



Universidad Autónoma de Querétaro
Facultad de Ingeniería
Maestría en Ciencias en Inteligencia Artificial

Reconstrucción tridimensional de objetos usando aprendizaje profundo
en perfilometría por cambio de fase con franjas de color

Tesis

Que como parte de los requisitos para obtener el Grado de
Maestro en Ciencias en Inteligencia Artificial

Presenta

Joaquín Omar Castillo Salazar

Dirigido por:

Dr. Jesús Carlos Pedraza Ortega

Nombre del Sinodal

Presidente: Dr. Jesús Carlos Pedraza Ortega

Nombre del Sinodal

Secretario: Dr. Marco Antonio Aceves Fernández

Nombre del Sinodal

Vocal: Dr. Saúl Tovar Arriaga

Nombre del Sinodal

Suplente: Dr. Juan Manuel Ramos Arreguin

Nombre del Sinodal

Suplente: MCIA Osmar Antonio Espinosa

Centro Universitario, Querétaro, Qro.

Octubre 2024

México

La presente obra está bajo la licencia:
<https://creativecommons.org/licenses/by-nc-nd/4.0/deed.es>



CC BY-NC-ND 4.0 DEED

Atribución-NoComercial-SinDerivadas 4.0 Internacional

Usted es libre de:

Compartir — copiar y redistribuir el material en cualquier medio o formato

La licenciante no puede revocar estas libertades en tanto usted siga los términos de la licencia

Bajo los siguientes términos:



Atribución — Usted debe dar [crédito de manera adecuada](#), brindar un enlace a la licencia, e [indicar si se han realizado cambios](#). Puede hacerlo en cualquier forma razonable, pero no de forma tal que sugiera que usted o su uso tienen el apoyo de la licenciante.



NoComercial — Usted no puede hacer uso del material con [propósitos comerciales](#).



SinDerivadas — Si [remezcla, transforma o crea a partir](#) del material, no podrá distribuir el material modificado.

No hay restricciones adicionales — No puede aplicar términos legales ni [medidas tecnológicas](#) que restrinjan legalmente a otras a hacer cualquier uso permitido por la licencia.

Avisos:

No tiene que cumplir con la licencia para elementos del material en el dominio público o cuando su uso esté permitido por una [excepción o limitación](#) aplicable.

No se dan garantías. La licencia podría no darle todos los permisos que necesita para el uso que tenga previsto. Por ejemplo, otros derechos como [publicidad, privacidad, o derechos morales](#) pueden limitar la forma en que utilice el material.

A mis padres, con su cariño y ejemplo, me motivan a ser mejor.

Agradecimientos

Quiero comenzar agradeciendo a mis padres, Aurora Salazar Cervantes y José Joaquín Castillo Salazar, por brindarme su amor, confianza y apoyo en cada etapa de mi vida. En particular agradezco a mi madre por siempre alentarme a buscar mi mejor versión, y a mi padre, que con su ejemplo me ha enseñado el valor del trabajo duro.

También agradecer a mis hermanas, Paty, Lorena y Denisse, por haberme acompañado en cada etapa de mi vida. Gracias por brindarme una infancia feliz. Gracias a mis sobrinas Vale, Helena y Danna por alegrar la vida de todos.

Agradezco a mis amigos, he tenido la fortuna de conocer la verdadera amistad. Han sido un pilar fundamental en el alcance de mis metas. Especialmente gracias a Miguel y Eduardo por acompañarme a lo largo de estos años.

Quiero agradecer al Doctor Jesús Carlos Pedraza, por su constante guía y apoyo. Además de ser un excelente investigador es un gran ser humano.

Agradecimientos especiales a la Universidad Autónoma de Querétaro por la oportunidad de realizar mis estudios de maestría. Y al CONAHCYT por el financiamiento recibido para la realización de este proyecto.

Contenido

Índice de Figuras	IV
Índice de Tablas	VI
Abreviaturas	VII
Resumen	1
Abstract	2
1 Introducción	3
2 Antecedentes	6
2.1 Reconstrucción tridimensional	6
2.1.1 Técnicas de luz estructurada	7
2.1.1.1 Perfilometría por cambio de fase	8
2.1.1.2 Perfilometría por cambio de fase con imágenes RGB	9
2.1.1.3 Desenvolvimiento de fase	10
2.1.1.4 Desenvolvimiento de fase temporal	13
2.2 Inteligencia Artificial	15
2.2.1 Redes Neuronales	16
2.2.2 Machine Learning	17
2.2.3 Deep Learning	18
2.2.3.1 Redes Neuronales Convolucionales	18
2.2.3.2 U-Net	20
2.2.3.3 Generative Adversarial Network	21
2.2.3.4 Patch GAN	22
2.2.3.5 Conditional Generative Adversarial Network	23

+2.3 Métricas	24
2.3.1 Error Cuadrático Medio Inverso (MSE)	24
2.3.2 Relación Señal-Ruido de Pico (PSNR)	24
2.4 Métricas	26
3 Hipótesis	29
4 Objetivos	30
4.1 Objetivo General	30
4.2 Objetivos Especifico	30
5 Metodología	31
5.1 Creación de Base de Datos	31
5.1.1 Técnicas de luz estructurada	32
5.5.1.1 Adquisición de imágenes	32
5.5.1.2 Adquisición de objetos de referencia	35
5.2 Pre Procesamiento	36
5.2.1 Preparación de imágenes	36
5.2.2 Calculo de amplitud y fase	36
5.3 Desenvolvimiento de fase Temporal	37
5.3.1 Preparación de imágenes	37
5.3.2 Desenvolvimiento de Fase Temporal Iterativo	38
5.3.3 Almacenamiento	39
5.4 Conditional Generative Adversarial Network	40
5.4.1 Material	40
6 Resultados y Discusión	44
6.1 Creación de base de datos de objetos digitales	44
6.2 Desenvolvimiento de fase temporal	46
6.3 C-GAN	54
6.4 Discusión de resultados	61

7 Conclusiones	63
Bibliografía	64
Anexos	68

Índice de Figuras

- 2.1 Técnicas de luz estructurada
- 2.2 Diagrama de adquisición del método PSP
- 2.3 Algoritmo base del desenvolvimiento de fase
- 2.4 Áreas de estudio la inteligencia artificial
- 2.5 Diagrama básico de la red Neuronal
- 2.6 Lógica de la programación clásica vs el aprendizaje máquina
- 2.7 Arquitectura base de una CNN
- 2.8 Arquitectura base de la U-Net
- 2.9 Diagrama de funcionamiento de una C-GAN
- 2.10 Arquitectura de la C-GAN
- 5.1 Metodología general
- 5.2 Disposición de los elementos en el escenario de proyección
- 5.3 Entrada y salidas del escenario de proyección de patrones de franjas senoidales
- 5.4 Script de captura de imágenes automatizado
- 5.5 Ejemplo de set de imágenes de objeto con patrones proyectados
- 5.6 Proceso de TPU iterativo
- 6.1 Ejemplo de objetos sintéticos
- 6.2 Proceso de la reconstrucción a través del TPU
- 6.3 Distribución de la reducción de MSE por iteración
- 6.4 Disminución del Error por iteración
- 6.5 Distribución del aumento del PSNR por iteración
- 6.6 Aumento de similitud por iteración
- 6.7 Reconstrucciones obtenidas por número de iteración
- 6.8 Reconstrucciones obtenidas por iteración (corte transversal)
- 6.9 Comparación Clásico vs TPU5 vs Ground Truth (Parte A)
- 6.10 Comparación Clásico vs TPU5 vs Ground Truth (Parte B)
- 6.11 Distribución de la reducción de MSE, TPU VS TPU+GAN

- 6.12 Comparación TPU5 vs TPU5+C-GAN vs Ground Truth (Parte A)
- 6.13 Comparación TPU5 vs TPU5+C-GAN vs Ground Truth (Parte B)
- 6.14 Corte Transversal de los objetos reconstruidos.
- 6.15 Corte Transversal de los objetos reconstruidos (Zoom)

Índice de Tablas

2.1 Trabajos sobre Reconstrucción 3D mediante PSP espacial

2.2 Trabajos sobre desenvolvimiento de fase temporal

2.3 Trabajos sobre Desenvolvimiento de fase temporal de Fase usando DL

5.1 Especificaciones del equipo de cómputo

5.2 Versiones de Software utilizados

6.1 Imágenes Adquiridas

6.2 MSE de objetos reconstruidos con TPU

6.3 PSNR de objetos reconstruidos con TPU

6.4 MSE método clásico vs TPU vs TPU+C-GAN

6.5 PSNR método clásico vs TPU vs TPU+C-GAN

Abreviaturas

PSP	Phase-Shifting-Profilometry
TPU	Temporal Phase Unwrapping
AI	Artificial Intelligence
ML	Machine Learning
DL	Deep Learning
CNN	Convolutional Neural Network
C-GAN	Conditional Generative Adversarial Network
CSV	Comma-Separated Values

Resumen

En este trabajo se implementó una variante de la Perfilometría por cambio de fase con franjas de color. Esta técnica, a pesar de ser ampliamente utilizada cuando se necesita realizar una medición o reconstrucción tridimensional con un alto nivel de detalle, presenta dos importantes limitaciones, la primera es respecto al número de franjas que se pueden proyectar y la segunda esta ligada a la complejidad de la superficie a reconstruir. Estas limitaciones originan errores en la etapa conocida como desenvolvimiento de fase. Por lo que en este trabajo se planteó como objetivo principal implementar técnicas de Aprendizaje Profundo para superar estas limitaciones y de esta manera conseguir reconstrucciones más precisas en comparación a los métodos convencionales. En concreto se utilizaron dos métodos, siendo el primero el uso de desenvolvimiento de fase temporal, el cual permite aumentar el número de franjas proyectadas y evitar errores en superficies complejas y el segundo método fue el uso de una Red Generativa Adversarial Condicional, la cual sirvió para corregir errores en la topología de las superficies y como filtrado de las deformaciones inherentes a la técnica de perfilometría por cambio de fase. Los resultados alcanzados fueron positivos, donde el error cuadrático medio de las reconstrucciones obtenidos corresponde a un 10% del error medido de las reconstrucciones obtenidas con el método clásico.

Abstract

In this work, a variant of phase shifting profilometry with color fringes was implemented. This technique, in spite of being widely used when a three-dimensional measurement or reconstruction with a high level of detail is needed, has two important limitations, the first one is related to the number of fringes that can be projected and the second one is linked to the complexity of the surface to be reconstructed. These limitations cause errors in the stage known as phase unwrapping. Therefore, the main objective of this work was to implement Deep Learning techniques to overcome these limitations and thus achieve more accurate reconstructions compared to conventional methods. Specifically, two methods were used, the first one being the use of temporal phase unwrapping, which allows increasing the number of projected fringes and avoiding errors in complex surfaces and the second method was the use of a Conditional Adversarial Generative Network, which served to correct errors in the topology of the surfaces and as a filtering of the deformations inherent to the phase change profilometry technique. The results achieved were positive, where the mean square error of the reconstructions obtained corresponds to 10% of the measured error of the reconstructions obtained with the classical method.

1. Introducción

Replicar la visión humana en máquinas no es una tarea sencilla [1] [2], ya que a pesar de que la mayoría de personas son capaces de reconocer objetos, rostros, números, etc. sin ningún esfuerzo, es una habilidad que ha llevado millones de años de evolución [3], Además de que solo se conoce la estructura que lleva a cabo la visión, pero no se conoce exactamente la forma en la que funciona.

Estas complicaciones han originado que el estudio de la visión por computadora se subdivida en áreas conocidas como los problemas clásicos de la visión por computadora. Estos problemas incluyen: reconocimiento de objetos, seguimiento de objetos, reconstrucción tridimensional, etc.

La reconstrucción tridimensional consiste en crear modelos tridimensionales de un objeto o de una escena a partir de imágenes planas (2D) [4]. Para lo cual existen diversos métodos los cuales tienen más o menos ventajas dependiendo el objetivo.

Dentro de los métodos existentes para realizar una reconstrucción tridimensional se encuentran: Fotogrametría (basada en la respuesta fotópica del ojo), visión estereoscópica, triangulación y los que emplean luz [2] [5].

Las técnicas de luz estructurada presentan ventajas sobre los otros métodos como precisión, rapidez, no son invasivas y se pueden automatizar. Una de las técnicas de luz estructurada más empleadas es la perfilometría por cambio de fase (PSP). Usar PSP permite llevar a cabo reconstrucciones y mediciones tridimensionales de objetos y superficies precisas, de forma veloz, todo ello sin contacto con la superficie [4] [6].

Se han desarrollado muchas variaciones del algoritmo base para mejorar la calidad de la reconstrucción [7] [8]. Una de estas variaciones emplea imágenes coloreadas o RGB, en esta variante se utilizan tres cambios de fase, donde cada fase se

proyecta a través de cada canal de la imagen [9]. Utilizar esta variante permite adquirir tres capas de información por imagen capturada [10].

Desafortunadamente, toda técnica de PSP presenta un fenómeno llamado fase envuelta [10] Esto se debe a que al recuperar la fase se utiliza la función arco tangente, la cual tiene límites de su dominio entre $-\pi$ y π [11] [12]. Por lo tanto, es necesario corregir esta limitación para interpretar la superficie adecuadamente, esto se conoce como desdoblamiento de fase [12] [13] [14].

Se han propuesto varios algoritmos que realizan el desdoblamiento de fases en el dominio espacial [15] [16]. Este tipo de algoritmos son útiles para superficies simples, pero al realizar el desarrollo de fases en superficies discontinuas o complejas, se generan errores o artefactos a lo largo de la medición [17].

Para superar los inconvenientes generados por las superficies discontinuas, se desarrollaron algoritmos de desdoblamiento de fase temporal (TPU) [18] [19]. Como su nombre lo indica, llevan a cabo un desdoblamiento de fases en el dominio del tiempo [20], haciendo uso de dos o más patrones de franjas. El caso más general es cuando se utilizan dos patrones, el primero con pocas rayas, un patrón de baja frecuencia, y el segundo con mayor número de rayas, un patrón de alta frecuencia. Donde la fase desdoblada del patrón de baja frecuencia se utiliza como guía para desdoblarse la fase del patrón de alta frecuencia [21].

Algunos trabajos han mostrado resultados óptimos utilizando el desdoblamiento de la fase temporal tanto cuando se utilizan de forma independiente como con métodos híbridos utilizando Deep Learning [22] [23].

La importancia de realizar el desdoblamiento de fase de forma correcta, incluso de superficies complejas y recuperando los pequeños detalles, recae en la variedad de aplicaciones que tienen las reconstrucciones tridimensionales [24] en donde solo se permite un pequeño margen de error [13] [14].

En este trabajo se estudió el efecto de utilizar de forma iterativa una modificación del algoritmo de desenvolvimiento de fase temporal (TPU) Capitulo 5.3. Además, en el Capitulo 5.4 se analizó el efecto de procesar las reconstrucciones obtenidas al usar TPU con una red adversarial generativa condicional (C-GAN).

2. Antecedentes

2.1 Reconstrucción tridimensional

La reconstrucción tridimensional de imágenes hace referencia a la creación de modelos tridimensionales a partir de imágenes bidimensionales, lo cual es fundamental en la medición tridimensional precisa de objetos. Este proceso permite obtener la geometría tridimensional de los objetos, independientemente de su reflectividad superficial, distancia al sensor y condiciones de iluminación [25].

La reconstrucción tridimensional de imágenes implica la utilización de diversas técnicas, para obtener modelos 3D de alta calidad a partir de una, dos o más imágenes [26]. Con ellas es posible realizar la reconstrucción de modelos especializados, como cabezas, cuerpos o arquitectura.

Entre las técnicas existentes para llevar a cabo la reconstrucción tridimensional se listan las siguientes [27].

- Visión estéreo
- Holografía digital
- Interferometría
- Proyección de franjas y Técnicas de cambio de fase

2.1.1 Técnicas de luz estructurada

Un método ampliamente utilizado en la reconstrucción tridimensional se basa en el uso de luz estructurada, es decir, iluminación activa de la escena con un patrón de intensidad espacialmente variable en 2D [4].

Un modulador de luz espacial genera una iluminación estructurada 2D que varía espacialmente. A su vez, estas técnicas se dividen de acuerdo al tipo de patrón proyectado (Figura 2.1).

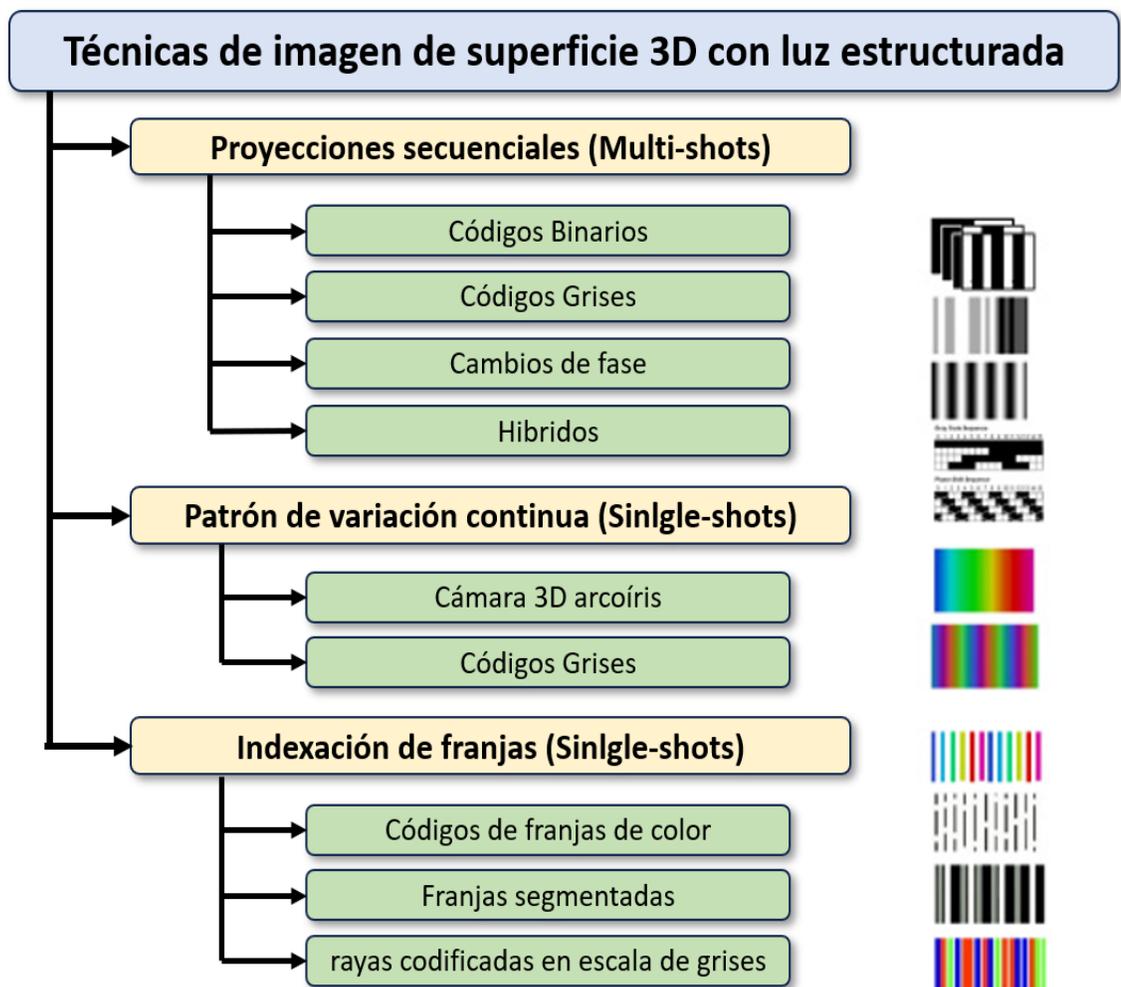


Figura 2.1 Técnicas de luz estructurada. Modificado de [4]

2.1.1.1 Perfilometría por cambio de fase

La perfilometría de cambio de fase (PSP) es un método comúnmente utilizado para medir superficies de objetos tridimensionales (3-D). El sistema PSP consta de un proyector y una cámara digital. Las técnicas de PSP incluyen dos principales pasos, proyectar y adquirir una secuencia de patrones de franjas sinusoidales desplazadas en fase [28]. Posteriormente se realiza una fase de desdoblamiento para determinar la morfología del objeto.

Modelo PSP:

Consideramos una sección transversal de la superficie del objeto para una coordenada dada. Entonces, la intensidad de los patrones de franjas capturados por la cámara y la función de distribución de altura se pueden expresar como una función con una sola variable x .

$s(x)$: Intensidad del patrón de franjas proyectadas.

$d(x)$: Patrón de franjas deformado

$h(x)$: Distribución de la altura de la superficie del objeto.

M : Número de patrones de franjas ($M \geq 3$)

Los patrones de franjas (M) se proyectan secuencialmente sobre el plano de referencia y la superficie del objeto y luego son capturados por la cámara.

Cada uno de los patrones de franjas M tiene una diferencia de fase de $2\pi/M$ sus patrones adyacentes. Por lo tanto, los patrones de franjas proyectadas $s_m(x)$ y $d_m(x)$ pueden ser expresados como se muestra en las ecuaciones (1) y (2):

$$s_m(x) = b_0 + b_1 \cos\left(2\pi f_0 x + \frac{2\pi(m-1)}{M}\right) \text{ for } m = 1, 2, \dots, M \quad (1)$$

$$d(x) = b_0 + b_1 \cos\left(2\pi f_0 x + \frac{2\pi(m-1)}{M}\right) + \phi(x) \text{ for } m = 1, 2, \dots, M \quad (2)$$

b_0 : Iluminación de fondo.

b_1 : Representación del contraste entre franjas claras y oscuras.

ϕ : Cambio de fase causado por la distribución de altura del objeto. (El mapa de fase).

f_0 : Frecuencia espacial de los patrones de franjas.

$h(x)$ puede ser calculada utilizando las ecuaciones (3), (4) y (5).

$$S = \tan(2\pi f_0 x) = -\frac{\sum_{m=1}^M s_m(x) \sin\left(\frac{2\pi(m-1)}{M}\right)}{\sum_{m=1}^M s_m(x) \cos\left(\frac{2\pi(m-1)}{M}\right)} \quad (3)$$

$$D = \tan(2\pi f_0 x + \phi(x)) = -\frac{\sum_{m=1}^M d_m(x) \sin\left(\frac{2\pi(m-1)}{M}\right)}{\sum_{m=1}^M d_m(x) \cos\left(\frac{2\pi(m-1)}{M}\right)} \quad (4)$$

$$h(x) = \frac{I_0 \phi(x)}{\phi(x) - 2\pi f_0 d_0} \quad (5)$$

S, D: Variables intermedias.

d_0 : Distancia de la cámara al proyector.

I_0 : Distancia de la cámara al plano de referencia.

2.1.1.2 Perfilometría por cambio de fase con imágenes RGB

PSP con franjas RGB puede disminuir la cantidad de patrones necesarios en PSP y mejorar la velocidad de medición [28].

En las técnicas de un solo disparo, múltiples patrones de franjas con cambio de fase se codifican en un solo patrón de color a través de los canales RGB, que luego se proyectan en un objeto cuya superficie se adquirirá en una sola imagen a color [7]

En la Figura 2.2 se muestra el modelo de adquisición empleado en la perfilometría por cambio de fase RGB.

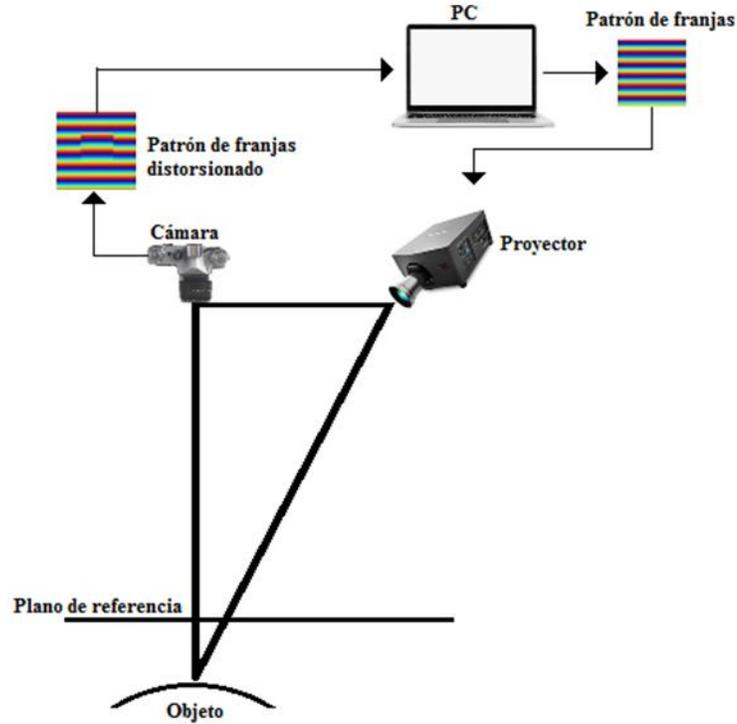


Figura 2.2 Diagrama de adquisición del método PSP.

De forma ideal, el patrón de franjas distorsionado que se captura al proyectar el patrón original sobre el objeto [1] [10], está dado por las siguientes ecuaciones (6), (7) y (8).

$$I_R(x, y) = a_R(x, y) + b_R(x, y) \cos \left[\phi(x, y) - \frac{2\pi}{3} \right] \quad (6)$$

$$I_G(x, y) = a_G(x, y) + b_G(x, y) \cos[\phi(x, y)] \quad (7)$$

$$I_B(x, y) = a_B(x, y) + b_B(x, y) \cos \left[\phi(x, y) + \frac{2\pi}{3} \right] \quad (8)$$

Dónde:

- $a(x, y)$ es una función asociada a la intensidad media.
- $b(x, y)$ es una función asociada con la modulación.
- (x, y) es el sistema de coordenadas de la cámara.
- $I_R(x, y), I_G(x, y), I_B(x, y)$ Son las intensidades de R, G y B.

Después de normalizar las intensidades de los patrones, el mapa de fase se puede obtener de la siguiente ecuación (9):

$$\phi(x, y) = \tan^{-1} \left[\sqrt{3} \frac{I_R(x, y) - I_B(x, y)}{2I_G(x, y) - I_R(x, y) - I_B(x, y)} \right] \quad (9)$$

2.1.1.3 Desenvolvimiento de fase

El problema del desenvolvimiento de fase bidimensional es un tema crucial en diversas aplicaciones como el radar de apertura sintética (SAR), la resonancia magnética (MRI) y el análisis de patrones de franjas [12]. Este problema es originado debido a que el rango de la función arco tangente, usada para recuperar la fase, está limitada entre $(-\pi, \pi)$. Por lo que las imágenes producidas por estas aplicaciones necesitan ser desenvueltas para formar un mapa de fase continuo [11].

El algoritmo básico para desenvolver la fase de una señal unidimensional [11] se puede visualizar en el diagrama de la Figura 3.

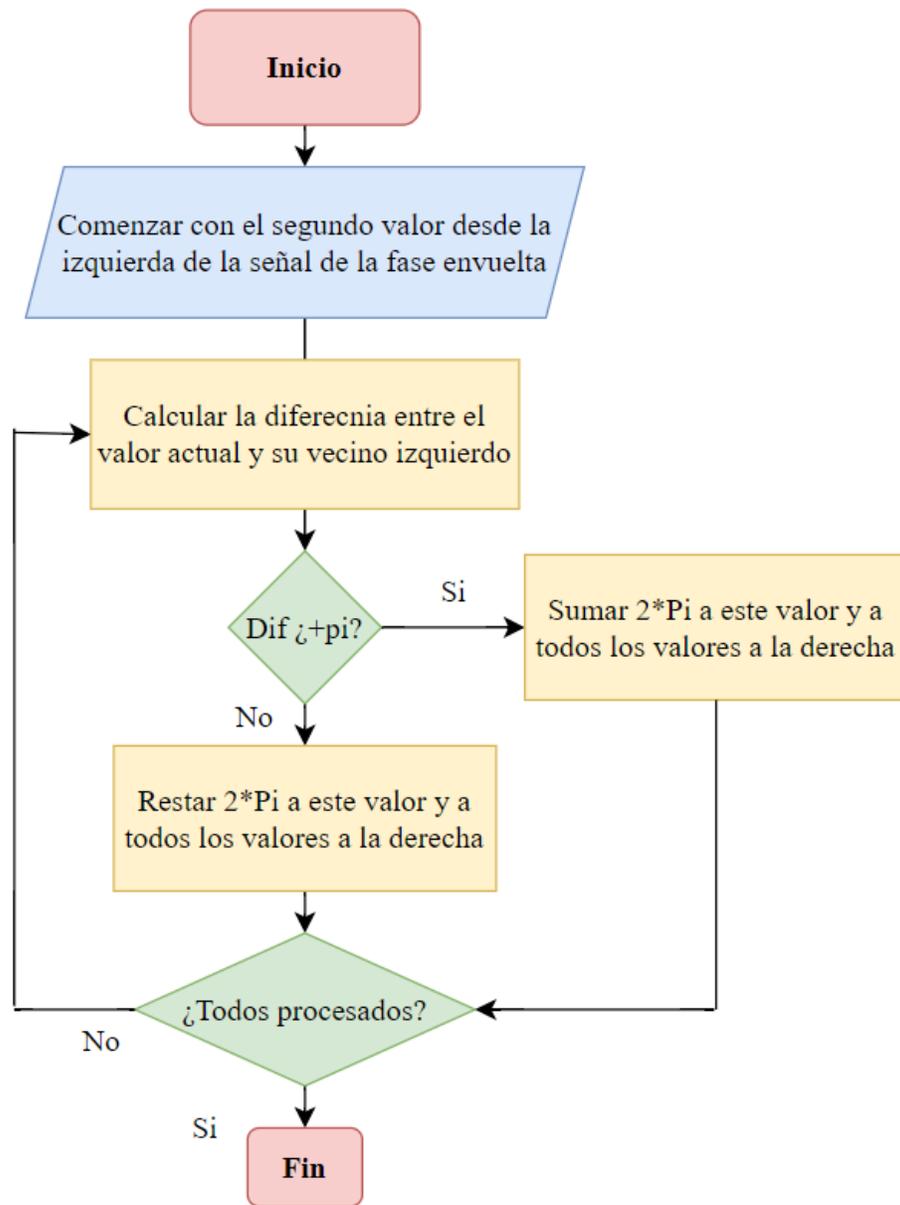


Figura 2.3 Algoritmo base del desenvolvimiento de fase.

La lógica del diagrama de la Figura 3 puede ser procesado a señales de dos dimensiones, como imágenes [17].

2.1.1.4 Desarrollo de fase temporal

Los métodos de desarrollo de fase espacial tienen como limitación los casos en donde se encuentran superficies discontinuas o múltiples objetos aislados, mientras que los métodos de desarrollo de fase temporal (TPU) no presentan estas limitaciones [29]. La idea en la que se basa el TPU es utilizar una secuencia de patrones de franjas con un pequeño cambio, que consiste en reducir la distancia entre las franjas proyectadas. Por tanto, el problema consiste en añadir un número suficientemente grande de franjas para reconstruir la forma completa [29] [18].

El caso básico de TPU emplea dos mapas de fase uno con un número menor de franjas proyectadas (l) y otro con un número mayor (h). Es posible llevar a cabo el algoritmo para realizar el TPU gracias a las relaciones existentes entre las fases expresadas en (10), a partir de las cuales es posible obtener k mediante operaciones algebraicas (11) [18] [30].

$$\left\{ \begin{array}{l} \Phi_l(x, y) = \phi_l(x, y) + 2\pi k_l(x, y) \\ \Phi_h(x, y) = \phi_h(x, y) + 2\pi k_h(x, y) \\ \Phi_h(x, y) = \left(\frac{f_h}{f_l}\right) \Phi_l(x, y) \end{array} \right\} \quad (10)$$

Dónde:

- l = Patrón de baja frecuencia.
- h = Patrón de alta frecuencia.
- $\phi(x, y)$ = Fase envuelta.
- $\Phi(x, y)$ = Fase desenvuelta.
- f = Frecuencia de los patrones.

Sustituyendo Φ de la tercera ecuación de (10) para obtener k_h .

$$\phi_h(x, y) + 2\pi k_h(x, y) = \left(\frac{f_h}{f_l}\right) \Phi_l(x, y) \quad (11)$$

$$k_h(x, y) = \text{Round} \left(\frac{\left(\frac{f_h}{f_l}\right) \Phi_l(x, y) - \phi_h(x, y)}{2\pi} \right) \quad (12)$$

Este enfoque aprovecha la independencia de la fase en cada píxel para evitar la propagación de errores a través de la imagen, lo que es especialmente ventajoso en superficies con discontinuidades [21].

2.2 Inteligencia Artificial

La inteligencia artificial (IA) nació en la década de 1950., esta puede definirse como el esfuerzo por automatizar tareas intelectuales normalmente realizadas por humanos [31].

Durante bastante tiempo, la mayoría de expertos creían que la IA a nivel humano podría lograrse al tener programas con un conjunto suficientemente grande de reglas explícitas para manipular conocimiento almacenado en bases de datos explícitas. Este enfoque se conoce como IA simbólica [31].

Aunque la IA simbólica demostró ser adecuada para resolver problemas lógicos bien definidos, resultó ser intratable descubrir reglas explícitas para resolver problemas más complejos y confusos, como la clasificación de imágenes, el reconocimiento de voz o la traducción de lenguaje natural. Por lo que surgió un nuevo enfoque para tomar el lugar de la IA simbólica: aprendizaje automático (Machine Learning) [31].

En la Figura 4 se puede observar un diagrama de la Inteligencia artificial y dos de sus ramas principales.

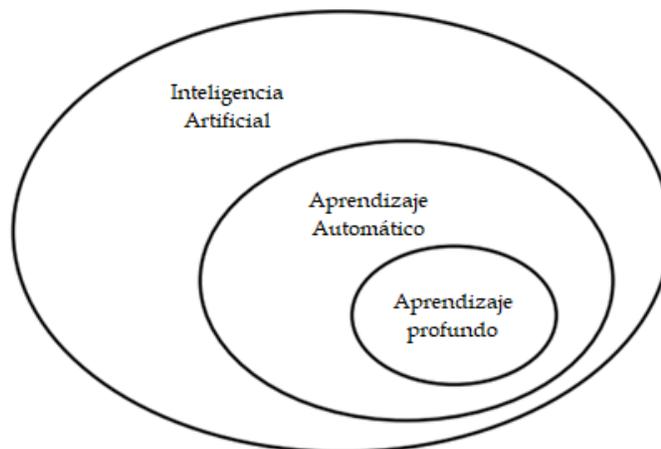


Figura 2.4 Áreas de estudio la inteligencia artificial. modificado de [31].

2.2.1 Redes Neuronales

Las redes neuronales son un tipo de modelo computacional (Figura 5) inspirado en el funcionamiento del cerebro humano, que consiste en un conjunto de unidades de procesamiento llamadas neuronas artificiales, conectadas entre sí mediante pesos sinápticos que determinan la intensidad de la señal transmitida. Las redes neuronales pueden aprender de los datos y adaptarse a nuevas situaciones, mediante un proceso de entrenamiento que ajusta los pesos sinápticos en función del error cometido entre la salida deseada y la obtenida [32].

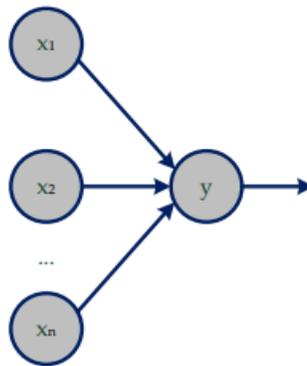


Figura 2.5 Diagrama básico de la red Neuronal. Obtenido de [32]

Existen varios tipos de redes neuronales. Las redes feedforward, también conocidas como perceptrones multicapa, se caracterizan por una estructura en capas donde la información fluye desde las entradas hacia las salidas sin retroalimentación, lo que las hace adecuadas para tareas de clasificación y regresión [33]. Las redes neuronales recurrentes, por otro lado, presentan conexiones cicladas que permiten el procesamiento de secuencias y datos temporales, convirtiéndolas en una elección natural para tareas como el procesamiento de lenguaje natural y la generación de secuencias [34]. Las redes neuronales convolucionales se diseñaron específicamente para capturar patrones espaciales en datos visuales, aprovechando las capas convolucionales para extraer características locales y jerárquicas en imágenes.

2.2.2 Machine Learning

La forma habitual de hacer que una computadora haga un trabajo útil es tener un programador humano escribiendo reglas a seguir para convertir los datos de entrada en respuestas apropiadas [31]. El aprendizaje automático le da la vuelta a esto: la máquina ve los datos de entrada y las respuestas correspondientes, y determina cuáles deben ser las reglas [31].

Un sistema de Machine Learning es entrenado en lugar de explícitamente programado. Se presenta con muchos ejemplos relevantes para una tarea y encuentra una estructura estadística en estos ejemplos que finalmente permite que el sistema genere reglas para automatizar la tarea [31].

En los diagramas de la Figura 6 se muestra la manera de programación clásica y la del aprendizaje máquina para la resolución de tareas.

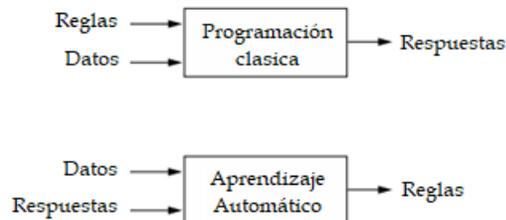


Figura 2.6 Lógica de la programación clásica vs el aprendizaje máquina.

El aprendizaje automático está relacionado con las estadísticas matemáticas, pero difiere de las estadísticas en varios aspectos importantes. A diferencia de la estadística, el aprendizaje automático tiende a manejar conjuntos de datos grandes y complejos para los cuales el análisis estadístico clásico, como el análisis bayesiano, no sería práctico. Como resultado, el aprendizaje automático, y especialmente el aprendizaje profundo (Deep Learning) [31].

El aprendizaje automático, entonces, se trata de hacer que las computadoras modifiquen o adapten sus acciones para que estas acciones sean más precisas, donde la precisión se mide por qué tan bien las acciones elegidas reflejan las correctas [35].

Proceso del aprendizaje automático [35]:

- Recolección de los datos y preprocesamiento.
- Selección de características.
- Selección de algoritmos
- Selección del modelo y parámetros.
- Entrenamiento
- Evaluación

2.2.3 Deep Learning

El deep learning, una subárea del aprendizaje automático, se basa en el uso de redes neuronales profundas para modelar y resolver problemas complejos. Estas redes, inspiradas en el cerebro humano, se componen de múltiples capas de neuronas artificiales que permiten el procesamiento de datos a diferentes niveles de abstracción [36].

2.2.3.1 Redes Neuronales Convolucionales

Las Redes Neuronales Convolucionales (CNN, por sus siglas en inglés) son un tipo de red neuronal especialmente efectiva para el procesamiento y análisis de datos con una estructura de cuadrícula, como las imágenes. Las CNN emplean capas convolucionales que aplican filtros (kernels) a la entrada para extraer características relevantes, seguidas de capas de pooling que reducen la dimensionalidad, conservando las características más importantes [37].

Las partes que conforman una CNN [38] se listan a continuación:

1. Capa de entrada (Input Layer): Esta capa toma la imagen de entrada y la pasa a través de la red. Cada imagen se representa como una matriz de píxeles, y cada píxel puede tener varios canales.
2. Capas Convolucionales (Convolutional Layers): Cada capa convolucional aplica múltiples filtros a la entrada para extraer características específicas. Estos filtros son pequeñas matrices que se deslizan por la imagen, calculando productos escalares en cada posición y produciendo mapas de características que resaltan regiones de interés.
3. Capas de Activación: Después de cada operación de convolución, se aplica una función de activación no lineal.
4. Capas de Agrupación (Pooling Layers): Estas capas reducen la dimensionalidad espacial de las representaciones de las características.
5. Capas Totalmente Conectadas (Fully Connected Layers): Estas capas se encargan de combinar las características extraídas para tomar decisiones finales sobre la clasificación o regresión del problema.
6. Capa de Salida (Output Layer): La capa de salida produce la salida final de la red, que depende del tipo de problema que estás resolviendo.

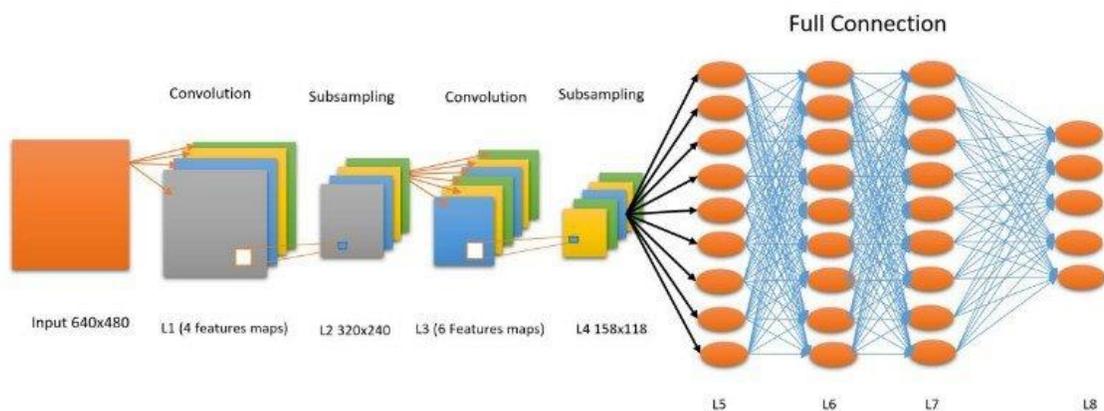


Figura 2.7 Arquitectura base de una CNN. Obtenida de [38]

Las CNN se han utilizado en una amplia variedad de aplicaciones, incluyendo el reconocimiento de imágenes, la detección de objetos y la segmentación semántica [39] [40]. En la Figura 2.5 se muestra una ilustración de la arquitectura básica de una CNN típica.

2.2.3.2 U-Net

U-Net es una arquitectura de red neuronal convolucional diseñada específicamente para la segmentación de imágenes biomédicas. Introducida por Ronneberger et al. en 2015, U-Net consta de una parte de contracción (encoder) y una parte de expansión (decoder), lo que permite capturar tanto el contexto global como los detalles locales de las imágenes [41].

La característica distintiva de U-Net es su estructura en forma de U Figura 2.6. Donde las características extraídas en la fase de contracción se combinan con las características de la fase de expansión mediante conexiones de salto. Esta combinación mejora la precisión de la segmentación, especialmente en tareas donde los bordes y las formas precisas son cruciales [42].

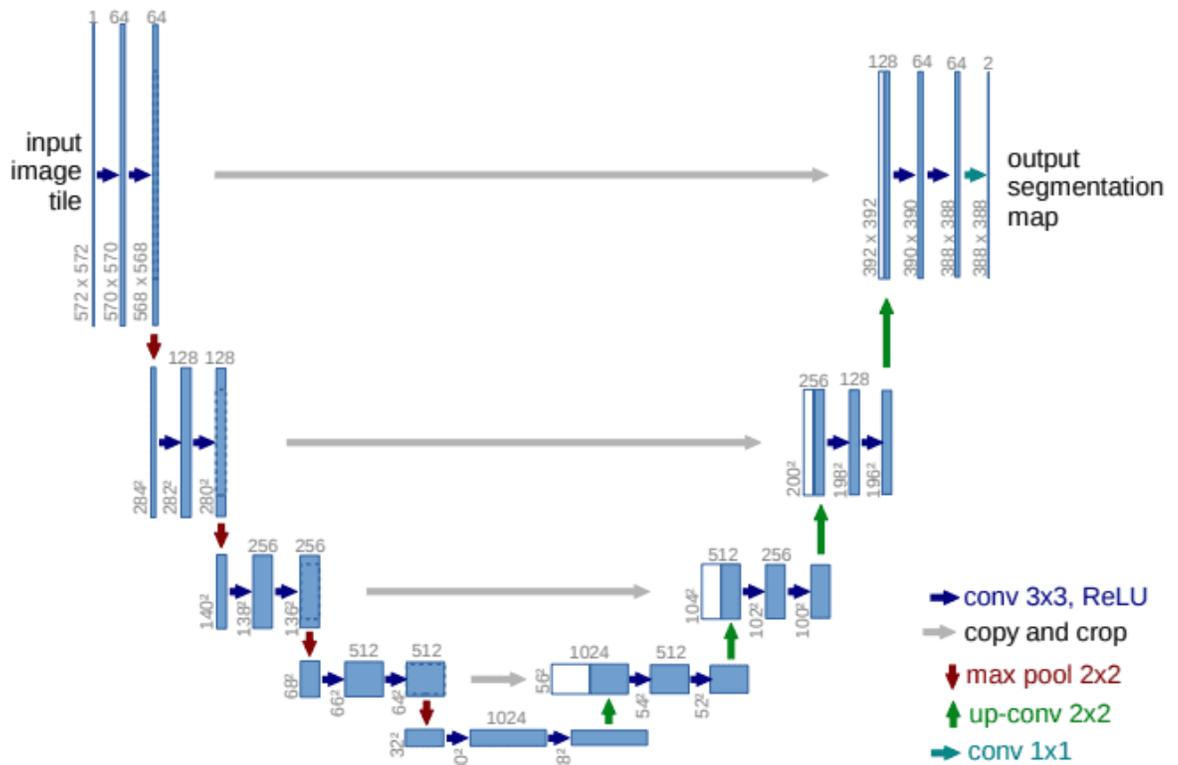


Figura 2.8 Arquitectura base de la U-Net. Obtenida de [43]

2.2.3.3 Generative Adversarial Network

Las Redes Generativas Adversariales (GAN, por sus siglas en inglés) son un tipo de modelo de deep learning introducido por Ian Goodfellow en 2014. Las GANs consisten en dos redes neuronales que compiten entre sí: el generador, que crea muestras falsas de datos, y el discriminador, que intenta distinguir entre muestras reales y falsas [44], cómo se observa en la Figura 2.7.

Esta estructura adversarial permite que las GANs generen datos sintéticos de alta calidad que son casi indistinguibles de los datos reales. Las aplicaciones de las GANs incluyen la generación de imágenes, la mejora de resolución de imágenes y la síntesis de datos [45].

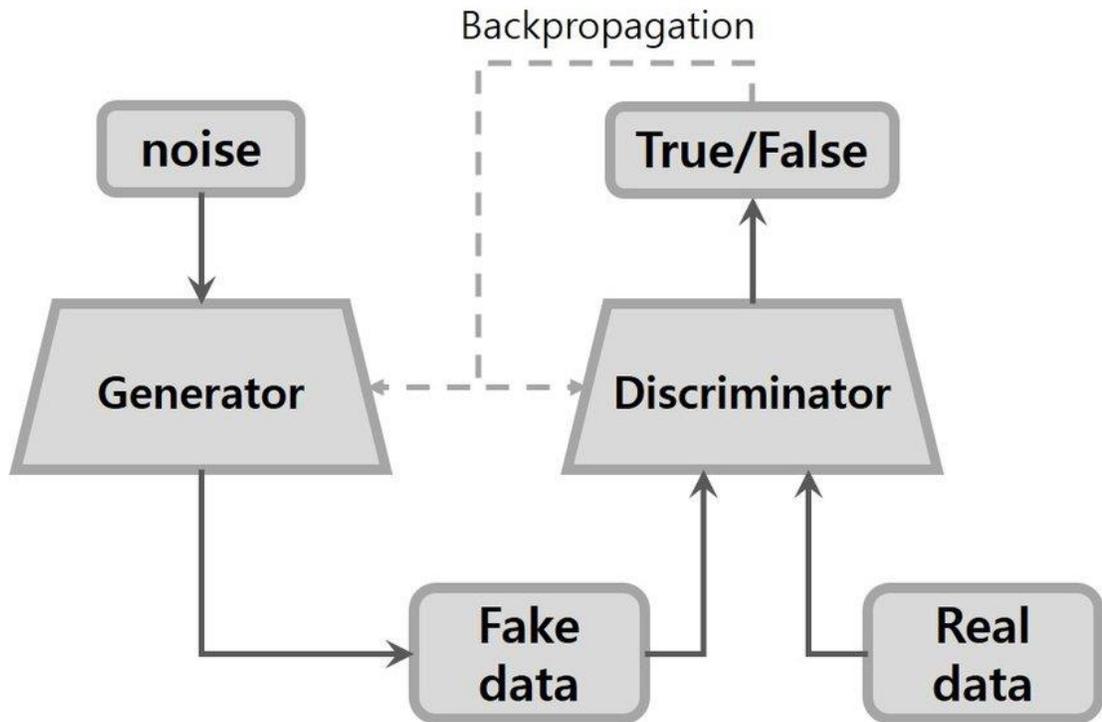


Figura 2.9 Diagrama de funcionamiento de una C-GAN. Obtenido de [46]

2.2.3.4 Patch GAN

La Patch GAN es una variante de las GANs en la que el discriminador opera sobre pequeños parches de la imagen en lugar de la imagen completa. Esta técnica es particularmente útil en aplicaciones donde es importante capturar la estructura local y los detalles finos de la imagen, como la síntesis de texturas y la superresolución de imágenes [47].

El discriminador de Patch GAN clasifica cada parche de la imagen de entrada como real o falso, lo que permite al generador enfocarse en producir detalles coherentes a nivel local. Esto resulta en imágenes más nítidas y detalladas comparadas con las generadas por GANs tradicionales [48].

2.2.3.5 Conditional Generative Adversarial Network

Las Redes Generativas Adversariales Condicionales (C-GAN) son una extensión de las GANs que permiten controlar el proceso de generación mediante la incorporación de información adicional, como etiquetas de clase o datos de entrada específicos. Las C-GAN pueden generar datos condicionados a una clase particular, mejorando la relevancia y la coherencia de los datos generados [49].

En lugar de generar datos desde cero, el generador de una C-GAN recibe tanto un vector de ruido como información condicional. Esto permite aplicaciones como la generación de imágenes etiquetadas, la traducción de imágenes a partir de descripciones textuales y la mejora de la resolución de imágenes condicional [50].

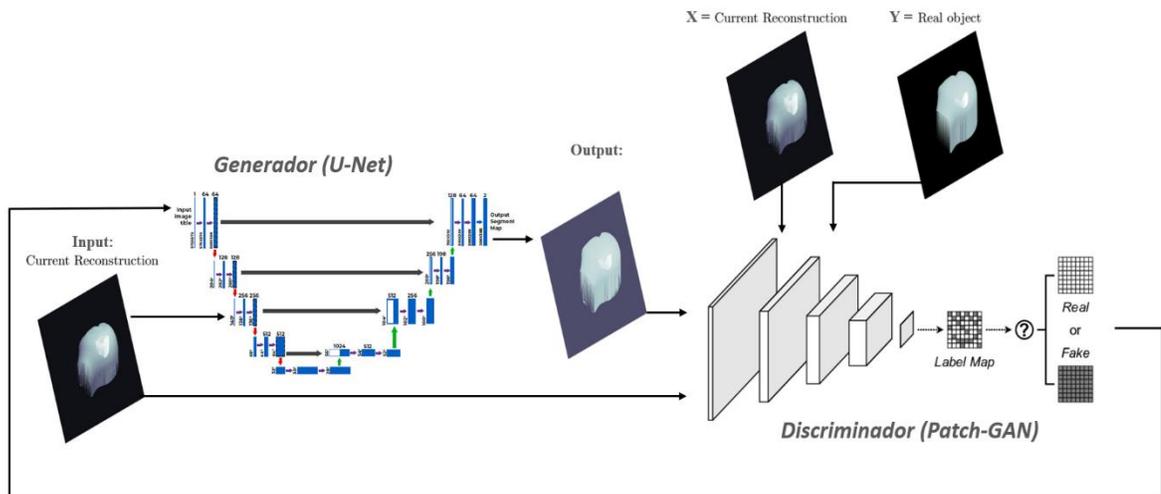


Figura 2.10 Arquitectura de la C-GAN. Elaboración propia.

2.3 Métricas

2.3.1 Error Cuadrático Medio (MSE)

El Error Cuadrático Medio Inverso (MSE) es una métrica utilizada para evaluar la calidad de las imágenes reconstruidas comparando la imagen original con la imagen generada. La fórmula del MSE se basa en el cálculo del error cuadrático medio (MSE) [51].

La fórmula del MSE es:

$$MSE = \frac{1}{mn} \sum_{i=1}^m \sum_{j=1}^n (I(i,j) - K(i,j))^2 \quad (13)$$

Donde:

- I : Imagen original
- K : Imagen procesada.
- m : Número de fila.
- n : Número de columna.

$K(i,j)$ es el valor del píxel correspondiente en la imagen reconstruida. El IMMSE proporciona una medida de cuán similares son las dos imágenes, siendo valores más altos indicativos de una mejor calidad [51]

2.3.2 Relación Señal-Ruido de Pico (PSNR)

La Relación Señal-Ruido de Pico (PSNR) es una métrica ampliamente utilizada para evaluar la calidad de las imágenes comprimidas o reconstruidas. El PSNR mide la relación entre la potencia máxima de una señal (la imagen original) y el ruido que afecta la calidad de su representación (la imagen reconstruida) [51].

Se define como:

$$PSNR = 10 * \log_{10}\left(\frac{Max^2}{MSE}\right) \quad (14)$$

Donde:

- *Max*: Valor máximo posible en un.
- *MSE*: Error Cuadrático Medio

Un PSNR más alto indica que la imagen reconstruida es más similar a la original, es decir, que tiene menos ruido. Valores típicos de PSNR para imágenes de alta calidad están en el rango de 30 a 50 dB.

2.4 Estado del Arte

La revisión del estado del arte se dividió en tres tipos de trabajos. El primer tipo correspondiente a métodos que no emplean desenvolvimiento de fase temporal, ni ocupan alguna arquitectura de DL, Tabla 2.1. El segundo tipo de trabajos revisados fueron aquellos en donde se emplea el TPU convencional o alguna de sus variantes, Tabla 2.2. Por último, se presentan los trabajos en los que se empleó alguna técnica de DL Tabla 2.3.

Tabla 2.1 Trabajos sobre Reconstrucción 3D mediante PSP espacial.

Artículo	Aportaciones	Enfoque	Resultados	Métricas
<i>Phase shifting algorithms for fringe projection profilometry: A review [20].</i>	Proporciona una revisión completa de los diferentes algoritmos de cambio de fase utilizados en técnicas de proyección de franjas.	Revisión de algoritmos de cambio de fase, enfatizando la exactitud y resolución espacial.	Identifica los algoritmos más eficaces para aplicaciones específicas de perfilometría.	Exactitud de $\pm 0.1\%$, resolución espacial de hasta 0.01 mm.
<i>Accurate 3D Shape Reconstruction from Single Structured-Light Image via Fringe-to-Fringe Network [1].</i>	Presenta una red neural profunda llamada Fringe-to-Fringe Network para la reconstrucción precisa de formas 3D a partir de una sola imagen de luz estructurada.	Implementación de redes neuronales profundas para analizar imágenes de luz estructurada	Logra reconstrucciones 3D de alta precisión usando datos mínimos de entrada.	Error medio cuadrático (RMSE) reducido a 0.05 mm.
<i>Fast and accurate phase unwrapping algorithm based on the transport of intensity equation [52].</i>	Desarrolla un algoritmo rápido y preciso para el desenvolvimiento de fase basado en la ecuación de transporte de intensidad.	Emplea la ecuación de transporte de intensidad para mejorar la rapidez y precisión del desenvolvimiento de fase.	Error medio cuadrático (RMSE) reducido a 0.05 mm.	Velocidad de procesamiento aumentada en 30%, precisión de ± 0.02 mm.

Tabla 1 Trabajos sobre desenvolvimiento de fase temporal.

Artículo	Aportaciones	Enfoque	Resultados	Métricas
<i>A novel phase-shifting profilometry to realize temporal phase unwrapping simultaneously with the least fringe patterns [15].</i>	Introduce un nuevo método de perfilometría por cambio de fase para realizar el desenvolvimiento temporal de fase con el menor número de patrones de franjas.	Utiliza técnicas de cambio de fase para mejorar la eficiencia en la captura de patrones.	Mejora la precisión y la eficiencia en la reconstrucción 3D con un número reducido de patrones de franjas.	Precisión mejorada en un 20%, reducción del 50% en patrones.
<i>Temporal phase unwrapping: application to surface profiling of discontinuous objects [21].</i>	Desarrolla una técnica de desempaquetado temporal de fase para analizar mapas de fase de objetos con discontinuidades de altura.	Utiliza la variación del desfase entre franjas a lo largo del tiempo para desempaquetar la fase en cada píxel de forma independiente.	Permite el perfilado preciso de objetos con discontinuidades de altura, disminuyendo el error en altura inversamente proporcional al número de mapas de fase utilizados.	Error en altura disminuido proporcionalmente al número de mapas de fase.
<i>Temporal phase unwrapping algorithms for fringe projection profilometry: A comparative review [29].</i>	Proporciona una revisión comparativa de diferentes algoritmos de desempaquetado temporal de fase usados en la proyección de franjas.	Compara múltiples algoritmos de desempaquetado temporal de fase, destacando sus ventajas y desventajas.	Ofrece una guía para la selección de algoritmos adecuados para diversas aplicaciones de perfilometría.	Exactitud mejorada hasta un 15%, variabilidad según el algoritmo específico.

Tabla 2 Trabajos sobre Desarrollo de fase temporal de Fase usando DL.

Artículo	Aportaciones	Enfoque	Resultados	Métricas
<i>Unifying temporal phase unwrapping framework using deep learning [22].</i>	Desarrolla un método de fase temporal utilizando técnicas de aprendizaje profundo.	Emplea redes neuronales profundas para aprender y predecir el desarrollo de fase.	Mejora la robustez y la precisión, especialmente en presencia de ruido y discontinuidades	Reducción del error medio cuadrático (RMSE) a 0.03 mm, aumento de velocidad de procesamiento en un 25%.
<i>Phase unwrapping based on channel transformer U-Net for 3D reconstruction [53].</i>	Propone un método de desarrollo de fase basado en un transformer de canal y U-Net para extraer valores de fase absolutos de patrones de fase envuelta.	Utiliza un modelo de aprendizaje profundo para desarrollar la fase directamente desde los patrones de fase envuelta.	Mejora la precisión en la reconstrucción 3D al manejar patrones de fase envuelta complejos.	Reducción del error de fase a ± 0.04 mm, mejora en la velocidad de procesamiento en un 30%.
<i>Deep Convolutional Neural Network Phase Unwrapping for Fringe Projection 3D Imaging [54].</i>	Desarrolla una red neuronal convolucional profunda para el desarrollo de fase preciso en la proyección de franjas.	Implementa un proceso de entrenamiento en dos pasos.	Mejora la precisión del desarrollo de fase en escenas complejas con ruido.	Reducción del RMSE a 0.05 mm,

3. Hipótesis

Combinando técnicas de Deep Learning y el método de perfilometría por cambio de fase con franjas RGB, se obtienen reconstrucciones tridimensionales con error reducido en comparación con el estado del arte.

4. Objetivos

4.1 Objetivo General

Implementar algoritmos que lleven a cabo la reconstrucción tridimensional de objetos combinando técnicas de Deep Learning y el método de perfilometría de proyección de franjas RGB para llevar a cabo el desenvolvimiento de fase en superficies complejas.

4.2 Objetivos Específicos

- Crear una base de datos conformada por imágenes de objetos sintéticos para probar los algoritmos de reconstrucción.
- Implementar algoritmos para el desenvolvimiento de la fase.
- Diseñar e implementar los modelos de Deep Learning con los cuales se llevará a cabo la reconstrucción tridimensional.

5. Metodología

En la Figura 5.1 se muestra la metodología general a manera de diagrama de flujo.

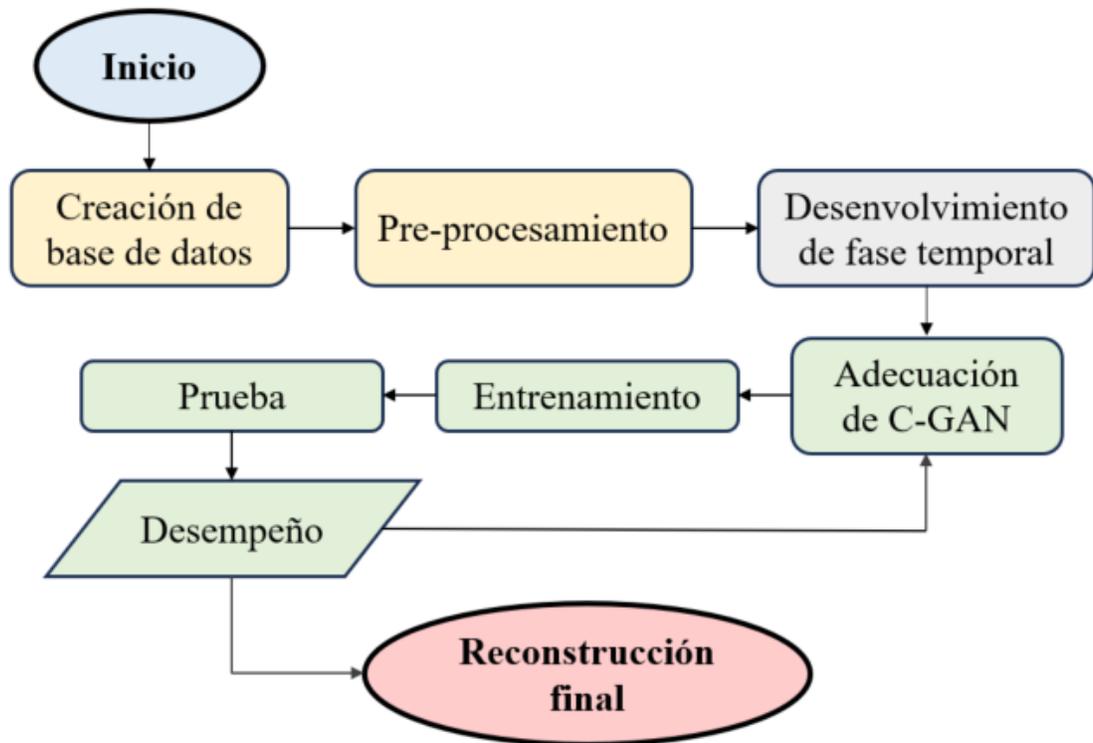


Figura 5.1 Metodología general.

5. 1 Creación de Base de Datos

Con el propósito de desarrollar y probar los algoritmos del desenvolvimiento de fase, se llevó a cabo la creación de una base de datos. En particular esta base de datos se compone de dos clases de datos, imágenes de objetos sintéticos e imágenes de objetos reales. En las siguientes secciones se describen de forma más específica las características, los materiales, y los protocolos de las imágenes adquiridas.

5.1.1 Base de Datos de Objetos Sintéticos

La base de datos de objetos sintéticos está compuesta por objetos tridimensionales renderizados, a los cuales les fueron proyectados patrones senoidales de acuerdo a la técnica de PSP-RGB.

5.1.1.1 Adquisición de imágenes con patrones proyectados

Los objetos sintéticos utilizados fueron descargados desde la página web *www.turbosquid.com*, mientras que la proyección de los patrones se realizó con *Blender*. A continuación, se listan los materiales empleados en la creación de esta base de datos.

1. Laptop Asus TUF Gaming F15.
2. *Software Blender 3.4*.
3. Modelos renderizados en formato. blend.

El escenario de proyección en *Blender* consta de los siguientes elementos.

- Cámara.
- Proyector con desfase de 0 grados (Rojo).
- Proyector con desfase de 120 grados (Verde).
- Proyector con desfase de 240 grados (Azul).
- Plano de referencia.

En la Figura 5.2 se ilustra el diagrama que muestra la disposición de los elementos anteriormente mencionados.

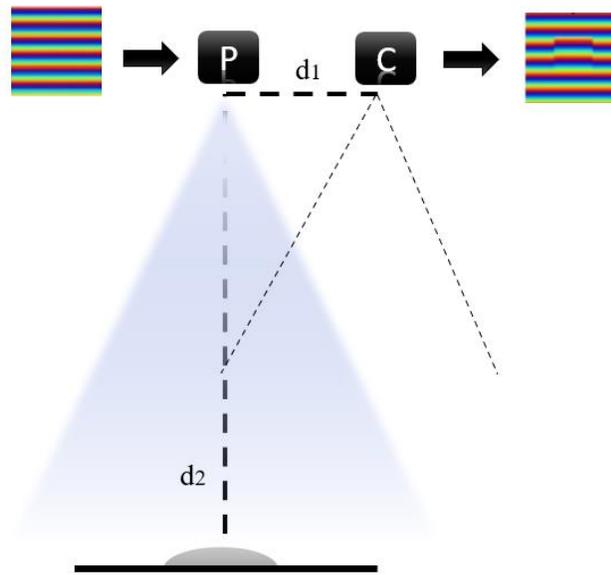


Figura 5.2 Disposición de los elementos en el escenario de proyección.
Donde d_1 y d_2 igual a 5cm y 15cm respectivamente.

Al capturar este escenario se obtiene una imagen de tres capas (RGB). Esto se ejemplifica en la Figura 5.3.

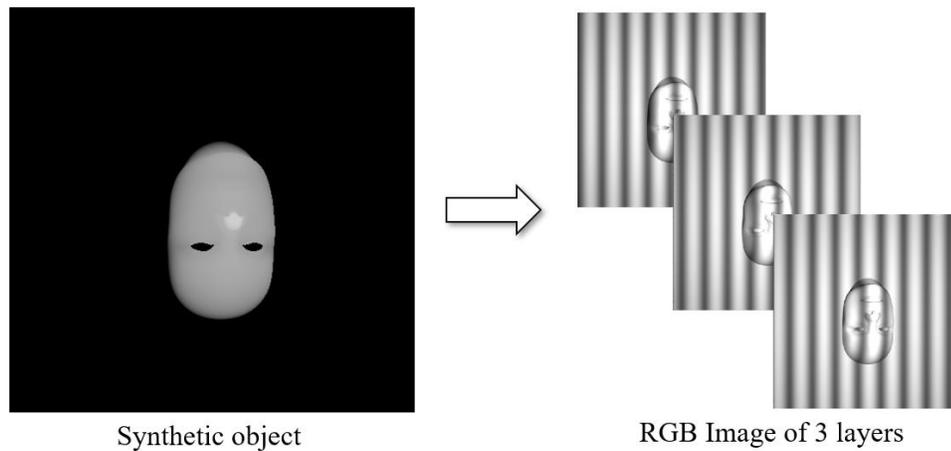


Figura 5.3 Entrada y salidas del escenario de proyección de patrones de franjas senoidales.

Debido a que el desarrollo del trabajo requiere obtener imágenes con patrones de diversas frecuencias proyectadas sobre un mismo objeto, se desarrolló un algoritmo en Python (Figura 5.4) que permite realizar lo anterior mencionado de forma automática.

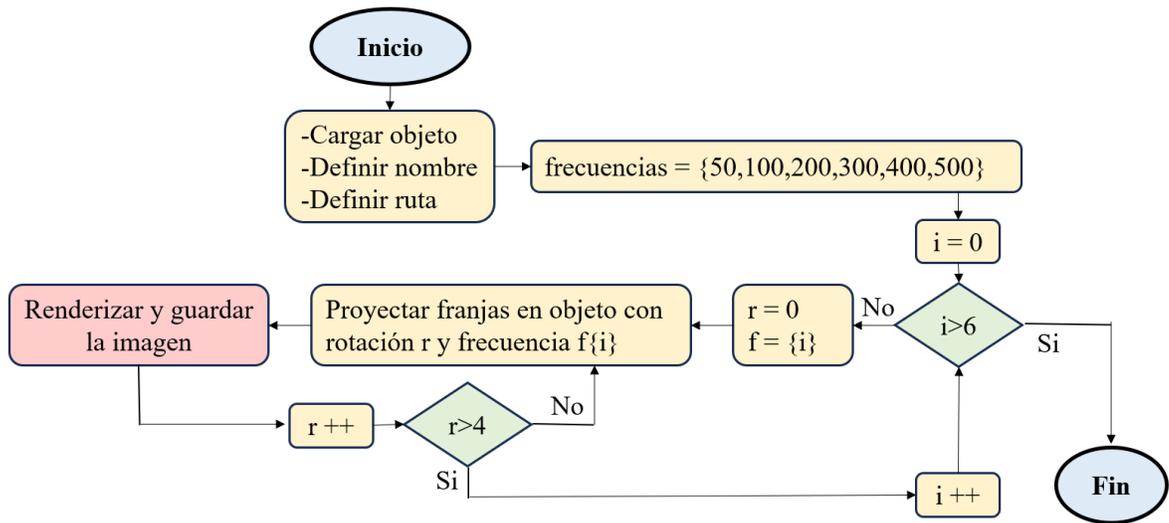


Figura 5.4 Script de captura de imágenes automatizado.

Cómo salida por objeto de este script de adquisición, se obtuvieron 6 imágenes por rotación. En este caso al estar programadas 4 rotaciones se generaron un total de 24 imágenes de 3 capas. Cada rotación se toma como un objeto independiente.

Por lo que cada objeto sintético produce 4 sets independientes (Figura 5.5) de 6 imágenes correspondientes a cada una de las frecuencias proyectadas.

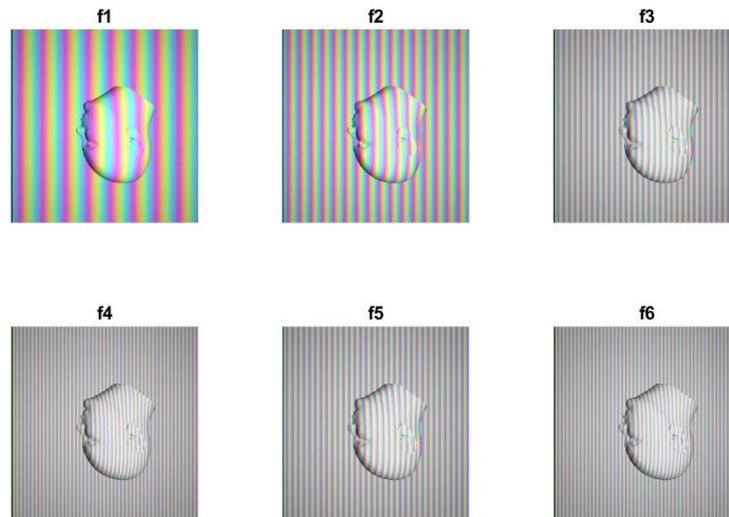


Figura 5.5 Ejemplo de set de imágenes de objeto con patrones proyectados.

5.1.1.2 Adquisición de objetos de referencia

Además de las imágenes con los patrones proyectados, es necesario obtener el objeto de referencia (*ground truth*), no solo para comparar el resultado, sino también como salida esperada en el entrenamiento de nuestra red.

Para poder utilizar el *ground truth* es necesario realizar las siguientes transformaciones para que su información sea interpretada correctamente en las etapas posteriores.

1. Cargar el objeto
2. Modificar el tipo de captura a ortogonal
3. Normalizar los valores
4. Cambiar la textura

5. 2 Pre Procesamiento

La etapa de Pre Procesamiento fue realizada usando el software *Matlab R2019b*. En ella se adecuaron las imágenes de entrada para poder calcular amplitud y fase.

Cada una de las etapas del preprocesamiento se realizó usando todas las imágenes que conforman un set (6 imágenes).

5.2.1 Preparación de imágenes

La secuencia que siguió cada una de las imágenes del set fue la siguiente:

1. Cargar set de imágenes con patrones proyectados (Figura 5.5)
2. Separar cada imagen en sus tres canales RGB.
3. Recorte de imágenes.
4. Conversión de tipo de dato, de *uint8* a *float*.
5. Cálculo del plano de referencia para cada frecuencia.

5.2.2 Calculo de amplitud y fase

Usando la ecuación 15, se calcula la fase de cada una de las imágenes del set, así como su plano de referencia correspondiente.

$$phase(x, y) = \arctan\left(\frac{I_R(x,y) - 2I_G(x,y) + I_B(x,y)}{I_R(x,y) - I_b(x,y)}\right) \quad (15)$$

Por lo que a la salida del preprocesamiento se tienen 6 fases envueltas correspondientes a cada objeto junto con 6 fases desenvueltas que corresponden a sus planos de referencia.

5. 3 Desenvolvimiento de fase Temporal

El algoritmo de desenvolvimiento de fase temporal (TPU) empleado es una variante del algoritmo descrito en el artículo *Temporal phase unwrapping using deep learning* [30].

El caso base de la implantación de la variante de TPU se describe a continuación. Donde la fase del patrón con menor número de franjas (l) sirve como guía para desenvolver la fase del patrón con mayor número de franjas (h). El proceso a seguir se enlista a continuación:

1. Obtener las fases de ambos patrones ϕ_l y ϕ_h
2. Aplicar desenvolvimiento de fase a ϕ_l $\rightarrow \Phi_l$.
3. Calcular la diferencia entre Φ_l y ϕ_h $\rightarrow diff$
4. Aplicar desenvolvimiento de fase a $diff$ $\rightarrow DIFF$
5. Sumar $DIFF + \phi_h$ $\rightarrow \Phi_h$

El algoritmo empleado para el desenvolvimiento de fase es explicado en la sección 5.3.1

5.3.1 Desenvolvimiento de fase

El algoritmo para llevar a cabo el desenvolvimiento de fase fue implementado en el Software *Matlab R2019b*. Esta implementación lleva el nombre de stiepu, fue descargada desde la página MathWorks y está basado en el artículo [52].

5.3.2 Desarrollo de Fase Temporal Iterativo

La implementación de TPU se realiza de forma iterativa. Esto se consigue utilizando cada fase desenvuelta como guía del patrón subsecuente a desenvolver en el siguiente diagrama (Figura 5.6) se muestra el proceso que se siguió para obtener la fase desenvuelta de cada patrón.

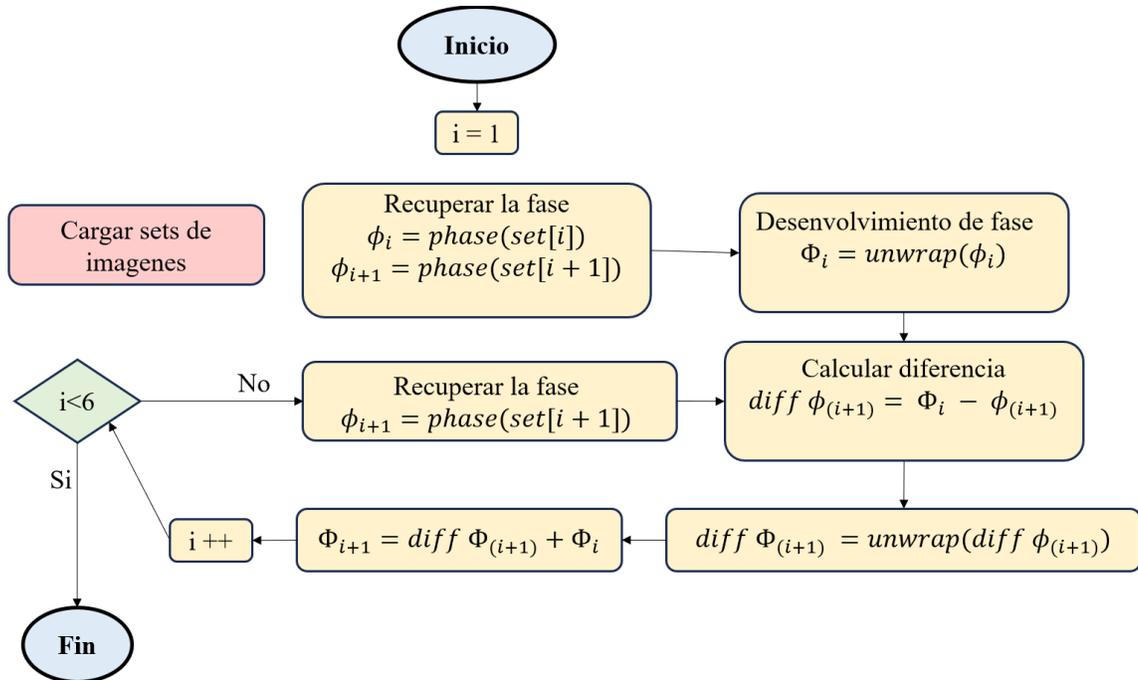
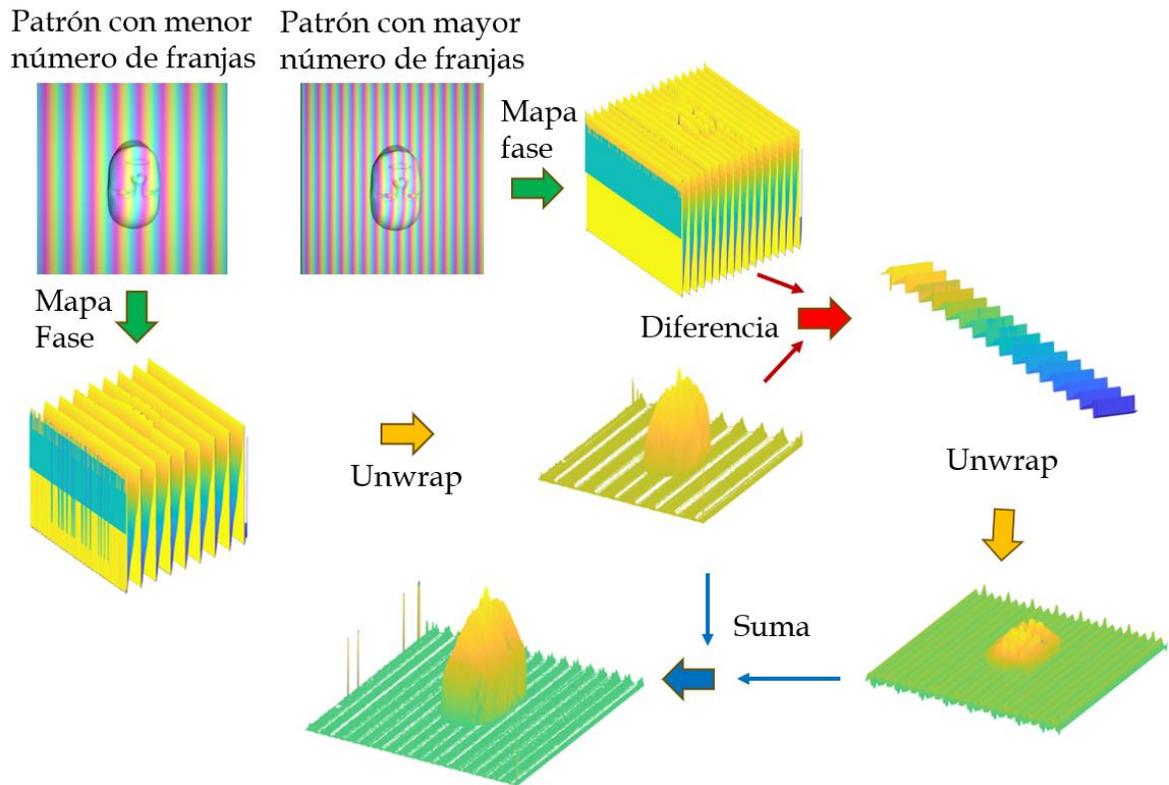


Figura 5.6 Proceso de TPU iterativo

Una vez que se tiene la fase desenvuelta del patrón con mayor número de franjas, se desenvuelve la fase de su plano correspondiente. Se procede a realizar la resta para así obtener la figura aislada, en la Figura 6.7 se ilustra el proceso descrito.



5.3.3 Almacenamiento

Para el posterior uso de las reconstrucciones obtenidas para entrenamiento y prueba de la C-GAN, se almacenaron las reconstrucciones obtenidas con el método de TPU de cinco iteraciones (TPU5) junto a su correspondiente Ground Truth. Ambas superficies fueron guardadas como archivo .csv.

5. 4 Conditional Generative Adversarial Network

Se importa el modelo de la C-GAN desde la página oficial de *TensorFlow* <https://www.tensorflow.org/tutorials/generative/pix2pix?hl=es-419> el cual está basado en el concepto de traducción de imágenes del artículo pix2pix [47].

5.4.1 Material

Los recursos de la computadora que implementan, entrenan y prueban el modelo se presenta en la Tabla 5.1.

Tabla 5.1 Especificaciones del equipo de cómputo.

Componente	Nombre	Especificaciones	Número
<i>Procesador</i>	Intel i7	4 núcleos a 4.2 GHz	1
<i>Tarjeta Gráfica</i>	GEFORCE GTX 1070	8 GB RAM	1
<i>Memoria RAM</i>	Asus	16 GB	4

Esto fue trabajado en el lenguaje de programación *Python*. Las especificaciones de los software y librerías utilizadas se encuentran en la Tabla 5.2

Tabla 3.2 Versiones de Software utilizados

Software	Versión
<i>Python</i>	3.8
<i>TensorFlow</i>	2.3.0
<i>Cuda</i>	10.1

5.4.2 Datos

5.4.2.1 Imágenes de Entrada y Etiquetas

Para el entrenamiento y prueba del modelo C-GAN, se emplearon dos conjuntos de imágenes:

- Entradas: Las reconstrucciones obtenidas mediante TPU después de 5 iteraciones.
- Etiquetas: Como etiquetas, se utilizaron imágenes *Ground Truth*, es decir, representaciones exactas o de alta fidelidad de los objetos.

5.4.2.2 Distribución de las imágenes

La distribución de las imágenes para las fases de entrenamiento y prueba se organizó de la siguiente manera: 90% de para Entrenamiento y 10% de las imágenes para Prueba.

5.4.2.3 Procesamiento de las imágenes

Antes de utilizar las imágenes para el entrenamiento y la evaluación del modelo C-GAN, se realizó un proceso de preparación y normalización para asegurar que las imágenes fueran adecuadas para el proceso de aprendizaje profundo. Este procesamiento constó de dos pasos:

- Padding de las Imágenes Para estandarizar el tamaño de todas las imágenes utilizadas y facilitar el procesamiento al utilizar dimensiones que fueran potencia de dos, se aplicó un padding a cada imagen para ajustarlas a un tamaño uniforme de 512x512 píxeles. Este paso garantiza que el modelo procese imágenes de dimensiones consistentes. El padding se realizó de manera que se preservara la relación de aspecto original de las imágenes,

añadiendo márgenes iguales en todos los lados de la imagen hasta alcanzar las dimensiones deseadas.

- Normalización: Posteriormente, todas las imágenes fueron normalizadas para tener valores de píxel en el rango de -1 a 1. Ya que mejora la convergencia del modelo durante el entrenamiento.

5.4.3 Arquitectura

La C-GAN utilizada consta de dos componentes Generador y Discriminador.

5.4.3.1 Generador

Generador: El generador de la C-GAN está diseñado con una arquitectura de red U-Net. El diagrama de la arquitectura completa se encuentre en la sección de Anexos (Anexo 3).

Utiliza dos funciones de pérdida:

- 1- La pérdida adversaria obliga al generador a crear imágenes que el discriminador clasifique como reales.
- 2- La pérdida L1 minimiza la diferencia absoluta entre las imágenes generadas y las imágenes objetivo.

Configuración Técnica:

- 1- Inicialización: Se utiliza un inicializador normal aleatorio con media 0 y desviación estándar de 0.02.
- 2- Activación Final: La capa de salida utiliza la función de activación *tanh*, que escala la salida a un rango de -1 a 1.
- 3- Downsample: Número de capas: Tamaño del Kernell: 7x7, Función de activación: Relu
- 4- Upsample: Número de capas: Tamaño del Kernell: 7x7, Función de activación: Relu

5.4.3.1 Discriminador

El discriminador emplea una arquitectura PatchGAN (Anexo 4), que clasifica segmentos de la imagen (parches) como reales o falsos, permitiendo una evaluación detallada y localizada de la autenticidad de las imágenes generadas.

Entradas:

El discriminador recibe dos imágenes: una imagen de entrada (inp) y una imagen objetivo (tar). Ambas tienen un tamaño de 512x512 píxeles y un solo canal.

Downsampling:

1. Primera Capa (down1): Utiliza 64 filtros y un kernel de 4x4, con un stride de 2 y sin normalización por lotes, reduciendo la resolución espacial a 128x128.
2. Segunda Capa (down2): Aumenta los filtros a 128 y sigue reduciendo la resolución espacial a 64x64, con normalización por lotes incluida.
3. Tercera Capa (down3): Continúa la tendencia con 256 filtros, disminuyendo la resolución a 32x32, también con normalización por lotes.

Procesamiento adicional:

1. Zero Pad 1: Añade un relleno de ceros alrededor de la salida de la tercera capa de downsampling.
2. Convolución: Aplica una convolución con 512 filtros y un kernel de 4x4, con un stride de 1.

Función de activación: Leaky ReLU

Salida: ltro con un kernel de 4x4 y un stride de 1 para producir una salida escalar.

6. Resultados y Discusión

6.1 Creación de base de datos de objetos digitales

A partir de 25 objetos sintéticos (Figura 6.1), con sus correspondientes cuatro rotaciones, se creó el primer base de datos, en la Tabla 6.1 se desglosa el contenido de la misma; tipo de imagen, número de elementos, peso, y terminación para su identificación.

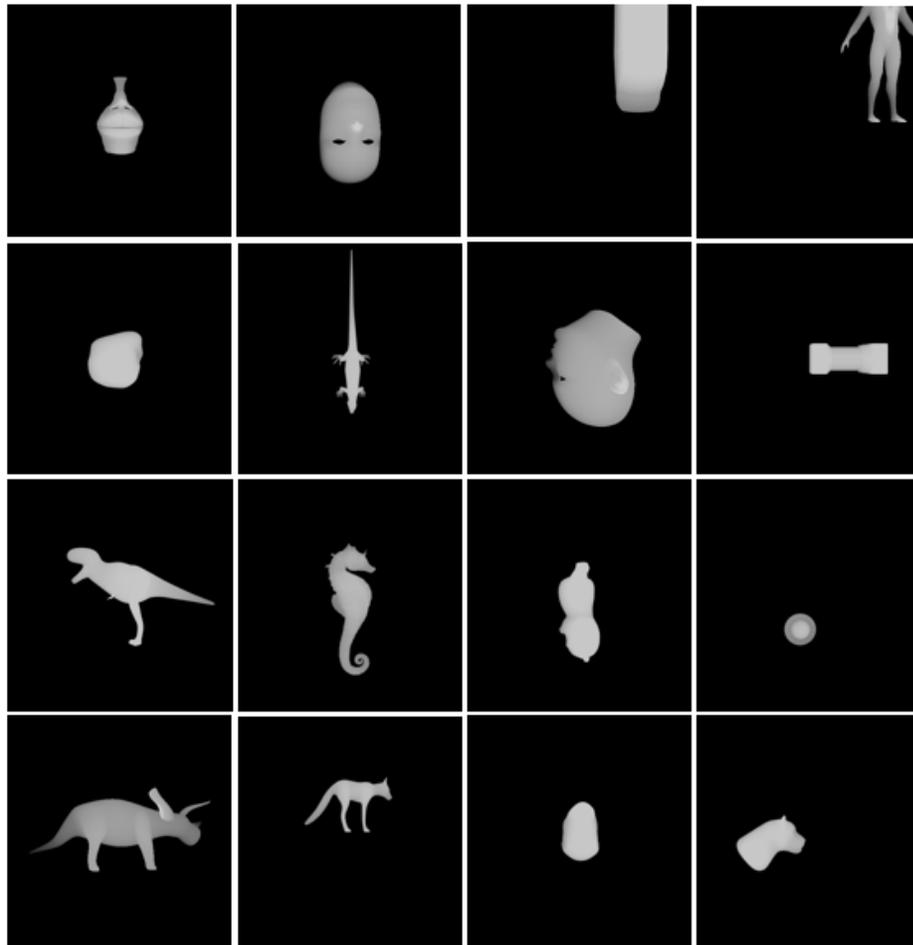


Figura 6.1 Ejemplo de objetos sintéticos.

Tabla 6.1 Imágenes Adquiridas.

Tipo	No. Imágenes	Terminación	Ejemplo
<i>Patrón R 8Hz</i>	100	r_50	
<i>Patrón G 8Hz</i>	100	g_50	
<i>Patrón B 8Hz</i>	100	b_50	
<i>Patrón R 16Hz</i>	100	r_100	
<i>Patrón G 16Hz</i>	100	g_100	
<i>Patrón B 16Hz</i>	100	b_100	
<i>Patrón R 32Hz</i>	100	r_200	
<i>Patrón G 32Hz</i>	100	g_200	
<i>Patrón B 32Hz</i>	100	b_200	
<i>Patrón R 48Hz</i>	10	r_300	
<i>Patrón G 48Hz</i>	100	g_300	
<i>Patrón B 48Hz</i>	100	b_300	
<i>Patrón R 64Hz</i>	100	r_400	
<i>Patrón G 64Hz</i>	100	g_400	
<i>Patrón B 64Hz</i>	100	b_400	
<i>Patrón R 80Hz</i>	100	r_500	
<i>Patrón G 80Hz</i>	100	g_500	
<i>Patrón B 80Hz</i>	100	b_500	
<i>Ground_Truth</i>	100	_real	
<i>Mascara</i>	100	_mask	

6.2 Desenvolvimiento de fase temporal

El funcionamiento del algoritmo de desenvolvimiento de fase temporal de una iteración se encuentra ilustrado en la Figura 6.2. En ella se ocuparon dos patrones, uno con menor número de franjas, del que se obtuvo la fase guía y el segundo con mayor número de franjas, fase objetivo. Donde a) la fase desenvuelta guía, b) que la diferencia entre la fase desenvuelta objetivo envuelta s y la fase desenvuelta objetivo, finalmente c) corresponde a la fase desenvuelta objetivo.

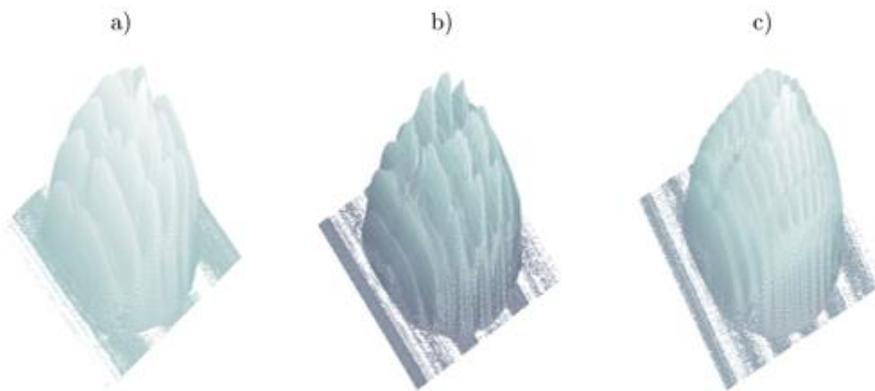


Figura 6.2 Proceso de la reconstrucción a través del TPU. a) Fase guía desenvuelta, b) diferencia entre la fase envuelta objetivo y la fase desenvuelta guía c) Fase desenvuelta objetivo.

Para determinar la calidad de las reconstrucciones obtenidas empleando el método clásico contra el desenvolvimiento de fase temporal con diverso número de iteraciones se comparó cada reconstrucción obtenida contra el *Ground Truth*, para así obtener el MSE (Tabla 7) y el PSNR (Tabla 8).

En las tablas mencionadas, las columnas corresponden a uno de los ocho objetos elegidos de forma aleatoria, mientras que las filas representan el método empleado. Las tablas completas con los valores de los 100 objetos reconstruidos se encuentran en la sección de Anexos (Anexos 1 y 2).

Tabla 6.2 MSE de objetos reconstruidos con TPU.

Método empleado	Face	Deadpool	Ear	Lizard	Pillar	Seahorse	T-rex	Body
Clásico	0.0963	0.2413	0.2140	0.2905	0.2044	0.2786	0.3386	0.2486
TPU1	0.0673	0.1754	0.1443	0.1578	0.1108	0.2396	0.2562	0.2233
TPU2	0.0568	0.1723	0.1244	0.1122	0.1211	0.1790	0.2695	0.1624
TPU3	0.0523	0.1722	0.1089	0.1208	0.0879	0.1761	0.2543	0.1603
TPU4	0.0507	0.1709	0.1033	0.1253	0.0716	0.1684	0.2509	0.1543
TPU5	0.0472	0.1664	0.0992	0.1118	0.0682	0.1648	0.2495	0.1564

Tabla 6.3 PSNR de objetos reconstruidos con TPU.

Método empleado	Face	Deadpool	Ear	Lizard	Pillar	Seahorse	T-rex	Body
Clásico	10.163	6.1744	6.696	5.368	6.894	5.550	4.703	6.0442
TPU1	11.720	7.5598	8.407	8.018	9.554	6.204	5.764	6.5109
TPU2	12.457	7.6370	9.052	8.498	9.169	7.441	5.694	8.1172
TPU3	12.818	7.6387	9.631	8.736	10.562	7.571	5.947	7.8953
TPU4	12.945	7.6718	9.857	9.021	11.449	7.735	6.005	8.0572
TPU5	13.258	7.7893	10.0348	9.514	11.2768	7.830	6.029	8.3658

El comportamiento del MSE y el valor promedio de cada método empleado muestran en las en la Figura 6.3 y 6.4 respectivamente.

Mientras que en la Figura 6.5 y 6.6 se muestra comportamiento del PSNR y el valor promedio de cada método.

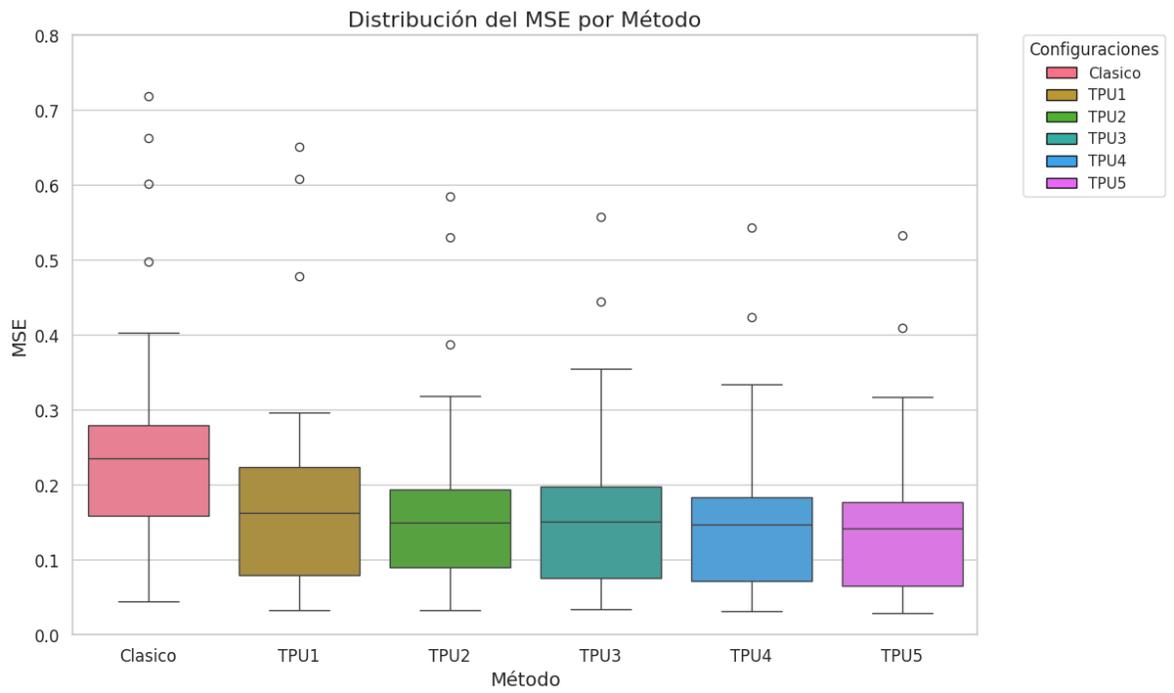


Figura 6.3 Comportamiento del MSE por método.

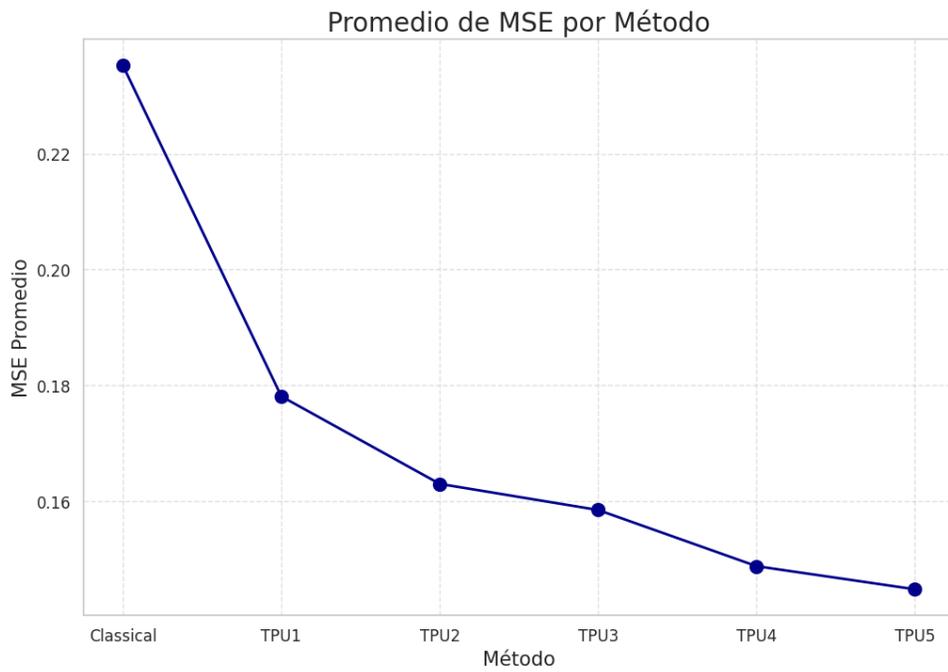


Figura 6.4 Valor promedio del MSE por método.

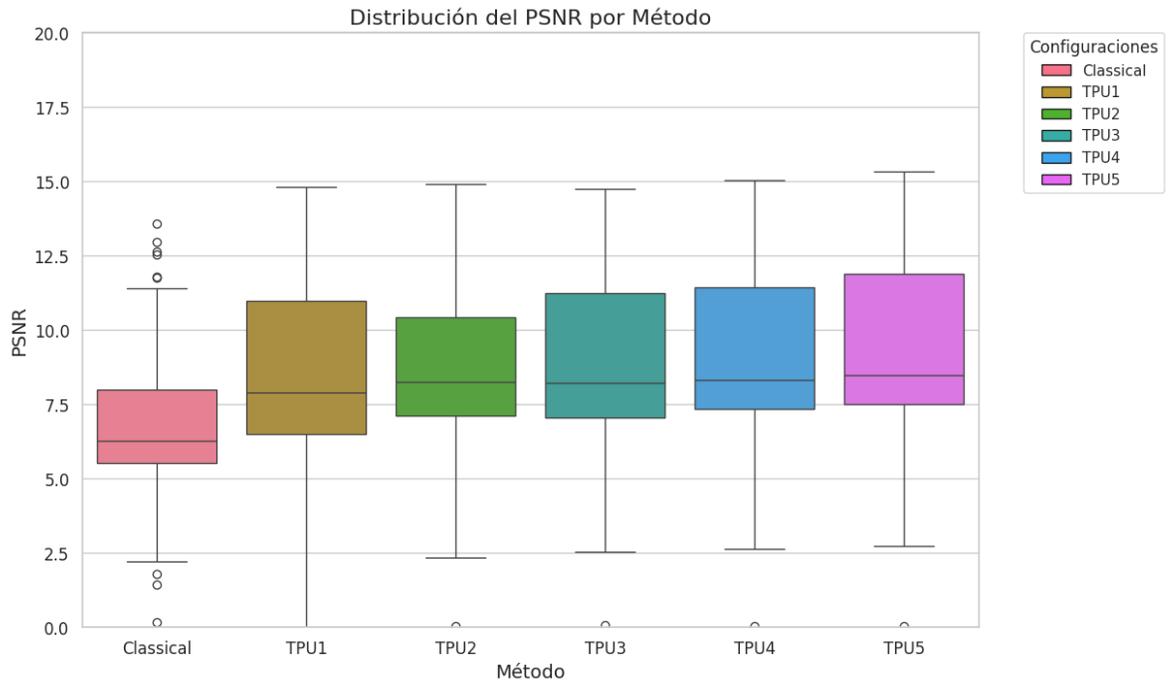


Figura 6.5 Comportamiento del PSNR por método

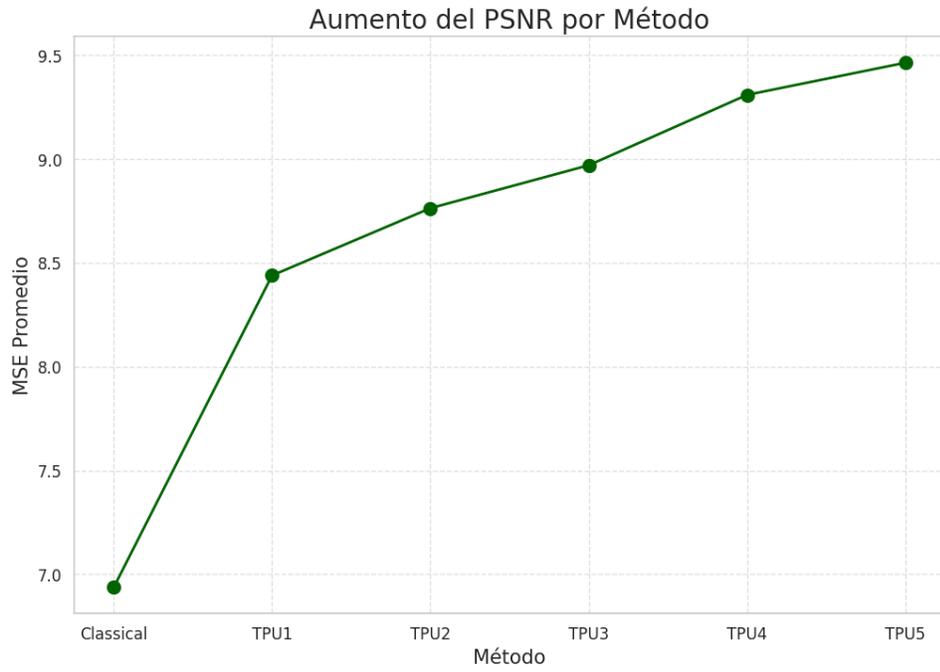


Figura 6.6 Valor promedio del PSNR por método.

En la Figura 6.7, para comparar el desempeño entre las reconstrucciones obtenidas con TPU respecto al método clásico, se estableció al MSE del método clásico como el 100% de MSE y se calculó el porcentaje al cual equivale el MSE de cada método.

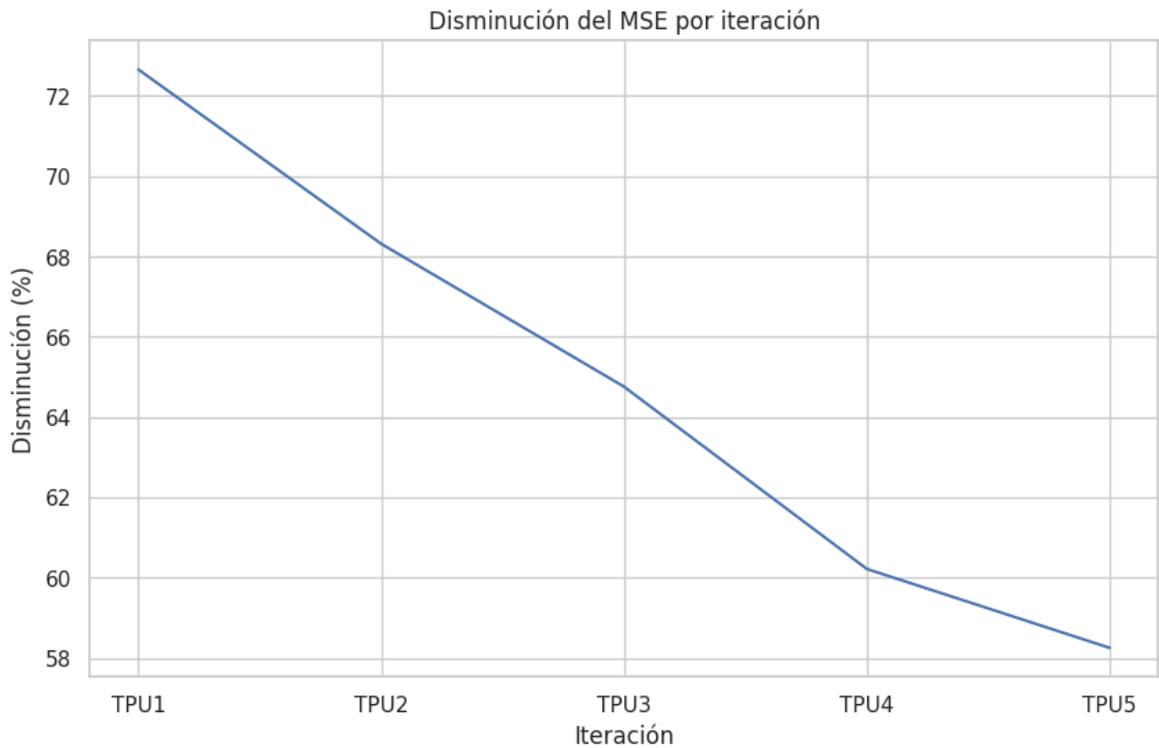


Figura 6.7 Reducción del MSE por Iteración.

En la Figura 6.8 se muestra explícitamente la evolución de cada reconstrucción del objeto *Face* obtenida de acuerdo al número de iteraciones empleadas. Así mismo se incluyó un acercamiento en una región de la reconstrucción con el fin de visualizar de forma más detallada la calidad de la figura obtenida.

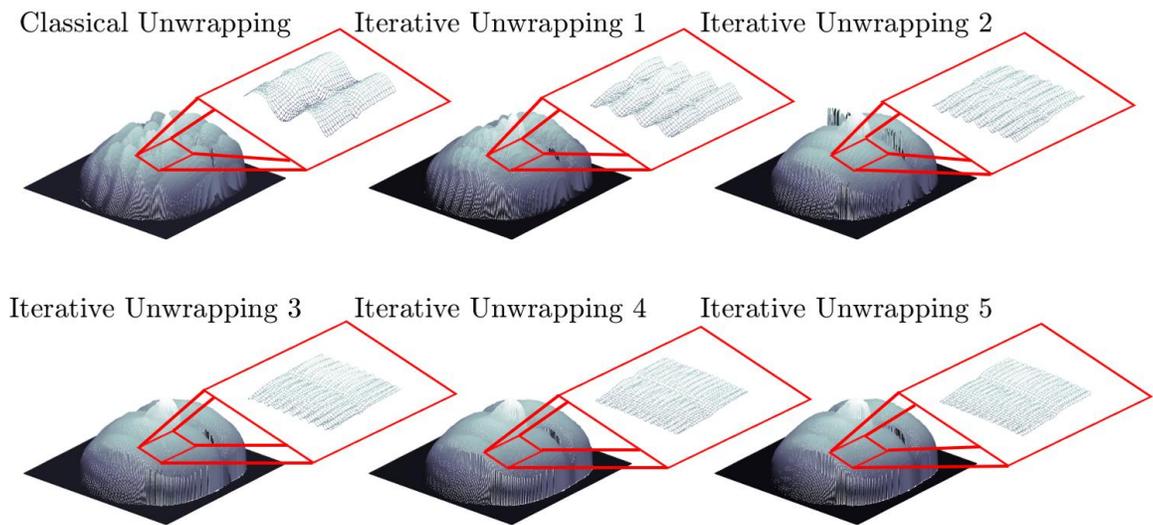


Figura 6.8 Reconstrucciones obtenidas por número de iteración.

En la Figura 6.9 se aprecia una visualización desde el plano sagital del objeto *face*, comparando el *Ground Truth* contra sus respectivas reconstrucciones.

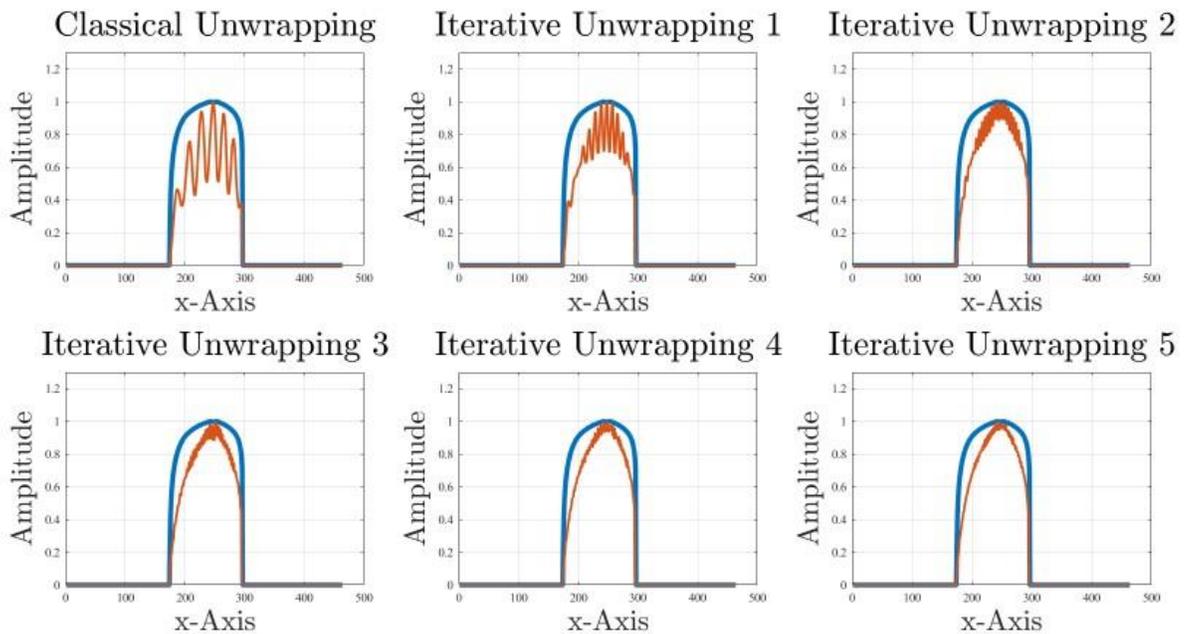


Figura 6.9 Reconstrucciones obtenidas por iteración (corte transversal).

Las Figuras 6.10 y 6.11 están compuesta por tres columnas, la primera correspondiente a las reconstrucciones obtenidas por el método clásico, la segunda se muestran las reconstrucciones obtenidas con cinco iteraciones de TPU y en la tercera columna se muestran los *Ground Truth*. Los objetos incluidos en las figuras corresponden a los ocho objetos referidos en la Tabla 6.

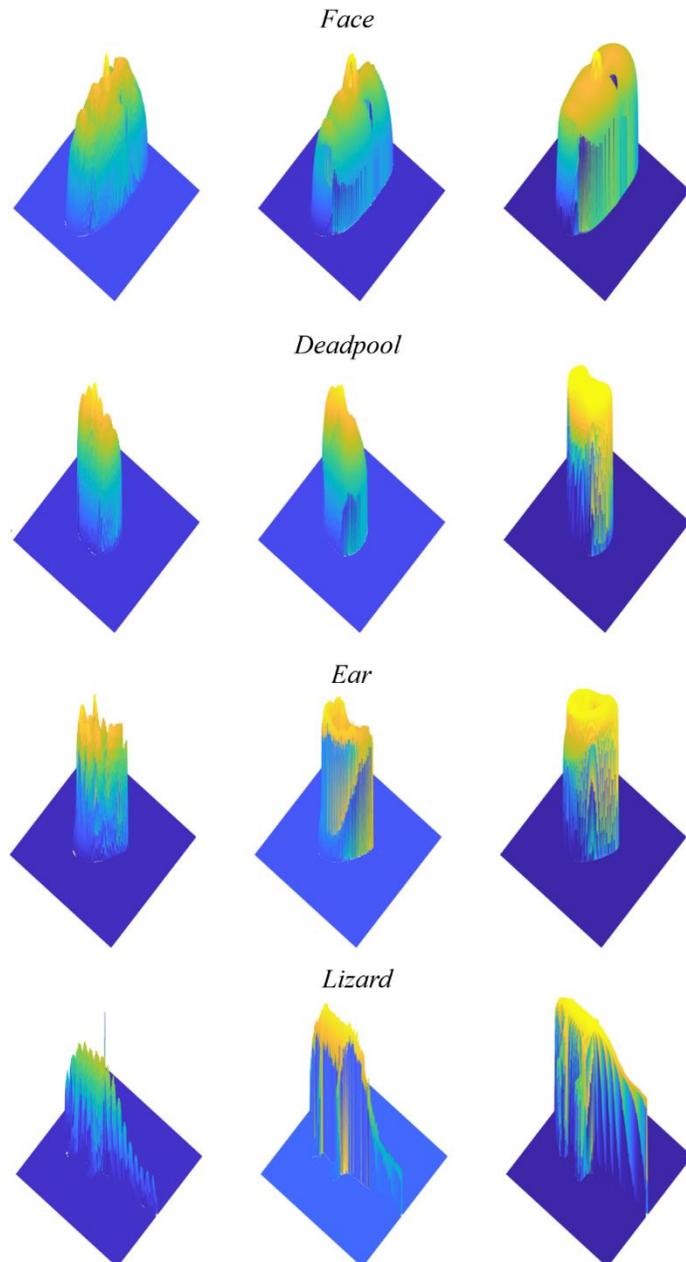


Figura 6.10 Comparación reconstrucción clásico vs TPU5 vs Ground Truth (Parte A).

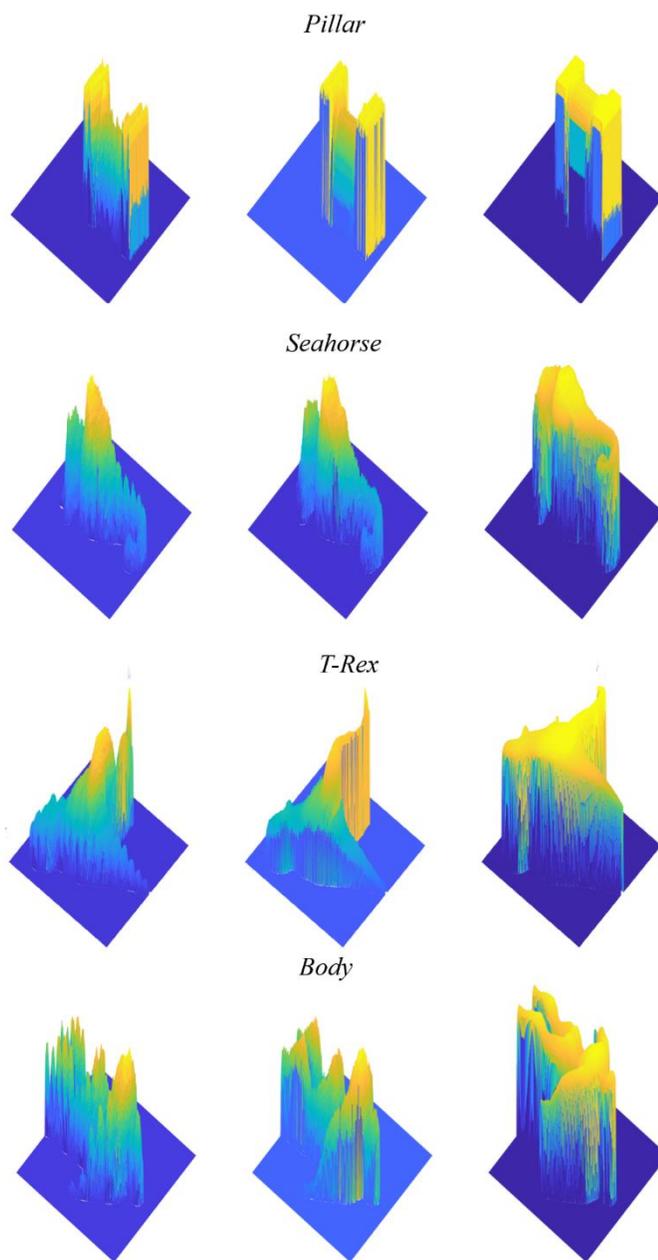


Figura 6.11 Comparación reconstrucción clásico vs TPU5 vs Ground Truth (Parte B).

6.3 C-GAN

Se analizó la calidad de las reconstrucciones obtenidas al emplear la C-GAN utilizando las mismas figuras mostradas en la sección 6.2 Tabla 6.2 como el conjunto de prueba de nuestra red, la cual fue entrenada con las 90 figuras restantes. Las Tablas 6.4 y 6.5 muestran el MSE y el PSNR de tres métodos de reconstrucción; método clásico, desenvolvimiento de fase Temporal de cinco iteraciones (TPU5), y desenvolvimiento de fase Temporal de cinco iteraciones post procesado con la C-GAN (TPU5 + C-GAN)

Tabla 6.4 MSE método clásico vs TPU vs TPU+C-GAN

Método	Face	Deadpool	Ear	Lizard	Pillar	Seahorse	T-rex	Body
Clásico	0.0963	0.2413	0.2140	0.2905	0.2044	0.2786	0.3386	0.2486
TPU 5	0.0472	0.1664	0.0992	0.1118	0.0682	0.1648	0.2495	0.1564
TPU5+CGAN	0.0018	0.0032	0.0061	0.0199	0.0030	0.0087	0.0049	0.0066

Tabla 6.5 PSNR método clásico vs TPU vs TPU+C-GAN

Método	Face	Deadpool	Ear	Lizard	Pillar	Seahorse	T-rex	Body
Clásico	10.163	6.174	6.696	5.368	6.894	5.550	4.703	6.0442
TPU5	13.258	7.789	10.034	9.514	11.276	7.830	6.029	8.3658
TPU5+CGAN	27.356	24.970	22.160	17.009	25.209	20.617	23.125	21.829

En la Figura 6.12 y 6.13 se muestra el MSE y el PSNR promedio de cada uno de los tres métodos de reconstrucción.

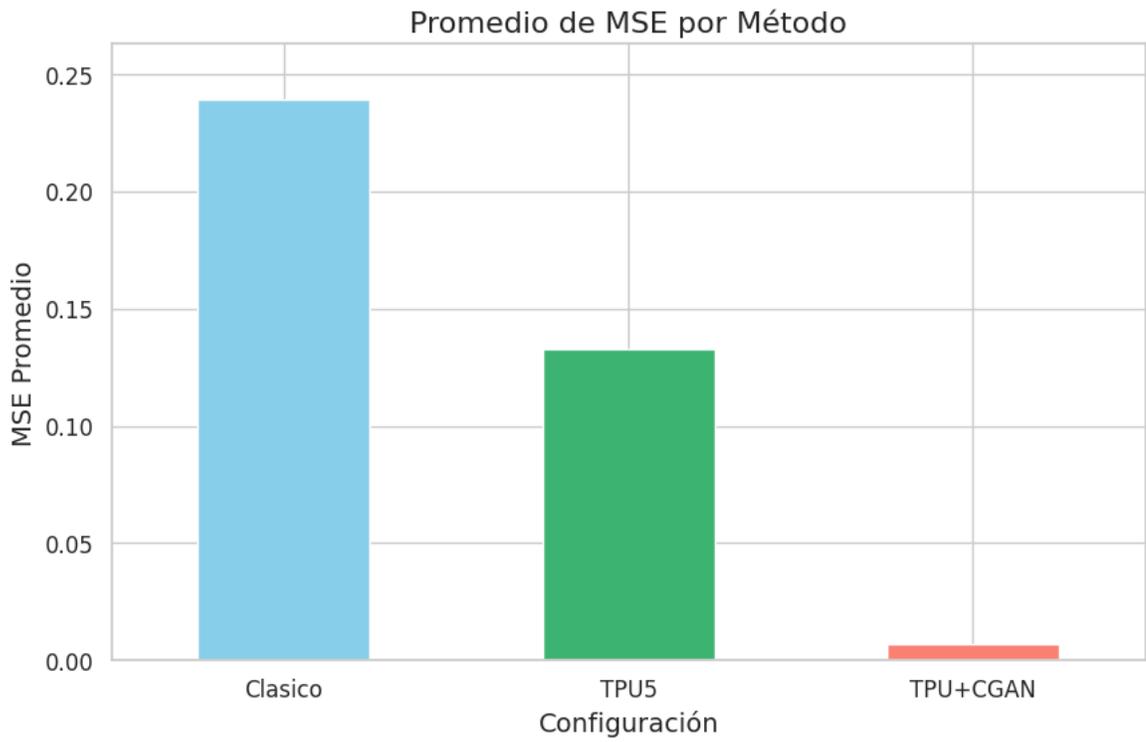


Figura 6.12 Valor promedio del MSE por método.

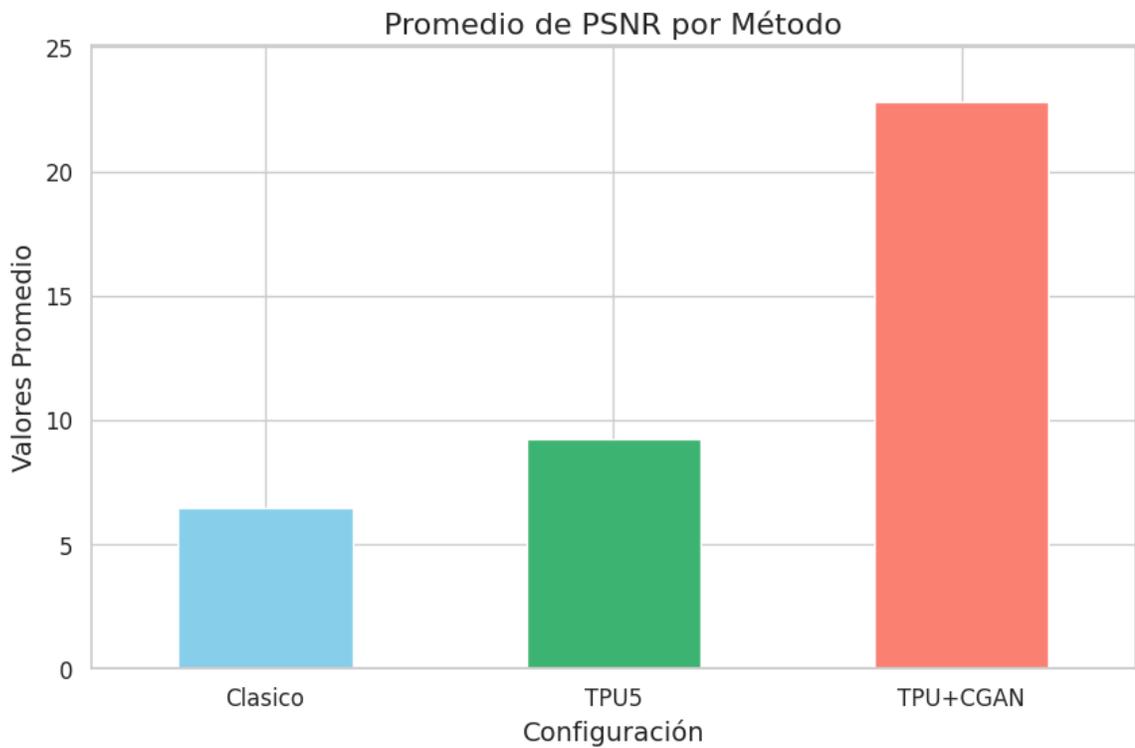


Figura 6.13 Valor promedio del PSNR por método

Se comparo el MSE obtenido cuando se usa TPU5 y TPU5 + C-GAN contra el método clásico, expresando esta disminución del error en porcentaje. En la Figura 6.14 se muestra la distribución obtenida para ambos casos.

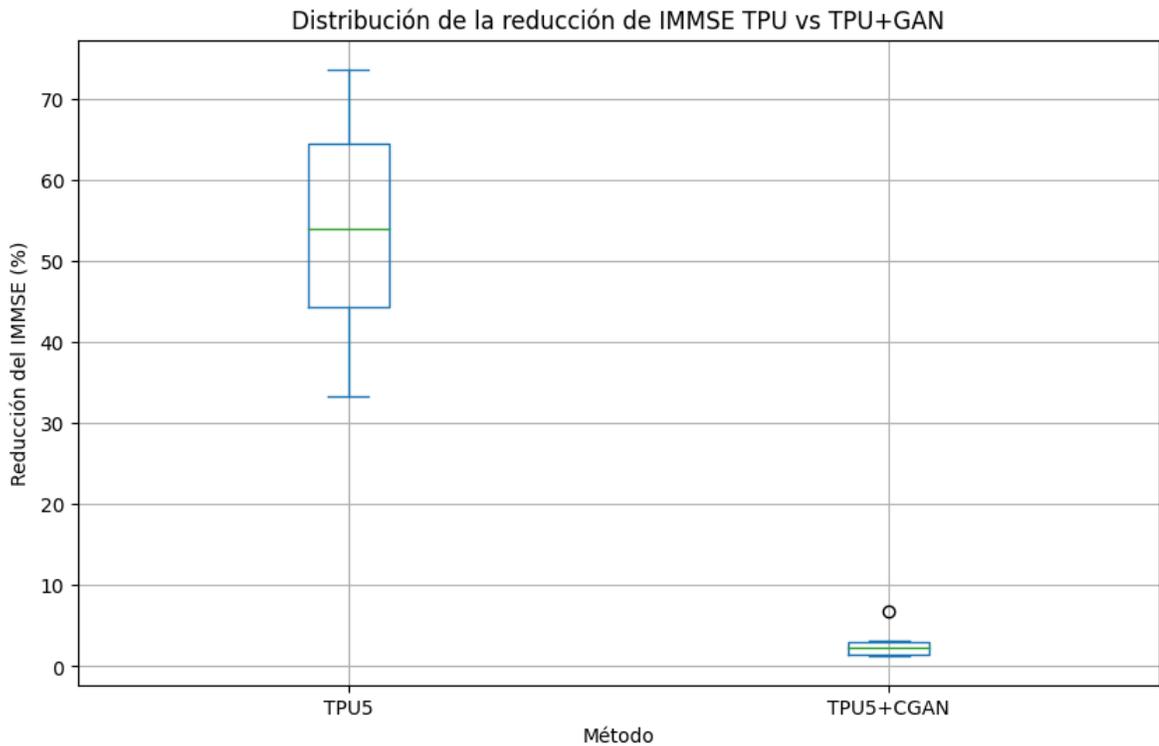


Figura 6.14 Distribución de la reducción de IMMSE, TPU VS TPU+GAN.

De forma análoga a la sección 6.2 en las Figuras 6.15 y 6.16 se muestran los ocho objetos referenciados en la Tabla 6.2. Pero esta vez la primera columna corresponde a las reconstrucciones obtenidas con TPU5, la segunda columna a las reconstrucciones obtenidas con cinco iteraciones de TPU +C-GAN y en la tercera columna se muestran los *Ground Truth*.

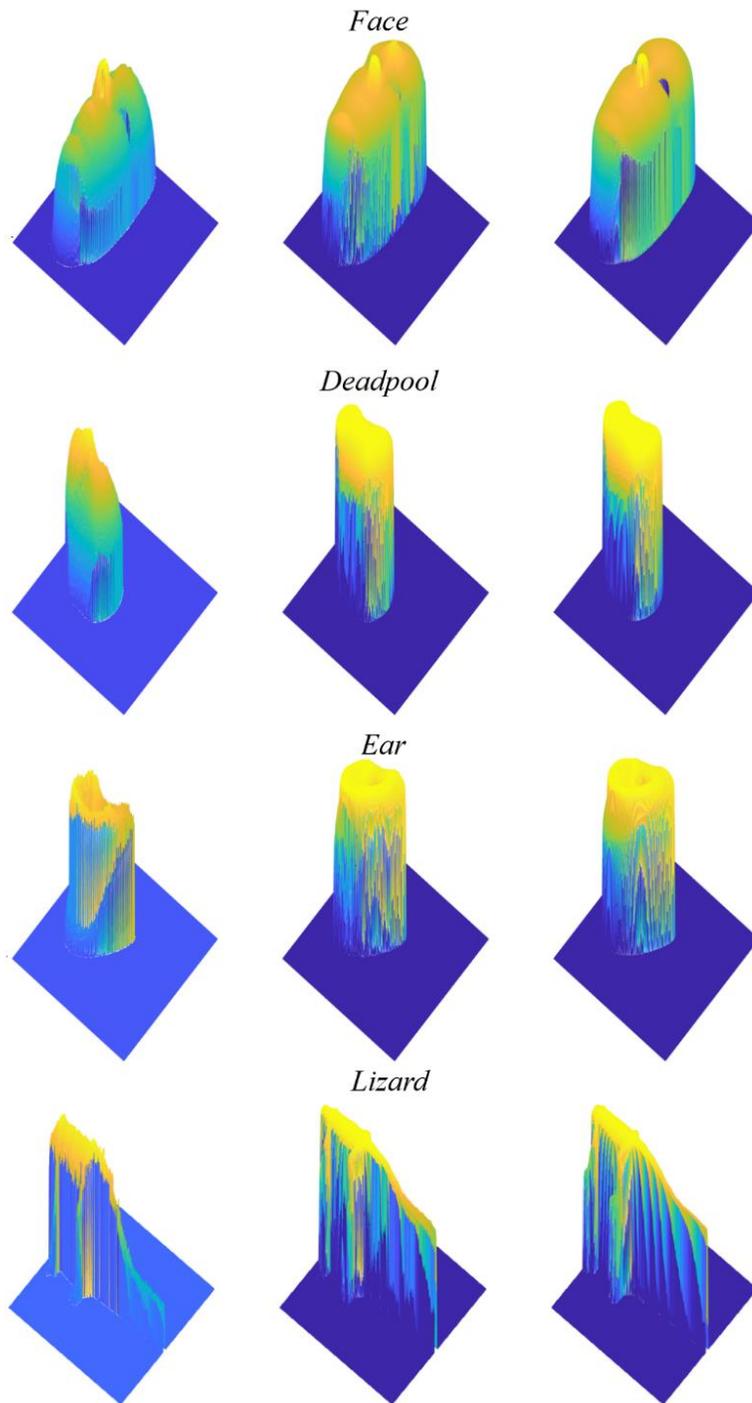


Figura 6.15 Comparación reconstrucción TPU5 vs TPU5+C-GAN vs Ground Truth (Parte A)

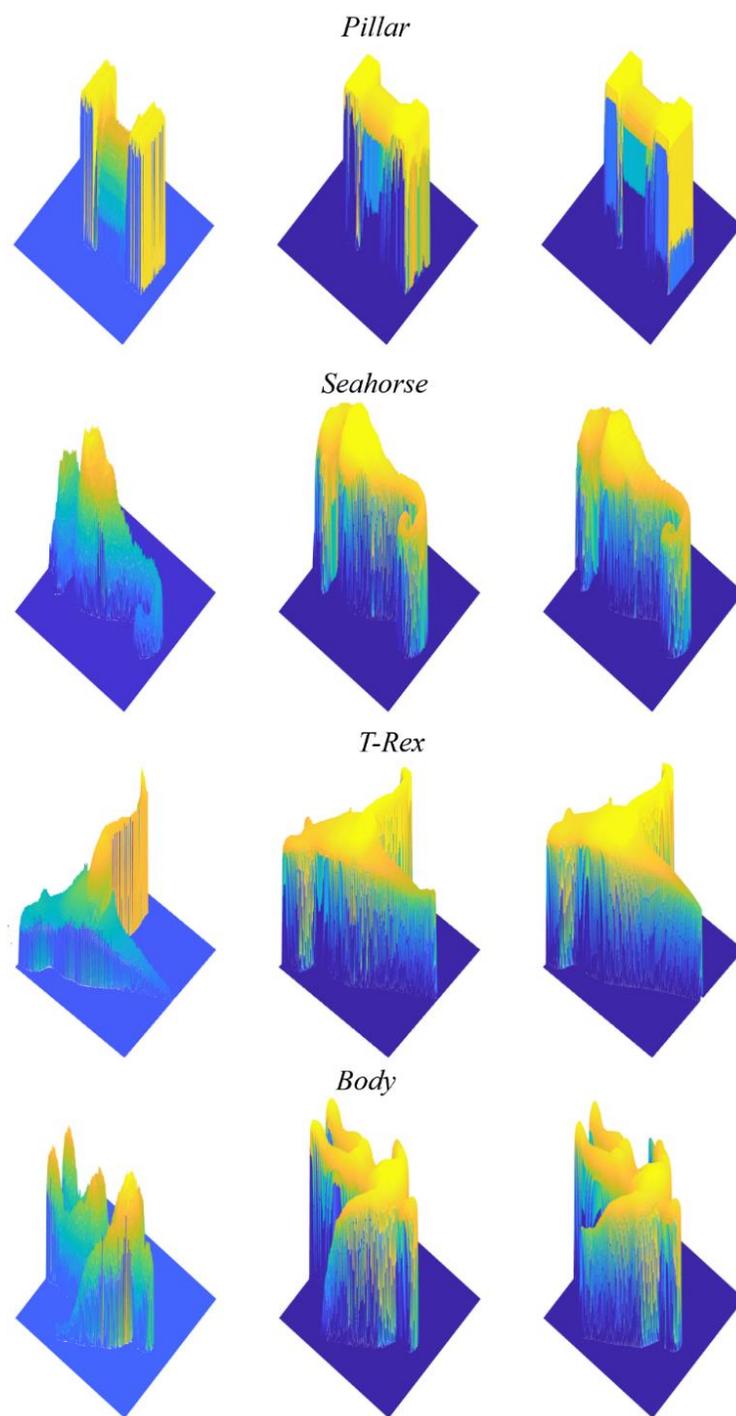


Figura 6.16 Comparación reconstrucción cTPU5 vs TPU5+C-GAN vs Ground Truth (Parte B)

En la Figura 6.17 se muestra un corte transversal de las figuras anteriormente presentadas. En ellas es posible observar las diferencias entre las alturas.

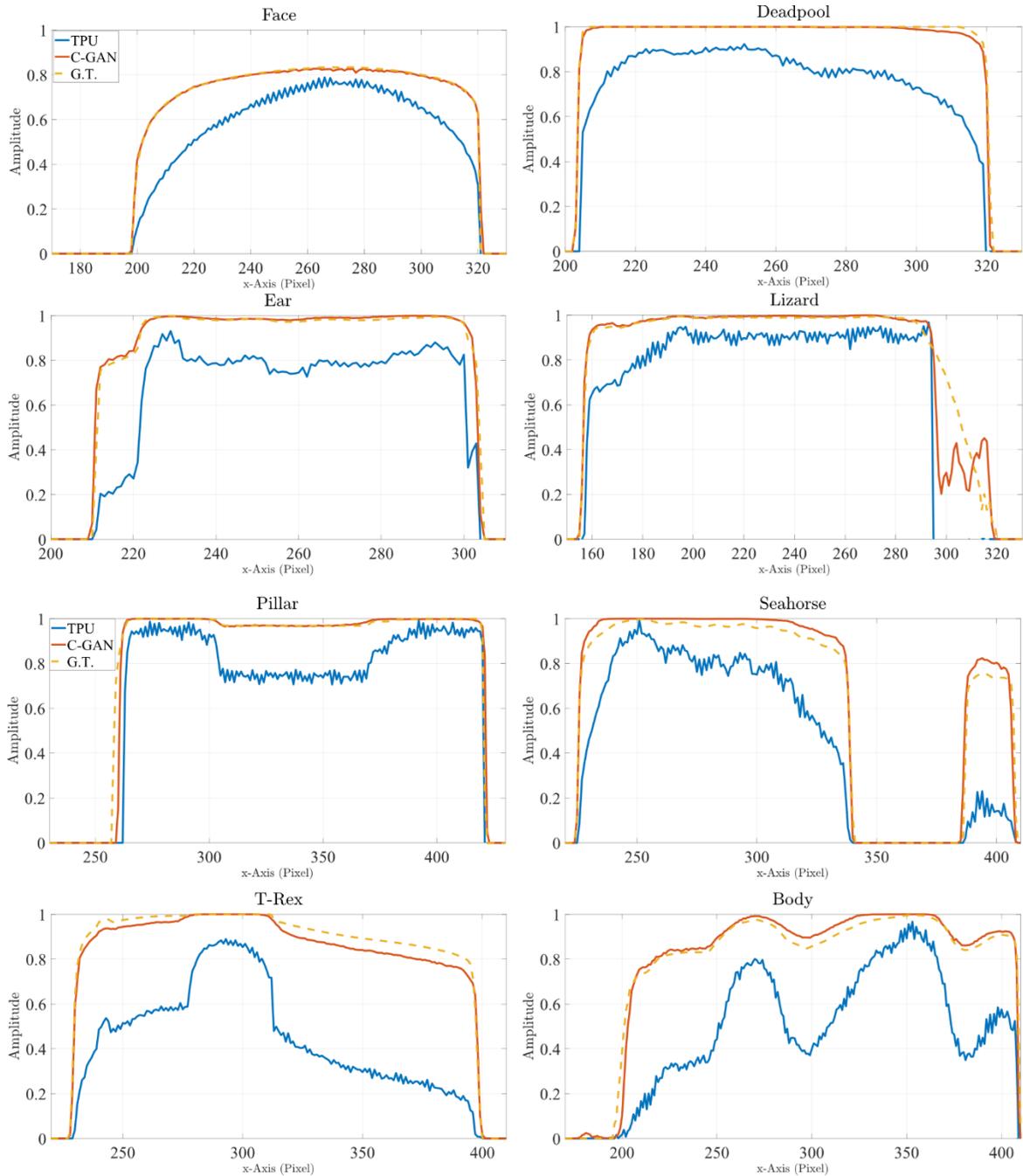


Figura 6.17 Corte Transversal de los objetos reconstruidos. Reconstrucción con TPU de 5 iteraciones (Azul), reconstrucción de TPU y C-GAN (Rojo) y Ground Truth (Amarillo).

En la Figura 6.18 se aplicó un zoom de los cortes mostrados en la Figura 6.17. Para destacar de forma detallada la diferencia entre las superficies obtenidas. El zoom aplicado abarca 20 píxeles de cada imagen a lo largo del eje x.

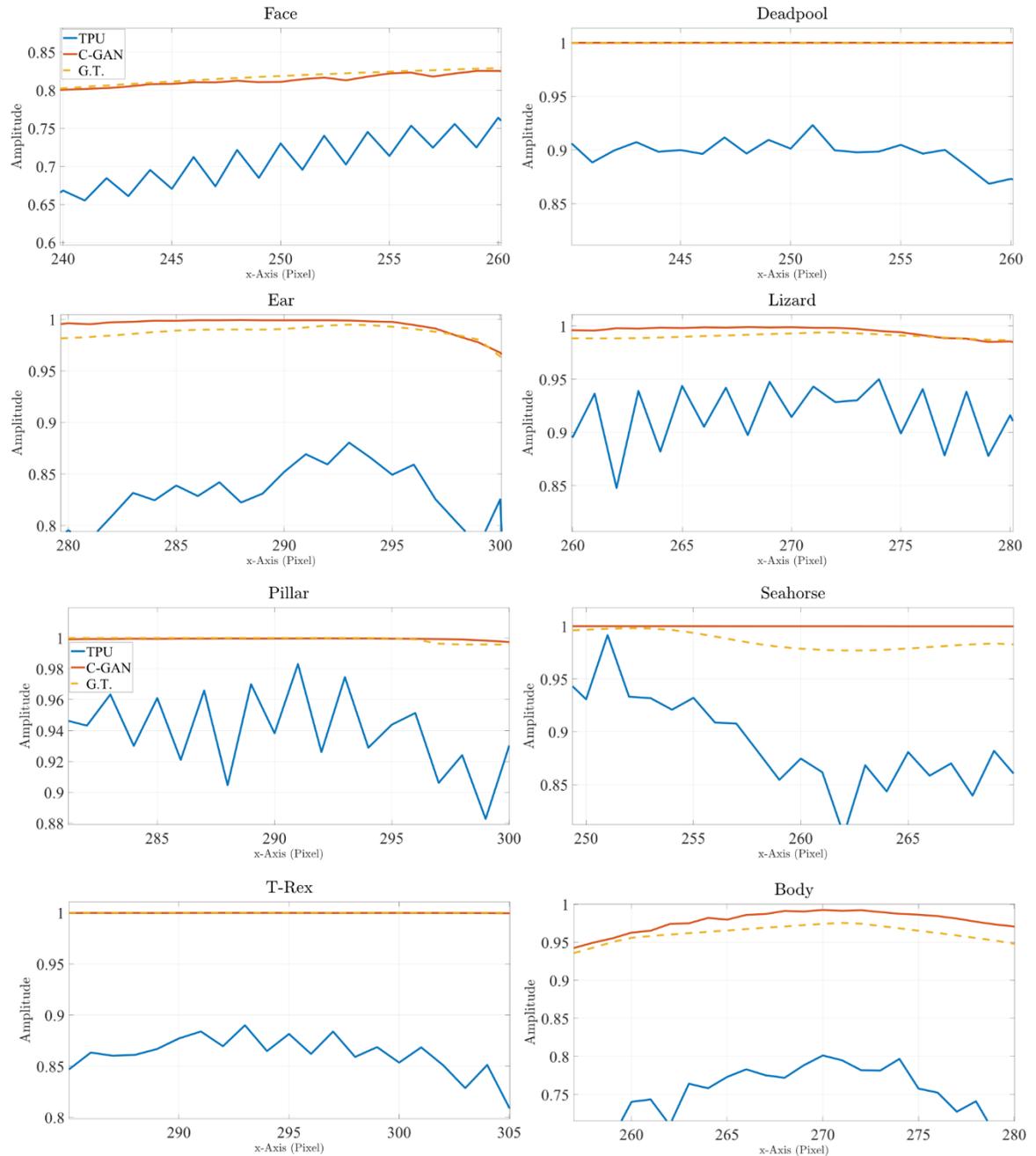


Figura 6.18 Corte Transversal de los objetos reconstruidos (Zoom).

6.4 Discusión de resultados

Emplear la técnica de Desarrollo de Fase Temporal presentó resultados positivos en la tarea de obtener reconstrucciones tridimensionales más fieles a los objetos originales. Se observó que es posible maximizar estos resultados repitiendo el proceso de forma iterativa. Esto se hace evidente al observar cómo disminuye el MSE entre mayor número de iteraciones de TPU son empleadas (Figuras 6.3 y 6.4), alcanzando una disminución de 58% del MSE cuando se emplean 5 iteraciones en comparación al método clásico.

Existe un fenómeno presente en los algoritmos de desarrollo de fase el cual modifica la altura de las superficies reconstruidas. En concreto este fenómeno disminuye la altura de las superficies de forma no lineal. Provocando que las regiones de mayor amplitud conserven su altura original mientras que entre menor la amplitud de ciertas regiones presentan una mayor reducción de su altura, es posible observar esto en las Figuras 6.8 y 6.9. En esa misma figura se puede observar que a pesar de la mejora en la reconstrucción este fenómeno sigue presente, esto debido a que el concepto del TPU es ir agregando detalles al mapa de fase base.

En las Figuras 6.10 y 6.11, gracias a la paleta de colores empleada se puede observar que a pesar de las evidentes mejoras entre las reconstrucciones de la columna 1 y las reconstrucciones de la columna 2, conservan amplitudes similares en todas las regiones.

En la sección 6.3, se observa que al emplear la C-GAN para procesar las reconstrucciones obtenidas después de 5 iteraciones del TPU, fue posible corregir el fenómeno de reducción de amplitudes. Al comparar las columnas 2 y 3 de las Figuras 6.15 y 6.16 se nota la similitud en las alturas a lo largo de toda la topología de la superficie medida. En donde las reconstrucciones procesadas con la C-GAN guardan mayor similitud con el *Ground Truth* que con las reconstrucciones antes de pasar por la C-GAN.

La corrección del fenómeno de reducción de amplitudes se traduce en la disminución del MSE Tabla 6.4, y una mejora en el PSNR Tabla 6.5. Para ponerlo en perspectiva el MSE obtenido al procesar las reconstrucciones obtenidas con TPU usando la C-GAN corresponde al 3 % del IMMSE del método clásico (Figura 6.14).

Además de la corrección de las alturas, el uso de la C-GAN mostro funcionar como una etapa de filtrado. Cómo se puede observar en las Figura 6.17, correspondientes a un acercamiento de un corte transversal, al procesar las reconstrucciones con la C-GAN las oscilaciones causadas por la proyección de franjas son disminuidas y en algunos casos corregidas.

7. Conclusiones

Al implementar una metodología híbrida combinando desarrollo de fase temporal y técnicas de aprendizaje profundo, en el método de perfilometría por cambio de fase se obtuvieron reconstrucciones mejoradas con respecto al método convencional.

Esencialmente la etapa de desarrollo de fase temporal aporta detalles a la reconstrucción evitando errores que son frecuentes cuando se usan patrones de 16 franjas o más. Mientras que etapa en la que se usa aprendizaje profundo (C-GAN) permite corregir las distorsiones de altura que pudieran existir, además de funcionar como filtrado eliminando las oscilaciones inherentes al método de perfilometría por cambio de fase.

La metodología híbrida presentada en la realización de este trabajo, tiene la ventaja de ser flexible, permitiéndonos sustituir el algoritmo de desarrollo de fase, así como variar el incremento en el número de franjas entre los patrones proyectados, con el fin de emplear menor número de patrones. Esta flexibilidad brinda la posibilidad de mejorar tanto la calidad de las reconstrucciones, como la eficiencia de los procesos aplicados.

Bibliografía

- [1] H. Nguyen y Z. Wang, «Accurate 3D Shape Reconstruction from Single Structured-Light Image via Fringe-to-Fringe Network,» *Photonics*, vol. 8, p. 459, October 2021.
- [2] Y. Li, J. Qian, S. Feng, Q. Chen y C. Zuo, «Composite fringe projection deep learning profilometry for single-shot absolute 3D shape measurement,» *Optics Express*, vol. 30, p. 3424, January 2022.
- [3] M. Nielsen, «Neural Networks and Deep Learning,» de *Machine Learning*, The MIT Press, 2021, p. 105–142.
- [4] J. Geng, «Structured-light 3D surface imaging: a tutorial,» *Advances in Optics and Photonics*, vol. 3, p. 128, March 2011.
- [5] Handbook of Optical Metrology: Principles and Applications, CRC Press, 2009.
- [6] B. Jähne, Handbook of Computer Vision and Applications, Three-Volume Set, Volume 1-3, Academic Press, 1999, p. 2541.
- [7] N. Tornero-Martinez, G. Trujillo-Schiaffino, M. Anguiano-Morales, P. G. Mendoza-Villegas, D. P. Salas-Peimbert y L. F. Corral-Martínez, «Color profilometry techniques: A review,» *Optica Pura y Aplicada*, vol. 51, p. 1–26, December 2018.
- [8] Fringe Pattern Analysis for Optical Metrology: Theory, Algorithms, and Applications, Wiley, 2014.
- [9] M. A. L. Cristerna, «Reconstrucción de objetos en 3-D por método de cambio de fase utilizando franjas RGB,» 2019.
- [10] J. L. Flores, J. A. Ferrari, G. G. Torales, R. Legarda-Saenz y A. Silva, «Color-fringe pattern profilometry using a generalized phase-shifting algorithm,» *Applied Optics*, vol. 54, p. 8827, October 2015.
- [11] M. Gdeisat, «One-Dimensional Phase Unwrapping Problem,» 2011.
- [12] T. R. Judge y P. J. Bryanston-Cross, «A review of phase unwrapping techniques in fringe analysis,» *Optics and Lasers in Engineering*, vol. 21, p. 199–239, January 1994.

- [13] S. Van der Jeught, J. A. M. Soons y J. J. J. Dirckx, «Real-time microscopic phase-shifting profilometry,» *Applied Optics*, vol. 54, p. 4953, May 2015.
- [14] P. Yao, S. Gai y F. Da, «Coding-Net: A multi-purpose neural network for Fringe Projection Profilometry,» *Optics Communications*, vol. 489, p. 126887, June 2021.
- [15] M. Costantini, «A novel phase unwrapping method based on network programming,» *IEEE Transactions on Geoscience and Remote Sensing*, vol. 36, p. 813–821, May 1998.
- [16] J. M. Bioucas-Dias y G. Valadao, «Phase Unwrapping via Graph Cuts,» *IEEE Transactions on Image Processing*, vol. 16, p. 698–709, March 2007.
- [17] M. Gdeisat, «Two-Dimensional Phase Unwrapping Problem,» June 2019.
- [18] J. M. Huntley y H. O. Saldner, «Shape measurement by temporal phase unwrapping: comparison of unwrapping algorithms,» *Measurement Science and Technology*, vol. 8, p. 986–992, September 1997.
- [19] H. Wu, Y. Cao, H. An, Y. Li, H. Li, C. Xu y N. Yang, «A novel phase-shifting profilometry to realize temporal phase unwrapping simultaneously with the least fringe patterns,» *Optics and Lasers in Engineering*, vol. 153, p. 107004, June 2022.
- [20] C. Zuo, S. Feng, L. Huang, T. Tao, W. Yin y Q. Chen, «Phase shifting algorithms for fringe projection profilometry: A review,» *Optics and Lasers in Engineering*, vol. 109, p. 23–59, October 2018.
- [21] H. O. Saldner y J. M. Huntley, «Temporal phase unwrapping: application to surface profiling of discontinuous objects,» *Applied Optics*, vol. 36, p. 2770, May 1997.
- [22] X. Guo, Y. Li, J. Qian, Y. Che, C. Zuo, Q. Chen, E. Y. Lam, H. Wang y S. Feng, «Unifying temporal phase unwrapping framework using deep learning,» *Optics Express*, vol. 31, p. 16659, May 2023.
- [23] X. Zhu, H. Zhao, M. Yuan, Z. Zhang, H. Wang y L. Song, «Phase unwrapping based on deep learning in light field fringe projection 3D measurement,» *Optoelectronics Letters*, vol. 19, p. 556–562, September 2023.
- [24] L. Sucar y G. Gómez, «Visión Computacional,» December 2022.
- [25] B. Jašhne, «Computer Vision and Applications: A Guide for Students and Practitioners,» *Journal of Electronic Imaging*, vol. 11, p. 115, January 2002.
- [26] R. Szeliski, *Computer Vision Algorithms and Applications*, Springer, 2010, p. 832.

- [27] S. Zhang, Handbook of 3D Machine Vision Optical Metrology and Imaging, Taylor & Francis Group, 2016.
- [28] M. Wu, N. Fan, G. Wu, S. Zhang y F. Liu, «An inverse error compensation method for color-fringe pattern profilometry,» *Journal of Optics*, vol. 22, p. 035705, February 2020.
- [29] C. Zuo, L. Huang, M. Zhang, Q. Chen y A. Asundi, «Temporal phase unwrapping algorithms for fringe projection profilometry: A comparative review,» *Optics and Lasers in Engineering*, vol. 85, p. 84–103, October 2016.
- [30] W. Yin, Q. Chen, S. Feng, T. Tao, L. Huang, M. Trusiak, A. Asundi y C. Zuo, «Temporal phase unwrapping using deep learning,» *Scientific Reports*, vol. 9, December 2019.
- [31] F. Chollet, Deep Learning with Python, Second Edition, Manning Publications Co. LLC, 2021, p. 400.
- [32] F. Berzal, Redes Neuronales and Deep Learning, Independently Published, 2018, p. 753.
- [33] A. Rodríguez-Sánchez, Redes neuronales: principios y aplicaciones, 2023.
- [34] N. S. Bajo y M. Á. Ballesteros, «Redes neuronales: concepto, aplicaciones y utilidad en medicina,» *Atención Primaria*, vol. 30, p. 119–120, 2002.
- [35] S. Marsland, Machine Learning: An Algorithmic Perspective, Chapman and Hall/CRC, 2014.
- [36] I. Goodfellow, Y. Bengio y A. Courville, Deep Learning, MIT Press, 2016, p. 800.
- [37] Y. LeCun, Y. Bengio y G. Hinton, «Deep learning,» *Nature*, vol. 521, p. 436–444, May 2015.
- [38] O. A. Montesinos López, A. Montesinos López y J. Crossa, «Convolutional Neural Networks,» de *Multivariate Statistical Machine Learning Methods for Genomic Prediction*, Springer International Publishing, 2022, p. 533–577.
- [39] A. Krizhevsky, I. Sutskever y G. E. Hinton, «ImageNet classification with deep convolutional neural networks,» *Communications of the ACM*, vol. 60, p. 84–90, May 2017.
- [40] K. Simonyan y A. Zisserman, *Very Deep Convolutional Networks for Large-Scale Image Recognition*, arXiv, 2014.
- [41] W. Weng y X. Zhu, «INet: Convolutional Networks for Biomedical Image Segmentation,» *IEEE Access*, vol. 9, p. 16591–16603, 2021.
- [42] Ö. Çiçek, A. Abdulkadir, S. S. Lienkamp, T. Brox y O. Ronneberger, «3D U-Net: Learning Dense Volumetric Segmentation from Sparse Annotation,» de *Lecture Notes in Computer Science*, Springer International Publishing, 2016, p. 424–432.

- [43] O. Ronneberger, P. Fischer y T. Brox, *U-Net: Convolutional Networks for Biomedical Image Segmentation*, arXiv, 2015.
- [44] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville y Y. Bengio, «Generative adversarial networks,» *Communications of the ACM*, vol. 63, p. 139–144, October 2020.
- [45] T. Karras, S. Laine y T. Aila, «A Style-Based Generator Architecture for Generative Adversarial Networks,» de *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.
- [46] Y. Li, B. Ku, S. Zhang, J.-K. Ahn y H. Ko, «Seismic Data Augmentation Based on Conditional Generative Adversarial Networks,» *Sensors*, vol. 20, p. 6850, November 2020.
- [47] P. Isola, J.-Y. Zhu, T. Zhou y A. A. Efros, «Image-to-Image Translation with Conditional Adversarial Networks,» de *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.
- [48] C. Ledig, L. Theis, F. Huszar, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang y W. Shi, «Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network,» de *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.
- [49] M. Mirza y S. Osindero, *Conditional Generative Adversarial Nets*, arXiv, 2014.
- [50] S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele y H. Lee, *Generative Adversarial Text to Image Synthesis*, arXiv, 2016.
- [51] J. Martinez-Carranza, K. Falaggis y T. Kozacki, «Fast and accurate phase-unwrapping algorithm based on the transport of intensity equation,» *Applied Optics*, vol. 56, p. 7079, August 2017.
- [52] B. Jahne, *Computer Vision and Applications*, Academic Press, 2000, p. 679.

Anexos

Anexo 1 MSE

	Classical	TPU1	TPU2	TPU3	TPU4	TPU5
0	0,26448	0,24616	0,22965	0,21457	0,20844	0,21018
1	0,29519	0,24503	0,23072	0,2258	0,21245	0,21453
2	0,29815	0,22016	0,22603	0,22119	0,20777	0,21846
3	0,096302	0,067291	0,056792	0,052263	0,050748	0,047227
4	0,083972	0,070187	0,059302	0,054535	0,051203	0,048094
5	0,096776	0,066989	0,056716	0,052369	0,051073	0,046817
6	0,086809	0,06898	0,057619	0,05418	0,050983	0,048089
7	0,14239	0,041224	0,088113	0,053586	0,042499	0,039391
8	0,11162	0,034605	0,12537	0,038498	0,031317	0,029193
9	0,60105	0,47784	0,38738	0,35448	0,33442	0,31723
10	0,12077	0,033329	0,086818	0,040281	0,037457	0,046697
11	0,2987	0,20975	0,15133	0,14023	0,13553	0,13112
12	0,20594	0,15818	0,17895	0,15218	0,14972	0,14796
13	0,28667	0,23532	0,25649	0,23228	0,23526	0,23372
14	0,25848	0,19542	0,23456	0,19763	0,17319	0,17
15	0,29567	0,21983	0,19796	0,20878	0,1822	0,17886
16	0,28301	0,26128	0,31915	0,30649	0,29908	0,2798
17	0,66246	0,6078	0,53084	0,4447	0,42342	0,40941
18	0,24868	0,2233	0,15427	0,16235	0,14565	0,15642
19	0,71818	0,65091	0,58452	0,55686	0,54379	0,53276
20	0,24131	0,15596	0,14964	0,14873	0,14838	0,14563
21	0,23542	0,15845	0,15303	0,15275	0,15163	0,15207
22	0,23005	0,15737	0,15361	0,15282	0,15083	0,15213
23	0,2424	0,15679	0,14958	0,15054	0,14717	0,14502
24	0,24968	0,17565	0,18135	0,17833	0,17121	0,16794
25	0,2413	0,17539	0,1723	0,17226	0,17093	0,16637
26	0,24869	0,17561	0,17972	0,17708	0,17139	0,16704
27	0,23886	0,18618	0,18275	0,17906	0,17643	0,17389
28	0,097196	0,079525	0,10188	0,083386	0,078016	0,078997
29	0,2043	0,12262	0,1019	0,11617	0,081013	0,083813
30	0,2215	0,13409	0,12005	0,099831	0,10497	0,099212
31	0,21399	0,14429	0,12437	0,10887	0,10334	0,10222
32	0,20828	0,13422	0,11486	0,10675	0,10749	0,09187
33	0,24474	0,20563	0,17551	0,17303	0,16765	0,16524
34	0,26506	0,22712	0,17273	0,21866	0,21198	0,17747

35	0,26056	0,22193	0,17442	0,22938	0,20046	0,17725
36	0,26181	0,22588	0,17119	0,2605	0,21124	0,17642
37	0,25775	0,224	0,17735	0,17544	0,16801	0,17007
38	0,22746	0,15585	0,1221	0,1043	0,11172	0,10803
39	0,20451	0,15268	0,12417	0,10325	0,10882	0,10888
40	0,22531	0,15514	0,1241	0,1018	0,1121	0,10875
41	0,213	0,14454	0,1205	0,097284	0,10652	0,10448
42	0,21897	0,167	0,12272	0,16429	0,1267	0,11444
43	0,21177	0,16209	0,11525	0,14396	0,12973	0,11312
44	0,21521	0,16673	0,12254	0,19711	0,13658	0,1155
45	0,2905	0,15782	0,11224	0,13378	0,12529	0,11185
46	0,25722	0,195	0,17871	0,17169	0,16124	0,16327
47	0,25304	0,19856	0,18215	0,1786	0,16149	0,16391
48	0,26474	0,19696	0,20237	0,20053	0,16634	0,16942
49	0,25246	0,18977	0,19791	0,19025	0,16064	0,16094
50	0,13613	0,071063	0,11285	0,057108	0,050059	0,048198
51	0,11758	0,06742	0,065973	0,05565	0,050262	0,048564
52	0,9648	1,0204	0,99263	0,98012	0,99052	0,99341
53	0,12314	0,066944	0,065765	0,056349	0,050419	0,049965
54	0,050451	0,037191	0,037558	0,035222	0,031903	0,032843
55	0,055545	0,041669	0,041199	0,036811	0,033467	0,032573
56	0,043888	0,033119	0,059865	0,046284	0,043997	0,039986
57	0,054269	0,043017	0,041675	0,037456	0,033902	0,032852
58	0,22605	0,15733	0,15155	0,14733	0,14635	0,13813
59	0,23735	0,15649	0,14702	0,14516	0,13884	0,13847
60	0,23327	0,1632	0,15344	0,1507	0,1474	0,14133
61	0,23221	0,16179	0,15301	0,15361	0,14973	0,14673
62	0,078814	0,053274	0,049797	0,053615	0,042343	0,03906
63	0,081149	0,052648	0,092636	0,067276	0,060847	0,050593
64	0,080153	0,052954	0,050437	0,040911	0,042259	0,040038
65	0,080415	0,052408	0,044511	0,04129	0,03961	0,038357
66	0,18549	0,1068	0,27058	0,12778	0,07884	0,074528
67	0,20445	0,1108	0,087853	0,12108	0,07163	0,068176
68	0,40243	0,11262	0,091892	0,15477	0,10157	0,091007
69	0,22851	0,1124	0,090415	0,11778	0,072939	0,068935
70	0,33855	0,26521	0,26947	0,25427	0,25089	0,24949
71	0,3164	0,26211	0,29433	0,251	0,25102	0,25168
72	0,3304	0,26468	0,2564	0,25395	0,25415	0,24804
73	0,31432	0,24175	0,24269	0,24483	0,23551	0,24264
74	0,19653	0,12896	0,077186	0,069167	0,076463	0,064873
75	0,18982	0,13396	0,08415	0,08825	0,080993	0,072904

76	0,2058	0,14139	0,09251	0,12737	0,084295	0,076753
77	0,20022	0,13199	0,090262	0,075101	0,07633	0,067873
78	0,27855	0,23962	0,17613	0,17899	0,16844	0,1648
79	0,2808	0,24064	0,17542	0,18427	0,17058	0,16199
80	0,27214	0,23999	0,1704	0,17292	0,16676	0,16152
81	0,26763	0,2402	0,16913	0,17819	0,16394	0,16409
82	0,15926	0,047565	0,063801	0,04794	0,045919	0,042781
83	0,31781	0,28462	0,24891	0,23575	0,23255	0,22553
84	0,49785	0,29442	0,27138	0,24184	0,2315	0,22867
85	0,29811	0,23165	0,25957	0,23131	0,21948	0,22583
86	0,32097	0,29496	0,24998	0,25935	0,24892	0,25444
87	0,30332	0,29335	0,24893	0,25696	0,24738	0,25203
88	0,31983	0,29698	0,24997	0,25659	0,24896	0,24984
89	0,32945	0,29068	0,24937	0,25806	0,24883	0,2514
90	0,066296	0,047512	0,032301	0,034403	0,031233	0,030679
91	0,066825	0,048357	0,033256	0,034016	0,032386	0,031096
92	0,072517	0,050065	0,033178	0,033625	0,031514	0,030584
93	0,073638	0,049577	0,032598	0,034077	0,031384	0,030431
94	0,28592	0,19999	0,19425	0,18814	0,18407	0,18249
95	0,2733	0,20162	0,19363	0,1925	0,19027	0,18826
96	0,28093	0,20111	0,19311	0,18982	0,18381	0,18228
97	0,28013	0,20157	0,19182	0,18838	0,18376	0,18087
98	0,090072	0,06872	0,08734	0,072439	0,068002	0,06325
99	0,09539	0,077908	0,086439	0,075395	0,06721	0,063383
100	0,14933	0,069788	0,04676	0,04566	0,044907	0,045004
101	0,24375	0,20868	0,14523	0,18545	0,16301	0,14561
102	0,23298	0,20193	0,13576	0,14701	0,14866	0,14077
103	0,25676	0,19719	0,13802	0,15	0,13969	0,14247
104	0,22331	0,20476	0,14062	0,14822	0,15257	0,14

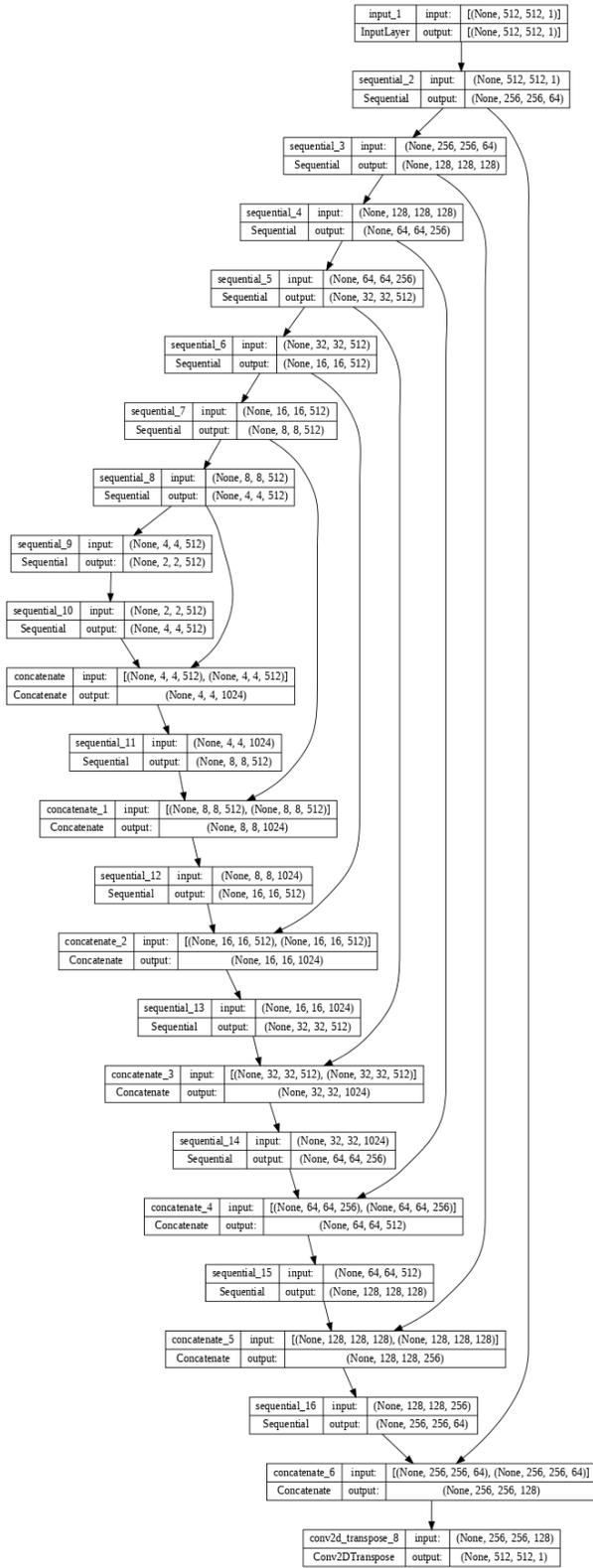
Anexo 2 PSNR

	Classical	TPU1	TPU2	TPU3	TPU4	TPU5
0	5,7761	6,0878	6,3894	6,6843	6,8101	6,774
1	5,2991	6,1077	6,3692	6,4628	6,7273	6,685
2	5,2556	6,5726	6,4583	6,5523	6,8242	6,6063
3	10,164	11,72	12,457	12,818	12,946	13,258
4	10,759	11,537	12,269	12,633	12,907	13,179
5	10,142	11,74	12,463	12,809	12,918	13,296
6	10,614	11,613	12,394	12,662	12,926	13,18
7	8,4652	13,848	10,55	12,709	13,716	14,046
8	9,5225	14,609	9,0181	14,146	15,042	15,347
9	2,2109	3,2072	4,1187	4,5041	4,7571	4,9862
10	9,1806	14,772	10,614	13,949	14,265	13,307
11	5,2477	6,7829	8,2009	8,5317	8,6796	8,8233
12	6,8626	8,0085	7,4727	8,1764	8,2471	8,2986
13	5,4262	6,2834	5,9092	6,3399	6,2845	6,313
14	5,8758	7,0904	6,2974	7,0414	7,6149	7,6956
15	5,292	6,5791	7,0341	6,803	7,3946	7,4748
16	5,482	5,8289	4,9601	5,1359	5,2421	5,5315
17	1,7884	2,1624	2,7503	3,5193	3,7323	3,8784
18	6,0436	6,5111	8,1173	7,8955	8,3667	8,0571
19	1,4376	1,8648	2,332	2,5425	2,6457	2,7347
20	6,1743	8,0699	8,2496	8,2759	8,2863	8,3675
21	6,2816	8,001	8,1524	8,1602	8,1922	8,1794
22	6,3817	8,0307	8,1359	8,1581	8,2152	8,1779
23	6,1547	8,0469	8,2514	8,2235	8,3219	8,3856
24	6,0261	7,5536	7,4149	7,4878	7,6648	7,7485
25	6,1744	7,5598	7,6371	7,6382	7,6718	7,7893
26	6,0434	7,5544	7,4541	7,5183	7,66	7,7717
27	6,2186	7,3007	7,3815	7,47	7,5342	7,5972
28	10,124	10,995	9,9192	10,789	11,078	11,024
29	6,8973	9,1144	9,9181	9,349	10,914	10,767
30	6,5463	8,726	9,2062	10,007	9,7893	10,034
31	6,6961	8,4075	9,0528	9,6309	9,8573	9,9044
32	6,8135	8,7218	9,3984	9,7163	9,6865	10,368
33	6,113	6,8692	7,557	7,6187	7,7561	7,8187
34	5,7666	6,4374	7,6264	6,6022	6,7371	7,5087
35	5,8409	6,5378	7,584	6,3945	6,9798	7,5142
36	5,8201	6,4613	7,6653	5,8419	6,7522	7,5344
37	5,8881	6,4976	7,5117	7,5588	7,7466	7,6938
38	6,4309	8,0729	9,133	9,8172	9,5187	9,6645

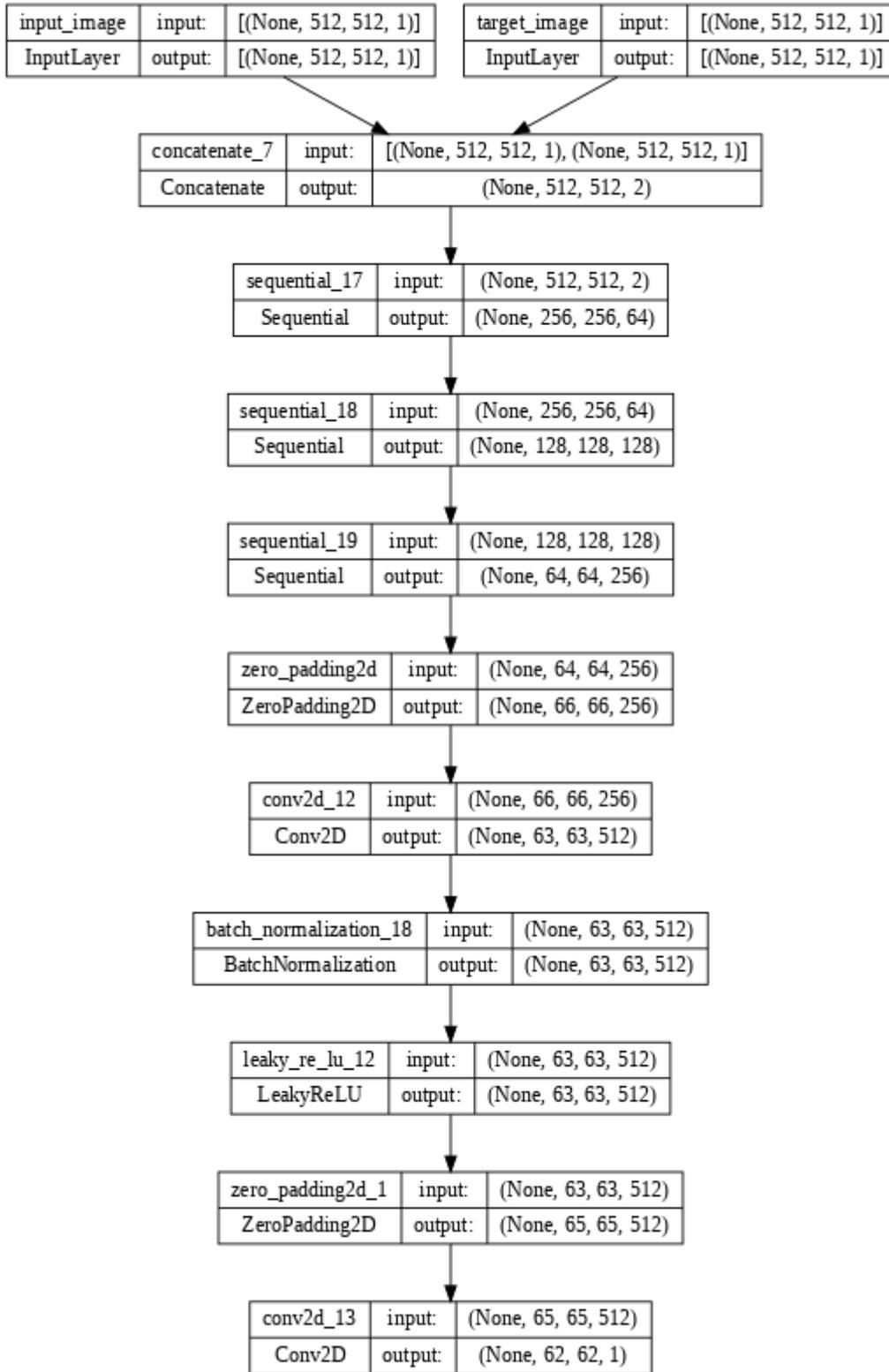
39	6,8928	8,1622	9,0598	9,861	9,633	9,6303
40	6,4722	8,0927	9,0622	9,9227	9,504	9,6356
41	6,7162	8,4003	9,1901	10,12	9,7258	9,8095
42	6,5961	7,7729	9,1108	7,8439	8,9721	9,4143
43	6,7413	7,9026	9,3835	8,4174	8,8697	9,4645
44	6,6714	7,7798	9,1171	7,0529	8,6463	9,3741
45	5,3685	8,0185	9,4986	8,736	9,021	9,5135
46	5,8969	7,0996	7,4784	7,6525	7,9252	7,8709
47	5,9682	7,0211	7,3957	7,4811	7,9186	7,8538
48	5,7718	7,0563	6,9386	6,9783	7,7901	7,7103
49	5,9782	7,2176	7,0353	7,2067	7,9414	7,9333
50	8,6604	11,484	9,4751	12,433	13,005	13,17
51	9,2965	11,712	11,806	12,545	12,988	13,137
52	0,15565	- 0,08786	0,032129	0,087208	0,041384	0,028705
53	9,0961	11,743	11,82	12,491	12,974	13,013
54	12,971	14,296	14,253	14,532	14,962	14,836
55	12,554	13,802	13,851	14,34	14,754	14,871
56	13,577	14,799	12,228	13,346	13,566	13,981
57	12,654	13,664	13,801	14,265	14,698	14,834
58	6,458	8,032	8,1946	8,317	8,346	8,5972
59	6,2461	8,0552	8,3264	8,3814	8,5747	8,5863
60	6,3213	7,8729	8,1406	8,2187	8,315	8,4977
61	6,3412	7,9105	8,1529	8,1357	8,247	8,3348
62	11,034	12,735	13,028	12,707	13,732	14,083
63	10,907	12,786	10,332	11,721	12,158	12,959
64	10,961	12,761	12,973	13,882	13,741	13,975
65	10,947	12,806	13,515	13,842	14,022	14,162
66	7,3169	9,7141	5,677	8,9354	11,033	11,277
67	6,8942	9,5545	10,562	9,1691	11,449	11,664
68	3,9531	9,4838	10,367	8,103	9,9322	10,409
69	6,411	9,4925	10,438	9,2894	11,37	11,616
70	4,7037	5,7641	5,6949	5,947	6,0051	6,0294
71	4,9976	5,8152	5,3117	6,0033	6,0029	5,9915
72	4,8096	5,7728	5,9108	5,9525	5,9491	6,0549
73	5,0262	6,1664	6,1495	6,1113	6,2799	6,1503
74	7,0656	8,8955	11,125	11,601	11,166	11,879
75	7,2167	8,7304	10,749	10,543	10,916	11,373
76	6,8656	8,4959	10,338	8,9493	10,742	11,149
77	6,9849	8,7947	10,445	11,244	11,173	11,683
78	5,5509	6,2048	7,5416	7,4718	7,7356	7,8304

79	5,516	6,1863	7,5592	7,3454	7,6806	7,905
80	5,6521	6,1981	7,6854	7,6215	7,7792	7,9179
81	5,7247	6,1942	7,7177	7,4912	7,8532	7,8492
82	7,9791	13,227	11,952	13,193	13,38	13,688
83	4,9784	5,4573	6,0396	6,2754	6,3348	6,4679
84	3,0291	5,3103	5,6642	6,1648	6,3545	6,4079
85	5,2563	6,3516	5,8575	6,3581	6,5861	6,4623
86	4,9354	5,3024	6,0209	5,8611	6,0393	5,9442
87	5,181	5,3261	6,0392	5,9014	6,0664	5,9854
88	4,9508	5,2728	6,0211	5,9076	6,0386	6,0233
89	4,8221	5,3659	6,0315	5,8827	6,0409	5,9964
90	11,785	13,232	14,908	14,634	15,054	15,132
91	11,751	13,155	14,781	14,683	14,896	15,073
92	11,396	13,005	14,792	14,733	15,015	15,145
93	11,329	13,047	14,868	14,675	15,033	15,167
94	5,4376	6,9899	7,1164	7,2551	7,3502	7,3875
95	5,6336	6,9547	7,1303	7,1557	7,2062	7,2524
96	5,514	6,9657	7,1419	7,2165	7,3563	7,3927
97	5,5263	6,9558	7,171	7,2496	7,3575	7,4263
98	10,454	11,629	10,588	11,4	11,675	11,989
99	10,205	11,084	10,633	11,227	11,726	11,98
100	8,2586	11,562	13,301	13,405	13,477	13,468
101	6,1305	6,8052	8,3793	7,3178	7,8778	8,368
102	6,3269	6,9481	8,6721	8,3265	8,2782	8,515
103	5,9047	7,0511	8,6005	8,2392	8,5485	8,4628
104	6,511	6,8876	8,5195	8,2909	8,1653	8,5388

Anexo 3 Arquitectura Generador



Anexo 4 Arquitectura del discriminador



Color Phase Shifting Profilometry using Multi-frequency Temporal Phase unwrapping

J. O. Castillo-Salazar¹, J. C. Pedraza-Ortega, O. A. Espinosa-Bernal, S. Tovar-Arriaga, and M.A. Aceves-Fernández

¹ Autonomous University of Queretaro, Faculty of Engineering, Cerro de las campanas
s/n, Queretaro, Mexico.

omar_0096@hotmail.com, caryoko@yahoo.com,
oespinosa07@alumnos.uaq.mx, saul.tovar@uaq.mx,
marco.aceves@gmail.com

Abstract. In this work, we analyzed the effect of implementing phase shifting profilometry with RGB images in an iterative way. This was done through a modification of the basic temporal phase unwrapping algorithm and the use of sine patterns with different frequencies for each projected pattern. It was concluded that by using iterative temporal phase unwrapping it is possible to overcome the limitation of the number of projected fringes, which translates into better reconstructions, being more faithful and with a higher degree of detail. Specifically, in this work, the ISSME was reduced by 43% compared to the conventional method.

Keywords: Phase-shifting-profilometry, phase-unwrapping, temporal-phase-unwrapping, RGB images.



ACCEPTANCE LETTER

Conferences and Workshops in Telematics and Computing
WITCOM 2024

Dear

Joaquin Omar Castillo Salazar, Jesus Carlos Pedraza Ortega, Osmar Antonio Espinosa Bernal, Saul Tovar Arriaga and Marco Antonio Aceves Fernandez

Paper ID: 1244

Title: Color Phase Shifting Profilometry using Multi-frequency Temporal Phase unwrapping

We are pleased to inform you that the above-referenced paper has been accepted for presentation at WITCOM 2024.

Congratulations on your acceptance!

Please carefully review the details in the acceptance email, as it contains crucial information regarding the inclusion of your paper in the Proceedings of WITCOM 2024, which will be published in the Springer CCIS Series (<https://www.springer.com/series/7899>).



UNIVERSIDAD AUTÓNOMA DE QUERÉTARO
FACULTAD DE LENGUAS Y LETRAS



A QUIEN CORRESPONDA:

La que suscribe, Directora de la Facultad de Lenguas y Letras, hace **C O N S T A R** que

CASTILLO SALAZAR JOAQUIN OMAR

Presentó el **Examen de Manejo de la Lengua** efectuado el día veintidós de mayo de dos mil veinticuatro, en el cual obtuvo la siguiente calificación:

8-

Se extiende la presente a petición de la parte interesada, para los fines escolares y legales que le convengan, en el Campus Aeropuerto de la Universidad Autónoma de Querétaro, el día veintidós de mayo de dos mil veinticuatro.



Atentamente,
"Enlazar Culturas por la Palabra"

DRA. ADELINA VELÁZQUEZ HERRERA

AVH/thb*CL*FLL-C.-3310



UNIVERSIDAD AUTÓNOMA DE QUERÉTARO
FACULTAD DE LENGUAS Y LETRAS



A QUIEN CORRESPONDA:

La que suscribe, Directora de la Facultad de Lenguas y Letras, hace **C O N S T A R** que

CASTILLO SALAZAR JOAQUIN OMAR

Presentó y acreditó el **Examen de Comprensión de Textos en Inglés** efectuado el día veintiocho de mayo de dos mil veinticuatro.

Se extiende la presente a petición de la parte interesada, para los fines escolares y legales que le convengan, en el Campus Aeropuerto de la Universidad Autónoma de Querétaro, el día veintiocho de mayo de dos mil veinticuatro.



Atentamente,
"Enlazar Culturas por la Palabra"

DRA. ADELINA VELÁZQUEZ HERRERA

AVH/thb*CL*FLL-C.-3336