

Reconstrucción de formas 3D mediante algoritmo de desplazamiento de fase estéreo usando proyección de franjas y una red neuronal convolucional

2023

Victor Manuel
Martínez Suárez



Universidad Autónoma de Querétaro

Facultad de Ingeniería

Reconstrucción de formas 3D mediante
algoritmo de desplazamiento de fase estéreo
usando proyección de franjas y una red
neuronal convolucional

Tesis

Que como parte de los requisitos para
obtener el grado de

Maestro en Ciencias en Inteligencia Artificial

Presenta:

Victor Manuel Martínez Suárez

Dirigido por:

Dr. Jesús Carlos Pedraza Ortega

Querétaro, Qro. a Junio 2023



Dirección General de Bibliotecas y Servicios Digitales
de Información



Reconstrucción de formas 3D mediante algoritmo de desplazamiento de fase estéreo usando proyección de franjas y una red neuronal convolucional

por

Víctor Manuel Martínez Suárez

se distribuye bajo una [Licencia Creative Commons Atribución-NoComercial-SinDerivadas 4.0 Internacional](#).

Clave RI: IGMAC-226262



Universidad Autónoma de Querétaro

Facultad de Ingeniería

Maestría en Ciencias en Inteligencia Artificial

Reconstrucción de formas 3D mediante algoritmo de desplazamiento de fase estéreo usando proyección de franjas y una red neuronal convolucional

TESIS

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

Presenta:

Ing. Víctor Manuel Martínez Suárez

Dirigido por:

Dr. Jesús Carlos Pedraza Ortega

SINODALES

Dr. Jesús Carlos Pedraza Ortega

Presidente

Firma

Dr. Juan Manuel Ramos Arreguín

Secretario

Firma

Dr. Marco Antonio Aceves Fernández

Vocal

Firma

Dr. Sebastian Salazar Colores

Suplente

Firma

Dr. Saúl Tovar Arriaga

Suplente

Firma

Dr. Manuel Toledano Ayala

Director de la Facultad

Dr. Juan Carlos Antonio Jáuregui Correa

Director de Investigación y Postgrado

Centro Universitario

Querétaro, QRO

México.

Junio 2023

© 2023 - Ing. Víctor Manuel Martínez Suárez

All rights reserved.

Esta tesis es dedicada a mi esposa, quien siempre me apoyó e impulsó para continuar mis estudios pese a las adversidades. A mi familia que, de no ser por ellos, no tendría las bases para haber comenzado mis estudios de posgrado. Y por último a mi director de tesis de quien parte la base del tema de investigación, agradeciendo su compromiso y guía.

Agradecimientos

Este trabajo fue apoyado en parte por el Consejo Nacional de Ciencia y Tecnología (CONACyT), México, en el Posgrado de la Facultad de Ingeniería por la Universidad Autónoma de Querétaro, bajo la beca CVU 1099400. Agradecimiento a los profesores del programa de la maestría, a quienes se les admira por su labor como docentes, resaltando el excelente trabajo del Dr. Jesus Carlos Pedraza Ortega en el constante seguimiento con sus tesis. A la M. en C. Maria Guadalupe Hernández Ramírez, que durante la investigación aportó consejos y su conocimiento en el tema de inteligencia artificial para la definición de la metodología y arquitecturas implementadas. Y por último mi familia, mi esposa, papá, mamá y mis hermanas que durante el propedéutico y primeros semestres fueron un sustento y soporte incondicional para siempre lograr más allá de mis objetivos.

Abstract

Fringe projection profilometry (FPP), among the non-contact 3D object reconstruction techniques, is one of the most used due to its easy implementation and cost reduction. Having a disadvantage because to efficiently recover the absolute phase it is necessary to obtain the reference pattern to perform a better unwrapping phase, which makes it complicated due to the ambiguities that exist in the wrapped phase. There are three methodologies used, such as spatial, temporal and geometric constraint unwrapping phase, the last being the most efficient and fastest without sacrificing measurement accuracy. Commonly this technique is called stereo phase unwrapping which can remove phase ambiguity without projecting any additional pattern, thus maximizing the efficiency of absolute phase recovery. Building on successes in implementing deep learning for phase analysis, we implement tools that make phase recovery, geometric constraints and phase unwrapping more organically efficient in a comprehensive framework. With extensive training data sets developed with a synthetic database, the neural network can gradually learn to transfer a pattern from high-frequency fringes to the physically meaningful and most likely absolute phase, rather than "step by step" as in conventional approaches. The experimental results obtained demonstrate that compared to traditional methods, we can more efficiently and stably develop phase imaging in a larger measurement volume with fewer camera views. Having a proposed approach that will lead us to high-speed, accurate, motion artifact-free 3D shape measurement for complicated objects from a single fringe pattern.

(Keywords: Convolutional Neural Networks, Fringe projection profilometry, Stereo phase unwrapping, depth constraints, feedback projection, 3-Step, 3D reconstruction.)

Resumen

La perfilometría de proyección de franjas (FPP), entre las técnicas de reconstrucción de objetos 3D sin contacto, es de las más utilizadas debido a su fácil implementación y reducción de costos. Teniendo una desventaja ya que para recuperar eficientemente la fase absoluta es necesario obtener el patrón de referencia para realizar una mejor fase de desenvolvimiento, lo que complica dadas las ambigüedades que existen en la fase envuelto. Se utilizan tres métodos como lo son la fase de desenvolvimiento de restricciones espaciales, temporales y de restricciones-geométricas, siendo la última la más eficiente y rápida sin sacrificar la precisión de la medición. Comúnmente, esta técnica se denomina desenvolvimiento de fase estéreo, que puede eliminar la ambigüedad de fase sin proyectar ningún patrón adicional, maximizando así la eficiencia de la recuperación de fase absoluta. En base a los éxitos en la implementación del aprendizaje profundo para el análisis de fase, implementamos herramientas que hacen que la recuperación de fase, las restricciones geométricas y el desenvolvimiento de fase sean más orgánicamente eficientes en un marco integral. Con extensos conjuntos de datos de entrenamiento desarrollados con una base de datos sintética, la red neuronal puede aprender gradualmente a transferir un patrón de franjas de alta frecuencia a la fase absoluta físicamente significativa y más probable, en lugar de "paso a paso" como en los enfoques convencionales. Los resultados obtenidos demuestran que, en comparación con los métodos tradicionales, podemos desarrollar imágenes de fase de manera más eficiente y estable en un volumen de medición más grande con menos vistas de cámara. Tener un enfoque propuesto que nos llevará a una medición de forma 3D sin artefactos de movimiento, precisa y de alta velocidad para objetos complicados a partir de un solo patrón de franjas.

Abreviaturas y siglas

SPU - Stereo Phase Unwrapping
TPU - Temporal Phase Unwrapping
CNN - Convolutional Neural Network
ADC - Adaptive Depth Constraint
GPU - Graphics Processing Unit
IA - Inteligencia Artificial
LSTM - Long Short Term Memory
ReLU - Rectified Linear Units
RGB - Red Green Blue
FPP - Fringe Projection Profilometry
PPF - Perfilometría de Proyección de Franjas
DL - Deep Learning
PSI - Phase Shifting Interferometry
PS - Phase Shifting
PSP - Phase Shifting Profilometry
CMOS - Complementary Metal Oxide Semiconductor
CCD - Charge Coupled Device
ESPI - Electronic Speckle Pattern Interferometry
DIC - Digital Image Correlation
LCD - Liquid Crystal Display
DLP - Digital Light Processing
ANN - Artificial Neural Network
ML - Machine Learning
DNN - Deep Neural Network
MAE - Mean Absolute Error
MSE - Mean Squared Error
PSNR - Peak Signal-to-Noise Ratio
SSIM - Structural Similarity Index
ZF - Zeiler and Fergus's
VGG - Visual Geometry Group
MLP - Multilayer Perceptron
FCN - Fully Convolutional Network
SRCNN - Super-Resolution Convolution Neural Network
CUDA - Compute Unified Device Architecture

CIF - Common Intermediate Format
VGA-SD - Video Graphics Array Standar Definition
ADAM - Adaptive Moment Estimation

Índice general

Agradecimientos	I
Abstract	III
Resumen	V
Abreviaturas y siglas	VII
Contenido	IX
Lista de Figuras	XI
Lista de Tablas	XV
1. Introducción	1
1.1. Descripción del problema	1
1.2. Justificación	2
1.3. Hipótesis	3
1.4. Objetivos	4
1.4.1. Objetivo general	4
1.4.2. Objetivos particulares	4
2. Antecedentes	5
2.1. Estado del arte	5
2.1.1. Antecedentes de la Interferometría	5
2.1.2. Problemas en los sistemas de metrología óptica	7
2.1.3. Estado del arte de los métodos de Desarrollo de Fase (PS)	7
2.1.4. Antecedentes de Redes Neuronales Convolucionales (CNN)	9
3. Metodología	17
3.1. Principio de la Perfilometría de proyección de franjas y desarrollo de fase estéreo	17
3.2. Métodos de metrología óptica	19
3.2.1. Proceso de reconstrucción 3D con imágenes sintéticas	19
3.2.2. Spatial phase unwrapping	21
3.2.3. Temporal phase unwrapping	25

3.2.4. Geometric phase unwrapping	26
3.3. Métodos y Materiales	28
3.4. Generación de ambiente emulado	29
3.4.1. Fase Envuelta con Imágenes sintéticas	32
3.5. Base de datos	33
3.6. Base de datos pública	34
3.7. Base de datos desarrollada	35
3.8. Diseño de arquitectura de CNN	36
3.8.1. Sobre las arquitecturas típica para tareas de clasificación de imagen	37
3.8.2. Estructura, diseño e implementación de una CNN	38
3.9. Metodología de trabajo implementando CNN para la reconstrucción de objetos 3D mediante el método SPU	41
3.9.1. Captura de los objetos y patrones de franjas	43
3.9.2. Pre-procesamiento de imagen	43
3.9.3. Mapas de fase	44
3.9.4. Calibración de las cámaras del Sistema Estéreo	45
3.9.5. Reconstrucción 3D	46
4. Resultados y Discusión	49
4.1. Resultados	49
4.1.1. Tiempos de renderizado	49
4.1.2. Evaluación cualitativa de los datos 3D generados	50
4.1.3. Entrenamiento, análisis y resultados	51
4.1.4. Evaluación del método tradicional y nuestra metodología	54
4.2. Importancia/Impacto	58
4.2.1. Impacto Social	58
4.2.2. Impacto Ambiental	58
4.2.3. Impacto económico	58
4.3. Publicaciones	59
4.4. Trabajo a futuro	59
5. Conclusión	61
References	69
.1. Registro de protocolo de investigación	70
.2. Dictamen de Comité de Ética Aplicada	71
.3. Examen de comprensión de textos en ingles	72
.4. Examen de manejo de la lengua	73
.5. Constancias WITCOM 2022	74
.6. Codigos CNN1 y CNN2	75

Índice de figuras

2.1. Proyección de franjas de una frecuencia portadora para los algoritmos de desplazamiento de fase [1].	5
2.2. Formación de imágenes en métodos típicos de metrología óptica [2].	8
2.3. Sistema para la adquisición de la imagen del método TPU. [4]	9
2.4. Representación matemática del perceptron de una capa simple [2].	11
2.5. Arquitectura de CNN para clasificación de imágenes [2].	13
2.6. Estructura para procesamiento de imagen SRCNN [2].	14
2.7. Estructura para procesamiento de imagen FCN [2].	15
2.8. Estructura para procesamiento de imagen U-net [2].	15
3.1. El Principio Básico de un sistema estéreo.	17
3.2. Background intensity de una proyección de franjas.	18
3.3. Representación 2D de la Fase envuelta (ϕ).	18
3.4. Representación gráfica del mapa de orden k del plano de referencia para la objeto de franjas sin objetos.	19
3.5. Algoritmos de desenvolvimiento de la Fase envuelta para proyección de franjas [2].	19
3.6. Diagrama de flujo del algoritmo de cambio de fase de 3 pasos.	20
3.7. Camino realizado para la detección de residuos. Los índices de dirección m y n son mostrados para indicar la dirección de los píxeles.	21
3.8. Imágenes de las franjas proyectadas en el plano de referencia(a) y el objeto(b) para 3-Step.	23
3.9. Residuos obtenidos con la ecuación 3.18, para la detección de los cambios de fase.	24
3.10. Resultados obtenidos implementando el método Spatial Phase Unwrapping en un objeto complejo.	24
3.11. Ilustración 2D de two-frequency temporal phase unwrapping ($\lambda_l = 1000$ píxeles, y $\lambda_h = 200$ píxeles) [3].	26
3.12. Diagrama de SPU y restricción de profundidad convencional (RPC) [95].	28
3.13. Ruta óptica de perfilometría de medición de fase en el entorno emulado.	30
3.14. El árbol principal de nodos de composición de este sistema Blender.	31
3.15. Modelos de TurboSquid.	31
3.16. Proyección de franjas renderizada en diferentes formatos de imagen. (a) Se genera con una resolución de 4k (resolución 3840×2160), (b) representa la resolución utilizada VGA-SD (resolución 640×480), y (c) de un formato CIF más pequeño (resolución 360×240 resolución).	32

3.17. Ejemplos de los tipos de captura según el escenario simulado, (a) Intensidad de fondo, (b) Plano completo considerando el fondo con el objeto, (c) Plano completo con fondo y sujeto considerando sombras, (d) Plano omitiendo el fondo proyectando solo el sujeto.	32
3.18. Fondo Intensidad que indica los puntos más altos con referencia al plano. (a) Representa el objeto más cercano a la cámara con una separación de $z = 6mm$ del plano, (b) una separación de $z = 4mm$, (c) una separación de $z = 2mm$	33
3.19. Fases envueltas obtenidas con imágenes renderizadas en (a) 4k (3840×2160píxeles), (b) VGA-SD (640×480píxeles) y (c) formatos CIF (380×210píxeles).	33
3.20. Ejemplos de imágenes que contiene la base de datos de Qian. (a) Las entradas de CNN1 (las imágenes marginales de 1000 escenas).(b) La salida de CNN1 (los términos numerador y denominador utilizados para calcular las fases envueltas). (c) Las entradas de CNN2 (los patrones marginales de 1000 escenas capturadas por dos cámaras, y la información del plano de referencia). (d) Las salidas de CNN2 (las órdenes de margen correspondientes a 1000 escenas).	34
3.21. Metodología de reconstrucción 3D por el método SPU implementando la base de datos Qian J. et. al 3.20.	35
3.22. Ejemplos de imagenes del ambiente emulado generado que contiene nuestra base de datos. (a) Escenas con PPF sin Background. (b) Escenas con PPF Originales. (c) Escenas con PPF sin Sombras. (d) Background Intensity. (e) Los términos numerador y denominador utilizados para calcular las fases envueltas. (f) Mascara del objeto de la escena con PPF.	36
3.23. Tres metodos tipicos en CNN para el sobremuestreo (upsampling).a) Unpooling, b) Transposed convolution, c) Sub pixel convolution. [2].	37
3.24. La arquitectura de la CNN1 propuesta que se construye modificando la U-Net. La entrada es un patrón de una sola franja [2].	39
3.25. Código de la función de los bloques para Downsample y Upsample.	40
3.26. Código usado para el entrenamiento de la red neuronal.	41
3.27. El comportamiento de aprendizaje, de los datos de perdida y la curva de distribución de perdida después del entrenamiento de la a)CNN1 y b)CNN2	42
3.28. Metodología de reconstrucción 3D por el método SPU implementando nuestra base de datos generada con el sistema emulado en Blender 3.22.	42
3.29. Capturas de la cámara 1 y 2 del sistema emulado en Blender.	43
3.30. Implementación de CNN1 para la obtención de mapa de fase D y M, a partir de una sola captura de cada cámara.	44
3.31. Numerador y Denominador de la salida de la CNN para la captura de la cámara Izquierda (L o 1) para calcular la Fase envuelta.	44
3.32. Numerador y Denominador de la salida de la CNN para la captura de la cámara Izquierda (L o 1) para calcular la Fase envuelta.	45
3.33. Numerador y Denominador de la salida de la CNN para la captura de la cámara Izquierda (L o 1) para calcular la Fase envuelta.	46
3.34. Visualización de la fase absoluta en 3D, en donde se encuentra en un plano inclinado debido al desenvolvimiento de la fase envuelta.	46
3.35. Reconstrucción 3D del mapa absoluto reacondicionado para la visualización lineal de los datos 3D obtenidos al desenvolver la fase envuelta.	47

4.1.	Triangulación de puntos característicos sin ángulo entre la cámara y el proyector. (a) Representa la captura de las franjas proyectadas, (b) la fase envuelta, (c) la reconstrucción de la fase absoluta [4].	50
4.2.	Triangulación de puntos característicos con ángulo entre cámara y proyector y gran proyección de sombras.(a)Representa la captura de las franjas proyectadas, (b) la fase envuelta, (c) la reconstrucción de la fase absoluta [4].	51
4.3.	Resultados de la evaluación de la generación de Ground-Truth usando el emulador con un objeto 3D simple. Realizando la reconstrucción del Objeto (a), para obtener los Datos 3D (b) y el Ground-Truth (c), con un corte en su centro (d) y obteniendo la diferencia de error (e) entre ambas reconstrucciones [4].	52
4.4.	Resultados de la evaluación de la generación de Ground-Truth usando el emulador con un objeto 3D complejo. Realizando la reconstrucción del Objeto (a), para obtener los Datos 3D (b) y el Ground-Truth (c), con un corte en la punta de la nariz del león (d) y obteniendo la diferencia de error (e) entre las dos reconstrucciones [4].	52
4.5.	Comparación de Mapa de orden generado con 12-Step para referencia y Mapa de orden predicho por nuestro modelo de la CNN2.	55
1.	Código utilizado para la construcción de la red neuronal (CNN1).	76
2.	Código utilizado para la construcción de la red neuronal (CNN1).	77

Índice de Tablas

2.1.	Inicios de desarrollo de técnicas PSI.	6
2.2.	Trabajos en los años 80's de técnicas PSI.	6
2.3.	Trabajos en los años 90's de técnicas PSI.	6
2.4.	Antecedentes de sistemas estéreo.	10
4.1.	Tiempos (segundos) promedios de renderizado de acuerdo a cada método y el tipo de objeto, a partir de nuestra base de datos de 125 objetos (No. de imágenes).	50
4.2.	Implementación preliminares para evaluación de resultados del modelo CNN2 (Entrenamiento de 20 épocas).	53
4.3.	Métricas e Hiper-parámetros de los resultados finales de los modelos CNN1 y CNN2.	54
4.4.	Métrica de evaluación de el error promedio en la reconstrucción de cada método y tipo de objeto simple.	56
4.5.	Métrica de evaluación de el error promedio en la reconstrucción de cada método y tipo de objeto complejo.	57
4.6.	Tiempos (segundos) de reconstrucción de cada método y tipo de objeto.	57

Introducción

1.1. Descripción del problema

Las técnicas ópticas de medición de formas tridimensionales (3D) sin contacto se han aplicado ampliamente en muchos aspectos, como es en la fabricación inteligente, la ingeniería inversa y la digitalización del patrimonio [5].

La FPP es una de las técnicas de medición óptica 3D sin contacto más populares debido a sus ventajas de alta precisión de medición, instalación de hardware simple e implementación flexible obteniendo gran calidad en la reconstrucción de objetos complejos [5], y adoptada con mayor frecuencia actualmente [6]. La demanda por los sistemas de medición rápida de formas 3D de alta resolución y sin contacto ha sido propulsada por los sectores médicos, industriales, de entretenimiento, hasta los fabricantes y los grupos de investigación para diseñar una gran variedad de técnicas de perfilometría óptica [7]. Para esta técnica, recuperar eficientemente la fase absoluta sigue siendo un gran desafío.

Las tecnologías de SPU basadas en restricciones geométricas pueden eliminar la ambigüedad de fase sin proyectar ningún patrón adicional [5], lo que maximiza la eficiencia de la recuperación de la fase absoluta a comparación de otros métodos tradicionales.

A partir de lo mencionado anteriormente, se sabe que aunque SPU es el más adecuado para la medición 3D en escenas de alta velocidad, todavía tiene algunos defectos, como un volumen de medición limitado, incapacidad para lograr un desenvolvimiento de fase robusto de imágenes de franjas de densas, pérdida de la eficiencia de la medición debido a la dependencia de los métodos de adquisición de fase de múltiples cuadros, la complejidad de la implementación del algoritmo, etc. [5].

Basándonos en los éxitos del aprendizaje profundo en FPP y los avances en las aplicaciones de restricción geométrica [8], sobre la base de los trabajos anteriores, Feng logro mejorar el método, al aplicarlo en conjunto en el desenvolvimiento de fase estéreo e incorporar restricciones geométricas en la red neuronal.

En su trabajo [5], las restricciones geométricas están implícitas en la red neuronal en lugar de utilizar directamente los parámetros de calibración, lo que simplifica todo el proceso de desenvolvimiento de fase y evita el complejo ajuste de varios parámetros. Con un amplio entrenamiento de datos, la red puede aprender a obtener la fase absoluta "físicamente significativa" de la proyección de un solo cuadro sin el cálculo convencional "paso a paso". En comparación con la SPU tradicional, el enfoque propuesto desenvuelve de manera robusta la fase de la frecuencia más alta con menos perspectivas en un rango más amplio [5].

Con base en lo ya descrito los métodos tradicionales son incapaces de obtener la fase completa sin la pérdida de datos debido a la ambigüedad de las fases, al ángulo de proyección y la sombra generada por la luz del proyector. Se ha detectado los problemas principales con lo que se pretende corregir aplicando

el método SPU y obtener la reconstrucción 3D en un solo disparo aplicando algoritmos complementarios y metodologías de aprendizaje profundo que mejoren la técnica convencional.

Una de las cosas que tienen en común la mayoría de los métodos mencionados anteriormente es la proyección de un haz de luz y la captura de los patrones generados en el objeto por el reflejo de la luz empleando un sensor o una cámara. La mayoría de las veces, conseguir el equipo indicado para realizar la medida óptica con alguna de estas técnicas no es sencillo y no es hasta la práctica que nos damos cuenta de diferentes factores que no se tuvieron en cuenta al principio, lo que obliga a cambiar algún equipo con otras características [9]. También se da el caso que montar el sistema básico como el que se muestra en la Figura 2.3, no resulta tan fácil ni rápido, donde se requiere hacer ajustes entre la cámara y el proyector, variar la intensidad de la onda generada para el patrón proyectado, cambiar la frecuencia de la onda generada entre otros. Disponer de un entorno que emule el sistema físico es útil tanto para generar grandes bases de datos en un tiempo preciso y asequible como para realizar pruebas de ajuste en un entorno controlado de forma práctica y sencilla.

A lo largo del trabajo se mostrarán los resultados obtenidos del entorno que emula un sistema PSP para un 3-Step, en el que es posible cambiar sus características para adaptarse a diferentes métodos como la corrección digital de imágenes o técnicas derivadas de la Perfilometría de proyección de franjas como desenvolvimiento de fase estéreo (SPU). Explicando así la metodología para poder generar dicho entorno que se pueda adaptar a aplicaciones concretas.

1.2. Justificación

Los cambios más significativos en los diferentes tipos de instrumentación durante los últimos 40 años han sido la integración de tecnologías como computadoras en la medición, y en la última década se implementaron más técnicas debido al poder computacional [10, 11].

Es así, que los sistemas y algoritmos complejos hacen que esta estrategia sea difícil de implementar y obtener la fase absoluta correcta para la primera medición, además que es difícil tratar con objetos que repentinamente aparecen fuera del rango de restricción de profundidad en escenas dinámica [6].

En trabajos recientes, se ha realizado la adquisición de fase con alta precisión a partir de un único patrón marginal mediante el uso de técnicas aprendizaje profundo. Sin embargo, muchos de estos trabajos solo obtienen la fase envuelta con más de un disparo, es decir, la adquisición de más de una imagen. Para realizar la medición 3D, se requiere desenvolver la fase, que es una de las operaciones en FPP que más afecta el resultado (eficiencia) de la medición [5].

Las técnicas basadas en perfilometría sólo requieren el uso de una cámara digital, o dos en el caso de las técnicas de visión estéreo (en algunas aplicaciones más de dos), y generalmente son configuraciones de bajo costo. Sin embargo, la precisión limitada de la medición en profundidad y los requisitos de procesamiento de señales digitales intensivos en computación reducen la facilidad de uso en configuraciones perfilométricas en tiempo real [7].

Es así que los métodos de desenvolvimiento de fase más utilizados son los algoritmos de TPU, que recuperan la fase absoluta con la ayuda de patrones de código de escala de grises o franjas de múltiples longitudes de onda [10]. Sin embargo, el requisito de patrones de frecuencias diferentes adicionales disminuye la eficiencia de la medición.

Para realizar la medición 3D, como mencionamos anteriormente, por lo general se requiere desenvolver la fase siendo una de las operaciones en FPP que mayor impacto tienen. El método del algoritmo de TPU, el cual determina el orden de la franja de píxeles con la distribución de fase envuelta única en el dominio del tiempo. Sin embargo, TPU generalmente necesita proyectar patrones codificados en escala de grises

adicionales o franjas de múltiples longitudes de onda, lo que disminuye en gran medida la eficiencia del desenvolvimiento de fase. El método TPU puede realizar el desenvolvimiento de fase solo mediante un mapa de fase envuelto de acuerdo con la información de fase de la vecindad. Pero fallará al medir objetos aislados [6].

Existen variantes del TPU, es así como el SPU mediante la integración de otra cámara la cual esta orientada en diferente posición a fin de detectar puntos ciegos que complementes la imagen principal mejorar los resultados obtenidos. Sin embargo, el método SPU convencional es generalmente insuficiente para desenvolver de manera robusta la fase de imágenes de franjas densas, mientras que aumentar la frecuencia de las franjas es crucial para la precisión de la medición [12].

La estrategia de restricción de profundidad (ADC, el algoritmo de restricción de profundidad más avanzado y complejo), que es un proceso de predecir el volumen de medición del sistema y excluir los candidatos 3D fuera de este rango de restricción. Es un método popular para mejorar la estabilidad de la SPU. Sin embargo, la estrategia de restricción de profundidad convencional solo puede desenvolver la fase en un rango de profundidad estrecho, y cómo establecer un rango de restricción de profundidad adecuado también es difícil. Aun así la estrategia de ADC puede ampliar el volumen de medición y seleccionar automáticamente el rango de restricción de profundidad [5].

Para mejorar la estabilidad de SPU, los métodos adoptados comúnmente son aumentar el número de vistas o aplicar la estrategia de restricción de profundidad [13].

- El primero, con mayores costos de hardware,, ya que requiere aún más datos 2D de la cámara 2, en la tercera o incluso en la cuarta cámara para la debido a que verifica la similitud de fase y excluye más datos 2D incorrectos.
- El segundo, a costa de una mayor complejidad algorítmica, puede eliminar por adelantado algunos datos 3D incorrectos que están fuera del rango de restricción de profundidad.

Sin embargo, el algoritmo de restricción de profundidad convencional solo es efectivo en un volumen estrecho. Generalmente, la SPU con al menos tres cámaras asistidas con ADC puede lograr un desenvolvimiento de fase robusto bajo la premisa de que se obtiene la fase absoluta correcta para la primera medición [13].

Las técnicas de Deep Learning (DL) nos han permitido resolver problemas desde una perspectiva de procesamiento digital, dejando de lado las limitaciones de hardware para los métodos ópticos.

Este enfoque propuesto con estas técnicas representa un importante paso adelante en la medición de forma 3D absoluta, libre de artefactos de movimiento, precisa y de alta velocidad para objetos marginales complicados de un solo patrón, por lo que es esencial tener acceso a nuestros datos, que se ajustan más a las aplicaciones que estamos desarrollando [5].

1.3. Hipótesis

La implementación del método de desenvolvimiento de fase estéreo mejora la eficiencia de una reconstrucción 3D aplicando restricciones geométricas implícitas, usando análisis de franjas y redes neuronales convolucionales.

1.4. Objetivos

1.4.1. Objetivo general

Diseñar e implementar el método de desenvolvimiento de fase estéreo utilizando dos cámaras y un proyector para la reconstrucción 3D aplicando una red neuronal convolucional entrenada para el procesamiento de las imágenes.

1.4.2. Objetivos particulares

- Proponer un prototipo base para el sistema estéreo para comparar los resultados obtenidos con algún método tradicional y el propuesto.
- Estudiar los métodos tradicionales de reconstrucción 3D, con la implementación del algoritmo FPP obteniendo métricas de comparación para la evaluación del método propuesto.
- Diseñar e implementar el método SPU con proyección de franjas, proponiendo una red neuronal convolucional para entrenar el modelo con alguna base de datos.
- Evaluar y validar los resultados obtenidos de método SPU mediante la comparación contra algún método establecido utilizando una base de datos sintética.

Antecedentes

2.1. Estado del arte

El cambio más grande en todos los tipos de instrumentación durante los últimos 30 años ha sido la integración de computadoras en el sistema de medición. La interferometría no es una excepción, y la principal manifestación de esta situación ha sido el desarrollo de interferometría de desplazamiento de fase (PSI por sus siglas en inglés). La PSI no es una configuración de hardware óptico específica, sino más bien un método de recopilación y análisis de datos que se puede aplicar a una gran variedad de situaciones de prueba, Figura 2.1.

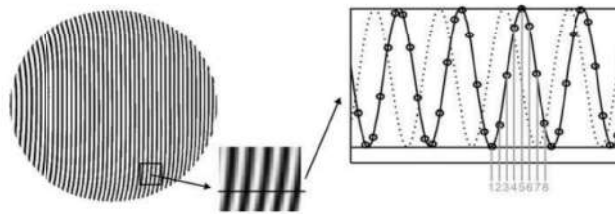


Figura 2.1: Proyección de franjas de una frecuencia portadora para los algoritmos de desplazamiento de fase [1].

Los métodos de metrología óptica generan imágenes, como la generación de patrones de franjas proyectadas para su procesamiento. Por tanto, la imagen se vuelve imprescindible para la reconstrucción del objeto a medir. En la mayoría de los métodos interferométricos, la imagen se forma por la superposición de las proyecciones de referencia y el objeto a medir, lo que da como resultado que los patrones de franjas del objeto sean modulados por una función armónica, lo que resulta en contrastes de luz y oscuridad [1].

2.1.1. Antecedentes de la Interferometría

A pesar de que el análisis de interferogramas estáticos se ha beneficiado de la informatización, adolece de la necesidad de encontrar centros marginales y el equilibrio resultante entre precisión y número de puntos de datos. Los datos se recopilan solo a lo largo de los centros marginales y la mayoría de los análisis necesitan los datos en una cuadrícula regular, por lo que requieren interpolación.

Este escaso muestreo espacial a lo largo de los centros marginales también limita la resolución lateral

del método. Con un interferograma estático, también se requiere información adicional para determinar la polaridad del frente de onda. PSI registra electrónicamente una serie de interferogramas mientras se cambia la fase de referencia del interferómetro. La fase del frente de onda se codifica en las variaciones en el patrón de intensidad de los interferogramas registrados, y un simple cálculo punto por punto recupera la fase. Se elimina la necesidad de ubicar los centros marginales, al igual que los problemas asociados.

Si bien la primera referencia a esta técnica se remonta a 1966 [14], el desarrollo de las técnicas de PSI realmente comenzó a principios de la década de 1970 como se muestra en la Tabla 2.1 los trabajos más relevantes.

Tabla 2.1: Inicios de desarrollo de técnicas PSI.

Cita	Autor	Año
[14]	Carre	1969
[15]	Moore	1973
[16]	Bruning et al.	1974
[17]	Wyant	1975
[18]	Johnson y Moore	1977
[19]	Hardy et al.	1977
[20]	Bruning	1978
[21]	Stump	1979
[22]	Massie et al.	1979

Con el advenimiento de la litografía óptica en los años 80, el desarrollo de PSI de esta década mostrada en la Tabla 2.2 recibió un impulso para mayor precisión y, por lo tanto, se desarrolló una teoría sobre algoritmos menos sensibles a las fuentes de error.

Tabla 2.2: Trabajos en los años 80's de técnicas PSI.

Cita	Autor	Año
[23]	Schwider et al.	1983
[24]	Creath	1988
[25]	Malacara	1990
[26]	Larkin y Ored	1992
[27]	Schwider et al.	1993

En los años 90, un tratamiento más riguroso del enfoque PSI dio como resultado varias teorías mostradas en la Tabla 2.3 sobre cómo desarrollar algoritmos PSI personalizados adaptados a las necesidades de las aplicaciones.

Tabla 2.3: Trabajos en los años 90's de técnicas PSI.

Cita	Autor	Año
[28],[29],[30],[31]	Surrel	1993-1996-1997-1998
[32]	Phillion	1997
[33],[34],[35]	Hibino et al.	1995-1997-1998-2004
[36],[37]	Malacara-Doblado et al.	2000-2001
[38]	Zhu y Gemma	2001

A lo largo de los años, esta técnica general ha sido conocida por varios nombres que incluyen interferometría de medición de fase, interferometría de barrido de franjas, interferometría en tiempo real, interferometría de CA e interferometría heterodino [1].

2.1.2. Problemas en los sistemas de metrología óptica

Realizar la reconstrucción del objeto a partir de las imágenes con franjas proyectadas ha representado un reto debido a las dificultades que presenta el método, teniendo entre sus ventajas alta precisión, sencilla instalación del hardware y flexibilidad de implementación obteniendo una excelente calidad en la reconstrucción de objetos complejos. Sin embargo, el método de desenvolvimiento de fase convencional generalmente no puede desenvolver la fase de manera robusta, sin la necesidad de algoritmos complementarios, siendo más complicado para la proyección de franjas densas, por lo que aumentar la frecuencia es crucial para la precisión, pero requiere una calibración compleja de los parámetros de fase [39], franjas proyectadas, y factores externos como el entorno, la luz, el ángulo de las cámaras, así como el ruido que pudiera generar el tiempo de exposición de la proyección para capturas rápidas y tener la mínima variación en la movimiento del objeto que aunque en su mayoría son estáticos, estas variaciones siguen estando presentes.

2.1.3. Estado del arte de los métodos de Desenvolvimiento de Fase (PS)

Los métodos de metrología óptica a menudo forman imágenes (por ejemplo, patrones de franjas/motas) para su procesamiento. Por lo tanto, la formación de imágenes es esencial para reconstruir varias cantidades. En la mayoría de los métodos metrológicos interferométricos, la imagen se forma por la superposición coherente del objeto y los haces de referencia. Como resultado, la intensidad bruta a través del objeto es modulada por una función armónica, lo que resulta en contrastes claros y oscuros, conocidos como patrones marginales.

Entre los métodos más conocidos de medición óptica se encuentran la interferometría clásica [24], la fotoelasticidad [40], el muaré geométrico [41], la interferometría holográfica [42], la holografía digital [43], la electrónica interferometría de patrón de motas [44], interferometría de corte de motas electrónicas [45], corrección de imagen digital [46] y perfilometría de proyección de franjas [47, 48, 49, 50]. Los procesos de formación de imágenes para los métodos típicos de metrología óptica se describen brevemente a continuación (Figura 2.2) [2]:

- a) **Interferometría clásica:** El patrón de franjas se forma mediante la superposición de dos frentes de onda, uno de los cuales suele ser un frente de onda de referencia plano o esférico y el otro un frente de onda distorsionado formado y dirigido por componentes ópticos
- b) **Fotoelasticidad:** Es una técnica de metrología óptica no destructiva de campo completo para medir la tensión desarrollada en objetos transparentes bajo carga.
- c) **Moiré geométrico/interferometría moiré:** La técnica moiré se define como la utilización del fenómeno moiré para medir la forma, la deformación o los desplazamientos de superficies
- d) **Interferometría holográfica:** Es una técnica que registra un patrón de interferencia y utiliza la difracción para reproducir un frente de onda, lo que da como resultado una imagen 3D que aún tiene la profundidad, el paralaje y otras propiedades de la escena original.
- e) **Holografía digital:** La holografía digital utiliza una cámara digital (CMOS o CCD) para registrar el holograma producido por la interferencia entre una onda de referencia y una onda de objeto que emana de la muestra
- f) **Interferometría de patrón de motas electrónicas (ESPI):** El objeto probado generalmente tiene una superficie ópticamente rugosa. Cuando se ilumina con un rayo láser coherente, creará un patrón moteado con fase, amplitud e intensidad aleatorias

- g) **Interferometría de corte de motas electrónicas (shearografía):** La interferometría de corte de motas electrónicas, comúnmente conocida como shearografía, es una técnica de medición óptica similar a ESPI.
- h) **Perfilometría/deflectometría de proyección de franjas:** La proyección de franjas es una técnica de metrología óptica no interferométrica ampliamente utilizada para medir la topografía de un objeto en un cierto ángulo entre el punto de observación y el de proyección.
- i) **Corrección de imagen digital (DIC)/estereovisión:** DIC es otro método importante de metrología óptica no interferométrica que emplea técnicas de correlación de imágenes para medir la forma, el desplazamiento y las tensiones de campo completo de la superficie de un objeto

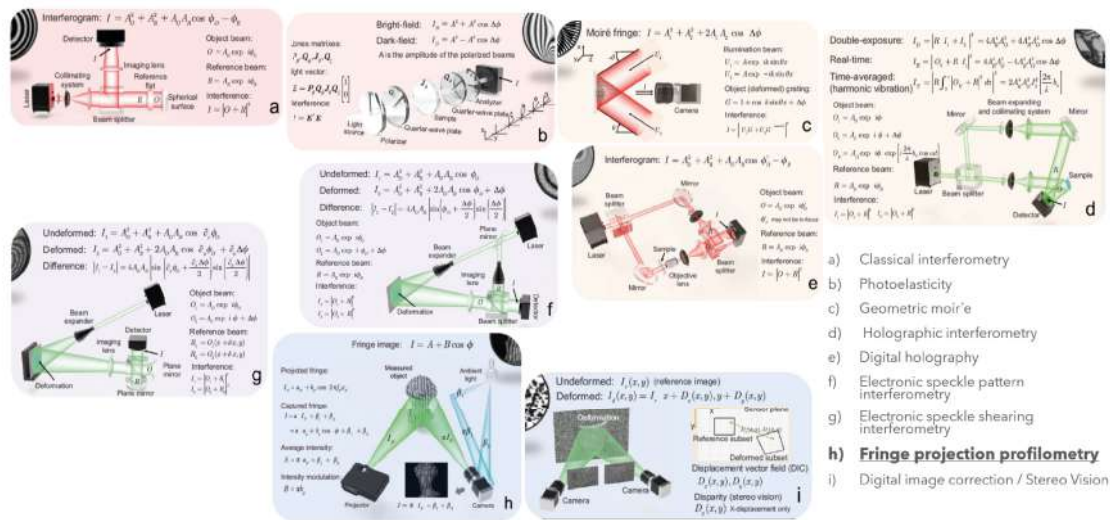


Figura 2.2: Formación de imágenes en métodos típicos de metrología óptica [2].

Dada la variedad existente para la medición de patrones de franjas, el estudio se centrará en la Deflectometría o mejor conocida como proyección de franjas.

La deflectometría no se ha quedado atrás, y el área principal ha sido el desarrollo de la Perfilometría de Desplazamiento de Fase (PSP por sus siglas en inglés) o también conocida como Perfilometría de Proyección Franjas (FPP por sus siglas en inglés). PSP es una configuración compuesta por diferentes tipos de hardware óptico, pero está más especializada en la recopilación y el análisis de datos que se pueden aplicar para una variedad de situaciones de medición que requieren precisión.

El sistema propuesto por Pan Ou [51] permite capturar de manera simultánea textura a color bidimensional y medición tridimensional de superficies en tiempo real. Este sistema propone reemplazar el proyector de pantalla de cristal líquido (LCD por sus siglas en inglés) usual por un proyector de procesamiento digital de luz (DLP por sus siglas en inglés) y una segunda cámara para obtener imágenes a color del objeto.

El método de Desenvolvimiento de Fase Temporal (TPU por sus siglas en inglés) generalmente necesita proyectar patrones de codificación gris adicionales o franjas de múltiples longitudes de onda, Figura 2.3, lo que disminuye en gran medida la eficiencia del desenvolvimiento de fase. El método de desenvolvimiento de fase espacial, puede realizar el desenvolvimiento de fase por solo un mapa de fase envuelto de acuerdo con la información de fase del vecindario. Pero fallará al medir objetos aislados.

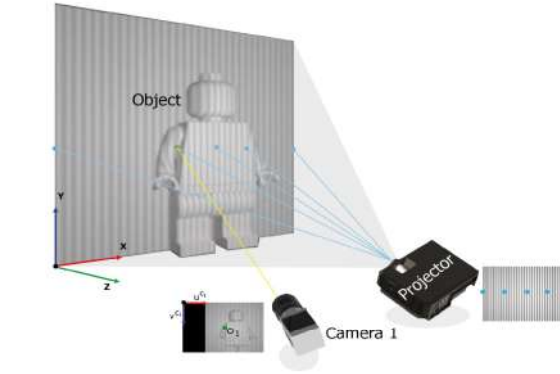


Figura 2.3: Sistema para la adquisición de la imagen del método TPU. [4]

Aunque el método de Desarrollo de Fase Estéreo (SPU por sus siglas en inglés) requiere más cámaras (al menos dos) en comparación con el sistema de Perfilometría de Proyección de Franjas tradicional, Figura 2.3, SPU maximiza la eficiencia del desenvolvimiento de fase y es muy adecuado para la medición 3D en tiempo real en escenarios dinámicos [6]. Sin embargo, la SPU convencional es generalmente insuficiente y a lo largo de los últimos años se han propuesto diferentes trabajos para resolver problemas particulares con algoritmos auxiliares que maximizan su desempeño.

En la segunda década del siglo XXI, la aplicación del método SPU ha ido en incremento gracias al gran avance del procesamiento de las computadoras lo que ha facilitado el desarrollo y experimentación de nuevos y mejores métodos para la reconstrucción 3D, en la Tabla 2.4 se hace un recopilado de los trabajos más relevantes que se han realizado hasta el momento.

2.1.4. Antecedentes de Redes Neuronales Convolucionales (CNN)

Deep Learning es un subconjunto de Machine Learning (ML), el cual lo definen como el uso de algoritmos específicos que permite el aprendizaje automático de patrones a partir de una gran cantidad de datos, que luego usará los patrones aprendidos para realizar una predicción de acuerdo a la entrada de datos [56]. El algoritmo clave utilizado en ML es ANN, que explora los datos de entrada $x \in X \subseteq \mathbb{R}^n$ para predecir una salida desconocida $y \in Y$. Las tareas que realiza las ANN se dividen en tareas de clasificación o tareas de regresión dependiendo de si la etiqueta es un valor continuo o discreto, en cambio el objetivo de ML es encontrar una función de mapeo $f : x \rightarrow y$. La elección de tales funciones viene dada por los modelos de redes neuronales con parámetros adicionales $\theta \in \Theta$: es decir, $\hat{y} = f(x, \theta) \approx y$ [57].

Artificial Neural Network (ANN)

Como se muestra en la Figura 2.4, la representación simple de una red neuronal se aprecia, la cual consiste de los siguientes elementos. Una red neuronal artificial tiene un bias b y un vector de pesos $w = (w_1, w_2, \dots, w_n)^T$ con parámetros $\theta = (b, w_1, w_2, \dots, w_n)^T$ para el mapa de entrada $x = (x_1, x_2, \dots, x_n)^T$ y para la salida $f_P(x)$ a través de una función de activación no lineal σ quedando una expresión como [58]:

$$f_P(x) = \sigma(w^T x + b) \quad (2.1)$$

Usualmente para las funciones de activación se usa la función señal $\sigma(x) = \text{sgn}(x)$, la función sigmoidea $\sigma = \frac{e^x - e^{-x}}{e^x + e^{-x}}$, y la unidad lineal rectificadora o mejor conocida como ReLU $\sigma(x) = \max(0, x)$ [59]. Una

Tabla 2.4: Antecedentes de sistemas estéreo.

Cita	Técnica	Uso de técnica	Aportación
[52]	Desenvolvimiento de fase absoluta asistida por estéreo para sistemas de luz estructurada estéreo.	Reconstrucción 3D de escenas dinámicas.	Hacer cumplir el punto de vista y la coherencia temporal da como resultado menos errores en las imágenes sin envolver y nubes de puntos más precisas.
[53]	Desenvolvimiento de fase por restricciones epipolares.	Reconstrucción 3D rápida basada en luz estructurada, en particular para objetos con geometría compleja.	Reduce en gran medida la complejidad de cualquier algoritmo de desenvolvimiento posterior, por lo que su rendimiento aumenta considerablemente.
[54]	Enfoque de restricción de profundidad adaptativa.	Medición de formas 3D en tiempo real de alta velocidad.	Los experimentos demuestran la capacidad del método para realizar mediciones de formas 3D en tiempo real con alta precisión, para superficies complejas y objetos espacialmente aislados, utilizando solo tres patrones de franjas de alta frecuencia.
[55]	Perfilometría de proyección de franjas con restricción de textura.	Sensor 3D de alta precisión para inspección de remaches.	Los experimentos demostraron la viabilidad y la alta precisión. La dimensión del remache se puede extraer automáticamente y, en comparación con el método tradicional, la precisión del diámetro se puede mejorar en un 50 %.
[6]	Desenvolvimiento de fase estéreo con la restricción de profundidad adaptativa basado en la proyección de retroalimentación.	Obtención rápida de la profundidad absoluta del objeto para corregir el rango de profundidad dinámico de la Restricción de Profundidad Adaptativa (ADC por sus siglas en inglés), guiando así la adquisición de información 3D de alta calidad.	Logra una medición 3D de alta velocidad, en tiempo real y de alta resolución con una velocidad de medición de 30 Hz bajo la premisa de utilizar dos perspectivas.
[5]	Aprendizaje profundo y desenvolvimiento de fases.	Medición de forma absoluta en 3D de un solo disparo.	Resuelve el problema de la ambigüedad de fase de las imágenes de franjas densas en un rango de medición más grande con menos información de perspectiva y algoritmos más simples.

red neuronal puede aproximar cualquier función continua $f(x)$ en un limitado conjunto \mathbb{R}^n , una red de capa simple esta representado como una combinación lineal de M neuronas individuales.

$$f_{1NN}(x) = \sum_{i=1}^M v_i \sigma w_i^T x + b_i \quad (2.2)$$

Donde v_i , es una combinación de pesos de i th neuronas. Así también podemos extender las especificaciones matemáticas de una red de capa simple a una red multicapa para la representación de funciones no lineales. De esta forma formalizaremos matemáticamente la aplicación de varias redes de capa simple para formar una red neuronal profunda (DNN) con N capas.

$$f_{DNN}(x) = f_{1NN}(f_{DNN}(\dots f_{DNN})) = f_{DNN} \circ f_{DNN} \dots \circ f_{DNN}(x) \quad (2.3)$$

La primera capa se referirá a la capa de entrada y la ultima como la capa de salida, y las capas entre la entrada y salida serán las capas ocultas. Nos referiremos para usar el termino "deep", cuando una red neuronal contenga muchas capas ocultas, dando el termino "deep learning".

Entrenamiento de Redes Neuronales

Con las bases de los conocimientos básicos sobre las redes neuronales y su topología básica, es requerido conocer como se entrenan, es decir, como determinar realmente sus parámetros θ . Para ello es necesario

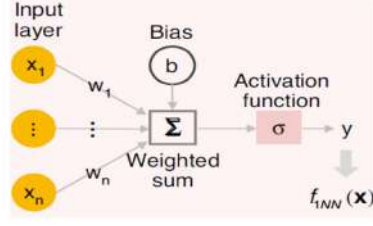


Figura 2.4: Representación matemática del perceptrón de una capa simple [2].

siempre elegir la topología de modelos adecuado para el problema a resolver y especificar los diversos parámetros asociados con el modelo (conocidos como *hiper-parámetros*. Además, definiremos una función que evalúe la calidad del conjunto de parámetros de la red θ , llamada "función de pérdida" L , la cual cuantifica el error entre el valor predicho $\hat{y} = f_{\theta}(x)$ y el valor observado y (etiqueta) [60].

Dependiendo del tipo de tarea realidad por la red, la función de pérdida se puede dividir en pérdida de clasificación y pérdida de regresión. Las funciones de pérdida usualmente incluyen pérdida de eje L_{Hinge} y pérdida de entropía cruzada L_{CE} [61].

$$L_{CE} = \sum_{i=1}^n [y_i \log \hat{y}_i + (1 - y_i) \log(1 - \hat{y}_i)] \quad (2.4)$$

$$L_{Hinge} = \sum_{i=1}^n \max[0, 1 - \text{sgn}(y_i) \hat{y}_i] \quad (2.5)$$

Dado que las tareas de metrología óptica involucradas pertenecen principalmente a tareas de regresión, nos enfocaremos en las funciones de pérdida de regresión. La pérdida del error absoluto medio (MAE por sus siglas en ingles) (L_{MAE}) y la pérdida del error cuadrático medio (MSE por sus siglas en ingles) (L_{MSE}) son las dos funciones mas utilizadas, que también se conoce como L_1 loss y L_2 loss, respectivamente.

$$L_{MAE} = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (2.6)$$

$$L_{MSE} = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (2.7)$$

En las tareas de procesamiento de imágenes, MSE generalmente se convierte en una métrica de relacion señal-ruido maxima (PSNR por sus siglas en ingles) (L_{PSNR}).

$$L_{PSNR} = 10 \log_{10} \frac{MAX^2}{L_{MSE}} \quad (2.8)$$

Donde MAX es el valor máximo de intensidad del píxel dentro del rango dinámico de la imagen sin procesar [62]. Otras variantes de L_1 y L_2 incluyen RMSE, pérdida Euclidiana, L_1 uniforme etc. [63]. Para imágenes naturales, el índice de similitud estructural (SSIM por sus siglas en ingles) es una medida de fidelidad de imagen respectiva, que juzga la similitud estructural de dos imágenes en función de tres métricas (luminosidad, constaste y estructura): $L_{SSIM} = l(y, \hat{y})c(y, \hat{y})s(y, \hat{y})$ [64], donde la funciones l , c y s son las similitudes de las tres métricas del parche local respectivamente. Con la función de pérdida definida, el objetivo detrás del proceso de entrenamiento de ANN se puede formalizar como un problema de optimización [65].

$$\hat{\theta} = \arg \theta_{\in \Theta} \min L(f_{\theta}(x, y)) \quad (2.9)$$

Es así que los esquemas de aprendizaje se pueden clasificar en términos generales en tres categorías, aprendizaje supervisado [66], semi-supervisado [67] y no supervisado [68].

- **Aprendizaje supervisado:** Es aplicada en la mayoría de las aplicaciones practicas, en las que se optimiza un modelo de red neuronal basado en una gran cantidad de datos de pares de datos etiquetados (x, y) , y el proceso de entrenamiento consiste en encontrar los parámetros del modelo $\hat{\theta}$ que predican mejor los datos basados en la función de perdida $L(\hat{y}, y)$.
- **Aprendizaje no supervisado:** Los algoritmos de entrenamiento procesan los datos de entrada x sin las etiquetas correspondientes, y la distribución subyacente de los datos debe modelarse en función de la entrada.
- **Aprendizaje semi-supervisado:** Se ubica entre los dos anteriores, donde una gran cantidad de datos de entrada x esta disponible y solo algunos de los datos están etiquetados.

Redes Neuronales Convolucionales (CNN)

Los conceptos de Machine Learning y Deep Learning comenzaron con la investigación de la red neuronal artificial, que se origino a partir del modelo matemático simplificado de neuronas biológicas estableció por McCulloch y Pitts en 1943 [69]. Sin embargo, los métodos clásicos para las tareas de clasificación y detección comienzan a ser largamente superados a partir del año 2012, específicamente con el trabajo que dio origen a la red AlexNet [70] para la tarea de clasificación. Para esto se usó una red neuronal convolucional (CNN por su sigla en inglés) con cinco capas convolucionales y tres capas completamente conectadas.

Surgieron numerosas investigaciones que profundizaron en el uso de las CNN para las tareas de clasificación entre las cuales destacan las redes conocidas como ZF [71], VGG [72] y ResNet [73]. Conjuntamente con el surgimiento de los sistemas de clasificación que emplean CNN, se desarrollaron sistemas de detección, algunos de los cuales se basaron en métodos del tipo región específica, que se fundamentan en la idea de que todos los objetos de interés en una imagen comparten características que los diferencian del fondo. De esta forma no es necesario hacer un recorrido exhaustivo por toda la imagen como sucede bajo el paradigma de ventana deslizante. Anteriormente en la Sección anterior hablamos de las DNN simples, e básicamente consistían en múltiples capas de neuronas, cada una completamente conectada a las de la capa adyacente, donde cada neurona recibía una entrada que se multiplicaban por sus pesos, con la no-linealidad aplicada a través de las funciones de activación. Las CNN representan una evolución en la arquitectura de MLP que se desarrollan para procesar datos en matrices únicas o múltiples siendo mas apropiadas para manejar entradas similares a imágenes.

Las CNN siguen el mismo patrón que MLP: las neuronas artificiales se apilan en capas ocultas una encima de la otra; los parámetros se aprenden durante el entrenamiento de la red con no-linealidad aplicada a través de funciones de activación; la perdida $L(\hat{y}, y)$ es calculada y retro-propaga para actualizar los parámetros de la red. Ahora la principal diferencia es que en lugar de capas regulares completamente conectadas, las CNN usa capas de convolución especializadas para modelar la localidad y la abstracción (Figura 2.5a). En cada capa, la imagen de entrada x (ordenada lexicográficamente) se convoluciona con un conjunto de filtros convolucionales W (La W representa la matriz de convolución de bloque Toeplitz) y sesgos agregados b para generar una nueva imagen, que esta sujeta a una función de activación no-lineal por elementos σ , y la misma estructura se repite para cada capa de convolución k .

$$x^k = \sigma(W_{k-1}x_{k-1} + b_{k-1}) \quad (2.10)$$

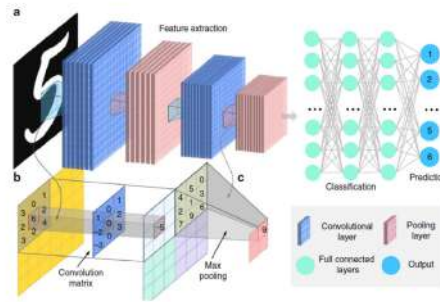


Figura 2.5: Arquitectura de CNN para clasificación de imágenes [2].

La segunda diferencia clave es la incorporación típica de capas de agrupación en CNN, donde los valores del píxel de las vecindades se agregan mediante la aplicación de una función invariante de permutación, como la operación máxima o media, para reducir la dimensionalidad de las capas convolucionales y que permite que las características significativas se propaguen abajo sin verse afectadas por píxeles vecinos 2.5c. La principal ventaja de las arquitecturas CNN es que explotan las dependencias espaciales en la imagen y solo consideran un vecindario local para cada neurona, en otras palabras los parámetros de la red se comparten de tal manera que la red realiza operaciones de convolución en las imágenes. La idea de una CNN es aprovechar una estructura piramidal para identificar primero características en el nivel mas bajo antes de pasar estas características a la siguiente capa, que a su vez crea características de un nivel superior. Dado que las estadísticas locales de las imágenes son invariantes a la ubicación, el modelo no necesita aprender pesos para la misma característica que ocurre en diferentes posiciones en una imagen, lo que hace que la red sea equivariante con respecto a las traducciones de la entrada. Hace que las CNN sean especialmente adecuadas para procesar imágenes capturadas en metrología óptica, siendo en nuestro caso un patrón de franjas que consiste en una señal sinusoidal repetida en diferentes ubicaciones de la imagen. Además, reduce drásticamente la cantidad de parámetros que debe aprender (la cantidad de pesos ya no depende del tamaño de la imagen de entrada).

En resumen como se muestra en la Figura 2.5, cada capa de una CNN transforma el volumen de entrada en un volumen de salida de activación de neuronas, lo que eventualmente conduce a las capas finales completamente conectadas, lo que da como resultado un mapeo de los datos de entrada a un vector de características 1D. Una configuración típica de CNN consta de una secuencia de capas de convolución y agrupación. Después de pasar por algunos pares de capas convolucionales y de agrupación, todas las características de la imagen se extrajeron y organizaron en un tubo largo. Al final del flujo convolucional de la red, generalmente se agregan varias capas completamente conectadas (es decir, arquitectura de red neuronal normal, MLP) para engordar las características en un vector, con el cual tareas, como clasificaciones, puede ser llevado a cabo.

Arquitectura Fully Convolutional Network (FCN) para el procesamiento de imagen

Debido a la presencia de una capa con muchos parámetros completamente conectada al final de la red, las CNN suelen descartar información espacial y producen salidas no espaciales, y para el procesamiento de imágenes en metrología óptica, deben tener la misma resolución o incluso mayor en comparación con la entrada, lo que se conoce como predicción densa. Es por eso que para este propósito se deben usar arquitecturas de red totalmente convolucionales sin capas totalmente conectadas y que acepten entradas de cualquier tamaño [74]. Existen tres arquitecturas de red representativas con tales características.

- SRCNN:** En la CNN convencional, el efecto de reducción de muestreo de la agrupación de capas da como resultado una salida con una resolución mucho más baja que la entrada. Por lo tanto, una solución relativamente ingenua y directa es simplemente apilar varias capas de convoluciones mientras se salta la agrupación de capas para conservar las dimensiones de entrada. Dong et al.[75] primero adoptan esta idea y proponen SRCNN para la tarea de superresolución de imágenes. SRCNN utiliza algoritmos tradicionales de sobremuestreo para obtener imágenes de baja resolución y luego refinarlas aprendiendo un mapeo de extremo a extremo desde imágenes gruesas interpoladas hasta imágenes de alta resolución de la misma dimensión pero con más detalles, como se ilustra en Figura 2.6. Debido a su simple ideal e implementación, SRCNN se ha convertido gradualmente en uno de los marcos más populares en superresolución de imágenes[76] y se ha extendido a muchas otras tareas, como la mejora de imágenes de radar [77], la visualización de imágenes submarinas de alta definición [78] y la tomografía computarizada [79]. Una desventaja importante de SRCNN es el costo de tiempo y espacio para mantener la resolución completa a través de toda la red, lo que limita SRCNN solo práctico para estructuras de red relativamente poco profundas.

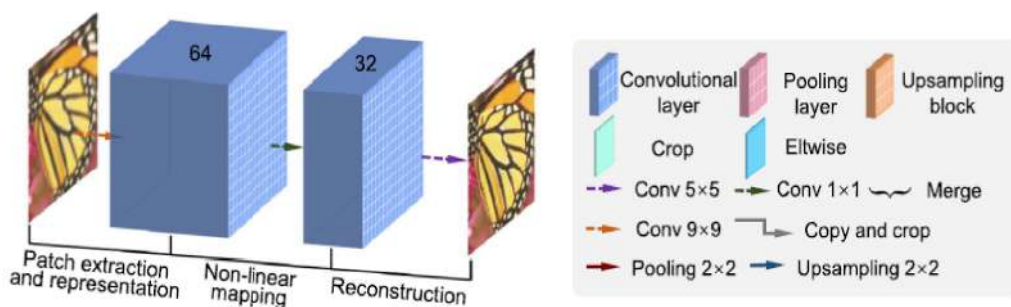


Figura 2.6: Estructura para procesamiento de imagen SRCNN [2].

- FCN:** La red totalmente convolucional (FCN) propuesta por Long et al. [80] es una estrategia popular y una línea de base para tareas de segmentación semántica. FCN se inspira en el hecho de que las capas totalmente conectadas en La clasificación CNN (Figura 2.5) también puede verse como circunvoluciones con núcleos que cubren la totalidad de sus regiones de entrada. Como se ilustra en la Figura 2.7, FCN utiliza la clasificación CNN existente como el módulo codificador de la red y reemplaza estas capas completamente conectadas en capas de convolución 11 (también denominadas capas de deconvolución) como módulo de decodificación, lo que permite que la CNN aumente la muestra de los mapas de características de entrada y obtener resultados de píxeles. En FCN, omitir conexiones combinando (simplemente agregando) información en capas finas y capas gruesas mejora la capacidad de localización de la red, lo que permite la reconstrucción de detalles precisos que respetan la estructura global. FCN y sus variantes han logrado un gran éxito en la aplicación de predicción de píxeles densos como se requiere en muchas tareas de computadoras avanzadas comprensión de la visión [81].
- U-net:** Ronneberger et al. [82] tomaron la idea de FCN y dieron un paso más y propusieron la arquitectura U-Net, que reemplaza la parte de muestreo ascendente de un paso con un montón de capas de convoluciones de muestreo ascendente complementarias, lo que da como resultado un codificador-decodificador casi simétrico del modelo de la arquitectura. Como se ilustra en la Figura 2.8, la estructura básica de U-Net consta de una rama contractiva y una rama expansiva, lo que permite el análisis multiresolución y transformaciones generales de imagen a imagen multiescala. La rama contractiva

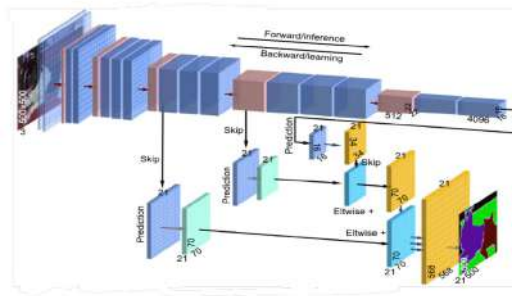


Figura 2.7: Estructura para procesamiento de imagen FCN [2].

(codificador) reduce la muestra de la imagen utilizando una convolución de zancadas convencional, produciendo una representación de características comprimidas de la imagen de entrada. el expansivo branch (decodificador), complementario a la rama contractiva, utiliza métodos de muestreo superior como la convolución de transposición para proporcionar el resultado procesado con el mismo tamaño que la entrada. Además, U-Net presenta conexiones de salto que concatenan la niveles de resolución coincidente de la rama contractiva y la rama expansiva. U-Net de Ronneberger es un gran avance hacia la segmentación automática de imágenes y se ha aplicado con éxito en muchas tareas que requieren transformaciones de imagen a imagen [83].

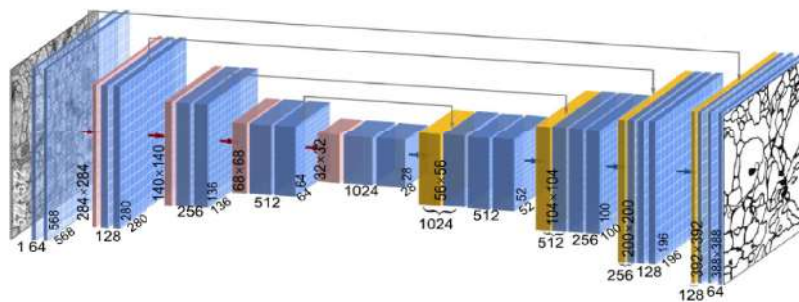


Figura 2.8: Estructura para procesamiento de imagen U-net [2].

Las aplicaciones de redes neuronales convolucionales comenzaron a tener aplicaciones mas específicas como lo que era clasificación, segmentación, entre muchas otras, lo que ha llevado su estudio a aplicaciones mas específicas en el área de la metrología óptica, y la necesidad de generar un procesamiento de imagen para mejorar la eficiencia en las mediciones por tecnologías de proyección de luz estructurada. La luz es caracterizada por propiedades fundamentales como lo es la amplitud, fase, longitud de onda, dirección, frecuencia, velocidad y polarización. Lo que lleva a generar un preprocesamiento para mejorar las mediciones realizadas en lo métodos y algoritmos existentes.

Metodología

3.1. Principio de la Perfilometría de proyección de franjas y desenvolvimiento de fase estéreo

Como se muestra en la Figura 3.1, un sistema básico consta de una cámara y un proyector. El algoritmo de cambio de fase de N pasos es un método conocido por su insensibilidad a la luz ambiental y la reflectividad de la superficie [84]. Se proyectan una serie de patrones desfasados con un cambio de fase y el número mínimo de imágenes es de al menos dos, entre ellos están los de 3-Paso, 4-Paso, 8-Paso, y 12-Paso entre otros, el último siendo más recomendable para adquirir la intensidad de fondo. Los patrones de franjas capturados por la cámara 1 para N -Step se pueden expresar como:

$$I_{cn}(u^c, v^c) = A_c(u^c, v^c) + B_c(u^c, v^c) \cos(\phi_c(u^c, v^c) + 2\pi n/N) \quad (3.1)$$

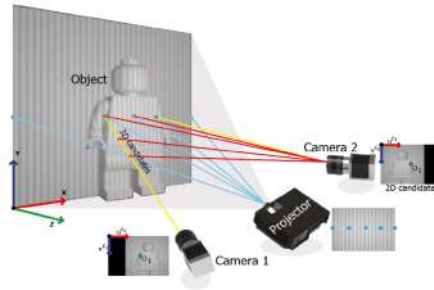


Figura 3.1: El Principio Básico de un sistema estéreo.

Donde I_{cn} representa la imagen capturada $(n+1)th$, $n \in [0, 2]$, (u, v) son las coordenadas de píxeles de la imagen y c correspondiente cámara ($1[L]$ a N -cámaras), A es la intensidad media (Figura 3.2), B denota la amplitud, ϕ es la fase, y $2\pi n/N$ es el cambio de fase que implementaremos para nuestro trabajo. Con la ortogonalidad de las funciones trigonométricas, la intensidad de fondo (promedio) se puede obtener como:

$$A_c(u^c, v^c) = \frac{1}{N} \sum_{n=0}^{N-1} I_{cn}(u^c, v^c) \quad (3.2)$$

El emulador nos permitirá trabajar bajo la premisa de que tanto la luz ambiental como la reflectividad permanecerán constantes una vez que se represente el entorno, los tres patrones de franjas de cambio de fase

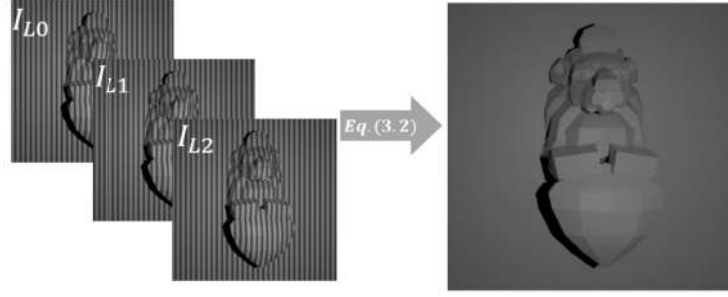


Figura 3.2: Background intensity de una proyección de franjas.

estándar con un cambio de $2\pi/3$ se capturan para la cámara 1, donde (u^c, v^c) se omite por conveniencia:

$$\begin{aligned} I_{L0} &= A_L + B_L \cos(\phi_L), \\ I_{L1} &= A_L + B_L \cos(\phi_L + 2\pi/3), \\ I_{L2} &= A_L + B_L \cos(\phi_L + 4\pi/3). \end{aligned} \quad (3.3)$$

Con el algoritmo PSP, la fase envuelta ϕ (Figura 3.3) para N -Step [1] se puede obtener con el método de mínimos cuadrados mediante:

$$\phi_L = \arctan \frac{\sum_{n=0}^{N-1} I_{cn} \sin(2\pi n/N)}{\sum_{n=0}^{N-1} I_{cn} \cos(2\pi n/N)} \quad (3.4)$$

Si es posible conocer la fase ϕ_L , la posición 3D se puede calcular usando los parámetros de calibración entre la cámara y el proyector [85]. Por lo tanto, la fase envuelta ϕ_L se puede calcular de la siguiente manera [86]:

$$\phi_L = \arctan \frac{\sqrt{3}(I_{L1} - I_{L2})}{2(I_{L0} - I_{L1} - I_{L2})} \quad (3.5)$$

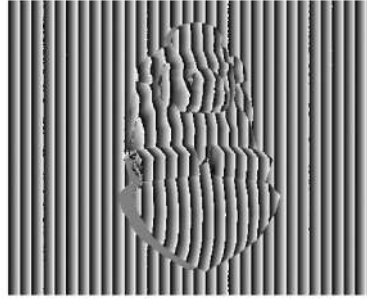


Figura 3.3: Representación 2D de la Fase envuelta (ϕ).

Las fases absoluta Φ_L y envuelta ϕ_L satisfacen la siguiente relación:

$$\Phi_L = \phi_L + 2k\pi \quad (3.6)$$

Donde k es el orden del período, $k \in [0, K - 1]$ y K denota el número de franjas proyectadas en el plano marginal del sistema. El orden marginal k se puede obtener mediante varios algoritmos de PSP (Ver [10, 11, 2]) que tienen el propósito de reducir el error de ambigüedad de fase para obtener la mejor calidad

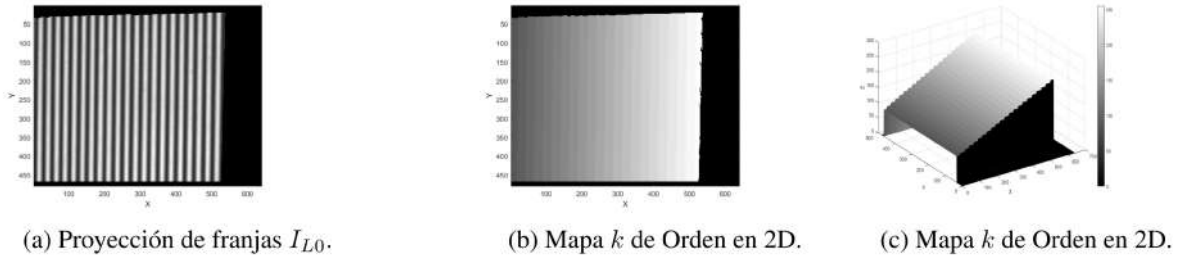


Figura 3.4: Representación gráfica del mapa de orden k del plano de referencia para la objeto de franjas sin objetos.

en el mapa de pedidos para desenvolver la fase correctamente. Sin embargo, debido a errores de calibración entre elementos y factores externos como la reflexión y la interferencia de la luz ambiental, algunos datos 2D pueden ser incorrectos, ya que pueden tener valores similares al punto de coincidencia O_c . Además, cuanto mayor sea el número de patrones proyectados, mayores serán las posibilidades de estos puntos. Por lo tanto, para resolver este problema, se recomienda la aplicación de un algoritmo PS de varios pasos que tendrá una mayor precisión de medición y robustez frente a la iluminación ambiental. Tampoco se recomiendan los patrones de franjas de alta frecuencia, ya que son susceptibles a problemas de alias [5].

3.2. Métodos de metrología óptica

Existen diversas metodologías para lograr desenvolver con éxito los mapas envueltos generados por la ecuación 3.6, es aquí donde el campo de estudio se vuelve muy amplio y la investigación desarrollada se ha enfocado en desarrollar métodos mas óptimos y que generen un mapa de fase absoluto con mejor calidad para la reconstrucción 3D del objeto. Trabajaremos con los tres métodos mas conocidos y generales para los Mapas de fase envueltos. Como se muestra en la Figura 3.5 un ejemplo de las diferencias entre estos tres, los cuales de detallaran en las siguientes secciones. Sin embargo, se explicara en la siguiente sección cual es la metodología para lograr obtener los mapas respectivos de cada paso como se muestra en la Figura 3.6.

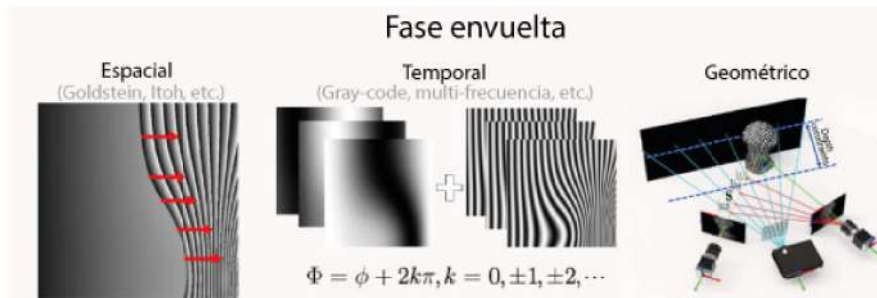


Figura 3.5: Algoritmos de desenvolvimiento de la Fase envuelta para proyección de franjas [2].

3.2.1. Proceso de reconstrucción 3D con imágenes sintéticas

El método de desenvolvimiento de fase generalmente requiere varias imágenes con franjas proyectadas desplazadas según el N-Step que se utilice. Sin embargo, los complicados algoritmos auxiliares requieren la

proyección de franjas densas y de alta frecuencia para desenvolver con éxito la fase. El Diagrama de Flujo de la Figura 3.6 muestra de manera general el procedimiento básico para obtener el desarrollo de la fase envuelta generada por las imágenes emuladas en el entorno gráfico.

Discutiremos brevemente el proceso para obtener la reconstrucción de objetos 3D.

- Paso 1: Para generar el patrón de franjas desplazadas, es necesario variar el valor de la señal $\sin(x + \delta)$ del cuadro 'Seno' del árbol de nodos en la Figura 3.14. Para nuestro caso, el desplazamiento generado fue $\delta = 0, 2\pi/3$ y $4\pi/3$, de manera que la lámpara emula el desplazamiento de las franjas.
- Paso 2: Procederemos a capturar cada proyección de la Lámpara con la cámara lo cual obtendremos por la fórmula (3.3).
- Step 3: Una vez que capture las imágenes con la fórmula (3.5), calcularemos lo que conocemos como la Fase Envuelta que está entre $-\pi$ y π .
- Paso 4: Para lograr un desarrollo de la fase envuelta existen varios métodos para obtener con éxito la fase absoluta, en nuestro trabajo se implementó el Algoritmo TPU, el cual no es rápido pero obtiene un Desenvolvimiento de Fase robusto.
- Paso 5: Finalmente, una vez obtenida la Fase Absoluta con el objeto, se requiere obtener la Fase Absoluta del plano de referencia (siguiendo el mismo procedimiento mencionado anteriormente) para obtener los datos del objeto 3D.

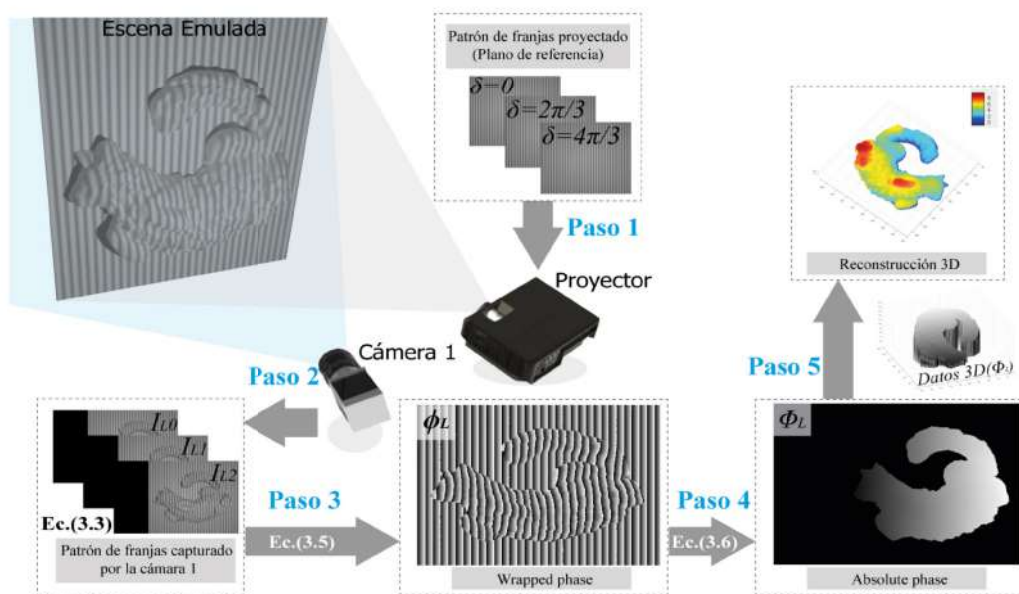


Figura 3.6: Diagrama de flujo del algoritmo de cambio de fase de 3 pasos.

El análisis de las imágenes es el componente central de la arquitectura para el procesamiento de la extracción de la información clave que contiene los parámetros que reflejan la cantidad física deseada que se mide a partir de las imágenes tomadas. En las técnicas de la medición de la fase envuelta, el análisis para desenvolver la fase, refiriéndose a la reconstrucción de la información a partir de las distribuciones de la intensidad modulada tipo franja, es decir, $f_{anal} : I \rightarrow \phi$.

El objetivo de la demodulación de fase o mejor conocido como análisis de franjas, es obtener el mapa de fase envuelto a partir de los patrones de franjas cuasiperiódico. Se han desarrollado varias técnicas para el análisis de franjas como se ha mencionado anteriormente, las cuales cumplen con los requisitos y las podemos clasificar en dos categorías, Spatial phase demodulation y Temporal phase demodulation por sus nombre en ingles que son mejor conocidas.

No importa que demodulación de fase se utilice la técnica, la distribución de fase recuperada se ajusta matemáticamente al valor principal de la función \arctan que oscila entre $-\pi$ y π . el resultado es lo que se conoce como una imagen de fase envuelta, y se debe realizar un desenvolvimiento de fase para eliminar cualquier discontinuidad de fase 2π . Y los algoritmos de desenvolvimiento de fase se pueden clasificar en términos generales en tres categorías, las cuales se detallan los métodos a continuación.

3.2.2. Spatial phase unwrapping

Anteriormente hemos hablado del desenvolvimiento de fase, el método Spatial Phase Unwrapping se deriva de la teoría base para dos-dimensiones. Teóricamente de la manera mas general el desenvolvimiento de fase de dos-dimensiones es un problema imposible. Esto se debe a que una función de fase desconocida ϕ corrompida por ruido y envuelta dentro de un intervalo $(-\pi, \pi]$ es imposible de recuperar inequívocamente. Sin embargo ciertas suposiciones del proceso subyacente puede hacer el problema de la fase envuelta tratable, pero no fácil.

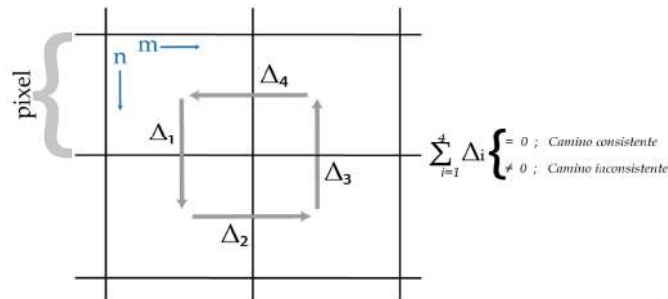


Figura 3.7: Camino realizado para la detección de residuos. Los índices de dirección m y n son mostrados para indicar la dirección de los píxeles.

La mayoría de las suposiciones es que se desea que la fase desenvuelta tenga una fase local que sea menor a π radianes en magnitud en todas lados. Se asume que conocemos la fase ϕ , el modulo 2π , en una cuadrícula discreta de puntos [87].

$$\Phi_{i,j} = \phi_{i,j} + 2\pi k, \quad (3.7)$$

k es un entero, $-\pi < \Phi_{i,j} \leq \pi$, $i = 0, \dots, M-1$, $j = 0, \dots, N-1$.

Utilizaremos la componente del gradiente discreto como una aproximación para la fase del gradiente continuo. Específicamente, utilizaremos 2 conjuntos de diferencia de fases, estas diferencias serán con respecto al índice i y con respecto al índice j . De la fase envuelta $\Phi_{i,j}$, podemos calcular la siguiente diferencia de la fase envuelta.

$$\begin{aligned} \Delta_{i,j}^x &= W\{\Phi_{i+1,j} - \Phi_{i,j}\}, \quad i = 0, \dots, M-2, \quad j = 0, \dots, N-1, \\ \Delta_{i,j}^x &= 0 \text{ de lo contrario} \end{aligned} \quad (3.8)$$

y

$$\begin{aligned}\Delta_{i,j}^y &= W\{\Phi_{i,j+1} - \Phi_{i,j}\}, \quad i = 0, \dots, M-1, \quad j = 0, \dots, N-2, \\ \Delta_{i,j}^y &= 0 \text{ de lo contrario}\end{aligned}\quad (3.9)$$

Donde los superíndices x y y se refieren a la diferencia envuelta en los índices i y j respectivamente. W define un operador de envoltimiento que envuelve todos los valores de sus argumentos en un rango de $(-\pi, \pi]$ para adicionar o sustraer un múltiplo entero de 2π de su argumento.

Para usar los residuos en el proceso del desenvolvimiento de fase, debemos ser capaces de localizarlos. Asumamos que tenemos un cadena la cual podría consistir en las muestras que se muestra en la Figura 3.7, la cual representa un ciclo de muestro de la fase. Ahora podemos considerar la integración del gradiente alrededor del camino. Una estimación del gradiente verdadero es obtenido por la diferencia envuelta de la fase envuelta, definiendo cada gradiente de la siguiente manera.

$$\begin{aligned}\Delta_1 &= W\{\Phi(m, n+1) - \Phi(m, n)\}, \\ \Delta_2 &= W\{\Phi(m+1, n+1) - \Phi(m, n+1)\}, \\ \Delta_3 &= W\{\Phi(m+1, n) - \Phi(m+1, n+1)\}, \\ \Delta_4 &= W\{\Phi(m, n) - \Phi(m+1, n)\},\end{aligned}\quad (3.10)$$

Ahora calcularemos la carga del residuo para sumar las diferencias de la fase alrededor del ciclo cerrado mostrado en la Figura 3.14. Si el valor obtenido es diferente de cero, entonces tenemos un camino inconsistente que nos estaría indicando una discontinuidad de la fase.

$$q = \sum_{i=1}^4 \Delta_i \quad (3.11)$$

Ahora asociaremos la Δ s con una (No envuelta) diferencia de fase de una continuidad subyacente de la función de la fase.

$$\begin{aligned}\Delta_1 &= W\{\phi(m, n+1) - \phi(m, n)\}, \\ \Delta_2 &= W\{\phi(m+1, n+1) - \phi(m, n+1)\}, \\ \Delta_3 &= W\{\phi(m+1, n) - \phi(m+1, n+1)\}, \\ \Delta_4 &= W\{\phi(m, n) - \phi(m+1, n)\},\end{aligned}\quad (3.12)$$

Mas información es obtenida si manipulamos la ecuación 3.12, Para obtener:

$$\begin{aligned}\sum_{i=1}^4 \Delta_i &= [\phi(m, n+1) - \phi(m, n)] \\ &+ [\phi(m+1, n+1) - \phi(m, n+1)] \\ &- [\phi(m+1, n+1) - \phi(m+1, n)] \\ &- [\phi(m+1, n) - \phi(m, n)].\end{aligned}\quad (3.13)$$

Los cuatro términos en sumatorio en el lado derecho de la ecuación 3.13 pueden ser categorizada de la siguiente manera: asociando al primer termino con la contra-parte discreta para una derivada parcial en la dirección de y . De este modo:

$$\phi(m, n+1) - \phi(m, n) \implies \phi_y(m, n) \quad (3.14)$$

Donde el sufijo y se refiere a la diferenciación parcial en la dirección y . Similarmente nosotros asociamos los términos remanentes en la ecuación 3.13 con la otra derivada parcial como se muestra:

$$\begin{aligned}\phi(m+1, n+1) - \phi(m, n+1) &\implies \phi_x(m, n+1), \\ \phi(m+1, n+1) - \phi(m+1, n) &\implies \phi_y(m+1, n), \\ \phi(m+1, n) - \phi(m, n) &\implies \phi_x(m, n).\end{aligned}\quad (3.15)$$

De este modo

$$\sum_{i=1}^4 \Delta_i = [\phi_x(m, n+1) - \phi_x(m, n)] - [\phi_y(m+1, n) - \phi_y(m, n)] \quad (3.16)$$

Si por un razonamiento similar, nosotros hacemos la siguiente asociación,

$$\begin{aligned} \phi_x(m, n+1) - \phi_x(m, n) &\implies \phi_{xy} = \frac{\partial \phi_x}{\partial y}, \\ \phi_y(m+1, n) - \phi_y(m, n) &\implies \phi_{yx} = \frac{\partial \phi_y}{\partial x} \end{aligned} \quad (3.17)$$

Encontramos que

$$q = \sum_{i=1}^4 \Delta_i = \phi_{yx} - \phi_{xy}. \quad (3.18)$$

De esta manera es obvio de la ecuación 3.18 que el camino cerrado de la integral es cero, si y solo si, las derivadas cruzadas son las mismas. La localización y las densidades de los residuos indican de la severidad del problema de la fase desvelada y gráficamente indica donde los problemas en el proceso de desenvolvimiento están ocurriendo. En adición las pistas provista por los residuos en cuanto a como corta la rama podría ser reconstruida o como la integración del camino podría ser seleccionada. Irónicamente los residuos son el mismo problema y la solución.

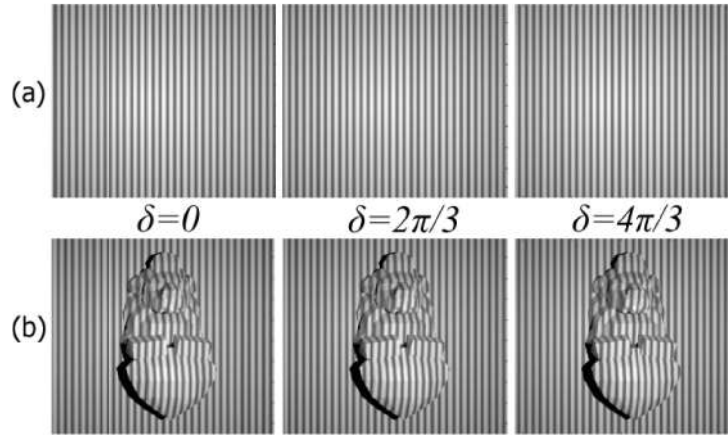


Figura 3.8: Imágenes de las franjas proyectadas en el plano de referencia(a) y el objeto(b) para 3-Step.

A lo largo de la metodología se ha explicado a grandes rasgos los métodos de desenvolvimiento generales, y la implementación que se realizara con la finalidad de comparar los resultados obtenidos con estos tres métodos sera con el objeto de la Figura 3.2, el cual no se considera una figura simple, pero tampoco una muy compleja al tener rasgos muy definidos pero con forma no muy uniforme, lo cual servirá para explicar y demostrar las deficiencias de los métodos sin el auxilio de algoritmos complementarios que mejoren la técnica implementada. Las imágenes que se usaron para implementar el método Spatial Phase Unwrapping con 3-Step, son las mostradas a continuación:

Para realizar el desenvolvimiento de la fase envuelta con el método Spatial, no requiere de algún plano de referencia como veremos mas adelante, basta simplemente con el conjunto de las 3 imágenes proyectada del objeto en la Figura 3.8b y aplicando la ecuación 3.18 podemos obtener los cambios de fase con la ayuda del gradiente y de esta manera detectar el orden de el mapa. A que nos referimos, examinando a detalle la Figura 3.9, vemos una secuencia de franjas blancas sobre un fondo gris, que podría representar las

franjas negras que se proyectaron en el objeto en un inicio, como se muestra los píxeles blancos representan una inconsistencia en los alrededores de ese punto, el cual indicaría que para desenvolver la fase envuelta correctamente en ese punto específico es necesario realizar un brinco positivo de 2π . Y el píxel negro al igual que el blanco representa una inconsistencia en el camino que se realiza como se muestra en la Figura 3.14, sin embargo, a diferencia del anterior este representa una cambio de fase en el mapa de orden negativo, que nos indicaría que en este punto para lograr desenvolver de manera correcta es necesario un brinco de -2π . Los píxeles de color gris simplemente representan que hay consistencia en sus alrededores y no es requerido un desenvolvimiento de la fase en estos puntos.

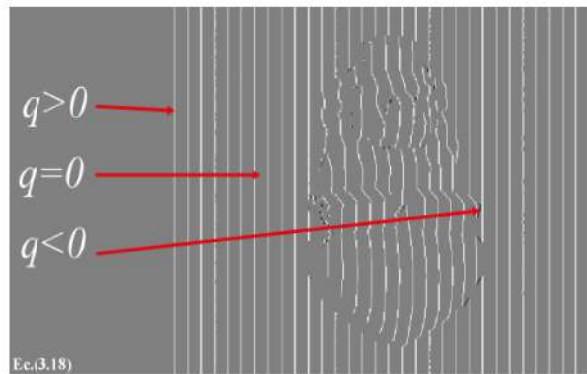


Figura 3.9: Residuos obtenidos con la ecuación 3.18, para la detección de los cambios de fase.

El procedimiento para desenvolver y obtener el resultado obtenido en la Figura 3.10 es diferente al planteado , ya que una vez obtenido el mapa de orden, procedemos a sumar el valor del escalón por 2π y la fase envuelta y de esta manera obtener el mapa absoluto como se muestra. Cabe recordar que este ultimo mapa esta representado como un plano inclinado, para ver el resultado en una visualización 3D es requerido restarle el mapa absoluto del plano de referencia sin el objeto para obtener el resultado mostrado.

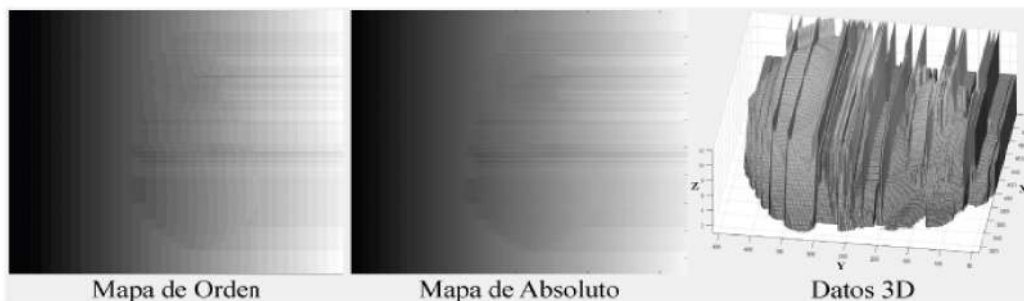


Figura 3.10: Resultados obtenidos implementando el método Spatial Phase Unwrapping en un objeto complejo.

Lo que se puede apreciar son varias franjas que atraviesan al objeto de lado a lado, estos se deben a los residuos que fueron calculados erróneamente causando una indeterminación y un cálculo erróneo en el mapa de orden, que al momento de desenvolver la fase se arrastra a lo largo del eje X, dado que el análisis se realiza de izquierda a derecha. Este tipo de errores se llega a tener en algoritmos similares como el de Itoh, el cual para mejorar su implementación ha requerido añadir algoritmos que complementen y corrijan el mal desenvolvimiento de la fase.

Para mejorar este principio de un desenvolvimiento de fase en 2 dimensiones se han propuesto diferentes métodos con la finalidad de complementar el principio básico.

3.2.3. Temporal phase unwrapping

El principio básico que ha analizado en la sección anterior para el desenvolvimiento de la fase. Sin embargo, en posteriores avances de la técnica surgieron 3 diferentes grupos de algoritmos de desenvolvimiento de fase temporal, llamados, algoritmos de *Multi – frequency*, *multi – wavelenth*, y *numbertheoretical*. Los últimos dos métodos vienen siendo una derivación del primero, aunque sus implementaciones proponen diferentes enfoques, aunque la idea común de estos tres algoritmos es desenvolver la fase con ayuda de una (o mas) adicional fase envuelta con diferente periodo de franjas. Para explicar la metodología detallaremos el algoritmo de *Multi – frequency*, el cual requerirá al menos dos diferentes proyecciones de franjas, una de franjas densas y otra de alta frecuencia.

Estos dos mapas de fase desenvueltos se recuperan por el algoritmo de desplazamiento de fase (ecuación 3.6)u otros enfoques de detección de la fase comprendidos en el rango de $-\pi$ a π . Los dos mapas de fase envueltos son presentados como ϕ_l y ϕ_h , con un longitud de onda de franja de λ_l y λ_h , respectivamente ($\lambda_h < \lambda_l$; el sufijo 'h' y 'l' denotan 'alta frecuencia' y 'baja frecuencia' respectivamente). Los mapas de fase continuos correspondientes a ϕ_l y ϕ_h son denotados como Φ_l y Φ_h respectivamente. Esto es fácil probar ya que las dos mapas de fase continuos deberán tener la siguiente relación:

$$\Phi_h(x, y) = \left(\frac{\lambda_l}{\lambda_h}\right)\Phi_l(x, y). \quad (3.19)$$

Multi-frequency(Jerárquico) temporal phase unwrapping

El método fue propuesto por primera vez por Huntley [45] y Saldner[88] en 1993, y posteriormente investigado por otros. En esta estrategia el patrón de franjas con diferentes densidad de franjas son proyectadas, y el patrón de franjas mas gruesas tiene solo una franja en total, de la cual la fase sin envolver (por que sus valores no exceden del rango de $[-\pi, \pi)$) se usa como información fundamental para desenvolver mejor la fase. Los otros mapas de fase se desenvuelven basándose en sus mapas de fase desenvueltos anteriores uno por uno según la relación de sus frecuencias o números de franjas. Dado que las fases se desenvuelven desde la capa mas gruesa hasta la mas fina, este método de clasificación también se conoce como enfoque de desenvolvimiento jerárquico'.

Se han propuesto varios algoritmos de desenvolvimiento de fase temporal multi-frecuencia para maximizar la precisión del desenvolvimiento y minimizar la cantidad necesaria de datos [88, 89]. Según el tipo de secuencia marginal utilizada, estos algoritmos se pueden dividir en varias categorías: secuencia lineal [88], secuencia exponencial [90], secuencia exponencial inversa [91], secuencia exponencial modificada [92], secuencia exponencial inversa generalizada [93], secuencia exponencial generalizada [94], etc. Desafortunadamente, para la mayoría de los algoritmos con el fin de lograr una buena precisión de reconstrucción 3D, el número total de mapas de fase relativa suele ser bastante grande (normalmente 5 mapas). En las aplicaciones en las que es necesario acortar el tiempo de medición, el número necesario de mapas puede reducirse a 2, como lo demuestra Zhao et al. [89]. Basado en la ecuación 3.19, las relaciones entre los mapas de fase (Φ_h o Φ_l) y los mapas de fase envueltos (ϕ_h o ϕ_l) se pueden escribir como:

$$\begin{aligned} \Phi_h(x, y) &= \phi_h(x, y) + 2\pi k_h(x, y), \\ \Phi_l(x, y) &= \phi_l(x, y) + 2\pi k_l(x, y). \end{aligned} \quad (3.20)$$

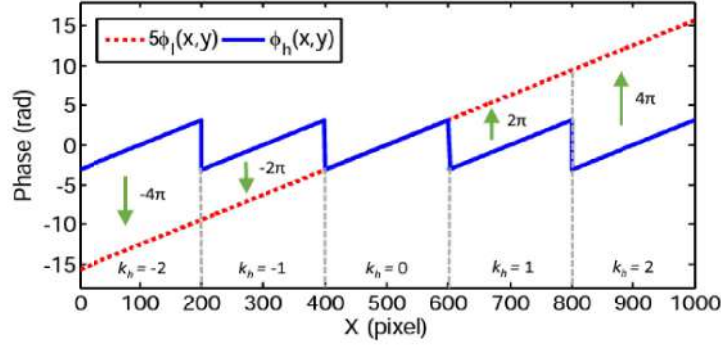


Figura 3.11: Ilustración 2D de two-frequency temporal phase unwrapping ($\lambda_l = 1000$ píxeles, y $\lambda_h = 200$ píxeles) [3].

Donde k_h y k_l son los respectivos órdenes de franjas enteras. En el desenvolvimiento de fase temporal de dos frecuencias, la distribución de fase de baja resolución ϕ_l se recupera mediante el uso de un conjunto de patrones de frecuencia unitaria y, por lo tanto, no se requiere desenvolvimiento de fase para ϕ_l , es decir, $\Phi_l = \phi_l$. Luego, refiriéndose a las ecuaciones 3.19 y 3.20, el orden de la franja k_h para cada píxel se puede determinar fácilmente:

$$k_h(x, y) = \text{Round}\left[\frac{(\lambda_l/\lambda_h)\phi_l(x,y) - \phi_h(x,y)}{2\pi}\right] \quad (3.21)$$

Donde $\text{Round}[\]$ denota obtener el valor entero más cercano. De esta forma, la fase de alta frecuencia ϕ_h se puede desenvolver, como se ilustra en la Figura 3.11. Dado que el número total de imágenes adquiridas se reduce considerablemente, este enfoque de dos frecuencias también se denomina enfoque jerárquico reducido [95]. En términos generales, el algoritmo se denomina enfoque de desenvolvimiento de fase multi-frecuencia en las siguientes comparaciones.

3.2.4. Geometric phase unwrapping

Para maximizar la eficiencia de la medición del PSP, Weise et al. [12] introdujo un novedoso método para el desenvolvimiento de la fase donde las restricciones geométricas entre las vistas (dos cámaras y el proyector) son usadas para recuperar la fase absoluta. No hay restricciones en la forma de la medición del objeto y no requiere una proyección extra.

El método es mejor conocido como Stereo phase unwrapping (SPU) ha ido incrementando sus aplicaciones para perfilometría de proyección de franjas de alta velocidad en tiempo real por que es capaz de recuperar la fase absoluta sin necesidad de adquirir un patrón adicional de franjas como es el caso de TPU. Basándonos en una predefinición de volumen, los máximos/mínimos artificiales del mapa de fase se puede crear solamente usando restricciones geométricas del sistema, permitiendo desenvolver la fase en un procedimiento de píxel por píxel. Para una medición precisa del objeto es recomendable aumentar la frecuencia de las franjas proyectadas, sin embargo, las ambigüedades de la fase se verán incrementadas lo cual hace que SPU no sera confiable. Es debido a esto que se han implementado técnicas complementarias para mejorar la robustez del método como el Adaptive Depth Constraint (ADC) que veremos a lo largo de esta sección, pero su aplicación aun requiere de mejoras.

La clave del problema es como elegir el rango de profundidad. Lo primero que se realizara sera un análisis de las características estáticas de la profundidad bruta del mapa, y analizar todos los datos del

volumen global de profundidad. Mientras tanto, algunos valores atípicos se pueden filtrar para obtener un mapa de profundidad refinado. Con base en este mapa de profundidad refinado, luego nos enfocaremos en el mapa de profundidad de un vecindario de cada píxel y creamos un mapa de volumen de profundidad por píxel para que sea una entrada de la restricción de profundidad en el próximo ciclo de medición 3D. La verificación de coherencia izquierda-derecha simplificada tiende a ayudar a garantizar la confiabilidad del mapa de profundidad sin procesar.

Principio básico del SPU

El sistema típico esta compuesto por dos cámaras y un proyector, donde el patrón de franjas son proyectadas en el objeto a medir por el proyectos, y deformadas por el objeto, para finalmente capturarlas por las dos cámaras. Basándonos en el principio básico de la Perfilometría de Proyección de Franjas, las expresiones utilizadas para las 3 capturas se basara en la ecuación 3.3, omitiendo por conveniencia las coordenadas de los píxeles (u, v) . Tomando la captura para un 3-Step, quedaría expresado cada ecuación con su cámara respectiva.

$$\begin{aligned}
 I_{L0} &= A_L + B_L \cos(\phi_L), \\
 I_{L1} &= A_L + B_L \cos(\phi_L + 2\pi/3), \\
 I_{L2} &= A_L + B_L \cos(\phi_L + 4\pi/3). \\
 I_{R0} &= A_R + B_R \cos(\phi_R), \\
 I_{R1} &= A_R + B_R \cos(\phi_R + 2\pi/3), \\
 I_{R2} &= A_R + B_R \cos(\phi_R + 4\pi/3).
 \end{aligned} \tag{3.22}$$

Donde $I_{L0,L1,L2}$ y $I_{R0,R1,R2}$ corresponden a los mapas de intensidad capturados de cada cámara, A_L y A_R son los mapas de intensidad media, B_L y B_R son los mapas de amplitud mientras que ϕ_L y ϕ_R son los mapas de fase absoluta deformada. Si las franjas son perpendiculares al eje horizontal del proyector, la coordenada horizontal del punto coincidente de (u, v) se puede determinar por:

$$u^P(u, v) = \frac{\phi(u,v)R}{2N\pi} \tag{3.23}$$

Donde R es la resolución horizontal del proyector, N es el periodo del numero de franjas. Entonces las coordenadas 3D (X^w, Y^w, Z^w) de (u, v) se puede recuperar por las siguientes expresiones:

$$\begin{aligned}
 Z^w(u, v) &= D^p(u, v) + \frac{E^p(u, v)}{F^p(u, v)u^p(u, v)+1}, \\
 X^w(u, v) &= G^p(u, v)Z(u, v) + J^p(u, v), \\
 Y^w(u, v) &= L^p(u, v)Z(u, v) + M^p(u, v)
 \end{aligned} \tag{3.24}$$

Donde $D^p, E^p, F^p, G^p, J^p, L^p$ son los parámetros de las matrices derivadas de los parámetros de calibración entre la cámara y el proyector [96]. El proceso de las ecuaciones 3.24 también se pueden implementar entre dos cámaras, siempre que se obtenga el punto de coincidencia en otra cámara. Sin embargo, debido a la limitación inherente de la función arco-tangente, solo la fase envuelta $\phi(u, v)$ puede obtenerse de las ecuaciones 3.22. Teniendo como resultado la ecuación ya presentada para el método 3.5.

La relación entre $\phi(u, v)$ y $\Phi(u, v)$ satisface igualmente la ecuación 3.6. En SPU, el k se recupera con restricción geométrica. Se utiliza por primera vez en el trabajo de Weise [12]. La Figura 1 muestra el principio básico de SPU. o^L es un punto arbitrario en la primera cámara (L) con la coordenada (u^L, v^L) y la fase envuelta $\phi^L(u^L, v^L)$. Para abreviar, o^L se utiliza para sustituir a (u^L, v^L) en algunas expresiones. Primero asignamos secuencialmente los enteros dentro del intervalo $[0, N-1]$ para $k^L(o^L)$, y anotar $k^L(o^L)$ asignado con diferentes valores como $k_0^L(o^L), k_1^L(o^L) \dots k_n^L(o^L) \dots k_{N-1}^L(o^L)$. Cada $k_n^L(o^L)$ corresponde a un

$\Phi^L(o^L, k_n^L(o^L))$ según la ecuación 3.6 $k_n^L(o^L)$ se simplificará como k_n^L . Entonces se puede derivar un total de N puntos 3D a partir de las ecuaciones 3.23 y 3.24, y el n -ésimo punto 3D se anota como o^L ; k_n^L con la coordenada $(X(o^L, k_n^L(o^L)), Y(o^L, k_n^L(o^L)), Z(o^L, k_n^L(o^L)))$

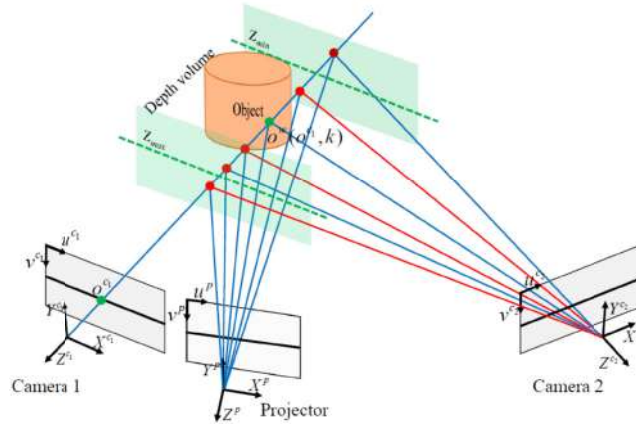


Figura 3.12: Diagrama de SPU y restricción de profundidad convencional (RPC) [95].

Todos estos puntos 3D se denominan candidatos 3D de o^L y el único punto coincidente 3D se incluye en estos candidatos. Las líneas azules del proyector en la Figura 3.12 denotan parte de los rayos con la misma fase envuelta $\phi^L(o^L)$ pero la diferente fase absoluta $\Phi^L(o^L, k_n^L)$. Estas líneas azules del proyector y la cámara L se cruzan en diferentes $o(o^L, k_n^L)$. Los candidatos 3D se proyectan en la segunda cámara R para obtener sus candidatos 2D correspondientes $o(o^L, k_n^L)$ con las coordenadas $(u^R(o^L, k_n^L), v^R(o^L, k_n^L))$. También hay un punto de coincidencia 2D dentro de estos candidatos 2D. o^L y su punto de coincidencia 2D debe tener propiedades similares, como la fase, la textura, etc. Tenga esto en cuenta, el candidato 2D con la fase envuelta más cercana a $\phi^L(o^L)$ se elige como punto de coincidencia. Una vez que se determina el punto de coincidencia 2D, se conocerá el orden de las franjas de o^L . Este es el principio básico de SPU.

Sin embargo, debemos considerar los efectos de la propiedad discreta de la cámara, la calibración imperfecta del sistema y el ruido en un experimento real. Todos estos efectos introducirán errores en nuestro sistema y disminuirán la similitud entre o^L y su punto de coincidencia 2D. Por otro lado, puede existir más de un candidato 2D que tenga la fase similar a $\phi^L(o^L)$ si se utilizan las franjas de alta frecuencia. En este caso, surgirán algunos errores de coincidencia durante el proceso de SPU. Podemos usar un esquema llamado verificación de consistencia de fase [84] para rechazar algunos candidatos 2D cuya diferencia de fase con $\phi^L(o^L)$ es mayor que un umbral. Como se muestra en la Figura 3.12, los candidatos 3D correspondientes a las líneas rojas de c^R se rechazan, mientras que los candidatos correspondientes a las líneas azules se invierten. Se requieren algunas técnicas adicionales para distinguir el punto coincidente de los candidatos reservados. Se han propuesto varios tipos de técnicas, y la revisión de estas técnicas se puede ver en la introducción. Solo se describirá en detalle la restricción de profundidad para explicar tanto algunos principios básicos como la motivación inicial de nuestro método.

3.3. Métodos y Materiales

Se desarrollará un entorno simulado para la adquisición de imágenes con software libre como Blender y configuraremos las condiciones físicas de la proyección de franjas donde experimentaremos con las posicio-

nes de las cámaras, la luz de la proyección y las configuraciones de renderizado. que nos permitirá obtener la mejor calidad de las imágenes generadas con un entorno completamente controlado.

La implementación del entorno emulado se desarrolló en Blender (<https://www.blender.org/>), que es un software libre para cualquier propósito, incluidos fines comerciales o educativos. Es una suite de creación 3D que nos permite modelar, ensamblar, animar, simular, renderizar, componer y seguir el movimiento entre otras opciones. En los últimos años, Blender ha demostrado ser una excelente opción para los desarrolladores, incluso para las empresas de desarrollo de juegos y animación de gran prestigio. Por lo tanto, su accesibilidad, flexibilidad y facilidad de uso nos brindan una excelente herramienta de desarrollo para estos entornos emulados que pueden generar imágenes sintéticas para nuestras bases de datos.

Para la extracción y generación de figuras 3D se utilizaron figuras desarrolladas por otros profesionales, las cuales fueron obtenidas de páginas como TurboSquid (<https://www.turbosquid.com/>). Los modelos TurboSquid son utilizados por desarrolladores y profesionales de todo el mundo. Esto tiene como objetivo ahorrar a los usuarios el tiempo de desarrollar modelos 3D grandes y complejos y, en cambio, permitirles dedicar su tiempo a otras áreas de su proceso de producción.

El entorno de simulación de desarrollo para el entorno emulado de Blender se implementó en una NVIDIA GeForce RTX 2060, 8 GB de RAM y i5-CPU 2.4Ghz.

La implementación de los modelos y los entrenamientos de las CNN aplicadas se desarrollo en Google Colab con la versión PRO, con esta versión tuvimos acceso a un Hardware remoto con un GPU modelo Tesla P100 con 16GB de RAM, y el uso de High-RAM que nos proporciona 27.3 GB, para la mayoría de nuestras aplicaciones fue mas que suficiente pero para la aplicación mas especifica e implementación de la red de Qian Jiaming et. al [5], era requerido mejores capacidades para obtener resultados mas óptimos durante los entrenamientos o incluso durante el procesamiento de la base de datos, donde se tuvieron diversas limitaciones. Con respecto a los algoritmos para el desenvolvimiento de la Fase se implementaron en Matlab.

La compatibilidad de tensorflow con el GPU que ocupes, tendrán que instalarse con las versiones respectivamente correctas de CUDA. En nuestra aplicación usamos tensorflow ver 2.2 y keras ver 2.3.1 con las que tenemos compatibilidad con el hardware que estamos ocupando

3.4. Generación de ambiente emulado

La colección de escenas que nos permite usar Blender está compuesta por cámaras, objetos 3D y luces. El entorno de desarrollo lo desarrollamos emulando un sistema de perfilometría de proyección de franjas, el cual consta de un proyector que está representado por una lámpara con la configuración que se describe a continuación, y una cámara con una distancia focal de 28 mm con una lente tipo 'Perspectiva' según la configuración de la cámara. , como se muestra en la Figura 3.13 la trayectoria óptica de la perfilometría de medición de fase, donde P es el centro de proyección del proyector, $C1$ es el centro de imágenes de la cámara, D es un punto en el objeto probado y área de proyección efectiva de 15,4 mm representada por l_0 . En nuestro experimento, la distancia d entre la cámara y el proyector es de 3 mm, y el plano de referencia se coloca frente al proyecto a unos 15 mm representado por l . El entorno emula el desarrollo de es por Blender ver 3.1.0.

El árbol de nodos genera un plano de proyección, mostrado en la Figura 3.14, seguido de un bloque con el nombre CLIP donde se establece una limitación de la proyección de las lámparas a un área específica. Debajo del bloque anterior tenemos un bloque SINE, este es el que genera la proyección de la franja, y es aquí donde podemos modificar el desfase según el N-Step que estemos implementando.

Los objetos virtuales que se usaron se pueden encontrar en la referencia del sitio que usamos o usar modelos 3d existentes con extensión .obj. También se pueden obtener de bases de datos existentes [97]

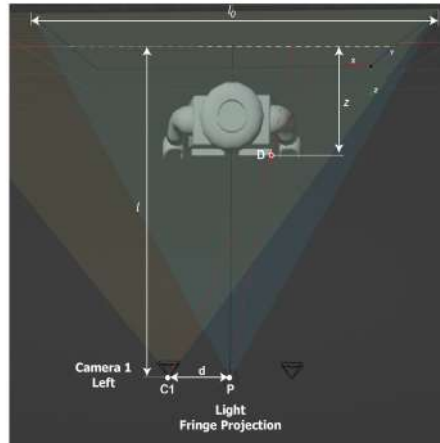


Figura 3.13: Ruta óptica de perfilometría de medición de fase en el entorno emulado.

como ModelNet, ShapeNet, ABC, Thingi10K, etc. Se realizó una variada selección de objetos simples a los más complejos con el fin de observar el comportamiento de las franjas proyectadas sobre estos objetos, como se muestra en la Figura 3.15.

Las imágenes capturadas son de 640×480 píxeles para generar imágenes con buena resolución, aunque el hecho de que Blender nos permita generar hasta 4k de resolución, esto no es recomendable dado que este tipo de imágenes son muy pesadas a la hora de entrenar redes profundas lo que complica la formación de nuestros modelos si no disponemos del hardware adecuado. Una baja resolución puede provocar una pérdida en la calidad de las franjas que se presenta en los resultados.

La generación de cada imagen renderizada tarda una media de 2,8 segundos con la resolución elegida, pero puede variar según el tipo de objeto y la configuración de la luz. Para una resolución 4k, el renderizado puede tardar hasta 60 segundos, lo que para la generación de grandes bases de datos puede resultar lento dada la cantidad de imágenes que se generan según el paso de desenvolvimiento que estemos realizando. Como se muestra en la Figura 3.16, la resolución del renderizado puede afectar la calidad de la proyección de las franjas, lo que puede representar una pérdida de información en la fase y una baja calidad en la fase desenvuelta. El número de franjas proyectadas en el plano es 32.

Se obtiene mejor calidad del plano con mayor número de franjas y así evitar el ruido de moiré que se genera, es recomendable aplicar un método de 12 pasos, en el que también se obtiene una mejor reconstrucción del objeto al desenvolverlo la fase [8].

En nuestro trabajo, para cada escena 3D del objeto, requerimos tres imágenes con franjas desplazadas $2\pi/3$ para hacer los cálculos. Además de cada escena, se trabajó con una sola perspectiva como se muestra: la Figura 3.17b representa el plano de referencia con la respectiva franja de proyección, y la Figura 3.17c consiste en tomar la escena considerando el fondo con el objeto y manteniendo las sombras naturales generadas por el ángulo de la cámara y el proyector. La Figura 3.17a es una imagen generada que se comparará con la intensidad de fondo generada por la fórmula (3.2).

Las imágenes obtenidas en la Figura 3.18 por la fórmula (3.2), que calcula la intensidad del Fondo. En el cuadro de zoom se puede ver un punto que indica la distancia z a la que se encuentra con referencia al plano ($z = 0$). Con esto podemos determinar que las luces generadas por las lámparas que emulan los proyectos, cuanto más cerca estén del proyector, mayor será la intensidad de luz que generarán en la región. Provocando la generación de un patrón de franjas que, cuanto mayor sea la iluminación del objeto, más resaltará.

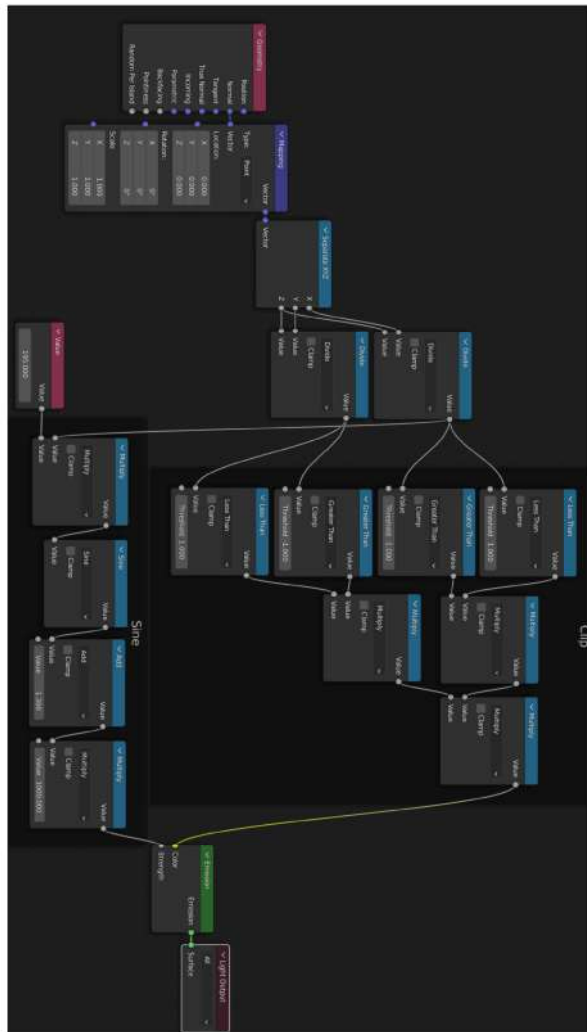


Figura 3.14: El árbol principal de nodos de composición de este sistema Blender.



Figura 3.15: Modelos de TurboSquid.

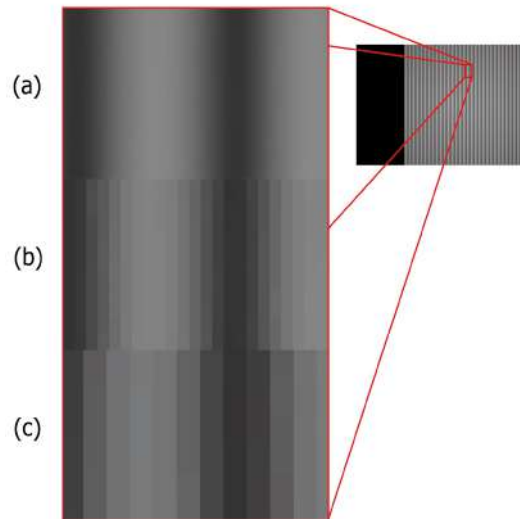


Figura 3.16: Proyección de franjas renderizada en diferentes formatos de imagen. (a) Se genera con una resolución de 4k (resolución 3840×2160), (b) representa la resolución utilizada VGA-SD (resolución 640×480), y (c) de un formato CIF más pequeño (resolución 360×240 resolución).

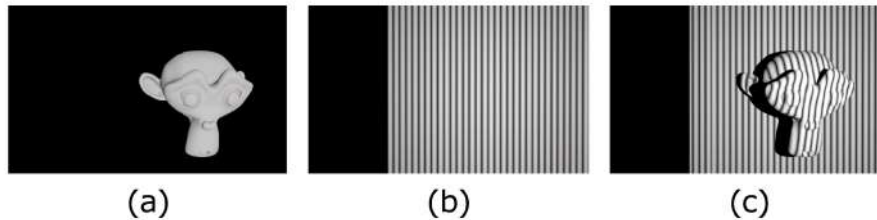


Figura 3.17: Ejemplos de los tipos de captura según el escenario simulado, (a) Intensidad de fondo, (b) Plano completo considerando el fondo con el objeto, (c) Plano completo con fondo y sujeto considerando sombras, (d) Plano omitiendo el fondo proyectando solo el sujeto.

Si es necesario modificar este parámetro, es recomendable modificar el valor en el árbol de nodos en lugar de modificar la potencia de la lámpara, ya que una mayor potencia de iluminación hará que el patrón resalte más y en la reconstrucción 3D del objeto, se generará un relieve ondulado.

3.4.1. Fase Envuelta con Imágenes sintéticas

En la sección anterior, comentamos sobre la resolución de renderizado, como se muestra en la Figura 3.16, para una gran cantidad de franjas proyectadas podemos perder calidad debido a los píxeles y generar un efecto de aliasing. Sin embargo, para nuestro estudio es importante determinar la calidad más óptima para la reconstrucción final del objeto, donde no se genere demasiada ambigüedad entre las fases, pero no requiera una resolución muy alta que a futuro complique el entrenamiento de nuestros modelos, y que tiene el fin de reducir la resolución de cada imagen.

Como se muestra en la Figura 3.19, tenemos las fases envueltas obtenidas con la Ecuación 3.5, podemos apreciar con mayor detalle lo que genera una baja resolución cuando calculamos la fase envolvente. La figura 8a muestra una mejor uniformidad en el contorno del objeto. Sin embargo, esta resolución será muy

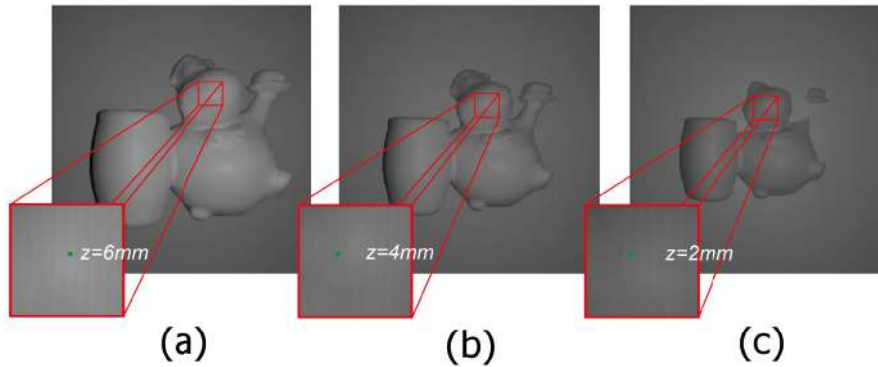


Figura 3.18: Fondo Intensidad que indica los puntos más altos con referencia al plano. (a) Representa el objeto más cercano a la cámara con una separación de $z = 6mm$ del plano, (b) una separación de $z = 4mm$, (c) una separación de $z = 2mm$.

pesada para el cálculo de la fase envolvente. En el caso de la Figura 8c, la generalización de los píxeles provocará una alta probabilidad de generar una ambigüedad en el salto de fase. Se determina de esta manera y en base a bases de datos similares [5] utilizar una resolución VGA como estándar.

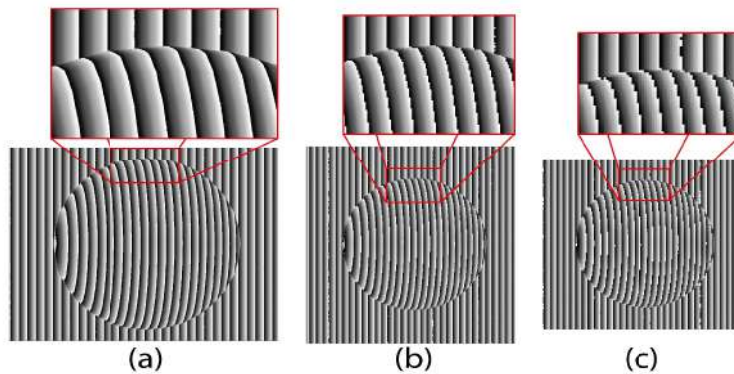


Figura 3.19: Fases envueltas obtenidas con imágenes renderizadas en (a) 4k (3840×2160 píxeles), (b) VGA-SD (640×480 píxeles) y (c) formatos CIF (380×210 píxeles).

3.5. Base de datos

Uno de los retos y dificultades de implementar técnicas de aprendizaje automático y aprendizaje profundo es tener un conjunto extenso de datos de entrenamiento, y las limitadas bases de datos públicas hacen necesario desarrollar herramientas que permitan emular su desarrollo [97], ya que hacerlas puede ser costoso debido al tiempo, personal y materiales necesarios para su desarrollo [98].

Es bien sabido que las limitaciones en inteligencia artificial son los datos limitados que son de acceso público, lo que limita las aplicaciones de redes neuronales convolucionales (CNN) para entrenar con las pocas bases de datos que se han desarrollado en este tipo de aplicación muy específica y que si han sido lanzadas públicamente, a la vez puede ser muy limitada su aplicación para la que se desarrolló y no considere las posibles variantes que pueden existir al realizar una medición con diferentes parámetros.

3.6. Base de datos pública

La base de datos que se implemento para entrenar nuestras arquitecturas de redes convolucionales, fue la desarrollada por Qian J. et al. [5], en la cual desarrollan una aplicación llamada Deep-learning-enabled geometric constraint con la finalidad de simplificar el desenvolvimiento de la fase y la obtención de mapa absoluto con técnicas de Deep Learning y con un solo disparo para la medición de la forma 3D final.

La base de datos presentada en la Figura 3.20 se describirá a continuación. Las imágenes (a) y (b) son las utilizadas para la CNN1, la primera representa la imagen capturada de una sola proyección de franjas sin sus desfases y la segunda es la misma proyección de franjas pero con su desfase para obtener el cos y sen que serán mejor conocidas como D (Denominador) y N (Numerador). Para las imágenes (c) y (d) son las utilizadas en la CNN2, en este caso tenemos 5 imágenes de entrada, las proyecciones de franjas simples son las capturadas por la cámara 1 y 2, y la que se encuentra debajo es el mapa de orden k generado con estas ultimas dos proyecciones de la referencia, y las proyecciones de franjas en la figura representa la captura tanto de la cámara 1 como la cámara 2 complementaria, la intención es obtener la mayor cantidad de referencias para que la red pueda aprender a predecir el mapa de orden (d), en el cual para generar el Ground-Truth se usaron varios métodos para obtener la mejor calidad.

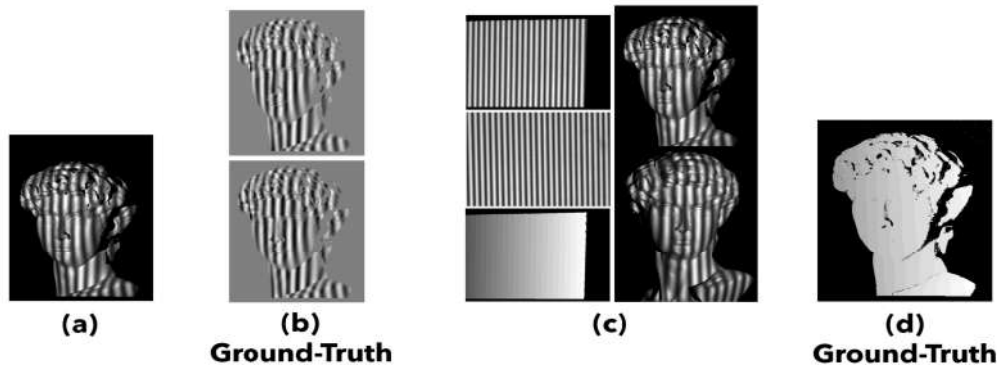


Figura 3.20: Ejemplos de imágenes que contiene la base de datos de Qian. (a) Las entradas de CNN1 (las imágenes marginales de 1000 escenas). (b) La salida de CNN1 (los términos numerador y denominador utilizados para calcular las fases envueltas). (c) Las entradas de CNN2 (los patrones marginales de 1000 escenas capturadas por dos cámaras, y la información del plano de referencia). (d) Las salidas de CNN2 (las órdenes de margen correspondientes a 1000 escenas).

Cabe mencionar que las recientes aplicaciones para la reconstrucción 3D mediante PPF, se han podido realizar punto a punto, sin la necesidad de realizar el desenvolvimiento de la fase envuelta y entrenando la red con una imagen con franjas y una salida en 3D. Sin embargo la precisión en la medición es muy importante en este tipo de metodologías y el realizar los procedimientos intermedios entre la proyección y la reconstrucción 3D ha dado mejores resultados en la calidad de la imagen resultante. La metodología que plantea tiene dos objetivos, el primero es realizar un preprocesamiento de las imágenes de entrada y la segunda es obtener la predicción de la imagen de entrada para generar el D y el N y de esta manera generar con la mejor calidad el mapa envuelto de la fase para el caso de la CNN1, con relación a la CNN2 la predicción que genera es el mapa de orden k el cual resulta ser muy importante y donde la mayoría de las investigaciones se han enfocada debido a que este mapa es el que nos permite desenvolver el mapa envuelto. Así como apreciamos en la Figura 3.21, los pasos desde la captura de una solo imagen de la cámara 1 y 2, así como las proyecciones en el plano de referencia que sirven de entradas a las CNN's. La predicción nos

genera las proyección de las franjas con el desfase \cos y \sin para obtener la fase envuelta de cada cámara (π_1), y en el caso de la CNN2 realizara la predicción del mapa de orden k de la cámara 1 que es nuestra referencia de medición, donde la cámara 2 es la complementaria para realizar la restricción de profundidad, y obtengamos la fase desenvuelta y mejor conocida como fase absoluta. Al final para obtener la reconstrucción 3D realizaremos la diferencia entre el mapa absoluta de la proyección con el objeto y el plano de referencia sin objeto.

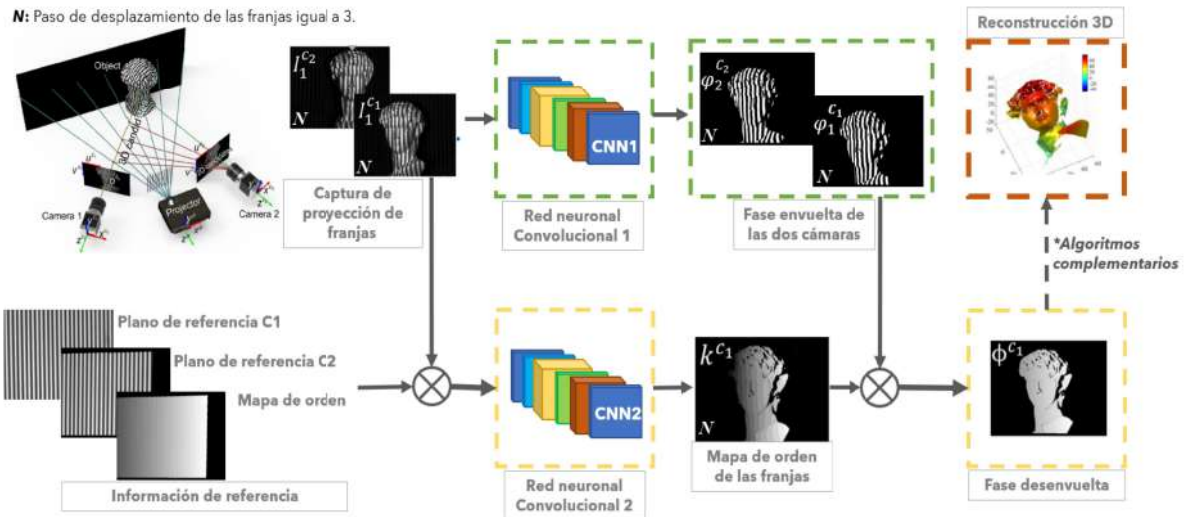


Figura 3.21: Metodología de reconstrucción 3D por el método SPU implementando la base de datos Qian J. et. al 3.20.

3.7. Base de datos desarrollada

Como se menciona anteriormente la base de datos de Qian J. es limitada a su aplicación. Sobre todo si no se desarrolla la implementación de las CNN's, no se puede utilizar ya que faltan imágenes para aplicar su metodología, siendo una de las base de datos mas completas que son publicas resulta difícil utilizarlas si no se desarrolla y para estudio de los métodos de desenvolvimiento es inservible. Generar estas imágenes en el mundo real resulta complicado por su calibración inicial, costoso por la cantidad de objetos que se requieren para generar una base de datos extensa. Además, de que se requiere tener experiencia en el tipo de objeto que se usa y el tiempo de desarrollo. Con la finalidad de simplificar este proceso y generar de manera mas fácil y sin costo alguno se desarrollo el ambiente emulado antes descrito y con ello una base de datos que nos ha permitido trabajar con diferentes objetos y configuraciones del sistema de manera accesible para cualquiera que cuente con una computadora.

La flexibilidad de uso de las imágenes que se generaron nos permite realizar diversas aplicaciones dentro del área de Deep Learning, por lo que no podemos generalizar sobre que imágenes son las de entradas o salidas por lo que describiremos las imágenes con las que cuenta nuestra base de datos generada, como se muestra en la Figura 3.22. (a) Es la proyección de las franjas sobre el objeto omitiendo el plano de referencia lo que nos da una mejor apreciación del objeto y evita la implementación de la eliminación del fondo por algoritmos de restricciones de profundidad o con eliminación por mascara generada. (b) Es la proyección de las franjas sobre el objeto manteniendo el plano de referencia detrás del objeto, es importante mencionar que

Blender nos permite configurar el comportamiento de la luz donde mantenemos la irreflexión de la luz que de forma natural deja proyectada una sombra en el plano, esta captura representaría la captura típica que se obtendría en el mundo real. (c) Esta imagen es similar a la anterior (b), con la diferencia omitir la sombra que genera el objeto en el plano por medio del software y que nos ayudara a obtener mejores mapas envueltos sin obtener datos que el algoritmo no sabe interpretar debido a que se pierde la referencia de la franja debido a la sombra. (d) Es la representación del Background intensity el cual solo se puede obtener o por una toma sin franjas y el manejo de luz adecuado, o por la combinación de las proyecciones de N-Step mediante la ecuación 3.2. (e) Con la finalidad de obtener el D y N sin requerir un preprocesamiento o entrenamiento de alguna red, se genero la imagen con una proyección de franjas con desfase \cos y \sin . (f) Y por ultimo para no aplicar algoritmo complementario que genere una mascara del objeto con la proyección, se genero una imagen que nos da una mascara del objeto de la escena emulada.

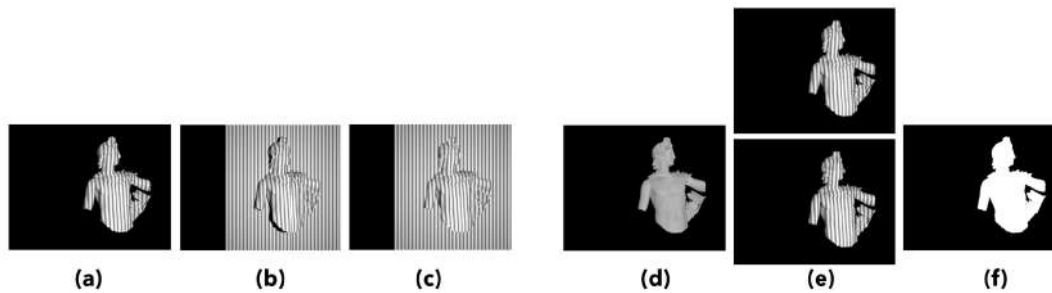


Figura 3.22: Ejemplos de imágenes del ambiente emulado generado que contiene nuestra base de datos. (a) Escenas con PPF sin Background. (b) Escenas con PPF Originales. (c) Escenas con PPF sin Sombras. (d) Background Intensity. (e) Los términos numerador y denominador utilizados para calcular las fases envueltas. (f) Mascara del objeto de la escena con PPF.

Las aplicaciones que nos ha permitido implementar nuestra base de datos han sido amplias y versátiles, y la generación del sistema emulado en Blender para la proyección de franjas nos ha servido para realizar ejercicios de configuración y experimentos para mejorar los algoritmos ya desarrollados. Es así que en la siguientes secciones detallaremos su implementación para nuestro trabajo.

3.8. Diseño de arquitectura de CNN

Por mucho, no hay un paradigma estándar para la apropiada selección arquitecturas CNN por que esto requiere un amplio conocimiento de la topología, métodos de entrenamiento y otros parámetros. En la practica, usualmente determinamos la estructura de las redes por diferentes evaluaciones de los modelos implementados que están disponibles, o comparando tareas similares de modelos específicos para entrenar con ellos con diferentes configuraciones de hiper-parámetros (capas de red, unidades neuronales, y funciones de activación) en una especifica base de datos de validación.

Sin embargo, la extensa cantidad de modelos de aprendizaje profundo limita la evaluación de solo algunos de los modelos mas confiables, lo que puede conducir a resultados subóptimos. Por lo tanto, debemos aprender a reducir rápida y eficientemente el rango de modelos disponibles para encontrar aquellos que probablemente tengan mejor desempeño en un tipo especifico de problema. Entrenar una CNN generalmente es laborioso y requiere mucho tiempo, y empeora mas con los ajustes repetitivos de la arquitectura de la red o los hiper-parámetros para evitar problemas de sobre-aprendizaje y convergencia.

En esta sección detallaremos los componentes y bloques que constituirán la arquitectura de la CNN para construir nuestros modelos y evaluar los resultados obtenidos por cada uno.

3.8.1. Sobre las arquitecturas típica para tareas de clasificación de imagen

Lo primero que habrá que analizar será la extracción de características las cuales solo se realiza en un espacio de baja dimensión, la computación y la complejidad espacial de las redes estructuradas de codificador-decodificador como lo son FCN y U-Net pueden reducirse mucho. Es por eso que se han convertido en la corriente principal para la segmentación y reconstrucción de imágenes [99].

El codificador suele ser una CNN clásica (Alexnet, VGG, Resnet, etc.) en el que se adopta la reducción de muestreo (agrupación de capas) para reducir la dimensión de entrada y generar mapas de características de baja resolución. El decodificador intenta reflejar el codificador para aumentar la muestra de estas representaciones de características y restaurar el tamaño original de la imagen. Es por eso que la forma de hacer el sobremuestreo es de gran importancia.

Cabe mencionar que aunque los métodos de muestreo ascendente tradicionales como la interpolación de vecino mas próximo, bilineal y bicúbica, son fáciles de implementar, los métodos de muestreo ascendentes basados en el aprendizaje profundo, por ejemplo: unpooling [71], transpose convolution [80], subpixel convolution [100], gradualmente se han convertido en tendencia. Todos estos enfoques se pueden combinar con el modelo mencionado anteriormente para evitar la disminución de la resolución y obtener una salida de imagen de resolución completa.

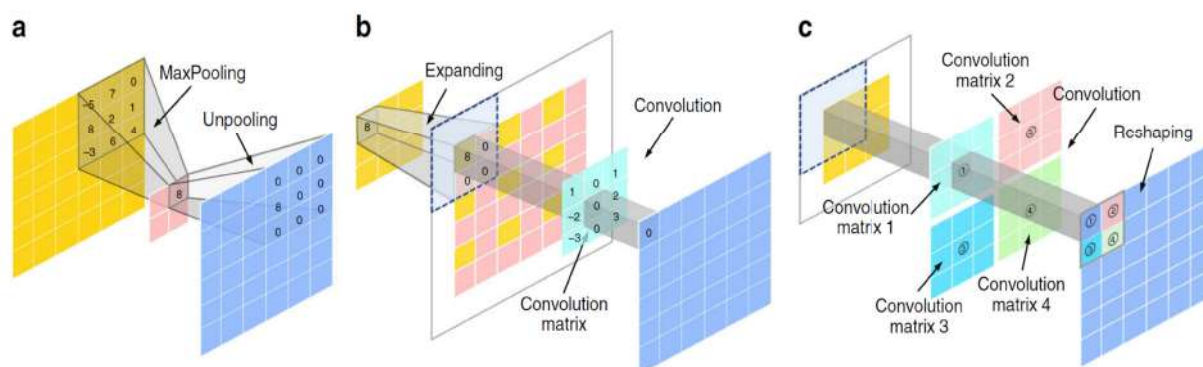


Figura 3.23: Tres metodos tipicos en CNN para el sobremuestreo (upsampling). a) Unpooling, b) Transposed convolution, c) Sub pixel convolution. [2].

- **Unpooling upsampling:** La desagrupación de muestreo superior revierte la agrupación máxima al recordar la ubicación de los máximos en las capas de agrupación máxima y en las capas de desagrupación copiar el valor exactamente a esta ubicación, como se muestra en la Figura 3.23a.
- **Transposed convolution:** Lo contrario de las capas convolucionales son las capas de convolución transpuestas (también malinterpretadas como capas de deconvolución), es decir, predecir la posible entrada basada en mapas de características de tamaño similar a la salida de convolución. Específicamente, aumenta la resolución de la imagen al expandir la imagen insertando ceros y realizando la convolución, como se muestra en la Figura 3.23b.
- **Sub pixel convolution:** La capa de subpíxeles realiza un muestreo superior al generar una pluralidad de canales por convolución y luego remodelarlos, como muestra la Figura 3.23c. Dentro de esta capa,

una convolución es en primer lugar aplicado para producir salidas con M canales, donde M es el factor de escala. Después de eso, se realiza la operación de remodelación (también conocida como reproducción aleatoria) para producir salidas con un tamaño M veces mayor que el original.

Los algoritmos en procesamiento de imágenes utilizados en metrología óptica comparten una característica común: pueden considerarse como un operador de mapeo que transforma el contenido de entrada de tamaño arbitrario en píxeles, salidas de nivel que encajan exactamente con las CNN's. En principio cualquier arquitectura puede utilizarse para realizar diferentes tipos de tareas que encontramos en metrología óptica. Dando una alternativa para procesar imágenes y obtener resultados similares o incluso superiores a los operadores de procesamiento de imágenes convencionales o sus combinados. Aunque también hay muchos otros factores deseables para sustituirlos, como la precisión, velocidad, generalidad y simplicidad. Siendo todos estos factores cruciales para permitir el rápido aumento del DL.

3.8.2. Estructura, diseño e implementación de una CNN

Nuestra CNN está construida en base a la arquitectura de U-Net, que es una red totalmente convolucional con un codificador y un decodificador. La Figura 3.24 muestra la arquitectura de la CNN, en la que agregamos capas de dropout antes de cada capa convolucional. La entrada de la red es una imagen con las proyección de las franjas sobre el objeto y entrenamos el modelo para que aprenda a predecir el numerador M y el denominador D . Para ser específicos, la imagen de franja de entrada en nuestros experimentos tiene una resolución de $W \times H$. Luego es manejada por el codificador, donde los canales C de características se extraen del patrón de franja ($C = 64$) por la capa convolucional que es activado por la unidad lineal rectificadora (ReLU) como función de activación, es decir, $ReLU(x) = \max(0, x)$. Luego, las características extraídas se muestrean a la mitad a lo largo de las direcciones x e y . Con los siguientes bloques convolucionales de muestreo descendente similares que extraen más detalles de la imagen a diferentes escalas, el codificador obtiene por fin características de alto nivel de $\frac{W}{16} \times \frac{H}{16} \times 16C$. Luego son manejados por el decodificador que realiza un muestreo ascendente para sintetizar y recuperar los resultados finales del tamaño original de la imagen de entrada. El muestreo ascendente se lleva a cabo mediante interpolación bilineal seguida de algunas capas convolucionales de hormigón. Se puede encontrar una conexión de salto en cada paso del decodificador, que se utiliza para concatenar la salida de las capas convolucionales con los mapas de características del codificador al mismo nivel. Las características en diferentes niveles se pueden recopilar al mismo tiempo con esta arquitectura. La última capa de nuestro CNN es una capa fusionada que incluye dos capas convolucionales. Uno se activa linealmente y aprende a generar el numerador y el denominador. El otro se activa mediante la función softplus (es decir, $softplus(x) = \log[\exp(x) + 1]$) y aprende a predecir la varianza del numerador y el denominador. La función de pérdida se muestra en la ecuación 3.25. Cabe señalar que solo necesitamos proporcionar los datos reales del numerador y el denominador y la varianza se puede aprender automáticamente.

$$L_{CNN} = -\frac{1}{K} \sum_{i=1}^K \log p = (y^i | x^i, w) + \frac{1}{K} KL[q(w) || p(w|X, Y)] \quad (3.25)$$

Donde el primer término es la función de verosimilitud logarítmica negativa mencionada y el segundo término es la divergencia de Kullback-Leibler (KL) que se utiliza para minimizar la diferencia entre la distribución simple seleccionada y la distribución real [101]. La tasa de caída se aprende con la convergencia de la divergencia KL . En cuanto al segundo término de L_{CNN} , dado que adoptamos una distribución simple $q(w)$ para aproximar la compleja $p(w|X, Y)$. La distribución $q(w)$ se puede escribir como:

$$q(w) = \prod_{j=1} q_{w_j^q(w_j)} \quad (3.26)$$

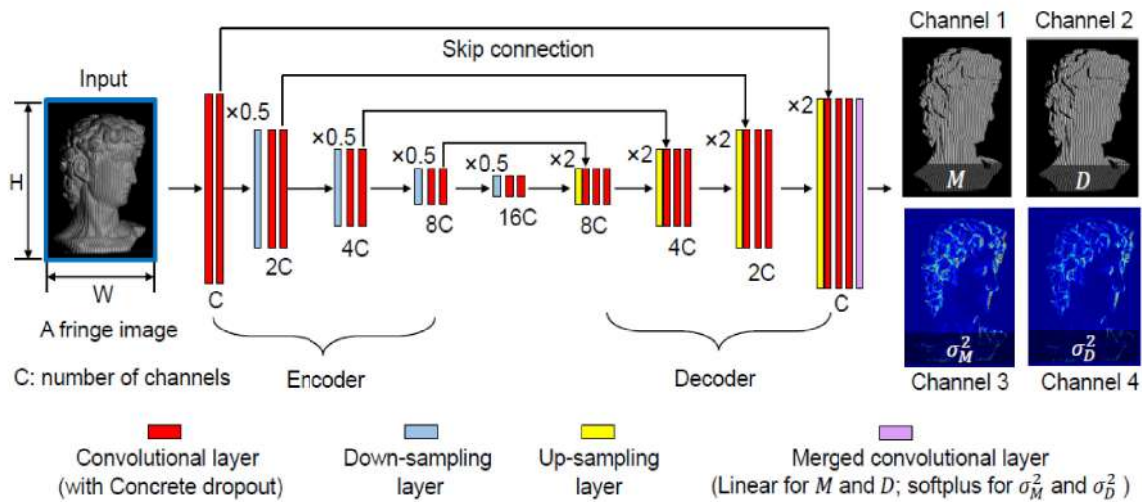


Figura 3.24: La arquitectura de la CNN1 propuesta que se construye modificando la U-Net. La entrada es un patrón de una sola franja [2].

Para entrenar la CNN, utilizaremos la base de datos de Qian y la complementaremos con nuestros conjuntos de datos desarrollados de manera sintética. Nuestro conjunto de datos de entrenamiento consistió en 984 patrones de franjas diferentes y 125 patrones de franjas pero con 3 desfases diferentes, en los que aproximadamente el 80 % de las imágenes se usaron para el entrenamiento y el resto para la validación. Las etiquetas de Ground-Truth para cada patrón de franjas se calcularon con el algoritmo PS de 12 pasos. Antes de ingresar a la CNN, el patrón de franjas de entrada se dividió por 255 para la normalización, lo que puede facilitar el aprendizaje para la red aunque provocara que los tensores generados ocupen mas memoria RAM. En la etapa de entrenamiento de CNN, se utilizó la estimación de momento adaptativo (ADAM) para ajustar los parámetros para encontrar el mínimo de la función de pérdida. En la implementación de ADAM, iniciamos con una tasa de aprendizaje de 10^{-4} . Lo redujimos por un factor de 2 si la pérdida de validación dejó de mejorar durante 10 épocas, lo que ayuda a que la función de pérdida salga de los mínimos locales durante el entrenamiento. Todo el procesamiento de datos y el entrenamiento y las pruebas de CNN se implementaron en Python utilizando Keras e implementándolo en Google Colab. El número de muestras de MC fue de 50. En cada capa de CNN, la tasa de dropout se inicializó aleatoriamente de acuerdo con una distribución uniforme [0.2, 0.6] y luego se actualizó dentro del rango [0.0, 1.0] durante el proceso de entrenamiento.

La adaptación de la Arquitectura mostrada en la Figura 3.24 correspondiente a la CNN1 se puede checar en la sección de Apéndice Figura 1 y la CNN2 corresponde a la Figura 2, la cual es diferente en cuestión de los datos de entrada y salida pero mantiene la misma estructura.

Estructura

En lugar de adoptar un esquema de aprendizaje de extremo a extremo que vincule directamente la imagen de la franja de entrada al mapa de fase de salida, aquí elegimos predecir los términos del numerador y el denominador de la función arco-tangente del mapa de fase a partir de un patrón de franja de entrada basado en dos consideraciones básicas:

1. La predicción de la fase a partir de la función arco-tangente evita las dificultades asociadas con la reproducción de cambios de fase abruptos y, por lo tanto, obtiene una estimación de fase de alta

calidad;

2. Dicha estrategia se refiere al modelo físico del método tradicional de cambio de fase, que elimina la influencia de la reflectividad de la superficie por la operación de división, haciendo que el modelo entrenado sea adecuado para objetos con superficies complejas.

Se demostrará que la red neuronal profunda desarrollada y entrenada puede realizar la tarea de demodulación de fase de una manera precisa y eficiente, utilizando solo un patrón de franja único. Por lo tanto, es capaz de combinar la fuerza de un solo cuadro de los métodos de demodulación de fase espacial con la alta precisión de medición de los métodos de demodulación de fase temporal.

Diseño e implementación

A continuación se muestra parte del código utilizado para realizar el entrenamiento de la CNN, comenzando por la definición de las capas que estructuran la arquitectura desarrollada partiendo del modelo U-Net. El código del modelo de la CNN1 y CNN2 se puede buscar en la sección de Apéndice.

```
1     def double_conv_block(x, n_filters):
2     # Conv2D then ReLU activation
3         x = layers.Conv2D(n_filters, 3, padding = "same", activation = "relu",
4                           kernel_initializer = "he_normal")(x)
5     # Conv2D then ReLU activation
6         x = layers.Conv2D(n_filters, 3, padding = "same", activation = "
7           relu", kernel_initializer = "he_normal")(x)
8     return x
9
10    def downsample_block(x, n_filters):
11        f = double_conv_block(x, n_filters)
12        p = layers.MaxPool2D(2)(f)
13        p = layers.Dropout(0.5)(p)
14    return f, p
15
16    def upsample_block(x, conv_features, n_filters):
17    # upsample
18        x = layers.Conv2DTranspose(n_filters, 3, 2, padding="same")(x)
19    # concatenate
20        x = layers.concatenate([x, conv_features])
21    # dropout
22        x = layers.Dropout(0.5)(x)
23    # Conv2D twice with ReLU activation
24        x = double_conv_block(x, n_filters)
25    return x
```

Figura 3.25: Código de la función de los bloques para Downsample y Upsample.

Y la parte que declara la compilación del modelo y la definición para el entrenamiento, y los valores graficados para observar su comportamiento.

Se muestra además los resultados obtenidos durante las primeras épocas las cuales son las esenciales para determinar si nuestro modelo está aprendiendo o se podría estancar y provocar un sobre-aprendizaje


```

1  modelB = SPU_CNN1_Unet2()
2  modelB.load_weights('/content/drive/MyDrive/Colab Notebooks/Tesis/
   TrainSetSPU_Unet/Model data weights/
   CNN2_model_Unet2_3L4C64C_weight_000-040epoch_ver18.h5')
3
4  PROJECT_PATH_MODEL="/content/drive/MyDrive/Colab Notebooks/Nombre de
   carpeta..."
5
6  os.chdir(PROJECT_PATH_MODEL)
7
8  ### Load weights
9  def schedule(epoch, decay=0.9):
10     return base_lr * decay ** (epoch)
11     base_lr = 1e-5
12     callback = [keras.callbacks.ModelCheckpoint(
13                 './content/drive/MyDrive/Colab Notebooks/Nombre de carpeta
14                 //weights_sigma.{epoch:02d}-{val_loss:.4f}.hdf5', verbose=1,
15                 save_weights_only=True),
16                 keras.callbacks.LearningRateScheduler(schedule)]
17
18     history =
19         modelB.fit({"left":XL_train, "order":XOO_train, "orderleft":XOL_train},
20                 y=y_train,
21                 batch_size = 8,
22                 epochs=40,
23                 validation_data=
24                 ({"left":XL_test, "order":XOO_test, "orderleft":XOL_test}, y_test
25                 ),
26                 use_multiprocessing=True)

```

Figura 3.26: Código usado para el entrenamiento de la red neuronal.

que no servirá para nuestros resultados. Es importante que el modelo se establezca como se muestra en la Figura 3.27a, a partir de la época 200 y que se tenga un aprendizaje rápido durante las primeras 100 épocas, de lo contrario en base a la experiencia los resultados que se han obtenido de modelos que no lograron estabilizarse rápido generaban patrones no deseados en el mapa de orden predicho por la red. En la Figura 3.27b vemos un periodo de 40 épocas correspondiente a la época 200 a la 240. En el caso de la red CNN2 se entreno por partes debido a que la RAM se saturaba, fue requerido entrenar por periodos no mayores a 40 épocas para evitar desbordar la memoria.

3.9. Metodología de trabajo implementando CNN para la reconstrucción de objetos 3D mediante el método SPU

Al tener ya nuestros pesos obtenidos con el entrenamiento de la red para obtener de manera directa los mapas de fase D y M, podremos continuar con la metodología propuesta como se muestra en la Figura 3.28. Para describir los pasos y las imágenes utilizadas así como los resultados dividiremos en 4 secciones el proceso para detallar cada sección. Se requerirá la imágenes que generara nuestro sistema SPU emulado en

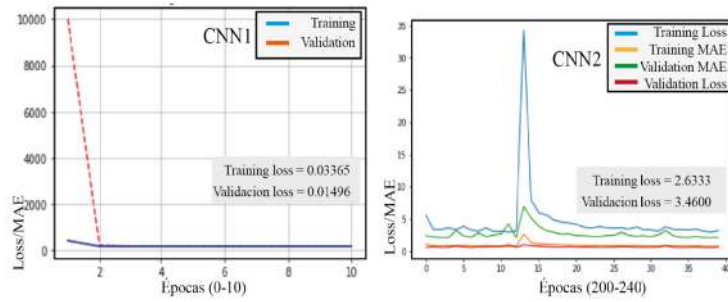


Figura 3.27: El comportamiento de aprendizaje, de los datos de perdida y la curva de distribución de perdida después del entrenamiento de la a)CNN1 y b)CNN2 .

Blender ya antes mencionada la cual servirá como entradas para nuestra CNN con la finalidad de realizar un pre-procesamiento a la imagen antes de aplicarle los algoritmos de desenvolvimiento de fase.

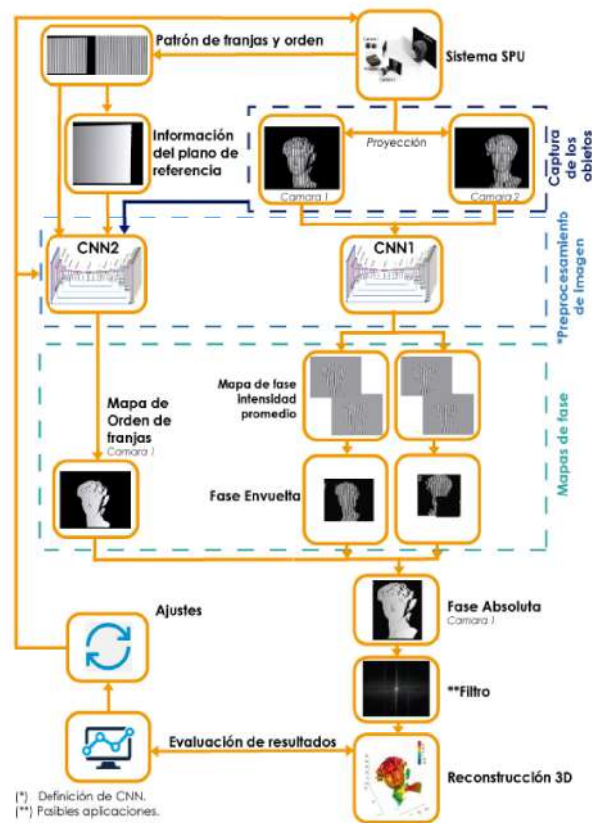


Figura 3.28: Metodología de reconstrucción 3D por el método SPU implementando nuestra base de datos generada con el sistema emulado en Blender 3.22.

3.9.1. Captura de los objetos y patrones de franjas

El sistema de nuestro prototipo desarrollado constara de una emulación de la proyección de franjas sobre un objeto, teniendo dos vistas que representaran cada una de las cámaras (L y R), Previamente se habrá de haber generado los patrones de franjas de cada cámara y haber generado un plano de referencia que sera el mapa de orden (k) de las franjas proyectadas sin ningún objeto.

Como observamos en la Figura 3.29 tenemos el objeto renderizado en Blender junto a las dos cámaras para obtener la vista del lado Izquierdo (L) y Derecho (R). Esto representaría el sistema SPU emulado para obtener las imágenes de la proyección de franjas sobre los objetos de manera digital y sin la compleja tarea de obtenerlas físicamente. Con esto obtendremos la proyección sobre el objeto de la cámara 1 y cámara 2.

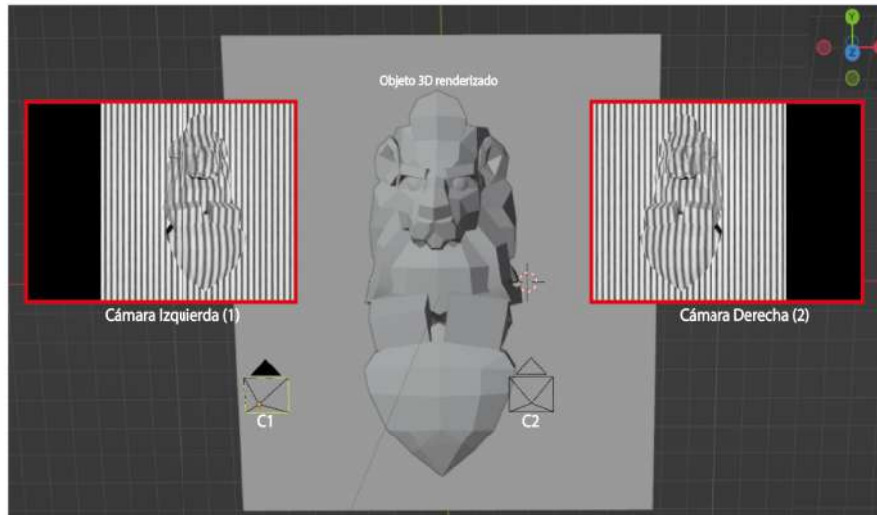


Figura 3.29: Capturas de la cámara 1 y 2 del sistema emulado en Blender.

Con los patrones de franjas y la generación de la información del plano de referencia, así como la captura de cada cámara tendremos la información de que sera pre-procesada por la CNN1 y CNN2 la cual se detallara a continuación.

3.9.2. Pre-procesamiento de imagen

El procesamiento de las imágenes de entrada con ayuda de las CNN nos servirá para dos cosas. Primero es obtener una mejor calidad en el desfase (CNN1) a partir de una sola imagen lo cual reducirá el tiempo que toma capturar cada una de los desfases de las proyecciones como un 3-Step y para un sistema estéreo es multiplicar este proceso por el numero de cámaras. Y segunda es predecir (CNN2) el mapa de orden de las franjas siendo este el proceso mas complejo y tardado que representa al desenvolvimiento de fase. Cabe mencionar que estos dos procesos con métodos tradicionales y sin algoritmos complementarios que ayuden en el desenvolvimiento de la fase del objeto no son suficientes para obtener la fase absoluta de manera correcta o con gran calidad, precisión y confiabilidad. Es por eso que se propuso un pre-procesamiento que ayude a mejorar la calidad de las capturadas previamente.

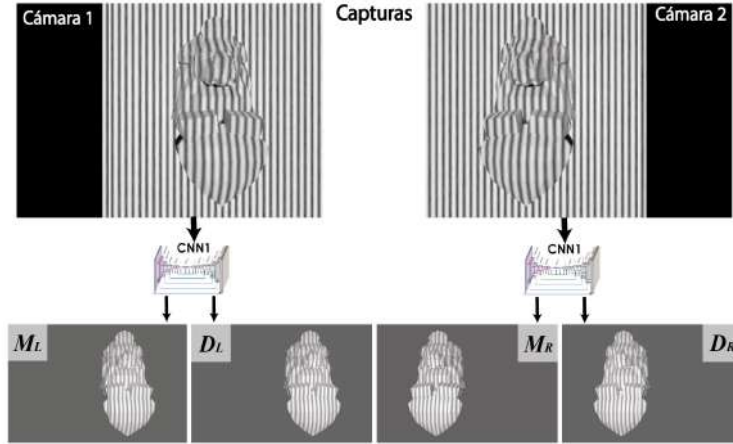


Figura 3.30: Implementación de CNN1 para la obtención de mapa de fase D y M, a partir de una sola captura de cada cámara.

3.9.3. Mapas de fase

Los mapas de fase se complementan con los dos resultados obtenidos de la CNN1 y CNN2. Siendo el mapa de fase de intensidad promedio o mejor conocido como D y M a lo largo del trabajo. Con este par de imágenes de cada cámara obtenemos la fase envuelta implementando el algoritmo de la ecuación 3.4, de esta manera obtendremos la fase envuelta. Como en el trabajo demostrado por Schimit [102], podemos expresar la ecuación 3.4 de la siguiente manera:

$$\phi = \arctan \frac{M}{D} = \arctan \frac{\sum_{n=0}^{N-1} I_{cn} \sin(2\pi n/N)}{\sum_{n=0}^{N-1} I_{cn} \cos(2\pi n/N)} \quad (3.27)$$

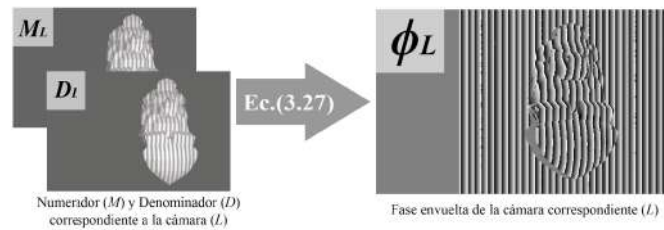


Figura 3.31: Numerador y Denominador de la salida de la CNN para la captura de la cámara Izquierda (L o 1) para calcular la Fase envuelta.

Recordemos que esta fase se encuentra expresada entre π y $-\pi$ y para lograr el desenvolvimiento requerimos el mapa de orden el cual hay muchos métodos. Sin embargo nuestra intención en obtener de una este mapa sin la necesidad de realizar algún método de desenvolvimiento de fase como el Espacial o Temporal que hasta cierto punto siguen resultado métodos complejos y que no son suficientes para desenvolver la fase.

Posterior a este punto es requerido implementar los algoritmo del método estéreo dado que tenemos dos imágenes, la izquierda y la derecha. La imagen de referencia sera la Izquierda y la Derecha solo nos servirá para remover algunas ambigüedades que se hayan filtrado al implementar el algoritmo.

3.9.4. Calibración de las cámaras del Sistema Estéreo

Un sistema estéreo al estar conformado por dos o mas cámaras es requerido generar los parámetros de calibración que nos indicaran la disparidad entre cada vista para ajustar la vista dimensional sobre los objetos, relacionando los puntos de una imagen con la otra. MATLAB posee una herramienta que nos permitirá genera estos parámetros. La aplicación como se muestra en la Figura 3.32 consta de la sección a), donde se cargan todas las imágenes tanto de la vista izquierda como derecha, automáticamente las relacionara de acuerdo al nombre que generemos por lo que se recomienda numerar cada imagen con su correspondiente así también podremos agregar mas o quitar imágenes que tengan un error promedio significativamente variado a los demás. La sección b) mostrara los puntos calibrados a partir de la mesa de ajedrez localizando y relacionando cada punto de las dos imágenes, y teniendo la orientación y punto de origen, es importante analizar cada imagen y determinar también con esta vista si la calibración se hizo correctamente o la imágenes puede meter ruido a nuestros parámetros. La sección c) nos muestra primero el conjunto de cada conjunto de imágenes relacionadas indicando su relación del error promedio, y después seguirán los demás conjuntos de pares, en este punto es importante mantener el menor error promedio y ademas que la relación entre los pares no varié demasiado entre la tendencia del error promedio general. Y por ultimo tenemos la sección d) esta esta dividida en dos parte pero la segunda se detallara mas adelante, la primera parte generara una imagen a partir de patrón centrado, es decir que nos mostrara el tablero de ajedrez como punto de origen a todas las capturas generadas por la cámara, deberemos tener en este gráfico solo la vista de un tablero y que no tenga desfase.

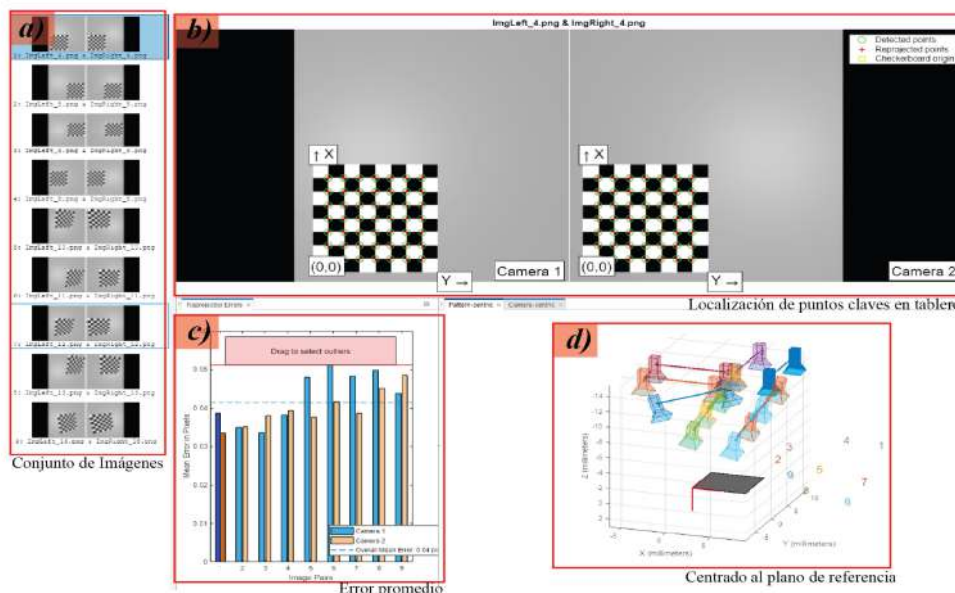


Figura 3.32: Numerador y Denominador de la salida de la CNN para la captura de la cámara Izquierda (L o 1) para calcular la Fase envuelta.

La segunda parte mencionada en la Figura 3.32d, seria la representación gráfica del centrado de las cámaras las cuales estarían observando todas las posiciones que tuvo el tablero de ajedrez, como observamos en la Figura 3.32, donde tenemos solo dos cámaras las cuales se encuentran posicionadas como en nuestro ambiente virtual (la distancia de 15mm del plano de referencia a la cámara y 8mm entre cámaras) y la vista renderizada de lado izquierda vista por la cámara 1

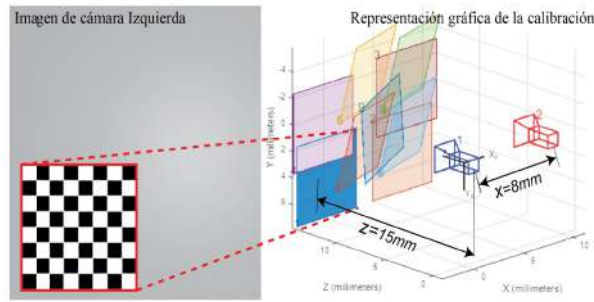


Figura 3.33: Numerador y Denominador de la salida de la CNN para la captura de la cámara Izquierda (L o 1) para calcular la Fase envuelta.

Para realizar la calibración para la obtención de los parámetros se recurrió a la creación de un tablero de ajedrez como objeto 3D en blender de tal manera que obtuviéramos sus parámetros de nuestro sistema emulado digital es importante menciona que para una relación con el mundo real se requerirá un escalamiento ya que el tablero tiene una dimensión de $8 \times 8 \text{ mm}$ teniendo un escalamiento de 1:10.

3.9.5. Reconstrucción 3D

Hasta este punto aun no se obtiene la reconstrucción 3D o la obtención de los datos 3D del objeto al cual se le proyectaron las franjas. Para ello requeriremos aplicar la ecuación 3.21, la cual hará una sumatorio de acuerdo al orden de la fase envuelta de la cámara Izquierda. Este desenvolvimiento generara una imagen de forma de rampa conocida como Fase Absoluta, donde se encontrara la información de nuestro objeto. Para obtener los datos de manera lineal o sobre nuestro plano es requerido restarle la información del plano de referencia que se genero desde un inicio.

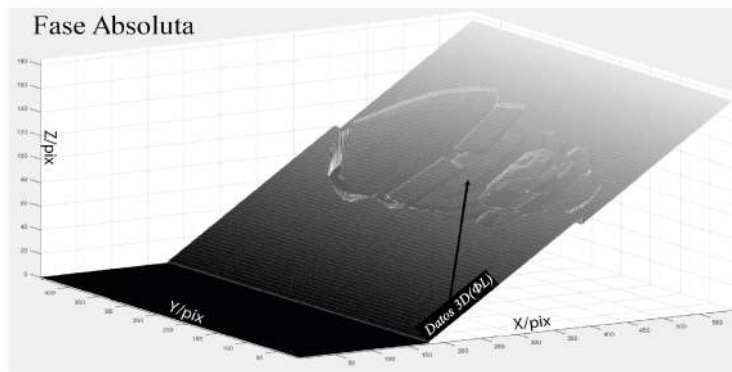


Figura 3.34: Visualización de la fase absoluta en 3D, en donde se encuentra en un plano inclinado debido al desenvolvimiento de la fase envuelta.

Se están proponiendo la implementación de filtros para mejorar la calidad del resultado y ayudarnos a mejorar la precisión o eliminar patrones no deseados como el ruido de Moiré el cual suele estar presente en este tipo de métodos y mas cuando son pocas las cantidades de imágenes generadas en un inicio siendo nuestro caso un 2-Step.

Como ultimo paso para ver la reconstrucción 3D del objeto sin el plano podemos aplicarle la resta de su mascara que convenientemente Blender nos ayuda a generar como se mostró en la base de datos que

generamos nosotros. Y como paso final queda analizar los datos como pueden ser cuantitativamente como cualitativamente sin embargo en nuestro proceso no es prioridad por el momento enfocarnos en la precisión si no en la calidad de los datos 3D obtenidos, y para fines demostrativos se mostraran datos cuantitativos que sirvan de comparación para trabajos futuros.

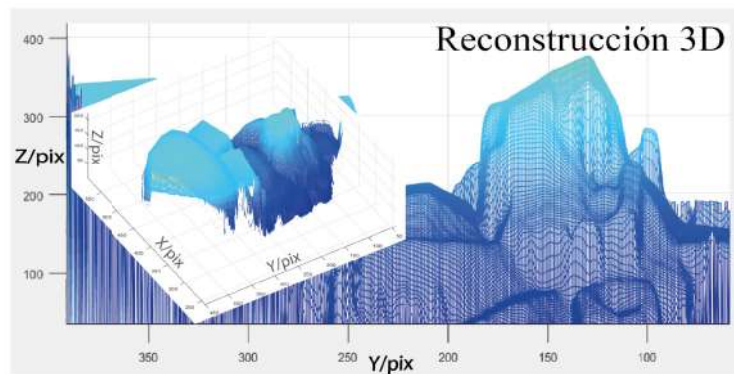


Figura 3.35: Reconstrucción 3D del mapa absoluto reacondicionado para la visualización lineal de los datos 3D obtenidos al desenvolver la fase envuelta.

Resultados y Discusión

Antes de realizar la evaluación de los resultados obtenidos, es importante mencionar que el Aprendizaje profundo en visión por computadora no es igual al Aprendizaje profundo en metrología óptica. El Aprendizaje profundo es en esencia el proceso de usar computadoras para ayudarnos a encontrar los patrones subyacentes dentro del conjunto de datos de entrenamiento. Ya que la información no puede nacer de la nada, las CNN no siempre pueden producir una solución correcta. En comparación con muchas tareas de visión por computadora, la metrología se enfoca más por la precisión, la confiabilidad, la repetibilidad y la trazabilidad [103]. Cuando se utiliza aprendizaje profundo para la inspección meteorológica óptica, se puede correr el riesgo de que un defecto en un componente sea suavizado y no se detecte por una CNN que está sobre-ajustada, lo que podría provocar una producción errónea en un proceso industrial. El éxito de las aplicaciones dependerá de las características comunes aprendidas y extraídas de las muestras de entrenamiento, lo que puede conducir a resultados insatisfactorios cuando se enfrente a muestras raras.

4.1. Resultados

4.1.1. Tiempos de renderizado

La generación de las imágenes con la proyección de franjas en los objetos, con respecto al mundo real tiene varios factores que se requieren ajustar para obtener la mejor calidad en la toma pero manteniendo la rapidez para capturar los otros desfases de las franjas proyectadas (Como el tiempo de exposición de la proyección y tiempo de espera para la captura entre cada desfase). Sin embargo nuestra captura de imágenes se han generado de manera sintética por lo tanto evaluaremos los tiempos de renderizado y la cantidad de imágenes que se requieren para cada técnica que se ha implementado. Tomaremos en cuenta el tiempo de renderizado por imagen y el tiempo total para generar todas las imágenes requeridas para obtener la reconstrucción.

Haciendo un análisis los tiempos de renderizado para generar reconstrucciones de la manera más precisa usando el método 12-Step, con el que podríamos obtener un Ground-Truth, obtendríamos un proceso muy largo ya que con un proceso automatizado que genere las imágenes en Blender tardaríamos al menos 1.5 minutos por cada objeto, simplemente es un proceso que otorga confiabilidad en la reconstrucción que buscamos obtener pero no optimiza el tiempo que toma en generarlas. La diferencia entre el 3-Step y TPU aunque son métodos similares el TPU requiere de la obtención de proyecciones de menor y mayor frecuencia de las franjas, incluso para mejores resultados entre mayor cantidad de frecuencias tomadas mejor el resultado obtenido. Y el método SPU con nuestra metodología solo requeriría la toma de dos capturas (D y

M) por cada cámara teniendo un total de 4 para lograr una reconstrucción 3D. En cuestión de tiempo no es mejor que el 3-Step pero el método SPU a demostrado tener mejores resultados en la reconstrucción final.

Tabla 4.1: Tiempos (segundos) promedios de renderizado de acuerdo a cada método y el tipo de objeto, a partir de nuestra base de datos de 125 objetos (No. de imágenes).

Tiempo de renderizado	Métodos tradicionales						Nuestro método	
	12-Step (x1)	Total (x12)	3-Step (x1)	Total (x3)	TPU (x1)	Total (x9)	SPU (x1)	Total (x4)
Objeto Simple	7.44	89.28	2.47	7.41	2.37	21.33	2.1	8.4
Objeto Complejo	9.32	111.84	2.63	7.89	2.61	23.49	2.77	11.08
Plano de referencia	6.18	74.16	2.39	7.17	2.28	20.52	2.14	8.56

4.1.2. Evaluación cualitativa de los datos 3D generados

Para corroborar la efectividad de las configuraciones de enfoque propuestas anteriormente, que consistían en una cámara (640×480 píxeles) y un proyector emulado por varias lámparas. En nuestro experimento se implementan patrones de franjas PS de 32 períodos. El tamaño efectivo de la proyección es de $15,4 \times 15,4$ mm. Evaluamos los resultados cualitativos de la configuración del ángulo entre la cámara y el proyector, y la evaluación de la obtención de una imagen de referencia que pueda servir como Ground-Truth [4].

Evaluación cualitativa del ángulo entre cámara y proyector

Para figuras más complejas en su forma, analizaremos dos perspectivas con la cámara, la luz de proyección seguirá manteniéndose en el centro y la cámara tomará dos posiciones en el entorno de simulación. Para la Figura 4.1, la posición de la cámara es de 3 mm en un ángulo de 0° con respecto al proyector, manteniendo ambas una distancia del plano de 15 mm y una distancia focal de 28 mm. Para la Figura 4.2 cambiamos la perspectiva para provocar lo que genera un ángulo muy grande entre la cámara y el proyector, la configuración modificó la posición de la cámara que ahora está a 7 mm con un ángulo de 12° con respecto y una distancia focal de 27 mm para capturar todo el marco de proyección [104].

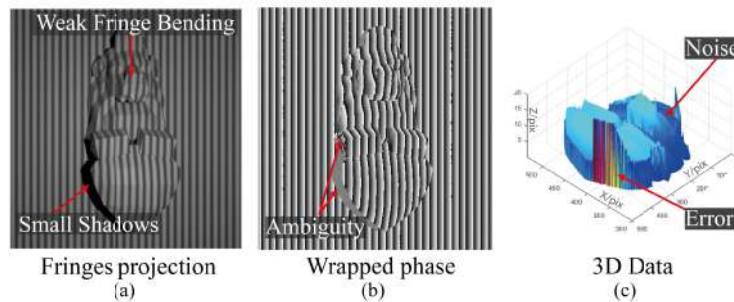


Figura 4.1: Triangulación de puntos característicos sin ángulo entre la cámara y el proyector. (a) Representa la captura de las franjas proyectadas, (b) la fase envuelta, (c) la reconstrucción de la fase absoluta [4].

Como se muestra en la Figura 4.1, para un ángulo de 0° entre la cámara y el proyector, al proyectar las franjas, las sombras naturales de la luz se forman en los modelos 3D, lo cual es conveniente para simular el mundo real y replicar un entorno similar a los problemas típicos de este método. Cabe señalar que el ángulo pequeño como se mencionó generará pequeñas sombras y ciertas características se verán como una

débil flexión de franjas en el modelo. Esto provocará que al calcular la fase envuelta obtendremos ciertas ambigüedades en la fase generadas por las discontinuidades de las franjas y al realizar la reconstrucción del objeto tendremos errores y algo de ruido por el tipo de triangulación entre un punto, cámara y proyector [4].

Sin embargo, a medida que aumenta el ángulo entre la cámara y el proyector, generará grandes sombras y una fuerte flexión de las franjas, así como una deformación del plano más pronunciada, como se muestra en la Figura 4.2a, por lo tanto, se generarán más ambigüedades en la fase envuelta (Figura 4.2b). y que se pueden eliminar con técnicas más avanzadas dejando grandes huecos ya que son datos perdidos. Para un ángulo entre 10° y 15° , es conveniente obtener un suavizado más natural en la reconstrucción 3D, sin embargo, como notamos en la Figura 4.2c que los Datos 3D que presentan mayores ambigüedades pueden generar errores durante la fase de desenvolvimiento.

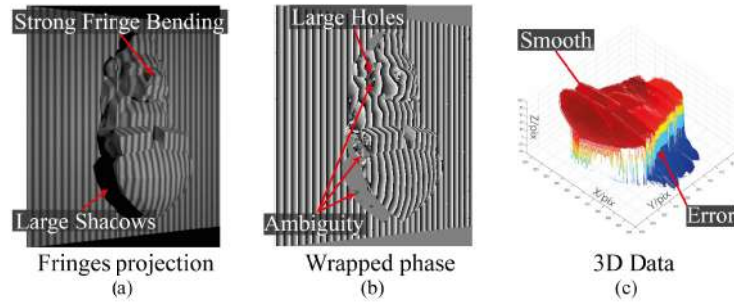


Figura 4.2: Triangulación de puntos característicos con ángulo entre cámara y proyector y gran proyección de sombras.(a)Representa la captura de las franjas proyectadas, (b) la fase envuelta, (c) la reconstrucción de la fase absoluta [4].

Evaluación cualitativa de la generación de datos 3D

Se presenta una propuesta con la que se pretende generar un Ground-Truth con el entorno simulado, se proyectó sobre dos esferas que se muestran en la Figura 4.3a, teniendo la reconstrucción 3D (Figura 4.3b) y nuestra imagen obtenida del emulador para obtener el Ground-Truth (Figura 4.3c). Evaluando las comparaciones entre los dos resultados, notamos la similitud en los resultados obtenidos para la forma de las esferas tanto del Ground-Truth sintético como de la reconstrucción de las esferas con imágenes sintéticas, el gráfico de la Figura 4.3d representa el corte en un punto central de las esferas referentes a Y [4].

En las reconstrucciones de los objetos que se realizaron se observó que la técnica que aplicamos en la configuración de los parámetros de Blender cuando se usaban objetos complejos como se muestra en la Figura 4.4, o con formas no uniformes como en la Figura 4.4a la generación del Ground-Truth (4.4b), no obtuvimos la reconstrucción correcta del objeto como lo hicimos en la Figura 4.4c. Y es en la gráfica de la Figura 4.4d, donde apreciamos que la saturación de luz del Ground-Truth establecía los colores en una escala de grises dentro de un rango, que variaba entre objetos. Y la mayor diferencia entre los objetos se encuentra en los límites del contorno que varían más en el objeto 3D en comparación con el Ground-Truth generado, que se muestra en la Figura 4.4e [4].

4.1.3. Entrenamiento, análisis y resultados

Durante la investigación de la literatura se encontraron diversos modelos de redes neuronales convolucionales que se ocupan actualmente, incluso redes las cuales usan bloques residuales para generar redes mas profundas que no sufren por degradación y perdida en extracción de características. Sin embargo, se

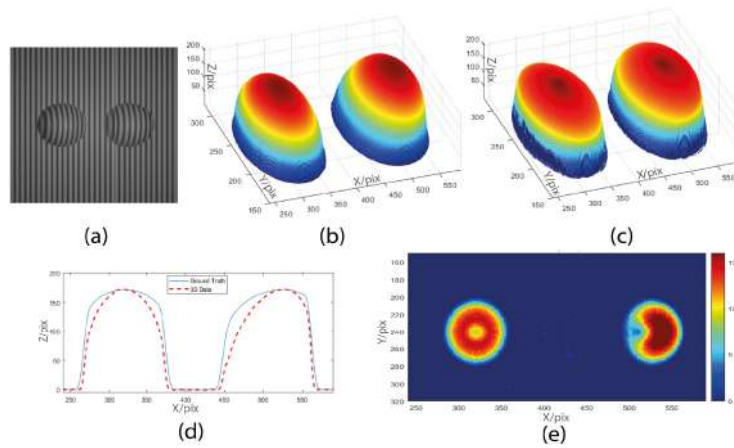


Figura 4.3: Resultados de la evaluación de la generación de Ground-Truth usando el emulador con un objeto 3D simple. Realizando la reconstrucción del Objeto (a), para obtener los Datos 3D (b) y el Ground-Truth (c), con un corte en su centro (d) y obteniendo la diferencia de error (e) entre ambas reconstrucciones [4].

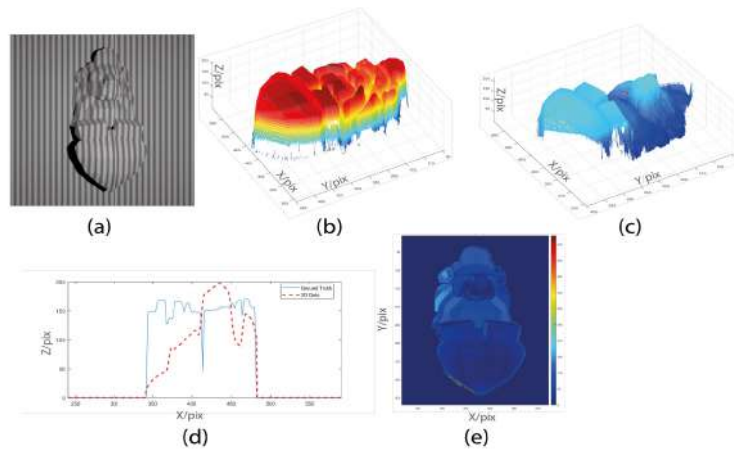





Figura 4.4: Resultados de la evaluación de la generación de Ground-Truth usando el emulador con un objeto 3D complejo. Realizando la reconstrucción del Objeto (a), para obtener los Datos 3D (b) y el Ground-Truth (c), con un corte en la punta de la nariz del león (d) y obteniendo la diferencia de error (e) entre las dos reconstrucciones [4].

han encontrado buenos resultados con redes mas simples como la FCN, SRCNN y la U-net siendo la ultima de las mas conocidas y utilizadas para las aplicaciones de metrología óptica. Algo que se menciona como conocimiento empírico [?], es que para un modelo de Aprendizaje profundo en metrología óptica si en las primeras épocas no tenemos una reducción del error de manera considerable recomienda modificar o cambiar Hiper-parámetros que nos ayuden a mejorar el aprendizaje del modelo, esto no significa que en las primeras épocas nuestro modelo sera capaz de predecir con precisión, es requerido un extenso entrenamiento para que pueda aprender de los diferentes patrones de los datos proporcionados.

En nuestra evaluación se entreno con tres modelos de CNN. FCN: es una excelente red para tareas de segmentación, dado que se quiere segmentar los patrones de las franjas en un Mapa de orden escalonado, se podría considerar como un excelente candidato, sin embargo, la estrategia implementada de proporcionar

la imagen de la cámara izquierda con los patrones de franjas, así como el mapa de orden y los patrones del plano de referencia, no generan el resultado esperado, segmentando la forma de la figura excluyendo los patrones de las franjas. SRCNN: esta red se considero dado que a través de sus capas mantiene la resolución original de la imagen de entrada y salida, lo cual esperaríamos obtener un análisis completo de cada detalle con una buena resolución, si embargo, la extracción de características en el resultado final solo mantiene la forma del objeto pero la proyección de franjas es excluida y refleja solo las franjas del plano de referencia. U-net: entre las redes mas usadas debido a el muestreo ascendente de un paso con varias capas complementarias nos permite extraer la mayor cantidad de características sin perder la información inicial, aunque el resultado pareciera el esperado a simple vista, es requerido aun realizar un extenso entrenamiento para mejorar la predicción del modelo.


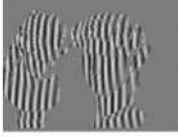


Tabla 4.2: Implementación preliminares para evaluación de resultados del modelo CNN2 (Entrenamiento de 20 épocas).

Modelo	MAE	Validación MAE	Tiempo de entrenamiento	Resultado preliminar
FCN	4.7981	8.2168	1 hr 4 min	
SRCNN	0.1107	0.1180	40 min	
U-net	0.9292	1.4878	56 min	

De acuerdo a la tabla 4.3, se presenta los hiper-parámetros, métricas y los resultados obtenidos así como el esperado. Comenzando con el modelo CNN1 en el cual se realizaron diferentes variantes de la red U-net con la finalidad de obtener una entrada y dos imágenes de salida con la apropiada relación de desfase que tiene un coseno y seno a partir de una sola proyección, el entrenamiento duro 16 horas y se contó con un dataset de 1125 imágenes, siendo 1000 de la base de datos publica de Qian y 125 tomadas de nuestra base de datos generada en el sistema emulado, una segmentación de 76/19/5 debido a que es de las relaciones mas utilizadas y dado que contamos con una base de datos grande, se entreno a través de 300 épocas en lapsos de 50 épocas debido a la limitante de hardware que se tenia y su implementación en Google Colab ante la necesidad de una RAM de al menos 64GB. El error absoluto promedio obtenido es realmente bajo lo cual si se observa los resultados tenemos una buena precision en la predicción de nuestra imagen M y D.

Relacionado al modelo CNN2 contamos con diversas modificaciones realizadas para adaptar el modelo a un mejor aprendizaje con toda la información posible como lo es un input de la captura de la proyección tomada por la cámara 1 (izquierda), la proyección de franjas sobre el plano de referencia que nos ayudara a tener la información base y el mapa de orden del objeto y del plano, es importante mencionar que la información de la cámara 2 se excluye ya que metía demasiado ruido al modelo el cual nunca lograba

Tabla 4.3: Métricas e Hiper-parámetros de los resultados finales de los modelos CNN1 y CNN2 .

Modelo	Training	Imágenes	Segmentación	Épocas	Learning rate	MAE	Validación MAE	Resultado	Ground-Truth
CNN1 Var.9	16 hr	1125	76/19/5	300	0.001	0.03365	0.01496		
CNN2 Var.18	62 hr	1125	76/19/5	250	0.001	2.6333	3.4600		

descender su error en las primeras épocas. Un factor importante es la normalización de los datos para los mapas de orden, esto con la finalidad de segmentar los escalones a un valor entero y que el modelo evite valores flotantes o intermedios, al final generara tensores mas pesados por la cantidad de información que poseemos al cambiar de un valor entero a flotante. Debido a este cambio nuestro entrenamiento duro 62 horas aproximadamente el cual fue dividido en varios entrenamientos de alrededor de 40 épocas debido a que un entrenamiento mas largo iba acumulando la información y mas tensores superando los 72 GB de RAM, siendo una de las mayores limitantes así como el GPU para modelos mas profundos. El learning rate se eligió de 0.001 tomando como estándar en la literatura, el cual recomiendan partir con este valor e ir ajustando de acuerdo a la necesidad y comportamiento del entrenamiento, en modelo CNN2 con un rate de 0.0001 provocaba que tardara mucho en aprender y dado que se recomendaba que nuestro modelo aprendiera rápido durante las primeras épocas, evitando una variación considerable para rate de 0.01. De las variantes realizadas a la red U-net en la cual se realizo la inclusión de bloques residuales en la parte de BONE, así como la estructura de datos concatenados en la entrada y una sola imagen de salida, el modelo U-net básico resulto tener las mejores métricas y resultados esperados. Sin embargo el MAE es de 2.633 representando un error alto dada la aplicación en la que se requiere precision y repetibilidad, para tener una reconstrucción confiable.

Aunque el Hardware actual no nos permite implementar redes mas profundas ya que nos limita a usar Batch pequeño de solo 1, lo cual no es recomendable en ningún sentido. Las áreas de oportunidad generadas y aprendidas con la implementación de diversos modelos y redes nos ha dado una perspectiva de hacia donde se puede realizar futuros desarrollos de modelos de aprendizaje para la metrología óptica.

4.1.4. Evaluación del método tradicional y nuestra metodología

Evaluación de la calidad de la reconstrucción

La calidad de nuestra reconstrucción final se vera afectada por la eficiencia y lo eficaz que pueda ser nuestro modelo para predecir el Mapa de Orden. Como observamos en la Figura 4.5, del lado izquierdo tenemos el mapa de orden calculado con los algoritmos de desenvolvimiento 12-Step que nos ha servido para encontrar los datos que usamos como Ground-Truth o como referencia ya que se obtiene la mejor calidad debido a la cantidad de imágenes usadas. Utilizando las ecuaciones presentadas en la sección 2 de Entrenamiento de Redes Neuronales, calcularemos el MSE y la relación PSNR. Con la finalidad de tener una métrica comparativa observamos que entre las dos imágenes a simple vista difieren en su estructura, la de la izquierda son escalones bien definidos, sin embargo, la derecha a pesar que sigue el mismo patrón de escalonamiento presenta un comportamiento sinusoidal.

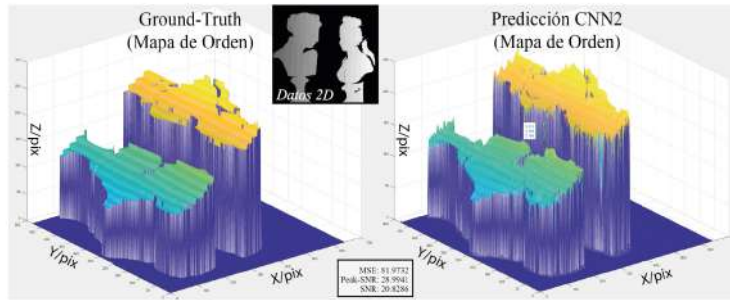


Figura 4.5: Comparación de Mapa de orden generado con 12-Step para referencia y Mapa de orden predicho por nuestro modelo de la CNN2.

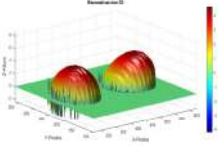
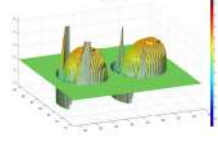
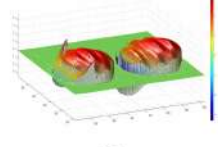
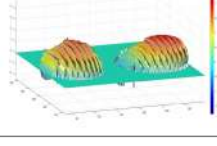
Un MSE de 81.9732 resulta un error muy bajo a comparación de un error total que podría llegar hasta los 48,000. Cabe mencionar que en términos de Metrología el error obtenido resulta ser alto cuando lo que se desea es tener una buena precisión que tenga un error que llegue a 0.0. EL PSNR obtenido es de 28.99 el cual es alto, esto podría provocar que se calcule erróneamente el paso de la franja causando un mal desenvolvimiento de nuestra fase. A continuación analizaremos los resultados obtenidos con nuestro método evaluando su desempeño para objetos simples y complejos.

Ya una vez analizado el background de nuestra predicción del modelo CNN2 para la obtención del Mapa de orden, compararemos los resultados obtenidos con los métodos tradicionales sin la implementación de algoritmos complementarios que ayuden a mejorar la calidad de la reconstrucción final. Comenzando con la Tabla 4.4, la cual esta conformada por el calculo del MSE y el PSNR que nos permitirán obtener una métrica de evaluación comparada con el 12-Step que sera nuestro método utilizado como Ground-Truth. EL MSE con menor indice estaría representado por el TPU y el peor por el 3-Step, la comparación entre estos dos métodos representaría el hecho de que el 3-Step al ser de los mas básicos y fáciles de implementar solo realiza la reconstrucción 3D con un solo patrón a diferencia del TPU que al menos requirió 3 diferentes patrones (frecuencia=1,8,57 Hz). Respecto a la métrica PSNR el que tuvo una disparidad menor fue el 3-Step y el que representa una mayor fue el TPU, caso contrario al anterior el 3-Step se desempeña de mejor manera al tener una aproximación mas cercana en sus datos al 12-Step,y por ultimo la métrica SNR tiene el caso similar al PSNR esto posiblemente al no reconstruir de manera eficaz la forma del objeto.

Entre todo nuestro método quedo en una posición intermedia esto debido a varios casos, aunque mantuvo la forma del objeto como el 3-Step, no fue capaz de predecir con exactitud el salto de cada fase presentando esas franjas periódicas muy pronunciadas que aunque se ven muy marcadas en el objeto en nuestra métrica es difícil apreciar una relación correcta que nos ayude a determinar si el objeto se reconstruyo de manera eficiente y eficaz con buena precisión y calidad, ya que existen diversos factores que determinaran esto. Además es importante resaltar que una de las ventajas que representa el SPU es la eliminación de los datos que se encuentren a cierta profundidad lo que puede ayudar a eliminar información no necesaria como la sombra del objeto que si observamos tanto en el 3-Step como en el TPU esos picos pueden distorsionar la información en la fase envuelta y provocar una reconstrucción errónea, la desventaja en estos dos métodos mencionados es que dependen del plano de referencia para saber en que Orden se desenvolverá de acuerdo al patrón y eso complica la eliminación de estos datos.

El ejemplo anterior mostrado en la Tabla 4.4, es un objeto representativo de una figura simple el cual es muy sencillo desenvolver con métodos básicos y algunos algoritmos extras para mejorar la reconstrucción final. Sin embargo, el reconstruir objetos complejos con patrones y formas no uniformes es lo que hace compleja las tareas de desenvolvimiento de la fase, como en la Tabla 4.5 analizaremos. Comenzando con

Tabla 4.4: Métrica de evaluación de el error promedio en la reconstrucción de cada método y tipo de objeto simple.

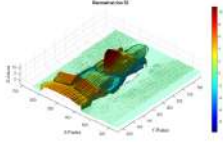
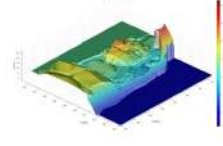
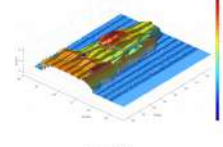
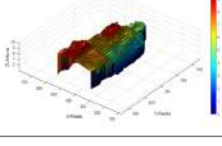
Método	Error (MSE)	Promedio	Relación señal-ruido máxima (PSNR)	Relación señal-ruido (SNR)	Reconstrucción 3D
12-Step	0.0	0.0	0.0	0.0	
3-Step	0.4234	3.7326	2.6375	2.6375	
TPU	0.01980	7.0339	5.9388	5.9388	
SPU	0.2731	5.6365	4.5414	4.5414	

el MSE el 3-Step presenta una métrica nada buena ya que es un error considerable, entre los factores que pueden estar ocasionando este error tan grande, no se refiere al objeto como tal si no a que en algún punto se perdió el Orden de la franja ocasionando un salto en el plano de referencia que como se observa de lado izquierdo de la figura tiene un color verde que representa un valor cercano a cero, y del lado derecho un color azul que representa un valor negativo por lo tanto al algoritmo del 3-Step tuvo complicaciones al perder el paso en la fase. Y al final podemos ver representado este plano negativo en las métricas del PSNR y SNR al tener una base de comparación de cero contra el valor negativo del plano de referencia.

Para el método TPU tenemos las mejores métricas de los tres métodos implementados, quizás solo batallando con la reconstrucción del plano de referencia el cual tiene una posición negativa lo cual también da sentido al valor negativo del PSNR que se obtuvo, por que en cuestión de altura y forma mantiene lo mas cercado al objeto. El contar con varios patrones de alta y baja frecuencia ayuda al algoritmo a determinar no solo los patrones complejos si no a determinar de una mejor manera la profundidad del objeto.

Y por ultimo el método SPU al igual que el objeto simple presenta los mismo detalles, el desfase en el Mapa de orden provoca unas franjas periódicas al patrón proyectado pero mantiene una representación gráfica muy similar al 12-Step sin embargo al determinar las alturas de los datos 3D la visión estereoscópica no fue suficiente para determinar y a su vez proporcionar información adicional a la cámara principal (Izquierda) para lograr una reconstrucción mas precisa

Tabla 4.5: Métrica de evaluación de el error promedio en la reconstrucción de cada método y tipo de objeto complejo.

Método	Error (MSE)	Promedio	Relación señal-ruido máxima (PSNR)	Relación señal-ruido (SNR)	Reconstrucción 3D
12-Step	0.0	0.0	0.0	0.0	
3-Step	73.6705	-13.0076	-18.6729		
TPU	2.6627	-4.2533	1.4120		
SPU	6.2177	-7.9363	-2.2710		

Evaluación del tiempo de reconstrucción

Se realizaron varias reconstrucciones, tomando como referencia dos tipos de objetos. Un objeto simple representado por dos circunferencias y un objeto con patrones mas complejos en sus estructura y texturas rectas como lo es el León. Esta selección nos ayudara a determinar que tanto es afectado los algoritmos cuando hay mas complejidad para desenvolver la fase.

En la siguiente tabla 4.6 tenemos los tiempos que se obtuvieron realizando las reconstrucción a partir de las imágenes ya obtenidas de nuestros sistema emulado. Es interesante como el tiempo del método 12-Step resulta ser mas rápido que métodos como 3-Step y TPU, sin embargo la desventaja la encontraremos en el proceso de obtener estas imágenes ya que nos tomara de 89 a 111 segundos, en cambio los otros método solo requieren de 8 a 23 segundos para la obtención de sus imágenes. Es esta una de las razones de por que se busca mejorar las técnicas como 3-Step para realizar una reconstrucción mas rápida pero que sea igual en calidad y precisión que un 12-Step el cual es uno de los métodos mas utilizados para la obtención de una referencia como Ground-Truth.

Tabla 4.6: Tiempos (segundos) de reconstrucción de cada método y tipo de objeto.

Objeto	12-Step	3-Step	TPU	SPU
Par de esferas	74.314	166.993	218.621	3.0819
León	89.191	144.354	227.643	4.044

En cuanto a nuestro método aunque resultado mejor en la reconstrucción rápida aun no es suficientemente

capaz de obtener una buena calidad en nuestra reconstrucción como se observa en la tabla 4.5 o 4.4. Se logra el objetivo de reconstruir rápidamente el objeto dado que este método es utilizado con este objetivo pero es requerido seguir usando algoritmos complementarios para mejorar los resultados.

4.2. Importancia/Impacto

Nuestro trabajo en términos generales deja la base de la aplicación de varios métodos de reconstrucción de imágenes basados en la proyección de franjas, así como un entendimiento amplio y básico para la aplicación de Deep Learning en la metrología óptica. Proporcionando además la generación de base de datos amplias para este tipo de aplicaciones, generadas de manera digital y que son útiles para nuestras investigaciones futuras y nos da un panorama mas amplio a nuestras aplicaciones y desarrollo de algoritmos complementarios a las técnicas vistas a lo largo del trabajo. Esto no solo beneficia de manera personal nuestras investigaciones si no que tiene impacto en varias áreas como lo son:

4.2.1. Impacto Social

El primer impacto social lo podemos ver y hasta el momento sera académico, dado que hasta el momento nuestras aplicaciones todavía se encuentran en fase de investigaciones, sin embargo en un futuro podfan ser las siguientes.

1. Agilidad en las investigaciones académicas en la UAQ relacionadas el tema.
2. Generación de la base de conocimientos para la trazabilidad de los trabajos anteriores.
3. Digitalización de objetos del mundo real en los nuevos ambientes digitales, con una gran precisión y calidad se podrán tomar en un futuro.

4.2.2. Impacto Ambiental

Un impacto ambiental resulta ser el no consumir o comprar objetos que solo serán usados una sola vez en nuestra investigación y que al final por la variedad de formas que se requiere genera un cumulo de material en nuestro espacio de trabajo. Ahora todos los objetos que se ocupan son digitales y no requieren de un tiempo y consumo de energía para su desarrollo ya que son descargados de fuentes gratuitas.

4.2.3. Impacto económico

La generación de imágenes sintéticas de la proyección de franjas generara un ahorro económico en investigaciones futuros con respecto a lo siguiente:

1. Materiales: En cuestión de búsqueda y compra de objetos que sean candidatos para proyección de franjas.
2. Equipo: No se requiere cámaras y proyectores de alta gama para generar imágenes de buena calidad.
3. Tiempo: Lo cual resulta conveniente dado lo complicado que resulta ajustar las cámaras y proyectores, así como la toma de las proyecciones de cada objeto y cada desfase.
4. Personal: No se requiere de varias personas para generar una base de datos extensa como lo era generarlas de forma manual.

4.3. Publicaciones

Durante el periodo de investigación de la tesis se desarrollaron los siguientes trabajos con el numero de referencia ORCID: <https://orcid.org/0000-0002-8006-1641>.

Siendo de autoría propia y sirviendo como trabajo complementario a la tesis el siguiente.

- Martínez-Suárez, V.M., Pedraza-Ortega, J.C., Salazar-Colores, S., Espinosa-Bernal, O.A., Ramos-Arreguin, J.M. (2022). Environment Emulation in 3D Graphics Software for Fringe Projection Profilometry. In: Mata-Rivera, M.F., Zagal-Flores, R., Barria-Huidobro, C. (eds) Telematics and Computing. WITCOM 2022. Communications in Computer and Information Science, vol 1659. Springer, Cham. https://doi.org/10.1007/978-3-031-18082-8_8

Y colaborador en el siguiente trabajo.

- Espinosa-Bernal, O.A., Pedraza-Ortega, J.C., Aceves-Fernández, M.A., Martínez-Suárez, V.M., Tovar-Arriaga, S. (2022). Adaptive Based Frequency Domain Filter for Periodic Noise Reduction in Images Acquired by Projection Fringes. In: Mata-Rivera, M.F., Zagal-Flores, R., Barria-Huidobro, C. (eds) Telematics and Computing. WITCOM 2022. Communications in Computer and Information Science, vol 1659. Springer, Cham. https://doi.org/10.1007/978-3-031-18082-8_2

Los dos trabajos publicados y presentados para el congreso en "The 11th International Congress in Telematics and Computing" WITCOM 2022.

4.4. Trabajo a futuro

Trabajos a futuro quedan planteados de la siguiente manera, tratando de abarcar las áreas de oportunidad generadas por la tesis desarrollada:

- Generación de base de datos con mayor cantidad de objetos y ampliando el tipo de paso generado hasta N-Step.
- Automatizar la generación de imágenes mediante Blender.
- Generación de otros ambientes emulados para diferentes tipos de luz estructurada.
- Implementación para reconstrucción de objetos en tiempo real.
- Implementación de las bases de datos desarrolladas en técnicas de Inteligencia Artificial.

Entre muchos otros trabajos en los cuales ya se han planteado para continuar la investigación.

Conclusión

El alcance de las pruebas realizadas durante este trabajo es la evaluación de las imágenes generadas por el entorno emulado de un sistema FPP con el que podemos obtener un método fiable para técnicas más avanzadas y que nos permite generar grandes conjuntos de datos fácilmente y sin invertir muchos recursos en la generación para la implementación de técnicas de Deep Learning, las cuales han demostrado ser una herramienta capaz de desarrollar la fase de forma rápida y robusta [2].

Evaluando que si bien el resultado obtenido mostrado en la Figura 4.3, es similar considerando la aplicación de un método de 3 pasos. Dada la diferencia de error que existía en la métrica MSE donde para las dos esferas (Figura 4.3) el resultado fue 72.1329 y para el León (Figura 4.4) fue 682.6826, donde el error se incrementó considerablemente y observando los resultados obtenidos con objetos 3D más complejos como la Figura 4.4 y esto se puede apreciar en la Figura 4.4d ya que se perdía por completo la figura con respecto a Ground-Truth que saturó los colores en el eje Z .

La implementación de técnicas como el método de 12 pasos para obtener un Ground-Truth de mayor calidad, el cual es más confiable para generar la reconstrucción de objetos para nuestros modelos de entrenamiento [105] seguirá siendo la opción mas confiable para obtener repetibilidad en nuestros resultados.

A pesar de la limitación de no poder obtener los datos 3D de la reconstrucción para usarlos como Ground-Truth, el entorno emulado cumple con las características de un sistema FPP, mostrado en la Figura [4.1,4.2], donde se mantienen los efectos característicos de la proyección de franjas y la relación con el ángulo de captura de la imagen. También es interesante que el entorno de Blender nos permite controlar la luz y las sombras generadas por este efecto, incluyendo la eliminación del fondo que se usaría en el futuro para técnicas más avanzadas de un sistema FPP.

Muchas de las investigaciones realizadas no profundizan en los entornos emulados que generan para sus aplicaciones, generalizando su construcción y no detallando sus características, lo que lleva a una extensa investigación para poder hacer uso de sus sistemas. De la misma forma, en el caso de las implementaciones de aprendizaje profundo, se requieren muchas pruebas para obtener los resultados que presentan y es hasta entonces que sus datos pueden ser modificados y adaptados a nuestras aplicaciones. El entorno emulado que se generó nos ha permitido realizar pruebas específicas en nuestras investigaciones sin la limitación de buscar y probar con datos de otras investigaciones que, además, no están hechas para ninguna condición ambiental. La facilidad para obtener imágenes sintéticas del entorno emulado ha permitido un estudio más profundo del PPF y la generación de bases de datos más adaptadas a los métodos implementados como Spatial Phase Unwrapping, Temporal Phase Unwrapping y Stereo Phase Unwrapping.

La limitante en estos sistemas digitalizados seguirá siendo la potencia gráfica que se posea para hacer uso del software, también lo es el tiempo que se requiera para la generación de la base de datos dependerá

de la tarjeta gráfica que se tenga y como sucede en los sistemas físicos de PPF se generan algoritmos complementarios para obtener las diferentes proyecciones de los patrones automáticamente.

Con respecto a los resultados obtenidos con las técnicas tradicionales sin algoritmos complementarios y la implementación de CNN's, resulta conveniente la forma que simplifica el trabajo e implementación acortando tiempos y procesos para obtención de imágenes y la generación de la reconstrucción de los objetos en forma de datos 3D. Aunque el tiempo de desarrollo de la arquitectura, base de datos y entrenamiento, resulta compleja y con tiempos extensos de desarrollo, nos permite obtener un modelo que no solo servirá para reconstruir objetos estáticos si no que por la velocidad y generación simple de imágenes que se ocupan en el algoritmo podemos implementarlo en técnicas de tiempo real para la reconstrucción de objetos en movimiento. Quitando carga de procesamiento al procesador para la generación rápida de datos 3D.

Nuestra Hipótesis la cual era implementar un método de desenvolvimiento de fase estéreo para reconstruir un objeto con datos 3D a a partir de datos 2D sustituyendo algoritmos complementarios como el ADC y implementando redes neuronales se ha cumplió parcialmente. Analizando los resultados obtenidos nuestro objetivo principal era mejorar la eficiencia del método SPU, con la finalidad de utilizar un método que nos permitiría utilizar una mínima cantidad de recursos, en términos de Deep Learning resulta contraproducente dada la cantidad de datos que se requirieron para realizar los entrenamientos con la base de datos, sin embargo, en términos algorítmicos hemos tenido una mejora en cuestión de recursos utilizados ya que disminuimos el tiempo de procesamiento a las imágenes mediante algoritmos complementarios ya que dejamos de analizar los datos y lo único que se realiza es la implementación de filtros que nos ayudan a pre-procesar la información antes de aplicar los métodos de PPF así como también reconstruyendo el objeto con una menor cantidad de imágenes.

Anteriormente se menciona que nuestra hipótesis se cumplía parcialmente, esto debido a que efectivamente nuestro método resulto mas eficiente en términos de recursos utilizados (Computo, imágenes, algoritmos), pero la poca precisión en la predicción del mapa de orden provoca una baja calidad en la reconstrucción final. Así teniendo una buena eficiencia en el resultado pero con poca eficacia.

Se han ubicado las áreas de oportunidad para el método desarrollado detalladas mas adelante en trabajos a futuro. En términos generales es requerido y como se detallo respecto a la calidad de la imagen, un equipo de computo mayor es vital todavía para seguir desarrollando modelos que nos ayuden a predecir los datos para una correcta reconstrucción del objeto. Esto se debe en primera con relación a los entrenamientos de las CNN's, al ser imágenes de una calidad de 640x480 pixeles es un formato que se puede considerar grande y dado que se requiere una normalización de los datos la memoria RAM que se va requiriendo conforme pasa cada época del entrenamiento va aumentando limitándonos en varios factores como la cantidad de épocas de entrenamiento antes de llenarse la RAM, un Batch muy pequeño de 2 y la cantidad de imágenes para el entrenamiento, validación y pruebas, las cuales son de mucha importancia ya que entre mayor cantidad de patrones de los objetos se pueda tener nuestro modelo sera capaz de predecir un mapa de orden y tener una mejor calidad en la reconstrucción final. Y segunda a que nos limita en el diseño de las CNN. En la literatura se ha observado que redes mas profundas han tenido mejores resultados pero esto conlleva a la necesidad de mayores recursos computacionales.

Bibliografía

- [1] D. Malacara, “Chapter 14. phase shifting interferometry,” *Optical Shop Testing*, pp. 547–655, 2007.
- [2] C. Zuo, J. Qian, S. Feng, W. Yin, Y. Li, P. Fan, J. Han, K. Qian, and Q. Chen, “Deep learning in optical metrology: a review.,” *Light Sci Appl*, vol. 11, 2022.
- [3] Z. Chao, H. Lei, Z. Minliang, C. Qian, and A. Anand, “Temporal phase unwrapping algorithms for fringe projection profilometry: A comparative review,” *Optics and Lasers in Engineering*, vol. 85, pp. 84–1033, 2016.
- [4] M.-S. V.M., P.-O. J.C., Salazar-Colores, S. Espinosa-Bernal, O.A., and R.-A. J.M., “Environment emulation in 3d graphics software for fringe projection profilometry,” *In: Mata-Rivera, M.F., Zagal-Flores, R., Barria-Huidobro, C. (eds) Telematics and Computing. WITCOM 2022*, vol. Communications in Computer and Information Science, vol 1659. Springer, Cham, 2022.
- [5] J. Qian, S. Feng, T. Tao, Y. Hu, Y. Li, Q. Chen, and C. Zuo, “Deep-learning-enabled geometric constraints and phase unwrapping for single-shot absolute 3d shape measurement.,” *APL Photonics* 5, 046105 (2020), 2020.
- [6] Y. Li, J. Qian, S. Feng, T. Tao, and C. Zuo, “Stereo phase unwrapping method based on feedback projection.,” *Proc. SPIE 11552, Optical Metrology and Inspection for Industrial Applications VII*, 1155209, 2015.
- [7] S. V. der Jeught and J. J. J. Dirckx, “Real-time structured light profilometry: a review,” *Optics and Lasers in Engineering*, vol. 87, pp. 18–31, 2016.
- [8] S. Feng, Q. Chen, G. Gu, T. Tao, L. Zhang, Y. Hu, W. Yin, and C. Zuo, “Fringe pattern analysis using deep learning.,” *Adv. Photon. 1(2) 025001.*, 2019.
- [9] G.-I. C.A. and A. O. Noé, “One shot profilometry using a composite fringe pattern.,” *Optics and Lasers in Engineering.*, vol. Volume 53, pp. Pages 25–30, 2014.
- [10] S. Feng, Q. Chen, G. Gu, T. Tao, L. Zhang, Y. Hu, W. Yin, , and C. Zuo, “Fringe pattern analysis using deep learning.,” *Adv. Photonics 1, 025001.*, 2016.
- [11] Z. C., F. S., H. L., T. T., Y. W., and C. Q., “Phase shifting algorithms for fringe projection profilometry: A review.,” *Optics and Lasers in Engineering, United States: N. p.*, 2018.
- [12] T. Weise, B. Leibe, and L. V. Goolr, “Fast 3d scanning with automatic motion compensation.,” *IEEE Conference on Computer Vision and Pattern Recognition.*, pp. 1–8, 2007.

- [13] T. Tao, Q. Chen, S. Feng, J. Qian, Y. Hu, L. Huang, and C. Zuo, "High-speed realtime 3d shape measurement based on adaptive depth constraint.," *Opt. Express* 26, 22440–22456, 2018.
- [14] P. Carré, "Installation et utilisation du comparateur photoélectrique et interférentiel du bureau international des poids et mesures," *Metrologia*, vol. 2, pp. 13–23, 1966.
- [15] A. Alvarado, C. Pedraza, R. Arreguin, E. Gorrostieta, and S. Tovar, "Gradient index optics design and tolerancing.," *Ph.D. Dissertation, Institute of Optics, University of Rochester, Rochester, NY.*, 1973.
- [16] J. Bruning, D. R. Herriott, J. E. Gallagher, D. P. R. A. D. White, and D. J. Brangaccio, "Digital wavefront measuring interferometer for testing optical surfaces.," *Lenses, Appl. Opt.*, 13, 2693, 1974.
- [17] W. J. C., "Use of an ac heterodyne lateral shear interferometer with real-time wavefront correction systems," *Appl. Opt.*, vol. 14, 2622, 1975.
- [18] J. G. W., D. C. Leiner, and D. T. Moore, "Phase-locked interferometry," *Proc. SPIE*, vol. 126, 152, 1977.
- [19] H. J. W., J. E. Lefebvre, and C. L. Koliopoulos, "Real-time atmospheric compensation," *JOSA*, vol. 67, 360, 1977.
- [20] B. J. H., "Fringe scanning interferometers, in optical shop testing," *D. Malacara Ed., 1st Edition, Wiley, New York*, 1978.
- [21] S. K. D., "Real-time interferometer," *Opt. Eng.*, vol. 18, 648, 1979.
- [22] M. N. A., R. Nelson, and S. Holly, "High-performance real-time heterodyne interferometry," *Appl. Opt.*, vol. 18, 1797, 1979.
- [23] S. J., R. Burow, K.-E. Elssner, J. Grzanna, R. Spolaczyk, and K. Merke, "Digital wave-front measuring interferometry: Some systematic error sources," *Appl. Opt.*, vol. 22, 3421, 1983.
- [24] C. K., "Phase-measurement interferometry techniques.," *Elsevier Science Publishers.*, vol. in: Progress in Optics Vol. XXVI, E. Wolf Ed., Amsterdam., p. 349–393, 1988.
- [25] M. D., "Analysis of the interferometric ronchi test.," *Appl. Opt.*, vol. 29, 3633, 1990.
- [26] L. K. G. and B. Oreb., "Design, assessment of symmetrical phase-shifting algorithms," *J. Opt. Soc. Am. A*, vol. 9, 1740, 1992.
- [27] S. J., O. R. Falkenstoerfer, H. Schreiber, A. Zoeller, and N. Streibl, "New compensating four-phase algorithm for phase-shift interferometry," *Opt. Eng.*, vol. 32, 1883, 1993.
- [28] S. Y., "Phase stepping: A new self-calibrating algorithm," *Appl. Opt.*, vol. 32, 3598, 1993.
- [29] S. Y., "Design of algorithms for phase measurements by the use of phase stepping," *Appl. Opt.*, vol. 35, 51., 1996.
- [30] S. Y., "Additive noise effect in digital phase detection," *Appl. Opt.*, vol. 36, 271–276., 1997.
- [31] S. Y., "Phase-shifting algorithms for nonlinear, spatially nonuniform phase shifts: Comment," *J. Opt. Soc. Am. A*, vol. 15, pp. 1227–1233, 1998.

- [32] P. D.W., “General methods for generating phase shifting interferometry algorithms.,” *Appl. Opt.*, vol. 36, 8098–8115, 1997.
- [33] H. K., B. F. Oreb, D. I. Farrant, and K. G. Larkin, “Phase-shifting algorithms for nonlinear, spatially nonuniform phase shifts.,” *J. Opt. Soc. Am. A*, vol. 14, 918., 1997.
- [34] H. K., K. G. Larkin, B. F. Oreb, and D. I. Farrant, “Phase-shifting algorithms for nonlinear, spatially nonuniform phase shifts: Reply to comment.,” *J. Opt. Soc. Am. A*, vol. 15, p. 1234–1235, 1998.
- [35] H. K., R. Hanayama, J. Burke, and B. Oreb, “Tunable phase-extraction formulae for simultaneous shape measurement of multiple surfaces with wavelength-shifting interferometry.,” *Opt. Express*, vol. 12, p. 5579–559, 2004.
- [36] M.-D. D. and B. Dorrío, “Family of detuning-insensitive phase-shifting algorithms.,” *Appl. Opt.*, vol. J. Opt. Soc. Am. A 17, p. 1857–1863, 2000.
- [37] Malacara-Doblado, M. F. Kuechel, R. Rojas-Laguna, and E. Vargas-Rodriguez, “Wide-range detuning insensitive phase-shifting algorithms.,” *Proc. SPIE*, vol. 4419, 269, 2001.
- [38] Z. Y. and T. Gemma, “Method for designing error-compensating phase-calculation algorithms for phase-shifting interferometry.,” *Appl. Opt.*, vol. 40, p. 4540–4546, 2001.
- [39] Z. M., C. Q., T. Tao, F. S., H. Y., and L. H. et al., “Robust and efficient multi-frequency temporal phase unwrapping: optimal fringe frequency and pattern sequence selection.,” *Opt. Express* 25, no. 20381-20400, 2017.
- [40] A. H. and G. C., “Integrated photoelasticity.,” *In Photoelasticity of Glass*, no. 86–101, 1993.
- [41] S. C. A., “The moiré method—a review.,” *Exp. Mech.*, vol. 22, no. 418–433, 1982.
- [42] G. D., “A new microscopic principle.,” *Nature*, vol. 161, no. 777–77, 1948.
- [43] C. E., B. F., and D. C., “Digital holography for quantitative phase-contrast imaging,” *Opt. Lett.*, vol. 24, no. 291–293, 1999.
- [44] L. O.J., “Electronic speckle pattern interferometry,” *Optical Metrology (ed. Soares, O. D. D.)*, no. 542–57, 1987.
- [45] H. Y., “Shearography: a new optical method for strain measurement and nondestructive testing,” *Optical Eng.*, vol. 21, no. 213-391, 1982.
- [46] P. B. et al., “Two-dimensional digital image correlation for in-plane displacement and strain measurement: a review.,” *Meas. Sci. Technol.*, vol. 20, no. 062001, 2009.
- [47] G. J., “Structured-light 3d surface imaging: a tutorial,” *Adv. Opt. Photonics*, vol. 3, no. 128–160, 2011.
- [48] P. O. J. C., R. M. J. W., B. R. L., G. H. E., S. J. T., and R. A. J. M. et al., “Image processing for 3d reconstruction using a modified fourier transform profilometry method.,” *Springer*, no. 705-712, 2007.

- [49] P.-O. J.C., G.-H. E., D.-R. M., C.-M. S.L., R.-A. J.M., and e. a. Aceves Fernandez M.A., "A 3d sensor based on a profilometrical approach," *Sensors*, vol. 9(12), no. 10326-10340, 2009.
- [50] L. C.V., S. C. S., K. K., P. J.C., and R. J.M., "Improving 3d reconstruction accuracy in wavelet transform profilometry by reducing shadow effects," *IET Image Processing*, vol. 14(2), no. 310-31, 2020.
- [51] O. P., L. B., W. Y., and Z. S., "Flexible real-time natural 2d color and 3d shape measurement," *Opt. Express*, vol. 21(14), p. 16736–16741, 2013.
- [52] G. Ricardo and Z. A., "Consistent stereo-assisted absolute phase unwrapping methods for structured light systems," *IEEE Journal of Selected Topics in Signal Processing*, vol. 6, pp. 411–424, 2012.
- [53] K. J., P. J., N. T., and S. D., "Improvement of phase unwrapping algorithms by epipolar constraints," *In Proceedings of the 10th International Conference on Computer Vision Theory and Applications*, vol. Volume 3: VISAPP, (VISIGRAPP 2015), no. ISBN 978-989-758-091-8, pp. 472–479, 2015.
- [54] T. Tao, Q. Chen, S. Feng, J. Qian, Y. Hu, L. Huang, and C. Zuo, "High-speed real-time 3d shape measurement based on adaptive depth constraint," *Opt. Express*, vol. 26, no. 22440-22456, 2018.
- [55] Y. Wang, H. Zhao, X. Li, and H. Jiang, "High-accuracy 3-d sensor for rivet inspection using fringe projection profilometry with texture constraint," *Sensors*, vol. 20, no. no. 24, p. 7270, 2020.
- [56] M. D., S. D., and T. Charles, "Machine learning, neural and statistical classification," *Technometrics*, vol. 37, 1999.
- [57] Z. X. D., *Machine learning. in A Matrix Algebra Approach to Artificial Intelligence*, vol. (cd. Zhang, X. D.). Springer, 2020.
- [58] F. Rosenblatt, "The perceptron: a probabilistic model for information storage and organization in the brain.," *Psychological review*, vol. 65 6, pp. 386–408, 1958.
- [59] N. V. and H. G. E., "Rectified linear units improve restricted boltzmann machines," *In Proceedings of the 27th International Conference on International Conference on Machine Learning*, p. 807–814, 2010.
- [60] K. M., F. S., and O. A., "Deep learning in business analytics and operations research: models, applications and managerial implications," *Eur. J.Operational*, vol. 281, p. 628–641, 2010.
- [61] Z. Z. L. and S. M. R., "Generalized cross entropy loss for training deep neural networks with noisy labels," *In Proceedings of the 32nd International Conference on Neural Information Processing Systems*, p. 8792–8802, 2018.
- [62] K. J. and Y. J. Y., "Peak signal-to-noise ratio revisited: is simple beautiful," *In Proceedings of the 4th International Workshop on Quality of Multimedia Experience*, p. 37–38, 2012.
- [63] G. R., "Fast r-cnn," *In Proceedings of 2015 IEEE International Conference on Computer Vision*, p. 1440–1448, 2015.
- [64] W. Z. et al., "Image quality assessment: from error visibility to structural similarity," *IEEE Trans. Image Process*, vol. 13, p. 600–612, 2004.

- [65] W. J. J. et al., “Deep learning for smart manufacturing: methods and applications,” *J. Manuf. Syst.*, vol. 48, p. 144–156, 2018.
- [66] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016. <http://www.deeplearningbook.org>.
- [67] K. D. P., R. D. J., M. Shakir, and W. Max, “Semi-supervised learning with deep generative models,” 2014.
- [68] A. Guillaume, B. Yoshua, Y. Li, Y. Jason, T.-L. Eric, Z. Saizheng, and V. Pascal, “Deep generative stochastic networks trainable by backprop,” 2013.
- [69] M. W. S. and P. W., “logical calculus of the ideas immanent in nervous activity,” *Bull. Math. Biophysics*, vol. 5, pp. 115–133, 1943.
- [70] K. A., S. I., and H. G., “Imagenet classification with deep convolutional neural networks,” *Neural Information Processing Systems*, vol. 25, 2012.
- [71] M. D. Zeiler and R. Fergus, “Visualizing and understanding convolutional networks,” 2013.
- [72] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” 2014.
- [73] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” 2015.
- [74] J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional networks for semantic segmentation,” 2014.
- [75] D. C. et al., “Image super-resolution using deep convolutional networks,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, p. 295–307, 2015.
- [76] W. Z. H., C. J., and H. S. C. H., “Deep learning for image super-resolution: a survey,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, p. 3365–3387, 2021.
- [77] D. Y. P. et al., “Srcnn-based enhanced imaging for low frequency radar,” *In 2018 Progress in Electromagnetics Research Symposium (PIERS-Toyama)*, p. 366–370, 2018.
- [78] L. Y. J. et al., “Underwater image high definition display using the multilayer perceptron and color feature-based srcnn,” *IEEE Access*, vol. 7, p. 83721–83728, 2019.
- [79] U. K., O. J., and I. T., “Application of super-resolution convolutional neural network for enhancing image resolution in chest ct,” *J. Digital Imaging*, vol. 31, p. 441–450, 2018.
- [80] L. J., S. E., and D. T., “Fully convolutional networks for semantic segmentation,” *In Proceedings of 2005 IEEE Conference on Computer Vision and Pattern Recognition.*, p. 3431–3440, 2015.
- [81] N. H., H. S., and H. B., “Learning deconvolution network for semantic segmentation,” *In Proceedings of 2015 IEEE International Conference on Computer Vision*, p. 1520–1528, 2015.
- [82] R. O., F. P., and B. T., “U-net: convolutional networks for biomedical image segmentation,” *In Proceedings of 18th International Conference on Medical Image Computing and Computer-Assisted Intervention*, p. 234–241, 2015.

- [83] F. T. et al., “U-net: deep learning for cell counting, detection, and morphometry.,” *Nat. Methods*, vol. 16, p. 67–70, 2019.
- [84] T. T., Q. C., S. F., Y. H., J. D., and C. Z., “High-precision real-time 3d shape measurement using a bi-frequency scheme and multi-view system,” *Appl. Opt.*, vol. 56, pp. 3646–3653, 2017.
- [85] Z. S. and H. P.S., “Novel method for structured light system calibration,” *Opt. Eng.*, vol. 45, no. 083601, 2006.
- [86] S. V., L. H.C., and H. M., “Automated phasemeasuring profilometry of 3-d diffuse objects,” *Appl. Opt.*, vol. 23, p. 3105–3108, 1984.
- [87] G. D. and P. M., *Two-dimensional phase unwrapping : Theory, algorithms and software*. New York: John Wiley and Sons, 1998.
- [88] J. M. Huntley and H. Saldner, “Temporal phase-unwrapping algorithm for automated interferogram analysis,” *Appl. Opt.*, vol. 32, no. 3047–3052, 1983.
- [89] H. Zhao, W. Chen, and Y. Tan, “Phase-unwrapping algorithm for the measurement of three-dimensional object shapes,” *Appl. Opt.*, vol. 33, pp. 4497–4500, 1994.
- [90] J. M. Huntley and H. O. Saldner, “Error-reduction methods for shape measurement by temporal phase unwrapping,” *J. Opt. Soc. Am. A*, vol. 14, no. 3188–3196, 1997.
- [91] J. M. Huntley and H. O. Saldner, ““shape measurement by temporal phase unwrapping: comparison of unwrapping algorithms,” *Measurement Science and Technology*, vol. 8, no. 986, 1997.
- [92] X. P. Z. Yang and H. Niu, “Multi-resolution reconstruction of 3-d image with modified temporal unwrapping algorithm,” *Optics Communications*, vol. 224, no. 35-44, 2003.
- [93] L. Kinell and M. Sjö odahl, “Robustness of reduced temporal phase unwrapping in the measurement of shape,” *Appl. Opt.*, vol. 40, p. 2297–2303, 2001.
- [94] J. T. X. Peng and X. Zhao, “A generalized temporal phase unwrapping algorithm for three-dimensional profilometry,” *Optics and Lasers in Engineering*, vol. 46, no. 336– 342, 2008.
- [95] C. Zuo, Q. Chen, G. Gu, S. Feng, F. Feng, R. Li, and G. Shen, “Highspeed three-dimensional shape measurement for dynamic scenes using bifrequency tripolar pulse-width-modulation fringe projection,” *Optics and Lasers in Engineering*, vol. 51, 2013.
- [96] L. K., W. Y., L. D.L., H. Q., and H. L.G., “Udual-frequency pattern scheme for high-speed 3-d shape measurement,” *Opt. Express*, vol. 18(5), p. 229–5244, 2010.
- [97] W. F., W. C., and G. Q., “Single-shot fringe projection profilometry based on deep learning and computer graphics,” *Opt. Express* 29, pp. 8024–8040, 2021.
- [98] R. B., J. V., V. S., and B. J., “Projection moiré profilometry simulation software for algorithm validation and setup optimisation.,” *Optical Measurement Techniques for Systems and Structures*, pp. 87–96, 2013.
- [99] B. V., K. A., and C. R., “Segnet: A deep convolutional encoder-decoder architecture for image segmentation,” *IEEE Trans Pattern Anal Mach Intell*, vol. 39(12), pp. 2481–2495, 2017.

- [100] S. W. Z. et al., “Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network.,” *In Proceedings of 2016 IEEE Conference on Computer Vision and Pattern Recognition*, no. 1874–1883, 2016.
- [101] Y. Gal, J. Hron, and A. Kendall, “Concrete dropout,” *arXiv preprint*, vol. arXiv:1705.07832, 2017.
- [102] J. Schmit and K. Creath, “Extended averaging technique for derivation of error-compensating algorithms in phase-shifting interferometry,” *Appl. Opt.* 34, pp. 3610–3619, 1995.
- [103] X. Zexiang, S. Kalyan, H. Sunil, and R. Ravi, “Deep image-based relighting from optimal sparse samples,” *ACM Transactions on Graphics (TOG)*, vol. 37, no. 4, p. 126, 2018.
- [104] Z. S., “High-speed 3d imaging with digital fringe projection techniques,” *CRC Press*, vol. ISBN: 978-1-4822-3433-6, 2016.
- [105] Q. J., F. S., L. Y., T. T., H. J., and C. Q. et al., “Single-shot absolute 3d shape measurement with deep-learning-based color fringe projection profilometry,” *Opt. Lett.*, vol. 45, pp. 1842–1845, 2020.

.1. Registro de protocolo de investigación



UNIVERSIDAD AUTÓNOMA DE QUERÉTARO
FACULTAD DE INGENIERÍA
DIVISIÓN DE INVESTIGACIÓN Y POSGRADO

REGISTRO DEL PROTOCOLO DE INVESTIGACIÓN DEL ESTUDIANTE DE POSGRADO

Los 2 Espacios oscuros exclusivos para la Dirección	No. Registro de Proyecto*: Fecha de Registro*:	12590 09-08-2021
	Fecha de inicio de proyecto:	25/06/2021
	Fecha de término de proyecto:	30/12/2022
1. DATOS DEL SOLICITANTE		
No. de expediente:	226262	
Apellido Paterno	Apellido Materno	Nombre(s)
MARTINEZ	SUAREZ	VICTOR MANUEL
Dirección:		
Calle y número	Colonia	C.P.
16 de Septiembre #47	Centro SRJ	76200
Estado	Teléfono (Incluir lada)	Correo Electrónico
22	4424884139	vmartinez21@alumnos.uaq.mx
2. DATOS DEL PROYECTO		
Facultad:	INGENIERÍA	
Programa:	MAESTRIA EN CIENCIAS EN INTELIGENCIA ARTIFICIAL	
Tema específico del proyecto:	Reconstrucción de formas 3D mediante algoritmo de desplazamiento de fase estéreo usando proyección de franjas y una red neuronal convolucional	
 PEDRAZA ORTEGA JESUS CARLOS Director de tesis	 TOVAR ARRIAGA SAUL Coordinador de programa	 MARTINEZ SUAREZ VICTOR MANUEL Alumno
Dr. Juan Carlos Jáuregui Correa Jefe de División de Investigación y Posgrado de la Fac. de Ing.	Dr. Manuel Toledano Ayala Director de Fac. Ing.	 Dra. Ma. Guadalupe Flavia Lobarca Piña Directora de Investigación y Posgrado UAQ



.2. Dictamen de Comité de Ética Aplicada



C.U., 10 de junio de 2021

Victor Manuel Martínez Suárez
Estudiante de Maestría en Ciencias en Inteligencia Artificial
Expediente 226262

Presente

El Comité de Ética Aplicada para la Investigación de la Facultad de Ingeniería ha revisado el protocolo del trabajo de tesis:

CEAIFI-089-2021-TP

**Reconstrucción de formas 3D mediante algoritmo de desplazamiento
de fase estéreo usando proyección de franjas y una red neuronal
convolucional**


Con apego a los lineamientos éticos de beneficencia, no maleficencia, justicia y autonomía, este comité ha dado el siguiente dictamen:

Exento de dictamen ético

El presente dictamen tiene vigencia de un año a partir de su fecha de emisión.

Sirva esta carta para los fines académicos que al interesado convengan.

Atentamente
"El Ingenio para Crear, No para Destruir"


Dra. Aurora Femat Díaz
Presidente del CEAIFI
afemat@uaq.mx



.3. Examen de comprensión de textos en inglés

 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

MARTINEZ SUAREZ VICTOR MANUEL

Presentó y acreditó el **Examen de Comprensión de Textos en Inglés** efectuado el día ocho de octubre de dos mil veintiuno.

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 veinticinco de octubre de dos mil veintiuno.

Atentamente,
"Enlazar Culturas por la Palabra"




DRA. ADELINA VELÁZQUEZ HERRERA

AVH/japa*CL*FLL-C.-2016



SOMOS UAQ
BUSCAR CRECE CONOCER

Campus Aeropuerto, Anillo Vial Fray Junípero Serra S/N, Querétaro, Qro., C.P. 76140
Tel. 442 192 12 00 Dirección Ext. 61010, Secretaría Administrativa Ext.61300, Pasadizo Ext. 61140,
Licenciatura Ext.61070, Centro de Lenguas Ext.61050, Secretaría Académica Ext.61100 y Platación Ext.61110

4. Examen de manejo de la lengua

  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

MARTINEZ SUAREZ VICTOR MANUEL

Presentó el **Examen de Manejo de la Lengua** efectuado el día treinta de noviembre de dos mil veintiuno, 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 diez de diciembre de dos mil veintiuno.

Atentamente,
"Enlazar Culturas por la Palabra"


DRA. ADELINA VELÁZQUEZ HERRERA

AVH/japa*CL*FLL-C.-2458

 SOMOS UAQ
INNOVACIÓN EDUCATIVA

Campus Aeropuerto, Avda. Vial Pray Julián S/N, Querétaro, Qro., C.P. 76140
Tel. 442 192 12 00 Dirección, Ext. 61010, Secretaría Administrativa Ext. 61200, Rectoría Ext. 61140,
Biblioteca Ext. 61070, Centro de Idiomas Ext. 61050, Secretaría Académica Ext. 61100 y Planeación Tel. 61110

.5. Constancias WITCOM 2022



.6. Codigos CNN1 y CNN2

```

1
2  from tensorflow.python.keras.activations import linear
3  from tensorflow.python.ops.gen_math_ops import Sigmoid
4
5  from math import log10
6  def SPU_CNN1_Unet(pretrained_weights = None):
7      dx,dy,dz = 480,640,1
8      C = 64
9      L_input = keras.Input(shape=(dx,dy,dz), name="left") # Variable-
          length sequence of ints
10
11 # encoder: contracting path - downsample
12 # 1 - downsample
13     f1, p1 = downsample_block(L_input, C)
14 # 2 - downsample
15     f2, p2 = downsample_block(p1, C)
16 # 3 - downsample
17     f3, p3 = downsample_block(p2, C)
18 # 4 - downsample
19     f4, p4 = downsample_block(p3, C)
20
21 # 5 - bottleneck
22     bottleneck = double_conv_block(p4, C)
23
24 # decoder: expanding path - upsample
25 # 6 - upsample
26     u6 = upsample_block(bottleneck, f4, C)
27 # 7 - upsample
28     u7 = upsample_block(u6, f3, C)
29 # 8 - upsample
30     u8 = upsample_block(u7, f2, C)
31 # 9 - upsample
32     u9 = upsample_block(u8, f1, C)
33
34     outI = Conv2D(2, 3, activation='linear',padding = 'same',
          kernel_initializer = 'he_normal')(u9)
35     outI1 = Dense(1)(outI)
36     outI2 = Dense(1)(outI)
37     model = Model(inputs = L_input, outputs = [outI1,outI2])
38
39     model.compile(optimizer= tf.keras.optimizers.Adam(learning_rate=1e
          -3), loss= 'mse', metrics=['mae'])
40     model.summary()
41
42     if(pretrained_weights):
43         model.load_weights(pretrained_weights)
44
45 return model

```

Figura 1: Código utilizado para la construcción de la red neuronal (CNN1).

```

1  from math import log10
2  def SPU_CNN1_Unet2(pretrained_weights = None):
3      dx,dy,dz = 480,640,1
4      C = 64
5      L_input = keras.Input(shape=(dx,dy,dz), name="left") # Variable-
        length sequence of ints
6      O_input = keras.Input(shape=(dx,dy,dz), name="order") # Variable-
        length sequence of ints
7      OL_input = keras.Input(shape=(dx,dy,dz), name="orderleft") #
        Variable-length sequence of ints
8      inputs = concatenate([L_input,O_input,OL_input], axis = 3)
9
10
11     # encoder: contracting path - downsample
12     # 1 - downsample
13     f1, p1 = downsample_block(inputs, C)
14     # 2 - downsample
15     f2, p2 = downsample_block(p1, C)
16     # 3 - downsample
17     f3, p3 = downsample_block(p2, C)
18     # 4 - downsample
19     f4, p4 = downsample_block(p3, C)
20
21     # 5 - bottleneck
22     bottleneck = double_conv_block(p4, C)
23
24     # decoder: expanding path - upsample
25     # 6 - upsample
26     u6 = upsample_block(bottleneck, f4, C)
27     # 7 - upsample
28     u7 = upsample_block(u6, f3, C)
29     # 8 - upsample
30     u8 = upsample_block(u7, f2, C)
31     # 9 - upsample
32     u9 = upsample_block(u8, f1, C)
33
34     outI = Conv2D(1, 3, activation='linear',padding = 'same',
        kernel_initializer = 'he_normal')(u9)
35
36     model = Model(inputs = [L_input,O_input,OL_input], outputs = outI)
37     model.compile(optimizer= tf.keras.optimizers.Adam(learning_rate=1e
        -4), loss= 'mse', metrics=['mae'])
38     model.summary()
39
40     if(pretrained_weights):
41         model.load_weights(pretrained_weights)
42     return model

```

Figura 2: Código utilizado para la construcción de la red neuronal (CNN1).

