

## Perfilometría de superficies mediante franjas desfocalizadas

 R. A. Montalvo\* y P. H. Rivera

*Universidad Nacional Mayor de San Marcos, Facultad de Ciencias Físicas, Lima, Perú*

Recibido 15 Ene 2025 – Aceptado 31 Mar 2025 – Publicado 17 Abr 2025

En los últimos años, el advenimiento de la potencia computacional para propósitos de cálculo ha tenido un gran impacto en la optimización de los procesos industriales automáticos relacionados a visión computacional, *machine learning*, robótica y máquinas autónomas. En este contexto, un análisis topográfico rápido de objetos 3D que abarcan de milímetros a decímetros juega un rol importante en los procesos industriales autónomos. Desde la óptica, la perfilometría de proyección de franjas ha sido una herramienta barata y versátil para este propósito. En este trabajo, realizamos una ingeniería reversa de esas técnicas encontradas en la literatura en las últimas décadas. Construimos un sistema de adquisición y desarrollamos algunos códigos en Python para la reconstrucción de imágenes de objetos no encontrados en la literatura.

**Palabras clave:** Fase desenvuelta, perfilometría de proyección de franjas, franjas desfocalizadas, transformada rápida de Fourier.

### Surface profilometry by defocalized fringes

In the last years, the advent of computational power for calculations purposes have a great impact in the optimization of automatic industrial process relating to computer vision, machine learning, robotics and autonomous machine decisions. In this context, the fast topographic analysis of 3D objects ranging from millimeters to decimeters plays an import rol in autonomous industrial process. From the Optics, the fringe projection profilometry has been a cheap and versatile tool for this purpose. In this work, we realize an reverse engineering of these techniques found in the literature in the last decades. We build an acquisition system and developed some Python code for the image reconstruction for objects no found in the literature.

**Keywords:** Phase unwrapping, fringe projection profilometry, defocusing fringes, fast Fourier transform.

## 1. Introducción

El análisis de la proyección de franjas sobre objetos ha sido una técnica relativamente barata y rápida para determinar el relieve tridimensional de los mismos. A esta técnica se le ha dado el nombre de perfilometría mediante la proyección de franjas, FPP, por sus siglas en inglés, *Fringe Projection Profilometry* [1].

La sugerencia inicial de esta técnica consistió en el uso de imágenes de patrones sinusoidales creados numéricamente para ser proyectados a través de un proyector sobre un objeto. El patrón sinusoidal sobre el objeto al ser tomado por la cámara, no presenta la simetría sinu-

soidal respecto a la amplitud de la señal debido al  $\gamma > 1$ <sup>1</sup> que el proyector posee desde el proceso de calibración industrial realizado por el fabricante.

Esto sugirió usar otros patrones periódicos para generar las franjas, entre estos sobresale los patrones binarios. La desfocalización de los patrones binarios durante la proyección permitió obtener imágenes que presentan una simetría sinusoidal respecto a la amplitud. Pero, al procesar las imágenes usando franjas con diferentes corrimientos de fase, usando secuencialmente el proceso de *wrapping* y *unwrapping* para obtener el mapa de fases entre  $-\pi$  y  $+\pi$ . Éstos presentan, en su forma 3D, un desnivel no existente en el relieve de los objetos.

\*rmontalvob@unmsm.edu.pe

© Los autores. Este es un artículo de acceso abierto, distribuido bajo los términos de la licencia Creative Commons Atribución 4.0 Internacional (CC BY 4.0) que permite el uso, distribución y reproducción en cualquier medio, siempre que la obra original sea debidamente citada de su fuente original.

<sup>1</sup>Relación entre la señal de entrada y la brillantez percibida en la imagen.



Para resolver este problema, se sugirió un procedimiento numérico partiendo de los patrones binarios y del corrimiento de fases para determinar unos patrones denominados *gray-code* que permitía corregir ese desnivel producido por inestabilidades numéricas producidas durante el proceso de *wrapping* y *unwrapping* [2]. Zhang realizó un estudio sistemático de este procedimiento y estableció el *stepping technique* que consigue resolver el problema de las esquinas de los mapas de fases, pero en dichos mapas se observa la presencia de ciertas líneas que se deben a inestabilidades producidas por el *code-word* o *gray-code* establecido en las fronteras de dos zonas de fases o en la transición de claros a oscuros repentinos, que involucra dos o tres píxeles en la frontera. Se sugiere que este problema debe ser resuelto antes del procesamiento del *unwrapping* y al parecer este es un problema aún no resuelto, porque se está sugiriendo técnicas de *machine learning* para resolver la aparición de estos errores. Cuando uno resuelve la aparición de estas líneas en el mapa de fases variando el *gray-code* desaparecen en la región previa y aparecen otras líneas de forma diferente en otras regiones del mapa de fases. Realizar una corrección manual de estos errores involucra mucho tiempo dependiendo del tamaño de las imágenes.

El objetivo del presente trabajo es realizar una ingeniería reversa de todos los procedimientos de manera cuantitativa para tener un *toolbox* para realizar el procesamiento de imágenes 3D a partir de las imágenes obtenidas mediante franjas con determinados corrimientos de fases y dos tipos de *escaleras* denominados por Zhang a los *code-words*. En la sección 2, explicamos los métodos numéricos involucrados; en la sección 3, explicamos la adquisición, corrección de las imágenes antes de su procesamiento usando patrones binarios, binarios modificados y sus respectivas desfocalizaciones. En la sección 4, realizamos una discusión de los métodos involucrados y finalmente las conclusiones.

## 2. Teoría

El método FPP de superficies mediante la proyección de franjas periódicas sobre los objetos es una técnica, aparentemente, de bajo costo y, relativamente, rápida para la caracterización de las topografías de superficies de objetos con geometrías complicadas.

### 2.1. Franjas binarias

Debido a que los patrones sinusoidales proyectados sobre los objetos y adquiridos por las cámaras crean dificultades numéricas de simetría debido al  $\gamma$  del proyector, el cual no podemos manipularlo porque vienen calibrados desde fábrica. Por ese motivo, en este trabajo, usamos las franjas binarias (FB) los cuales se generan usando la re-

lación

$$f(i) = \begin{cases} 0 & ; \text{ Si } \lfloor \frac{i}{P/2} \rfloor \text{ mód } 2 = 0, \\ 1 & ; \text{ Si } \lfloor \frac{i}{P/2} \rfloor \text{ mód } 1 = 1, \end{cases} \quad (1)$$

donde  $i$  es el índice que abarca toda la longitud de las franjas,  $P$  es el periodo de las franjas,  $\lfloor x \rfloor$  es la función *floor*, el cual da el entero más grande que es menor o igual a  $x$  y  $\text{mód}$  es la función módulo que resulta 1 ó 0. Construimos las FBs con los siguientes corrimientos de fase:  $0$ ,  $\pi/2$ ,  $\pi$  y  $3\pi/2$ .

El segundo conjunto de franjas se construye colocando una cierta distribución aleatoria en los píxeles que conforman las FBs, con sus respectivos corrimientos de fase, los cuales llamaremos como franjas binarias modificadas (FBM). El tercer y cuarto conjunto de franjas a ser proyectadas sobre los objetos se obtienen desfocalizando [3] el proyector al emitir las FBs y las FBMs, denominándolas franjas binarias desfocalizadas (FBD) y franjas binarias modificadas desfocalizadas (FBMD), con sus respectivos corrimientos de fases.

Con estas franjas proyectadas sobre los objetos obtenemos un conjunto de 4 imágenes para cada tipo de franjas: FBs, FBMs, FBDs y FBMDs. Estas imágenes son corregidas con los parámetros intrínsecos y extrínsecos de la cámara [4] y el proyector, obteniéndose las imágenes corregidas que se usan para la reconstrucción 3D del objeto. Las imágenes para cada corrimiento de fase podemos identificar como  $I_1$ ,  $I_2$ ,  $I_3$  y  $I_4$ , considerando los corrimientos de fase mencionados líneas arriba, respectivamente.

La fase envuelta se determina mediante

$$\phi(x, y) = \arctan \left( \frac{I_2 - I_4}{I_1 - I_3} \right), \quad (2)$$

donde  $\phi$  es la fase envuelta,  $(x, y)$  son las coordenadas cartesianas 2D de los píxeles de las imágenes.

Calculamos la fase desenvuelta siguiendo las sugerencias dadas en [5]. En primer lugar, trasladamos la información de las primeras derivadas, a lo largo de  $x$  e  $y$ , de la fase envolvente, entre  $-\pi$  y  $+\pi$ ; a los resultados obtenidos, nuevamente obtenemos la segunda derivada a lo largo de  $x$  e  $y$ , de la fase envolvente y sumamos las dos derivadas resultantes de modo que el resultado sea igual a una densidad, emulando la ecuación de Poisson. A esta densidad le sacamos la transformada rápida de Fourier discreta tipo coseno y la dividimos entre la función coseno 2D. Finalmente, a este último resultado le aplicamos la transformada inversa de Fourier discreta tipo coseno obteniendo la fase desenvuelta de la imagen del objeto 3D.

### 2.2. Escaleras

Por otro lado, se construyen dos escaleras periódicas de 10 periodos. Denominamos la escalera 1 con 8 peldaños de modo que cada peldaño tenga 16 píxeles, y la

escalera 2 con 4 peldaños y 32 pixeles por peldaño. Cada escalera tiene 4 ejemplares, cada uno con un corrimiento de fase igual a  $0, \pi/2, \pi$  y  $3\pi/2$ , respectivamente.

El conjunto de escaleras periódicas 1 y 2, se proyectan sobre el objeto y se adquieren las imágenes con la cámara. Estas imágenes también son procesadas con los parámetros intrínsecos y extrínsecos de la cámara y el proyector para obtener las imágenes corregidas de las distorsiones ópticas incorporadas por el proyector y la cámara en el momento de ser proyectadas y adquiridas. Estas imágenes corregidas son analizadas usando el siguiente procedimiento:

1. Se normalizan las fases de modo que las mismas queden entre  $-\pi$  y  $+\pi$ , usando la expresión siguiente [6]:

$$I_{i_{\text{norm}}} = -\pi + \left( \frac{I_i - I_{i_{\text{min}}}}{I_{i_{\text{max}}} - I_{i_{\text{min}}}} \right) 2\pi, \quad (3)$$

donde  $i = 1, \dots, 4$ , indica cada corrimiento de fase.

2. Luego se envuelven las imágenes usando la Ec.(2), obteniendo el mapa de fase envuelta  $\phi(x, y)$  para las escaleras.
3. La fase envuelta se cuantifica de la forma [6]:

$$k(x, y) = (N - 1) \left( \frac{\phi(x, y) - \phi_{\text{min}}}{\phi_{\text{max}} - \phi_{\text{min}}} \right) + 1, \quad (4)$$

donde  $N$  es el número de peldaños por escalera, es decir,  $N = 8$  para las escaleras 1 y  $N = 4$  para las escaleras 2.

4. La imagen representada por  $k(x, y)$  muestra una periodicidad de las escaleras proyectadas sobre el rostro. Mediante un código en Python, ver apéndice A, identificamos las interfaces entre dos escaleras vecinas identificando el pixel con máximo valor y el pixel vecino con mínimo valor de  $k(x, y)$  en la dirección  $x$  para observar la evolución del contraste claro-oscuro a lo largo de la dirección  $y$ .
5. Con la información de las interfaces donde se producen los saltos abruptos de máximo a mínimo de  $k(x, y)$  entre escaleras vecinas, procedemos a construir una imagen del objeto sobre una escalera continua, añadiendo el valor máximo de la escalera anterior a los valores de las escaleras posteriores, de manera sucesiva.
6. De esta última imagen, creamos una imagen que se traslada un corrimiento aproximadamente igual a medio peldaño de la escalera a lo largo de  $x$ . La imagen trasladada y la anterior se suman formando una nueva imagen, que llamaremos imagen reenvuelta de las escaleras.

7. Esta imagen reenvuelta resultante de las escaleras se suma a la imagen envuelta de las FBs, FBMs, FBDs y FBMDs.
8. A partir de este momento se procede a realizar el proceso habitual en los métodos de perfilometría por franjas periódicas [6], para obtener las imágenes reconstruidas, calculamos las fases desenvueltas.

### 3. Resultados y discusión

El sistema de adquisición de imágenes usado en este trabajo consiste de: 1) una cámara en blanco y negro de Thorlabs DCC1545M-GL, con un lente de 25 mm de distancia focal ubicado en un soporte de alineamiento 3D; 2) un proyector led LG PH30JG-GL y un proyector de lámpara EPSON Powerlite E28; y 3) una pantalla móvil instalada sobre un riel paralelo usado en las máquinas *Computer Numerical Control*, CNC, controlado manualmente y mediante un motor de paso. La pantalla se instala perpendicular al eje óptico de la cámara. Y el proyector es colocado de forma que el eje óptico del proyector forma un ángulo entre  $10^\circ$  y  $12^\circ$  con el eje óptico de la cámara y ambos ejes ópticos coinciden en el mismo punto de la pantalla aproximadamente. Este alineamiento lo conseguimos usando punteros laser de nivel, Fig(1).



**Figura 1:** Se muestra el alineamiento de la cámara y el proyector LG en un ángulo  $\sim 11^\circ$ .

El proceso de calibración de la cámara fue realizado usando el método del tablero de ajedrez y obtenemos los parámetros intrínsecos y extrínsecos usando la versión Python de las librerías del OpenCV [7]. Usamos 16 imágenes del tablero de ajedrez orientados en diferentes direcciones que hacen ángulos diferentes con respecto a la dirección del eje óptico de la cámara con la pantalla [4].

Las 16 imágenes corregidas con los parámetros intrínsecos y extrínsecos de la cámara son proyectadas sobre la pantalla. Adquirimos, nuevamente, las imágenes proyectadas en la pantalla con la cámara y éstas imágenes son usadas para calibrar el proyector, es decir, obtener los parámetros intrínsecos y extrínsecos del proyector usando las librerías OpenCV, considerando el proyector como una cámara inversa.

Este proceso de calibración de la cámara y el proyector es muy importante, porque con los parámetros intrínsecos y extrínsecos de la cámara y el proyector podemos corregir cualquier imagen que se adquiera mediante el sistema cámara-proyector para el análisis de la topografía de las superficies en objetos 3D.

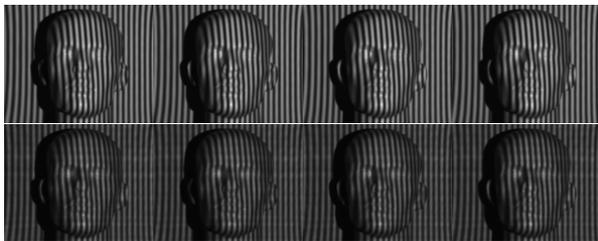
### 3.1. Franjas desfocalizadas

En primer lugar, las franjas proyectadas para este trabajo son las FBDs y las FBMDs que se proyectan sobre dos cabezas de maniquí usados en peluquería, el primero de sexo masculino con una superficie altamente reflectante y la segunda de sexo femenino cuyo color emula la piel humana con cabello artificial, Fig(2).



**Figura 2:** Se muestran las dos cabezas de maniquí usados en el presente trabajo.

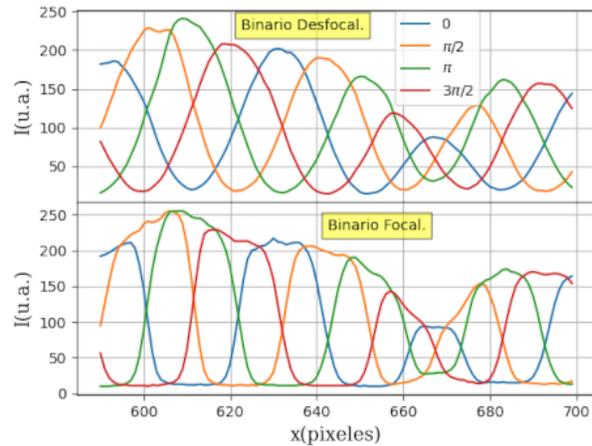
Las imágenes del primer objeto, con ambas franjas se muestran en la Fig.(3), mostrando las fases de  $0$ ,  $\pi/2$ ,  $\pi$  y  $3\pi/2$ , respectivamente.



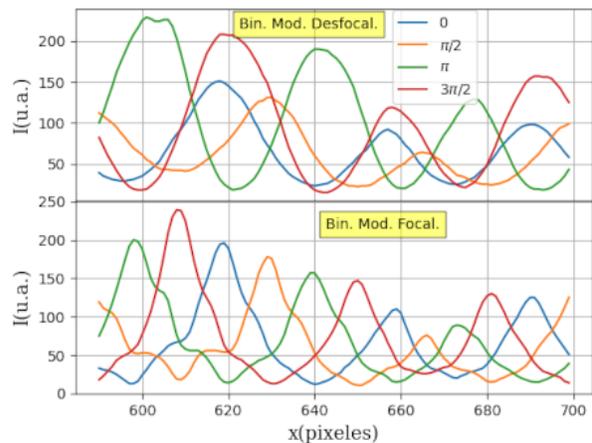
**Figura 3:** Las imágenes de la fila superior son las cuatro fases  $0$ ,  $\pi/2$ ,  $\pi$  y  $3\pi/2$  de las FBDs, mientras que las imágenes de la fila inferior son las fases mencionadas de las FBMDs, respectivamente.

Para verificar que el proceso de desfocalización *sinualiza* las franjas proyectadas sobre los objetos, mostramos en la Fig.(4), los perfiles de la región de la punta de la nariz con las franjas proyectadas sobre ella para una fa-

se de  $0^\circ$ , que corresponde a la proyección de las FBDS, parte superior, y a las FBs focalizadas, parte inferior.



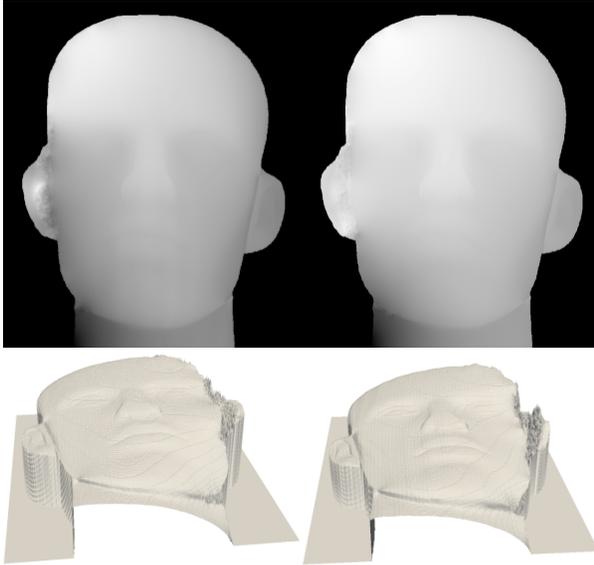
**Figura 4:** En la figura de la parte superior se muestra el perfil sinusoidal de la nariz del muñeco,  $x \in [590, 700]$ , con las FBDs cuando los píxeles están en  $y = 490$ , mientras que en la figura inferior se muestra el perfil de la nariz del muñeco con las FBs focalizadas en las mismas posiciones que la figura superior. Se observa claramente que el perfil binario desfocalizado presenta un comportamiento *sinusoidal*, sin interferencia del  $\gamma$  del proyector.



**Figura 5:** Se muestra el perfil sinusoidal de la nariz del muñeco, cuando  $x \in [590, 700]$  e  $y = 490$ . En la parte superior con las FBMDs, mientras que en la parte inferior con las FBMs focalizadas. El perfil binario modificado desfocalizado presenta un comportamiento *sinusoidal*, con poca interferencia del  $\gamma$  del proyector.

Lo mismo hacemos para las FBMDs y las comparamos con las FBMs focalizadas y también se observa la *sinualización* de las franjas desfocalizadas respecto a las focalizadas, tal como se observa en la Fig.(5), donde en la parte superior se muestra el patrón desfocalizado

mientras que en la parte inferior se muestra las FBMs focalizadas.



**Figura 6:** Las dos imágenes de la parte superior representan la reconstrucción 2D de la FPP usando FBDs (izquierda) y FBMDs (derecha). En la parte inferior se muestran las imágenes 3D correspondientes a los casos mencionados en la parte superior.

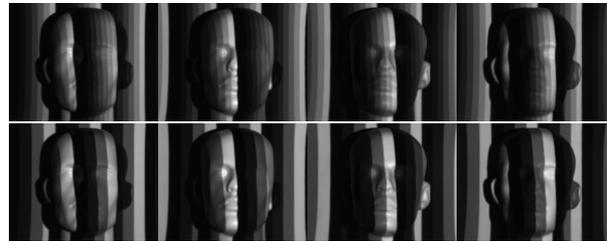
Cuando se procesan las imágenes de las diferentes fases para cada tipo de franjas proyectadas, se envuelve las fases y luego se desenvuelven. Lo mismo se realiza con las imágenes de las franjas proyectadas sin el objeto, se envuelve las fases y luego se desenvuelven. Las fases desenvueltas de las franjas con el objeto y sin el objeto se restan y el resultado de la diferencia se multiplican por la máscara del objeto y se obtienen las figuras mostradas en la Fig.(6). En la parte superior se muestra el mapa de fase 2D de las imágenes procesadas con las FBDs, en la parte izquierda, y con las FBMDs, en la parte derecha. Mientras que, en la parte inferior se muestra el mapa de fase 3D de las imágenes mencionadas en la parte superior. Se puede observar que las imágenes 3D presentan una reflexión ante un plano central de la imagen con respecto a las imágenes 2D. En las imágenes 2D, de la parte superior, las sombras se producen en el lado izquierdo del rostro, mientras que, en las 3D las sombras aparecen en el lado derecho. Las protuberancias observadas en el lado derecho de las imágenes 3D se deben estrictamente al efecto de las sombras que provocan inestabilidades numéricas en el cálculo del arco tangente de las intensidades. Ambas imágenes 3D evolucionan a valores pequeños en intensidad en el borde derecho de las dos imágenes, asociadas a las FBDs y a las FBMDs. Esta misma evolución se observa cuando se usan franjas sinusoidales, binarias completamente focalizadas, resultados no mostrados en

el presente artículo.

### 3.2. Escaleras de peldaños diferentes

En segundo lugar, procedemos a proyectar las escaleras periódicas 1 y 2 que hemos construido y lo proyectamos sobre los objetos de manera secuencial. Las escaleras tiene el mismo corrimiento de fase que hemos mencionado en la subsección anterior,  $0$ ,  $\pi/2$ ,  $\pi$  y  $3\pi/2$ .

Las imágenes obtenidas con la secuencia de corrimientos de fase mencionados para las escaleras 1 y 2, se muestran en la Fig.(7).

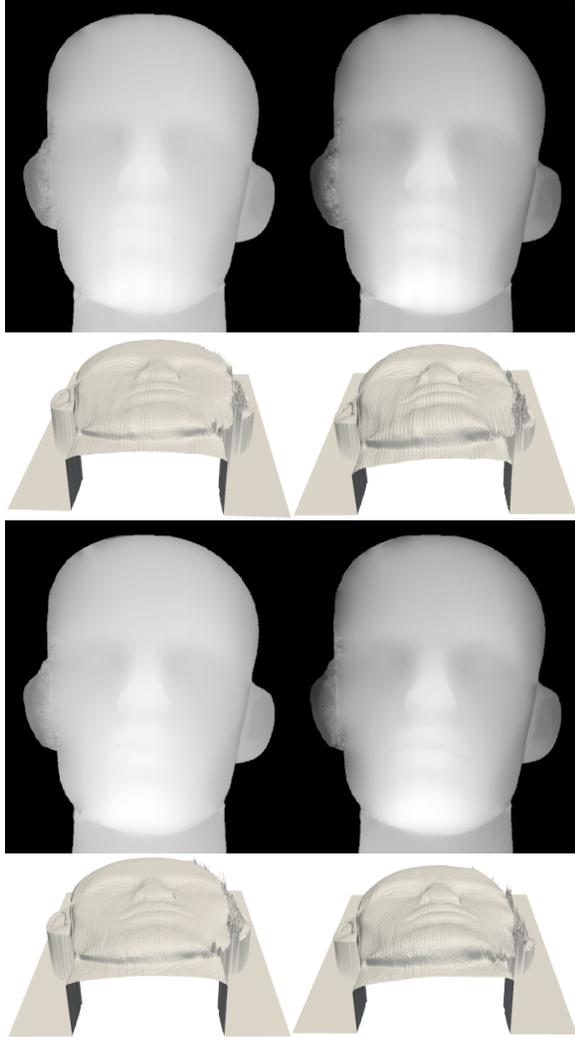


**Figura 7:** Se muestran las imágenes de la cabeza del maniquí masculino con las superposición de las escaleras 1 (superior) y 2 (inferior), con los respectivos corrimientos de fase descritos en el texto.

Las imágenes se procesan de acuerdo al protocolo descrito en la sección de la teoría referido a las escaleras y se obtienen, finalmente, las imágenes que se muestran en la Fig.(8). Se aprecia en la primera (escalera 1) y tercera fila (escalera 2) de la Fig.(8) las imágenes 2D reconstruidas del rostro masculino del maniquí con las FBDS (columna izquierda) y las FBMDs (columna derecha). En la segunda (escalera 1) y cuarta fila (escalera 2), se muestran las imágenes 3D de las FBDs (izquierda) y las FBMDs (derecha). Podemos observar que un 90 % del rostros están bien determinadas y comparadas con las imágenes 3D mostradas en la Fig.(6), no tienen los defectos mostrados en la parte inferior del lado derecho. Aún mantienen las inestabilidades numéricas debido a las sombras cerca de la oreja y en las mandíbulas derechas. Comparando las imágenes procesadas con las escaleras 1 y 2, se observa que la reconstrucción 3D observada para la escalera 2 presenta menos inestabilidades en las sombras de la oreja y la mandíbula derechas respecto a la escalera 1.

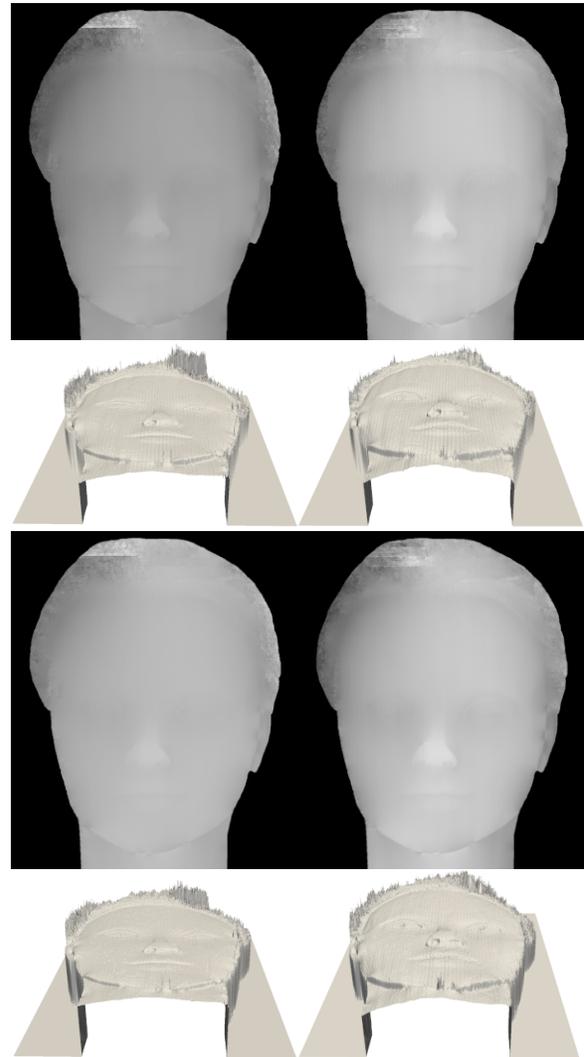
Finalmente, presentamos las imágenes reconstruidas para un conjunto de imágenes adquiridas del rostro femenino del maniquí de peluquería, con las FBDs y las FBMDs, usando las escaleras 1 y 2. Las cuales mostramos en la Fig.(9). Se observa que el cabello artificial tiene puntos de reflexión más intensos que otros y eso crea las inestabilidades numéricas observadas en la región de la cabeza. También podemos observar que la superficie del rostro no es tan reflectante como el rostro del maniquí masculino, debido a eso, el albedo de la parte del cuello

en el rostro femenino no permite una buena definición numérica en el mentón. También, observamos que la escalera 2 tiene menor inestabilidad numérica tanto en el cabello como en las fronteras del rostro.



**Figura 8:** Se muestran en la primera y tercera filas, las reconstrucciones 2D de las FBDs (columna izquierda) y las FBMDs (columna derecha) usando las escaleras 1 y 2, respectivamente. En la segunda y cuarta filas se muestran las imágenes 3D correspondientes a las FBDs (izquierda) y FBMDs (derecha), usando las escaleras 1 y 2, respectivamente. Todas las representan al rostro masculino del maniquí.

Cabe mencionar que las figuras mostradas están contruidas de los datos adquiridos y procesados de acuerdo a los procedimientos mencionados previamente en este artículo. Y las inestabilidades mostradas en las zonas próximas a las sombras y las fronteras del rostro, así como el cabello del maniquí deben ser minimizadas mediante un proceso denominado *post-processing*.



**Figura 9:** Se muestran en la primera y tercera filas, las reconstrucciones 2D de las FBDs (columna izquierda) y las FBMDs (columna derecha) usando las escaleras 1 y 2, respectivamente. En la segunda y cuarta filas se muestran las imágenes 3D correspondientes a las FBDs (izquierda) y FBMDs (derecha), usando las escaleras 1 y 2, respectivamente. Todas las representan al rostro femenino del maniquí.

En la literatura [8–12], se encuentran algunos intentos de reducir estas inestabilidades cuando se obtiene la función  $k(x, y)$  y se determinan las interfaces de los claros-oscuros. Realizar un proceso automático de este proceso en esta etapa de la reconstrucción, incluyendo las técnicas de *machine & deep learning* no son eficientes cuando los objetos para analizar son cambiantes. Pero son efectivos cuando se verifica la calidad de las superficies de un objeto que se contruyen durante un proceso industrial automatizado.

Un comentario adicional, para reconstruir una superficie 3D, ésta depende de las características ópticas del entorno, como la textura, la reflectancia y la iluminación; los sensores con llevan limitaciones en este aspecto, lo que resulta en una menor fiabilidad. Una posibilidad de resolver este problema en robótica es el uso de sensores visuales basados en luz estructurada que proporciona una reconstrucción 3D de alta precisión, velocidad y confiabilidad, como control de calidad de superficies para hallar grietas, depresiones o pendientes indeseables.

#### 4. Conclusiones

Hemos conseguido reproducir una técnica de reconstrucción de imágenes mediante la perfilometría de proyección de franjas binarias desfocalizadas para el análisis cuantitativo de la topografía de una superficie 3D, relativamente compleja.

El método estudiado no sólo es eficiente para superficies altamente reflexivas, también funciona adecuadamente para superficies coloreadas por lo que no afecta la precisión en la medición.

El uso del patrón binario permitió la reducción del  $\gamma \approx 1$ , resolviendo el problema de la no linealidad del proyector.

Conociendo el problema de la sincronización entre la cámara y el proyector, cuando un patrón sinusoidal es utilizado, esta sincronía no es necesaria usando patrones binarios debido a que el patrón sinusoidal, *in situ*, es producido externamente por la desfocalización del proyector.

#### A. Código de la escalera

```
nes = 4 # numero de steps en cada escalera
# Quantizando la fase del step
k_00 = (nes-1)*((ccc-ccc.min())/((ccc.max()-ccc.min()))+1

with open('inter_1.csv', newline='') as file:
    reader = csv.reader(file,
        quoting=csv.QUOTE_NONNUMERIC)
    ff=[]
    for row in reader:
        ff.append(row[:])

face = np.asarray(ff).astype(int)
```

```
z = k_00

z1 = np.zeros((nv,mh))

for j in range(nv):
    if (j<=nv):
        cc = np.max(z[j,5:])
        dd = np.min(z[j,5:])
        c1 = np.max(z[j,face[j,0]:face[j,0]+1])
        c2 = np.max(z[j,face[j,1]-5:face[j,1]])
        c3 = np.max(z[j,face[j,2]-5:face[j,2]])
        c4 = np.max(z[j,face[j,2]-5:face[j,3]])
        d1 = np.min(z[j,1:2])
        d2 = np.min(z[j,face[j,0]+1:face[j,0]+5])
        d3 = np.min(z[j,face[j,1]+1:face[j,1]+5])
        d4 = np.min(z[j,face[j,2]+1:face[j,2]+5])
        #print(j,c1,c2,c3,d1,d2,d3)

    for i in range(mh):
        if i < face[j,0] :
            z1[j,i] = z[j,i]
        elif i == face[j,0] :
            z1[j,i] = z[j,i]
        elif i > face[j,0] and i < face[j,1] :
            z1[j,i] = z[j,i] + (c1-d1)
        elif i == face[j,1] :
            z1[j,i] = z[j,i] + (c1-d1)
        elif i > face[j,1] and i < face[j,2] :
            z1[j,i] = z[j,i] + (c1-d1) + (c2-d2)
        elif i == face[j,2] :
            z1[j,i] = z[j,i] + (c1-d1) + (c2-d2)
        elif i > face[j,2] and i < face[j,3]:
            z1[j,i] = z[j,i] + (c1-d1) +
                (c2-d2) + (c3-d3)
        elif i == face[j,3] :
            z1[j,i] = z[j,i] + (c1-d1) +
                (c2-d2) + (c3-d3)
        elif i > face[j,3] :
            z1[j,i] = z[j,i] + (c1-d1) +
                (c2-d2) + (c3-d3) + (c4-d4)
        else:
            z1[j,i] = z[j,i]

plt.imshow(z1, cmap='gray')
plt.axis('off')
plt.show()
```

#### Referencias

- [1] Song Zhang; *High speed 3D imaging with digital fringe projection technique*, CRC Press, Taylor and Francis Group, Boca Raton (2016). <https://doi.org/10.1201/b19565>.
- [2] G. Sansoni, M. Carocci y R. Rodella; Appl. Opt. **38**(31) 6565 (1999). <https://doi.org/10.1364/AO.38.006565>.

- [3] S. Lei y S. Zhang; Opt. Lett. **34(20)**, 3080 (2009). <https://doi.org/10.1364/OL.34.003080>.
- [4] R. A. Montalvo y P. H. Rivera; Rev. Inv. Fis. **24(1)**, 17 (2021).
- [5] D. C. Ghiglia y M. D. Pritt; *Two-Dimensional Phase Unwrapping: Theory, Algorithms and Software*, John Wiley and Sons Ltd. (1998).
- [6] Z. Zeng, B. Li, Y. Fu y M. Chai; J. Euro. Opt. Soc.-Rap. Pub. **12**, 9 (2016). <https://doi.org/10.1186/s41476-016-0013-9>.
- [7] <https://opencv.org>; [https://docs.opencv.org/4.x/d6/d00/tutorial\\_py\\_root.html](https://docs.opencv.org/4.x/d6/d00/tutorial_py_root.html)
- [8] D. Zheng, F. Da, Q. Kemaoy y H. S. Seah; Opt. Express **25(5)**, 4700 (2017). <https://doi.org/10.1364/OE.25.004700>.
- [9] Song Zhang; Opt. Lett. **35(7)**, 934 (2010). <https://doi.org/10.1364/OL.35.000934>.
- [10] L. Huang y A. K. Asundi; Meas. Sci. Technol. **22**, 035304 (2011). <https://doi.org/10.1088/0957-0233/22/3/035304>.
- [11] S. Feng, Q. Chen, C. Zuo, R. Li, G. Shen y F. Feng; Opt. Eng. **52(1)**, 013605 (2013). <https://doi.org/10.1117/1.OE.52.1.013605>.
- [12] Y. Zhang, J. Tong, L. Lu, J. Xi, Y. Yu y Q. Guo; IEEE Access **9**, 23110 (2021). <https://doi.org/10.1109/ACCESS.2021.3056063>.