muestran las fases envueltas y desenrolladas del objeto y del plano de referencia respectivamente.

A continuación, se aplicó la ecuación 16 para encontrar la altura, y se aplicó un filtro gaussiano con la función *imgaussfilt*, con una desviación estándar de 8. En la figura 10 se muestra la superficie obtenida. Se puede observar que aun después de aplicar el filtro, queda ruido en las áreas que no son de interés. Por ello se aplicó la máscara.

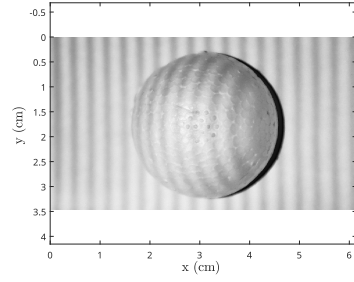


Figure 7: Objeto con escala real en plano transversal

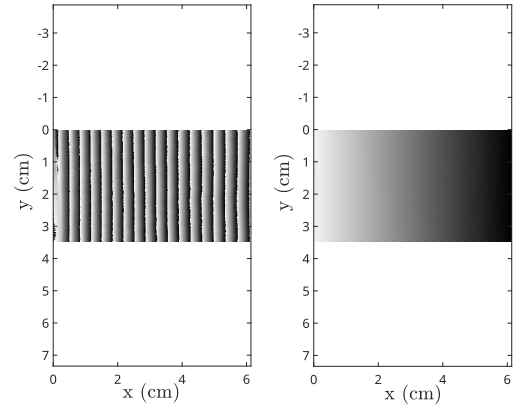


Figure 8: Desenrollamiento de fase del plano de referencia

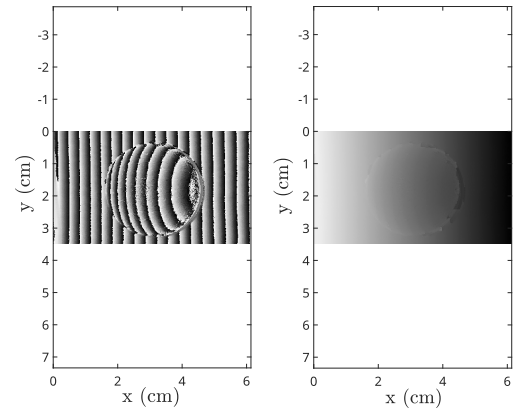


Figure 9: Desenrollamiento de fase del objeto

La figura 11 muestra la superficie después de aplicar la máscara con el radio teórico, mientras que la figura 12 muestra la máscara con el radio que se obtuvo a partir de la superficie reconstruida.

Calculando el máximo de esta superficie con máscara, se obtuvo una altura de 1.2524 cm. En la figura 13 se

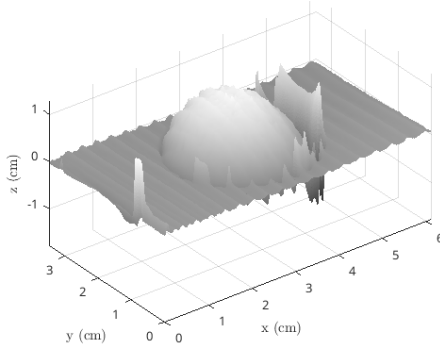


Figure 10: Superficie escalada y después de aplicar el filtro gaussiano

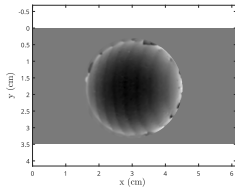


Figure 11: Máscara de prueba

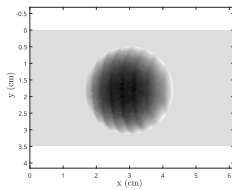


Figure 12: Máscara con radio obtenido

muestra la superficie final reconstruida, donde el punto negro tiene coordenadas (3,1.8,1.2524).

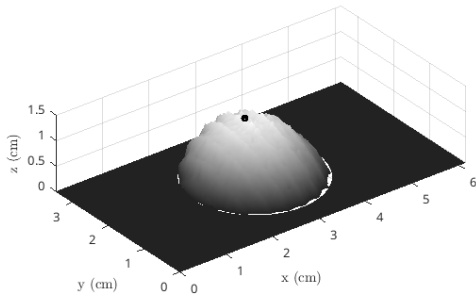


Figure 13: Superficie con máscara

VI. ANÁLISIS DE ERRORES

Se calcularon tres errores, primero el error entre las alturas máximas. El error absoluto en este caso fue:

$$Err_1 = |1.5 - 1.2524| = 0.2476 \quad (18)$$

Así mismo el error relativo porcentual fue:

$$Err_2 = \frac{|1.5 - 1.2524|}{1.5} \cdot 100 = 16.5039\% \quad (19)$$

Por otro lado se calculó el error absoluto promedio entre la superficie teórica y la superficie reconstruida. En la figura 14 se muestra la esfera con el radio teórico de 1.5 cm y en la figura 15 se muestra el error descrito, que se calculó con:

$$Err_3 = |Z_m - Z_t| \quad (20)$$

donde Z_m es la superficie con máscara (figura 13) y Z_t es la esfera teórica. El promedio de esta matriz fue de 0.17838.

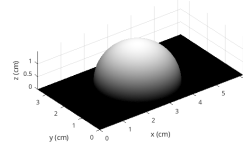


Figure 14: Esfera teórica

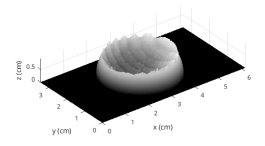


Figure 15: Error absoluto

VII. CONCLUSIONES

El procedimiento se llevó a cabo con éxito. Se logró obtener la reconstrucción tridimensional del objeto con una forma adecuada, aunque con un error considerable en la altura. Estos errores pueden deberse a varios factores. En primer lugar, no se pudo eliminar por completo la sombra que produce el objeto en el plano de observación, lo cual puede causar errores en la reconstrucción. Además, durante el proceso de corrimiento de fase, fue difícil minimizar el error experimental, pues las mediciones son muy sensibles a movimientos tanto de las rejillas, como del plano de observación.

Por otro lado, el método elegido tiene algunas limitantes. Al utilizar rejillas binarias superpuestas, se debe utilizar luz blanca. Las franjas generadas de esta forma tienen menor contraste que aquellas generadas por una fuente coherente como un láser. Además, la resolución de impresión de las rejillas, que en este caso fue de 300 dpi (*dots per inch*), puede jugar un papel en el contraste de las franjas y consecuentemente en la reconstrucción.

En cuanto a posibles mejoras, se podrían reducir tanto β como α . Con esto se podría minimizar la sombra. Además, durante las distintas pruebas que se

hicieron se observó que con ángulos β y α grandes se obtenían peores resultados. Por otro lado, en el algoritmo, los parámetros de contraste, brillo y desviación estándar del filtro gaussiano se optimizaron de forma manual, por lo que se podría decir que estos no fueron

el problema. También se probaron varios algoritmos de desdolvimiento de fase, y el de Constantini fue el que dio mejores resultados. Entonces, es probable que los resultados se puedan mejorar con franjas de mayor contraste, mejores ángulos y un corrimiento de fase más preciso.

-
- [1] E. Hecht. *Optics*. Addison Wesley, San Francisco, 2002.
 - [2] B. Luong. Costantini phase unwrapping, Aug 2009.
 - [3] A. Martínez, J. Rayas, H. Puga, and K. Genovese. Iterative estimation of the topography measurement by fringe-projection method with divergent illumination by considering the pitch variation along the x and z directions. *Optics and Lasers in Engineering*, 48(9):877–881, 2010.
 - [4] D. W. Robinson and G. T. Reid. Interferogram analysis: digital fringe pattern measurement techniques. (*No Title*), 1993.
 - [5] Y. Surrel. Phase stepping: a new self-calibrating algorithm. *Appl. Opt.*, 32(19):3598–3600, Jul 1993.

Appendix: Cálculo de fase

Sabemos que la intensidad es, para interferencia, un promedio temporal del valor absoluto al cuadrado del campo E que se propaga:

$$I = \langle \vec{E}^2 \rangle_T, \quad (\text{A.1})$$

donde $\vec{E} = \vec{E}_A + \vec{E}_B$ las cuales son ondas monocromáticas de la misma frecuencia. Por lo tanto:

$$\vec{E}^2 = \vec{E} \cdot \vec{E} \quad (\text{A.2})$$

$$\langle \vec{E}^2 \rangle_T = \langle \vec{E}_A^2 \rangle_T + \langle \vec{E}_B^2 \rangle_T + 2\langle \vec{E}_A \cdot \vec{E}_B \rangle_T. \quad (\text{A.3})$$

Aplicando la ecuación A.1:

$$I = I_A + I_B + I_{AB}, \quad (\text{A.4})$$

donde I_{12} es el término de interferencia, el cual es:

$$I_{AB} = E_A \cdot E_B \cos(\delta), \quad (\text{A.5})$$

y $\delta = (\vec{k}_1 \cdot \vec{r} + \epsilon_1 - \vec{k}_2 \cdot \vec{r} - \epsilon_2)$. Si los campos son perpendiculares $I_{12} = 0$, y si son paralelos $I_{12} = 2\sqrt{I_1 I_2} \cos(\delta)$. Para el caso del reto, los campos son paralelos, entonces:

$$I = I_A + I_B + 2\sqrt{I_A I_B} \cos(\delta), \quad (\text{A.6})$$

En el caso del reto, $\delta = \phi + \Delta\phi$, donde ϕ es la fase que queremos conocer, y $\Delta\phi$ es el corrimiento de fase

que conocemos tal que $\Delta\phi_k = (k - 1)\alpha(m, n)$ donde $k = 1, 2, 3, 4$, y $\alpha = \frac{\pi}{2}$, entonces al aplicar la técnica de 4 pasos, vamos a obtener 4 intensidades. Por lo tanto:

$$I_k = I_A + I_B + 2\sqrt{I_A I_B} \cos(\phi + \Delta\phi_k) \quad (\text{A.7})$$

$$I_1 = I_A + I_B + 2\sqrt{I_A I_B} \cos(\phi) \quad (\text{A.8})$$

$$I_2 = I_A + I_B + 2\sqrt{I_A I_B} \left(\cos\left(\phi + \frac{\pi}{2}\right) \right) \quad (\text{A.9})$$

$$I_3 = I_A + I_B + 2\sqrt{I_A I_B} \cos(\phi + \pi) \quad (\text{A.10})$$

$$I_4 = I_A + I_B + 2\sqrt{I_A I_B} \left(\cos\left(\phi + \frac{3\pi}{2}\right) \right). \quad (\text{A.11})$$

Sabiendo que $\cos(a + b) = \cos(a) \cos(b) - \sin(a) \sin(b)$, entonces:

$$I_1 = I_A + I_B + 2\sqrt{I_A I_B} \cos(\phi) \quad (\text{A.12})$$

$$I_2 = I_A + I_B + 2\sqrt{I_A I_B} (-\sin(\phi)) \quad (\text{A.13})$$

$$I_3 = I_A + I_B + 2\sqrt{I_A I_B} (-\cos(\phi)) \quad (\text{A.14})$$

$$I_4 = I_A + I_B + 2\sqrt{I_A I_B} \sin(\phi), \quad (\text{A.15})$$

Nos podemos dar cuenta de que las intensidades I_1 e I_3 tienen como factor común una función coseno, mientras que las intensidades I_2 e I_4 tienen en común una función seno. Por lo tanto, si restamos las intensidades $I_3 - I_1$ y $I_4 - I_2$ respectivamente para intentar despejar a la fase ϕ , obtenemos que:

$$I_3 - I_1 = 4\sqrt{I_A I_B} \cos(\phi) \quad (\text{A.16})$$

$$I_4 - I_2 = -4\sqrt{I_A I_B} \sin(\phi) \quad (\text{A.17})$$

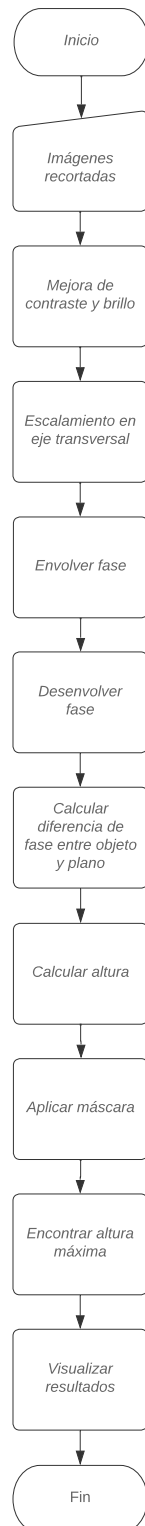
$$-(I_4 - I_2) = 4\sqrt{I_A I_B} \sin(\phi), \quad (\text{A.18})$$

Ahora, si dividimos la ecuación A.18 entre A.16 se obtiene lo siguiente:

$$\frac{-(I_4 - I_2)}{I_3 - I_1} = \tan \phi \quad (\text{A.19})$$

$$\therefore \phi = \arctan\left(\frac{-(I_4 - I_2)}{I_3 - I_1}\right) \quad (\text{A.20})$$

Appendix: Diagrama de flujo del procedimiento general de la rutina numérica



Appendix: Código implementado

```

clc; clear; close all

% Cargamos las imágenes ya recortadas
% junto con los factores de
% escalamiento y angulos del arreglo
% y visualizamos

load('WS_esfera5.mat')

% Parámetros
sigma = 8; % Desviación estándar para
% el filtro gaussiano
brillo = 0.2; % Para la funcion de
% imlocalbrighten
opcion_contraste = 1; % Función de
% MATLAB para mejorar contraste
gammaa = 0.8; % Factor en el ajuste de
% contraste para imadjust

montage(cat(3,Obj, Pln),'Size',[2 4])

%% Mejorar contraste y brillo
% Opciones de MATLAB
Obj1_imadjust = imadjust(Obj(:,:,1));
Obj1_histeq = histeq(Obj(:,:,1));
Obj1_adapthisteq = adapthisteq(Obj
(:,:,1));

% Comparación
figure;
montage({Obj(:,:,1),Obj1_imadjust,
Obj1_histeq,Obj1_adapthisteq},"
Size",[1 4])

% Mejora de contraste con la opción y
% parámetros elegidos
[OBJ,PLN] = contraste_brillo(Obj,Pln,
opcion_contraste,brillo,gammaa);

% Escalamiento en plano transversal
[xs,ys,Xs,Ys] = escalarT(OBJ(:,:,1),
factor_x,factor_y);

%% Envolver fase
Psi_obj = envolver(OBJ);
Psi_pln = envolver(PLN);

% Desenvolvimiento de fase
Phi_obj = cunwrap(Psi_obj);
Phi_pln = cunwrap(Psi_pln);

% Diferencia de fase
delta = Phi_obj - Phi_pln;

```

```

% Escalar superficie
Z = calcular_z(a,alph,bet,delta,sigma)
;

%% Aplicar máscara y encontrar altura máxima
% Primera estimación
r_t = 1.5; % radio teórico
Z_masked_prueba = enmascarar(Xs,Ys
,3,1.8,Z,r_t);
altura = max(-Z_masked_prueba(:));

figure;
imagesc(xs,ys,Z_masked_prueba), axis
equal, colormap gray;
xlabel('x (cm)',Interpreter='latex')
ylabel('y (cm)',Interpreter='latex')

% Coordenadas de la altura
[~, idx] = max(-Z_masked_prueba(:));
[fila_alt, col_alt] = ind2sub(size(Z),
idx);
CX = Xs(fila_alt, col_alt);
CY = Ys(fila_alt, col_alt);

disp("Altura: " + num2str(altura) + "
cm")

% Máscara con radio correcto
Z_masked_corregida = enmascarar(Xs,Ys
,3,1.8,Z,1.05*altura);

figure;
imagesc(xs,ys,Z_masked_corregida),
axis equal, colormap gray;
xlabel('x (cm)',Interpreter='latex')
ylabel('y (cm)',Interpreter='latex')

%% Visualización de resultados finales
ver_resultados(xs,ys,Psi_obj,Psi_pln,
Phi_obj,Phi_pln,-Z,-
Z_masked_corregida,CX,CY,altura);

%% Cálculo del error
% Esfera teórica
figure;
Zt = sqrt(max(r_t^2 - (Xs - 3).^2 - (
Ys - 1.8).^2, 0));
surf(Xs,Ys,Zt,edgecolor = "none"),
colormap gray, axis equal
xlabel('x (cm)',Interpreter='latex')
ylabel('y (cm)',Interpreter='latex')
zlabel('z (cm)',Interpreter='latex')

% Error absoluto en las alturas

```

```

figure;
err = abs(-Z_masked_corregida-Zt);
surf(xs,ys,err,edgecolor="none"), axis
equal, colormap gray
xlabel('x (cm)',Interpreter='latex')
ylabel('y (cm)',Interpreter='latex')
zlabel('z (cm)',Interpreter='latex')
disp("Error promedio: " + num2str(mean
(err(:))))

% Error relativo en la altura máxima
err_altura = abs(r_t-altura)/r_t*100;
disp("Error en la altura máxima: " +
num2str(err_altura) + "%")

%%
function [OBJ, PLN] = contraste_brillo
(Obj, Pln, opcion_contraste,brillo
,gammaa)

% Entradas:
% Obj: Arreglo con las imágenes del
objeto
% Pln: Arreglo con las imágenes del
plano de referencia
% opcion_contraste: entero entre 1 y 3
para elegir funcion de matlab
para ajustar contraste. 1-
imadjust, 2-histeq, 3-adapthisteq
% brillo: parámetro de imlocalbrighten
, número entre 0 y 1.
% gamma: parámetro de corrección para
imadjust
% Salidas:
% OBJ: arreglo con imagenes del objeto
con el contraste y brillo
mejorados
% PLN: arreglo con imagenes del plano
con el contraste y brillo
mejorados

OBJ = zeros(size(Obj));
PLN = zeros(size(Pln));

for ii = 1:4
    if opcion_contraste == 1
        OBJ(:, :, ii) = imlocalbrighten(
            imadjust(Obj(:, :, ii)
            , [], [], gammaa), brillo);
        PLN(:, :, ii) = imlocalbrighten(
            imadjust(Pln(:, :, ii)
            , [], [], gammaa), brillo);
    elseif opcion_contraste == 2
        OBJ(:, :, ii) = imlocalbrighten(
            histeq(Obj(:, :, ii)), brillo
        );
    end
end

```



```

        PLN(:,:,ii) = imlocalbrighten(
            histeq(Pln(:,:,ii)),brillo
        );
    else
        OBJ(:,:,ii) = imlocalbrighten(
            adapthisteq(Obj(:,:,ii)),
            brillo);
        PLN(:,:,ii) = imlocalbrighten(
            adapthisteq(Pln(:,:,ii)),
            brillo);
    end
end

% Comparación de antes y después de
% aplicar mejoras de contraste y
% brillo
figure;
montage({Obj(:,:,1),OBJ(:,:,1)},"Size
",[1 2])

end

function [xs,ys,Xs,Ys] = escalarT(Img,
    factor_x,factor_y)
% Escalamiento en plano transversal
% Entradas:
% Img: Matriz con perfil de intensidad
%       de franjas
% factor_x: factor de escalamiento en
%       x
% factor_y: factor de escalamiento en
%       y
% Salidas: Coordenadas en x y y en cm

px_x = size(Img,2); px_y = size(Img,1)
;
dim_x = px_x*factor_x;
dim_y = px_y*factor_y;
xs=linspace(0,dim_x,px_x);
ys=linspace(0,dim_y,px_y);
[Xs,Ys] = meshgrid(xs,ys);

figure;
imagesc(xs,ys,Img); axis equal;
colormap gray
xlabel('x (cm)',Interpreter='latex')
ylabel('y (cm)',Interpreter='latex')

end

function Psi = envolver(P)
% Encuentra la fase envuelta
% Entrada:
% P: arreglo con las 4 imágenes con
%     desfases de pi/2
% Salida:
% Psi: fase envuelta

P1 = P(:,:,1);
P2 = P(:,:,2);
P3 = P(:,:,3);
P4 = P(:,:,4);

Psi = atan2(-(P2-P4),(P1-P3));

end

function Z = calcular_z(a,alpha,bet,
    delta,sigma)
% Calcula la altura del objeto
% Entradas:
% a: periodo de las franjas (cm)
% alpha: ángulo de la fuente de
%       iluminación con respecto a la
%       normal del plano de observación (
%       rad)
% bet: ángulo de la cámara con
%       respecto a la normal del plano de
%       observación (rad)
% delta: arreglo con la diferencia de
%       fase entre el objeto y el plano de
%       referencia
% sigma: desviación estándar para el
%       filtro gaussiano
% Salida:
% Z: arreglo con altura del objeto en
%     cm (con filtro gaussiano aplicado)

z = (delta/(2*pi))*(a/(tan(alpha) + tan
    (bet)));
Z = imgaussfilt3(z,sigma);

end

function Z_masked = enmascarar(X,Y,CX,
    CY,Z,R)
% Aplica la máscara con forma circular
% Entradas:
% X: Matriz con las coordenadas x
% Y: Matriz con las coordenadas y
% CX: Coordenada X del centro del círculo
% CY: Coordenada Y del centro del círculo
% Z: Matriz con la altura de la
%     superficie
% R: Radio del círculo
% Salida:
% Z_masked: Superficie con máscara

circ = @(X,Y,R) ((X-CX).^2 + (Y-CY)
    .^2) < R^2;

```

```

Mask = circ(X,Y,R);
Z_masked = Mask .* Z;

end

function ver_resultados(xs,ys,P_env,
    P_env_ref,Q,Q_ref,Z,Z_masked,CX,CY,
    altura)
% Despliega las gráficas relevantes
% Entradas:
% xs: vector de coordenadas en x
% ys: vector de coordenadas en y
% P_env: Fase envuelta del objeto
% P_env_ref: Fase encuelta del plano
% Q: Fase desenvuelta del objeto
% Q_ref: Fase desenvuelta del plano
% Z: Superficie reconstruida
% Z_masked: Superficie reconstruida
% con máscara
% CX: Coordenada X del punto máximo de
% la superficie
% CY: Coordenada Y del punto máximo de
% la superficie
% altura: Valor en Z del punto máximo
% de la superficie

% Desenvolvimiento de fase %
% Objeto
figure;
subplot(1,2,1)
imagesc(xs,ys,P_env), axis equal,
    colormap gray
xlabel('x (cm)',Interpreter='latex')
ylabel('y (cm)',Interpreter='latex')

subplot(1,2,2)
imagesc(xs,ys,Q), axis equal, colormap
gray
xlabel('x (cm)',Interpreter='latex')
ylabel('y (cm)',Interpreter='latex')

% Referencia
figure;
subplot(1,2,1)
imagesc(xs,ys,P_env_ref), axis equal,
    colormap gray
xlabel('x (cm)',Interpreter='latex')
ylabel('y (cm)',Interpreter='latex')

subplot(1,2,2)
imagesc(xs,ys,Q_ref), axis equal,
    colormap gray
xlabel('x (cm)',Interpreter='latex')
ylabel('y (cm)',Interpreter='latex')

```

```

% Superficie escalada
figure;
surf(xs,ys,Z,'EdgeColor','none'), axis
    equal, colormap gray
xlabel('x (cm)',Interpreter='latex')
ylabel('y (cm)',Interpreter='latex')
zlabel('z (cm)',Interpreter='latex')

% Superficie con máscara
figure;
surf(xs,ys,Z_masked,'EdgeColor','none'
    ), axis equal, colormap gray
xlabel('x (cm)',Interpreter='latex')
ylabel('y (cm)',Interpreter='latex')
zlabel('z (cm)',Interpreter='latex')
hold on
scatter3(CX,CY,altura,'k','filled')
zlim([0 1.5])

end

```

Appendix: Código para imprimir las rejillas

```

function matriz = rejilla(dimension,
    lineas_por_pulgada)
% Genera una rejilla binaria
% Entradas:
% dimension: longitud de la rejilla en
% pulgadas
% lineas_por_pulgada: número de líneas
% en una pulgada

puntos = dimension*300; % Para una
    impresion a 300 dpis
num_lineas = dimension*
    lineas_por_pulgada;
puntos_por_linea = puntos/num_lineas;

if isinteger(puntos_por_linea) ==
    false
    disp("Combinacion no valida de
        dimension y líneas")
    matriz = NaN;
    return;
end

franjass = [zeros(1,puntos_por_linea/2)
    ones(1,puntos_por_linea/2)];
matriz = repmat(franjass,puntos,
    num_lineas);

name = ['rejilla_',num2str(
    lineas_por_pulgada),'lineas_',
    num2str(dimension),'in.bmp'];

```

```
location = "/home/isaac/Documentos/  
    Rejillas de difraccion/";  
filename = location + name;
```

```
imwrite(matriz,filename,"bmp");  
end
```

Appendix: Función de desenvolvimiento utilizada

Constantini phase unwrapping.