



**Tecnológico de Estudios Superiores
de Cuautitlán Izcalli**

Organismo Público Descentralizado del Estado de México

MAESTRÍA EN TECNOLOGÍAS DE LA INFORMACIÓN

**TRATAMIENTO DE IMÁGENES CON MACHINE LEARNING PARA
RECONSTRUCCIÓN TRIDIMENSIONAL DE OBJETOS UTILIZANDO
PROYECCIÓN DE FRANJAS**

TESIS

QUE PARA OBTENER EL GRADO DE:

MAESTRO EN TECNOLOGÍAS DE LA INFORMACIÓN

PRESENTA:

CYNTHIA WALLES GUTIERREZ

DIRECTOR DE TESIS:

DR. OMAR BARRAGAN PEREZ

CUAUTITLÁN IZCALLI, EDO. DE MÉXICO. NOV,2022.



"2022. Año del Quincentenario de Toluca, Capital del Estado de México".

Cuautitlán Izcalli, Estado de México a 17 de noviembre de 2022
TESCI/DIDT/202/XV/22

DIRECCIÓN ACADÉMICA
DEPARTAMENTO DE INVESTIGACIÓN Y DESARROLLO TECNOLÓGICO
COORDINACIÓN DE POSGRADO

INGENIERA
CYNTHIA WALLES GUTIERREZ
PRESENTE

Por este conducto me permito informarle que puede proceder a la digitalización del Trabajo de Tesis titulado:

“TRATAMIENTO DE IMÁGENES CON MACHINE LEARNING PARA RECONSTRUCCIÓN TRIDIMENSIONAL DE OBJETOS UTILIZANDO PROYECCIÓN DE FRANJAS”

Ya que la comisión encargada de revisar el trabajo que se presenta para efectos de titulación, han dado su autorización conforme a lo estipulado en el Lineamiento para la operación de los Estudios de Posgrado en el Sistema Nacional de Institutos Tecnológicos.

Sin nada más que agregar, quedo a sus órdenes para cualquier aclaración.

ATENTAMENTE



MTRA. ROCÍO ORTEGA JIMÉNEZ
DEPARTAMENTO DE
INVESTIGACIÓN Y DESARROLLO TECNOLÓGICO
COORDINACIÓN DE POSGRADO

c.c.p. Archivo
Departamento de Titulación
Expediente del alumno



SECRETARÍA DE EDUCACIÓN
SUBSECRETARÍA DE EDUCACIÓN SUPERIOR Y NORMAL
DIRECCIÓN GENERAL DE EDUCACIÓN SUPERIOR
TECNOLÓGICO DE ESTUDIOS SUPERIORES DE CUAUTITLÁN IZCALLI



Índice

Capítulo I	7
Resumen.....	7
Abstract	7
Objetivo general.....	8
Objetivos Específicos.....	8
Justificación	9
Introducción	10
Capítulo II	11
Marco teórico.....	11
Técnicas de Proyección de Franjas.....	11
Transformada de FFT.....	13
Transformada Inversa de Fourier	14
Método de la transformada de Fourier.....	16
Escala de grises.....	18
Filtrado Gaussiano.....	19
Histograma	20
Filtrado de Mediana	20
Imagen 3D	22
Desenvolvimiento de Fase	23
Machine Learning.....	24
Python	27
Dispositivos externos	28
Capítulo III	29

Estado del Arte	29
Capítulo IV	31
Metodología	31
Capítulo V	34
Diseño de la interfaz gráfica (GUI).....	34
<i>Diseño Tratamiento de Imágenes</i>	34
Capítulo VI	40
Funcionamiento de la interfaz gráfica (GUI)	40
Conclusiones.....	44
Bibliografía.....	45
Anexos	47

Índice de Imágenes

Figura 1: Sistema de Proyección de Franjas	12
Figura 2: Función de la transformada rápida de Fourier	14
Figura 3: Espectro de Fourier y frecuencia espacial	17
Figura 4: Escala de Grises proyección de Franjas	19
Figura 5: Imagen con ruido Gaussiano	20
Figura 6: Imagen de un histograma con curva normal	20
Figura 7: Imagen del filtro de media y filtro de mediana	21
Figura 8: Procesamiento de Imagen en 3D	23
Figura 9: Funcionamiento del Machine Learning	25
Figura 10: Tipos de Algoritmos Machine Learning	26
Figura 11: Modelado de un proceso Inteligencia Artificial	27
Figura 12: Lenguaje de código abierto Python	28
Figura 13: Funcionamiento del proyector de video para proyección de Franjas	28
Figura 14: Cámara CCD	29
Figura 15: Software para las expresiones matemáticas	29
Figura 16: Línea de Procesamiento de imágenes	32
Figura 17: Inicio de la interfaz Gráfica de Proyección de Franjas	34
Figura 18: Menú de la interfaz Gráfica Proyección de franjas	35
Figura 19: Menú de los Tratamiento que se pueden dar a una Imagen	35
Figura 20: Resultado del tratamiento de Transformada de Fourier	36
Figura 21: Resultado del Tratamiento a Escala de Grises	36
Figura 22: Resultado de Tratamiento de Filtrado de Gaussiano	37
Figura 23: Resultado de un tratamiento de imagen y su Histograma	37
Figura 24: Resultado de tratamiento de Filtrado de Mediana	38
Figura 25: Resultado de tratamiento de Contorno de Imagen	38
Figura 26: Resultado de tratamiento en una imagen en 3D	39
Figura 27: Interfaz desarrollada para procesamiento de imágenes.	40
Figura 28: Selección de imágenes para el procesamiento	40
Figura 29: Selección de imagen para su tratamiento	41

Figura 30: Aplicación de la fase Envuelta	41
Figura 31: Aplicación de la fase desenvuelta	42
Figura 32: Envolvimiento y desenvolvimiento de la fase	42
Figura 33: Imagen de la fase desenvuelta asociada a la forma 3D del objeto	43

Capítulo I

Resumen

El objetivo es implementar un software con machine learning para el análisis de imágenes y algoritmos a través de proyección de franjas, capturadas por escaneo láser y almacenadas en una computadora; con el fin de obtener una imagen tridimensional que permita la producción de un prototipo en el cual se pueda ver el volumen, textura y color de un objeto real. La reconstrucción tridimensional de la imagen se divide en técnicas de contacto y de no contacto siendo estas denominadas técnicas no invasivas y por tanto las más utilizadas en los sectores de la industria, dentro de esta categoría los sistemas que utilizan técnicas basadas en proyección de franjas son muy populares debido a su versatilidad, economía y precisión.

Palabras clave: Machine learning, Proyección de franjas, Reconstrucción tridimensional.

Abstract

The objective is to implement a software with machine learning for the analysis of images and algorithms through the projection of stripes, captured by laser scanning and stored in a computer; in order to obtain a three-dimensional image that allows the production of a prototype in which the volume, texture and color of a real object can be seen. The three-dimensional reconstruction of the image is divided into contact and non-contact techniques, these being called non-invasive techniques and therefore the most used in the industrial sectors, within this category the systems that use techniques based on stripe projection are very popular due to its versatility, economy and precision.

Keywords: Machine learning, Fringe projection, Three-dimensional reconstruction.

Objetivo general

El objetivo fundamental es aplicar el procesamiento digital de imágenes aplicando las técnicas necesarias para mejorar la calidad o facilitar la búsqueda de información, a través del uso de machine learning.

Objetivos Específicos

Los principales objetivos que se persiguen con la aplicación de filtros son:

- Suavizar la imagen: reducir la cantidad de variaciones de intensidad entre píxeles vecinos.
- Eliminar ruido: eliminar aquellos píxeles cuyo nivel de intensidad es muy diferente al de sus vecinos y cuyo origen puede estar tanto en el proceso de adquisición de la imagen como en el de transmisión.
- Realzar bordes: destacar los bordes que se localizan en una imagen.
- Detectar bordes: detectar los píxeles donde se produce un cambio brusco en la función intensidad. Por tanto, se consideran los filtros como operaciones que se aplican a los píxeles de una imagen digital para optimizarla, enfatizar cierta información o conseguir un efecto especial en ella.

Justificación

Con el propósito de tener un software adecuado para obtener imágenes en 3D con el uso de Machine Learning, se planteó este proyecto con el fin de poder usarlo en la cual permita el uso más fácil de lo que es el proceso de proyección de franjas por el método de Fourier o Takeda, ya que esto lleva a usar un software propio. para ello se elaboró este trabajo con la certeza de reducir costos y el uso fácil del mismo. El software está elaborado por medio de métodos matemáticos basándose en el uso de lenguaje de programación.

Introducción

El control dimensional y el prototipo rápido son necesidades de algunas industrias que se soportan en la generación de reconstrucciones tridimensionales (3D) de los objetos. Uno de los principales retos es la realización de esta reconstrucción 3D sin contacto mecánico y en el menor tiempo posible.

Se busca lograr imprimir o maquinar los objetos reconstruidos verificando sus dimensiones por medio de mediciones, aplicando el machine learning para un procesamiento de imágenes ágil y eficaz, se continúa trabajando en el desarrollo de métodos más eficientes para la captura, procesamiento de imágenes, involucramiento de la fase y desenvolvimiento, desarrollando un sistema semiautomático con interfaz para dicho proceso.

Considerando que en estudios recientes se ha mostrado el uso de Inteligencia artificial para hacer más eficiente dicho análisis. Donde utiliza visión por computadora e inteligencia artificial para digitalización de datos. Lo que permite hacer un análisis más profundo y con mayor exactitud lo que permitiría obtener calidad de reconstrucción adecuada en un corto tiempo. Muestra la importancia de la Inteligencia Artificial en aplicaciones de reconocimiento de imágenes donde se aplican técnicas como Deep Learning o aprendizaje. Que explica que la convolución es una herramienta primordial en el procesamiento digital de imágenes e incluso un pilar fundamental para la aplicación de Machine Learning. Es por ello que se toman en cuenta estos estudios para proponer un método que trabaje en conjunto con la técnica de proyección de franjas para reconstruir objetos en 3D.

Hoy en día se dispone de programas dedicados exclusivamente para el procesamiento de imágenes, existe un conjunto de herramientas que ayudan a desarrollar un modelo en dispositivos incorporados, móviles o de IoT (registro de datos), y que permita implementar el aprendizaje automático integrado en un dispositivo. Además, verán el desarrollo matemático de la implementación de la técnica de Takaeda ya que es muy importante para su aplicación en este trabajo, podrán observar el resultado obtenido en cada método que se implementó y los detalles de este mismo.

Capítulo II

Marco teórico

Técnicas de Proyección de Franjas

La técnica de proyección de franjas permite adquirir las imágenes necesarias para obtener una reconstrucción 3D con resolución micrométrica y precisión sub píxel. consiste en proyectar sobre el objeto que se va a reconstruir, franjas paralelas blancas y negras, que se deforman por la topografía superficial del objeto. (González, Proyección de franjas en metrología óptica facial¹, 2012)

El sistema consta de una unidad de proyección/adquisición y una unidad de procesamiento. La unidad de proyección/ adquisición está conformada por un dispositivo de proyección (tradicionalmente un video proyector) para proyectar las franjas y un dispositivo de observación para la adquisición de las imágenes (cámara CCD).

Las imágenes adquiridas son procesadas por la unidad de control, que permite adicionalmente sincronizar las adquisiciones y calibrar el dispositivo.

La proyección se realiza sobre una superficie plana donde se observan franjas paralelas y de paso al proyectarlas sobre un objeto, las irregularidades de la superficie (relacionadas directamente con la topografía superficial del objeto) quedan evidenciadas con la deformación de las franjas.



Figura 1: Sistema de Proyección de Franjas

En la actualidad el método de reconstrucción por franjas consta de un sistema de proyección, un sistema de observación y un plano de referencia, donde se localiza el sistema de observación, ubicado en la coordenada $Z_0 = 0$ del sistema coordenado (X_0, Y_0, Z_0) como se puede observar en la figura 1.1. Cuando se proyectan franjas sobre el plano de referencia, las franjas se ven paralelas entre sí.

Un patrón de franjas es sinusoidal, es proyectado sobre un plano de referencia y posteriormente sobre el objeto de estudio; ambas imágenes son almacenadas en la memoria de una computadora. La valoración de las diferencias de estos registros constituye el núcleo de la técnica de proyección de franjas, y en la actualidad se realiza por medios completamente digital lo que le brinda gran versatilidad y velocidad de desempeño.

Si las franjas proyectadas son cosenoidales, el desplazamiento del patrón de franjas se refleja en un cambio en el argumento de la función coseno, es decir, produce un cambio en fase que llamaremos $f(x)$. Este cambio de fase se puede relacionar con la distancia entre el plano de referencia y la superficie de interés del objeto z . El valor de z para cada uno de valores de x se puede hallar por medio de la evaluación de la fase $f(x)$ del sistema de franjas deformado por el punto x de interés.

$$Z = \frac{dx}{2\pi \tan \theta} f(x)$$

De acuerdo con la ecuación anterior el mapa de alturas z para las distintas posiciones del objeto está completamente determinado si se conoce el valor de la fase $f(x)$ y esta se puede obtener por medio de diversas metodologías: los métodos de la transformada de Fourier o cualquiera de las

técnicas de corrimiento de fase que figuran las más eficientes y ampliamente utilizadas. Una vez que se tiene la fase $f(x)$, se obtiene la altura z para cada x y de esta forma se reproduce la superficie del objeto en estudio. (Daniel Jaramillo Chamba, 2022)

Transformada de FFT

La transformada de Fourier convierte coordenadas espaciales en frecuencias. Cualquier curva o superficie se puede expresar como la suma de un cierto número (quizá infinito) de curvas del seno y del coseno. En el dominio de Fourier (llamado el dominio de la frecuencia) la imagen se representa con los parámetros de las funciones del seno y del coseno. La transformada de Fourier es el mecanismo matemático para moverse dentro y fuera del dominio de la frecuencia.

El dominio de la frecuencia se llama así porque los dos parámetros de una curva del seno son la amplitud y la frecuencia. El hecho de que una imagen se pueda convertir en una representación del dominio de la frecuencia implica que la imagen puede contener información de alta frecuencia o de baja frecuencia. Si el nivel gris de una cierta porción de la imagen cambia lentamente a través de las columnas, entonces sería representado en el dominio de la frecuencia como función del seno o del coseno teniendo una frecuencia baja. Una cierta cosa que cambia rápidamente, por ejemplo, un borde, tendrá componentes de alta frecuencia.

Es por lo tanto posible construir los filtros que quitarán o realzarán las frecuencias en la imagen, y éstas tendrán a veces un efecto reconstituyente.

De hecho, el ruido consiste principalmente en información de alta frecuencia, y así el filtrado de alta frecuencia debe tener un efecto de reducción del nivel de ruidos. Desafortunadamente también tiene un efecto de la reducción del borde.

La transformada de Fourier divide una imagen (en una dimensión, una señal) en un conjunto de componentes del seno y del coseno. Es importante mantener estas componentes separadas, y

así un vector de la forma (coseno, seno) es utilizado en cada punto del dominio de la frecuencia de la imagen; es decir, los valores de los píxeles en el dominio de la frecuencia de la imagen son vectores componentes. Una manera conveniente de representarlos es mediante números complejos.

Cada número complejo tiene una parte real y una parte imaginaria, se puede interpretar tal cual como un vector. Un número complejo típico podría ser escrito como se muestra en la ecuación.

$z = (x, jy) = x + jy$ donde j es imaginaria. El exponencial de un imaginario se puede representar como la suma de un seno y un coseno, que es exactamente como se muestra en la ecuación.

Esta forma polar es la que se usará a partir de ahora, pero es importante recordar que realmente es la suma del seno y del coseno. En una dimensión, la transformada de Fourier de $f(x)$ de una función continua es representada en la ecuación.

Si se ha muestreado la función de modo que ahora es discreta, la integral se convierte en la suma de los puntos muestreados como en la ecuación.

Si la función $f(k)$ es una curva muestreada del seno, entonces la transformada de Fourier $F(w)$ debe producir un solo punto que muestra los parámetros de la curva. (Gonzalez, 2006)

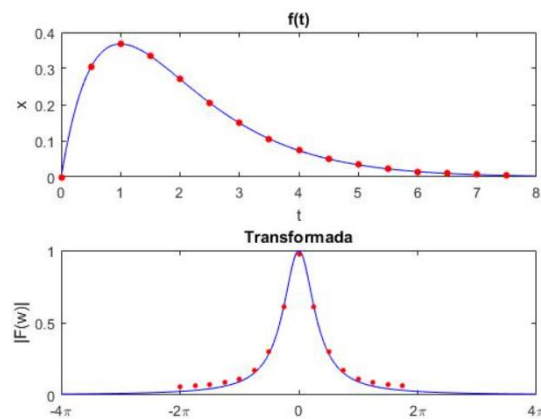


Figura 2: Función de la transformada rápida de Fourier

Transformada Inversa de Fourier

La inversa de la transformada de Fourier deshacerá la transformación; cuando se da la transformada de Fourier de un sistema de datos, lo contrario de la transformada reconstruirá los datos originales. La función logarítmica y la función de exponencial tienen una relación similar, donde una deshace la otra. (VÁSQUEZ, 2014)

La fórmula para el cálculo de la transformada discreta de Fourier se muestra en la ecuación:

Que se diferencia de la transformada directa en el signo del exponente.

Ahora puede ser el momento de precisar que el factor constante $1/n$ es algo flexible. Algunos lo aplican a la transformada directa, otros lo parten entre la directa y la transformada inversa y otros lo aplican solamente a la transformada inversa. Desde luego, el algoritmo rápido puede aplicarse a la transformada inversa, así un sistema de datos puede ser transformado tal cual fácilmente en cualquier dirección. En las rutinas de la librería de `fftlib` en lenguaje C, la función inversa del 1D FFT es la misma que la función directa de FFT; hay un parámetro que especificará una dirección directa o inversa.

La Transformada de Fourier de una imagen f es calcular primero la transformada de Fourier de cada fila, dando una imagen f' . Entonces se calcula la transformada de Fourier de cada columna de f' , dando F , la transformada de la imagen. Esto permite que utilice los métodos de cálculo de la FFT de una dimensión discutidos ya para calcular la transformada de 2 dimensiones.

$$f(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} F(x) e^{-ixt} dx$$

Además del uso de la transformada de Fourier para permitir la filtración de ciertas frecuencias, hay otras propiedades útiles que llegan a ser absolutamente importantes al discutir sobre datos de dos dimensiones. Lo más importante desde el punto de vista del procesamiento de una imagen es que la convolución es mucho más fácil de hacer en el dominio de la frecuencia que en el dominio espacial. En verdad, una convolución es simplemente el producto de un elemento por un elemento de la transformada de Fourier de las dos imágenes que son convolucionadas. (Olavarría, 2002)

Método de la transformada de Fourier

La imagen de una rejilla deformada por una ecuación puede ser interpretada como múltiples señales moduladas en fase $\phi(x, y)$, en una amplitud $r(x, y)$ y con frecuencia espacial portadora. Basado en que la fase posee información acerca de la forma tridimensional a medir, el problema es como separar $\phi(x, y)$ del resto de la información.

Se usará la notación utilizada en diferentes literaturas, que es una representación en forma exponencial y no en forma de expansión de series de Fourier. El propósito de usar esta notación, es el de dar una explicación más clara en los procedimientos del método de la Transformada de Fourier, para ello se muestra la ecuación.

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

En donde para respetar las variables propuestas por los textos, $a(x, y)$ es la distribución no uniforme de la reflectividad sobre la superficie del objeto. $b(x, y)$, $\phi(x, y)$ y f_0 , la intensidad, la fase y la frecuencia espacial de la malla (patrón de líneas deformadas) respectivamente. Ahora convirtiendo exponencialmente se tiene la ecuación

$$g(x, y) = a(x, y) + c(x, y)e^{2\pi f_0 x} + c(x, y)e^{-2\pi f_0 x}$$

Usando el algoritmo de FFT se calcula la transformada unidimensional de Fourier de la ecuación anterior para cada variable de x únicamente, con y manteniéndose fija obtenemos la ecuación.

$$G(x, y) = A(f, y) + C(f + f_0, y) + C(f - f_0, y)$$

En donde las letras mayúsculas representan el espectro de Fourier y f representa la frecuencia espacial en la dirección de x y y siendo tratada como una variable fija. Si asumo que $a(x, y)$, $b(x, y)$ y $\phi(x, y)$ tienen frecuencias mucho menores que f_0 , entonces la ecuación anterior se observa en la figura 1.2.

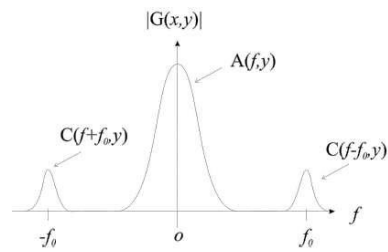


Figura 3: Espectro de Fourier y frecuencia espacial

Para poder separar el índice de reflectividad $a(x, y)$ se necesita pasar la señal por un filtro pasa-bandas el cual debe de estar centrado en f_0 . Para realizar lo anterior es posible utilizar un filtro FIR, FII o un filtro ideal (mascara) ya que la información no se está procesando en tiempo real.

La componente que se va a obtener filtrándola de las demás es la primera componente armónica. Después de dicho filtrado el espectro espacial de Fourier de la señal.

Después de filtrar la señal, esta sea trasladada al origen para que $C(f + f_0, y)$ se convierta en $C(f, y)$ esto se logra a partir de una sencilla operación basada en un teorema de la transformada de Fourier discreta como se observa en la ecuación.

$$e^{i\omega_0 n} x[n] \textcircled{1} X(e^{i(\omega-\omega_0)})$$

Entonces para hacer el dicho corrimiento se tiene que obtener primero la FFT inversa de la señal, una vez mas, esta transformada se realiza en una dimensión, es decir, avanza en x mientras que en y se mantiene como un parámetro constante. Después de lo anterior, se hace la multiplicación por el exponencial para poder enviar la señal al origen en el espectro espacial de Fourier mostrada en la ecuación.

$$\hat{g}(x, y) = \hat{g}'(x, y) e^{i\omega_0 n} = 1 b(x, y) e^{i\phi(x,y)}$$

En donde $\hat{g}(x, y)$ es la transformada inversa de Fourier de la información ya filtrada.

En el caso de los ejes ópticos cruzados, se realizan los mismos pasos para la imagen que solo contiene el fondo (plano de referencia) y así se obtiene la ecuación.

La información concerniente a las franjas, pero de esta información la única que interesa es la fase ya que contiene los datos de la profundidad del objeto. Al igual que en otros pasos, se proponen métodos para la separación n , que en realidad son equivalentes, uno de los métodos es el de por logaritmos complejos y el otro por arco-tangente. Ambos proveen resultados idénticos y además son igual de sencillos de implementar. (Olavarría, 2002)

Escala de grises

Una escala de grises o una escala de grises es aquella en la que el valor de cada píxel es una sola muestra que representa solo una cantidad de luz, es decir, solo transporta información de intensidad. Las imágenes de este tipo, también conocidas como blanco y negro o monocromáticas, están compuestas exclusivamente por tonos de gris, que varían desde el negro en la intensidad más débil hasta el blanco en el más fuerte.

Las imágenes en escala de grises son distintas de las imágenes en blanco y negro de un bit, que en el contexto de imágenes por computadora son imágenes con solo dos colores, blanco y negro (también llamadas imágenes binarias o binarias). Las imágenes en escala de grises tienen muchos tonos de gris en el medio.

Las imágenes en escala de grises pueden ser el resultado de medir la intensidad de la luz en cada píxel según una combinación ponderada particular de frecuencias o longitudes de onda, y en tales casos son monocromáticas cuando solo una frecuencia (en la práctica, una banda estrecha de frecuencias) está capturado. Las frecuencias pueden, en principio, ser de cualquier parte del espectro electromagnético (p. Ej., Infrarrojo, luz visible, ultravioleta, etc.).

Una imagen colorimétrica (o más específicamente fotométrica) en escala de grises es una imagen que tiene un espacio de color definido en escala de grises, que mapea los valores de muestra numérica almacenados al canal acromático de un espacio de color estándar, que se basa en las propiedades medidas de la visión humana (Cedeño, 2006).

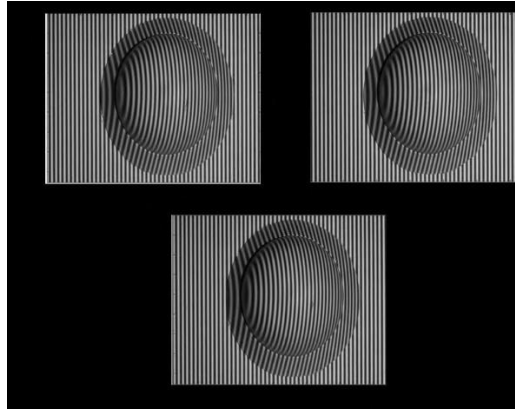


Figura 4: Escala de Grises proyección de Franjas

Filtrado Gaussiano

El filtro gaussiano se usa para emborronar imágenes y eliminar ruido. Es similar al filtro de media, pero se usa una máscara diferente, modelizando la función gaussiana.

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

Las ventajas del filtro gaussiano frente al filtro de media son:

- ✓ Es separable: es decir, en lugar de realizar una convolución bidimensional, podemos realizar dos convoluciones unidimensionales. Una en sentido horizontal y otra en sentido vertical.
- ✓ El filtro gaussiano produce un suavizado más uniforme que el filtro de media. (Cedeño, 2006)

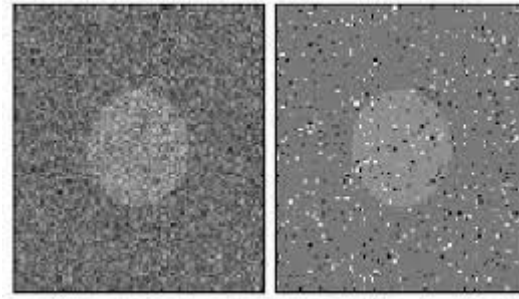


Figura 5: Imagen con ruido Gaussiano

Histograma

Un histograma es una función unidimensional de valores de píxeles que representa la distribución de los valores de píxeles en una ventana. Un histograma típico se caracteriza por la presencia de picos, o modos, que representan los valores de píxeles hallados en las características de la ventana dominante. Los efectos de ruidos provenientes de diversas fuentes, por ejemplo, error de cuantización espacial, impresión despareja, iluminación irregular o ruido eléctrico, hacen que estos picos se propaguen. Las aplicaciones de visión artificial dependen de la herramienta Histograma para determinar el conteo de píxeles, el contraste y el brillo. (Cedeño, 2006)

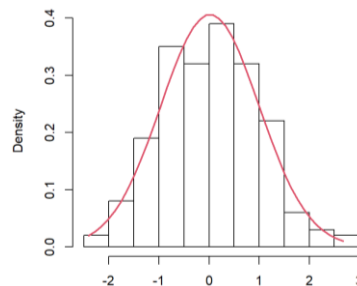


Figura 6: Imagen de un histograma con curva normal

Filtrado de Mediana

Se visita cada píxel de la imagen y se reemplaza por la mediana de los píxeles vecinos. La mediana se calcula Filtro de la mediana (median filter).

Filtros en el dominio del espacio: filtros no lineales por la mediana de los píxeles vecinos.

La mediana se calcula ordenando los valores de los píxeles vecinos en orden y seleccionando el que queda en medio.

Filtro de la mediana (median filter) Desventajas: Este filtro no es lineal. Dadas dos imágenes A y B, $\text{mediana}(A+B) \neq \text{mediana}(A) + \text{mediana}(B)$ Filtros en el dominio del espacio: filtros no lineales Ventajas: Da muy buenos resultados en caso de ruido sal y pimienta. (Cedeño, 2006)



Figura 7: Imagen del filtro de media y filtro de mediana

Contorno de Imagen

La detección de contorno es parte de un proceso de aislamiento (segmentación), que consiste en la identificación de objetos dentro de una imagen.

El contorno es simplemente un cambio en el nivel de gris que ocurre en una ubicación específica. Cuanto mayor es el cambio de nivel, más fácil resulta detectar el contorno, pero en el caso ideal cualquier cambio de nivel puede ser visto fácilmente.

La primera complicación ocurre debido a la digitalización. Es poco probable que una imagen sea muestreada de manera tal que todos los contornos correspondan exactamente con un píxel del borde. Por lo general, el cambio de nivel puede extenderse sobre varios píxeles. La posición actual del contorno se considera como el centro de la rampa que conecta el nivel bajo de gris con el nivel

alto. Esta es una rampa sólo en el mundo matemático, dado que después de que la imagen ha sido digitalizada (muestreada), la rampa tiene una apariencia escalonada.

La segunda complicación es el clásico problema del ruido. Debido a una gran cantidad de factores como la intensidad de la luz, el tipo de cámaras y lentes, el movimiento, la temperatura, efectos atmosféricos, polvo y otros; difícilmente dos píxeles que correspondan precisamente al mismo nivel de gris en la escena tengan el mismo nivel en la imagen.

El ruido es un efecto aleatorio, y es caracterizado sólo estadísticamente. El resultado del ruido en una imagen es que produce variaciones aleatorias en el nivel de un píxel a otro, y entonces las líneas suaves y rampas de los contornos ideales nunca son encontradas en las imágenes reales. (Cedeño, 2006)

Imagen 3D

El procesamiento de imágenes 3D se suele emplear en las imágenes médicas a fin de analizar imágenes DICOM o NIFTI procedentes de fuentes radiográficas como resonancias magnéticas o tomografías computarizadas. También se pueden aplicar técnicas de procesamiento de imágenes 3D en la microscopía a fin de detectar y analizar muestras de tejido o de rastrear neuronas.

Al utilizar técnicas de procesamiento de imágenes 3D para procesar las inspecciones de equipaje mediante escáner o para analizar materiales escaneados a fin de comprender su estructura.

Existen muchas técnicas que se pueden emplear al procesar datos de imágenes 3D. Estas técnicas varían según las tareas en cuestión; algunas de ellas son: importación, visualización, procesamiento y análisis de datos.

En este diagrama se indican los componentes clave de un flujo de trabajo de procesamiento de imágenes 3D. (MathWorks, 2022)

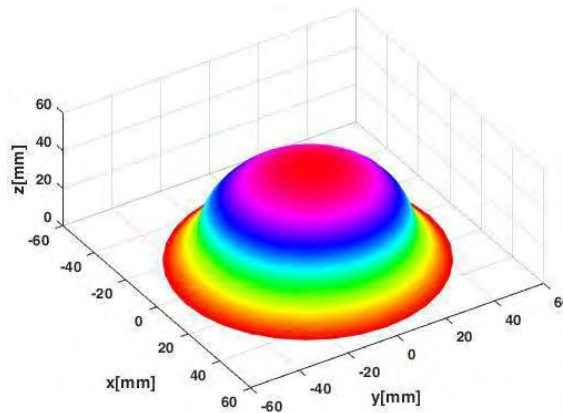


Figura 8: Procesamiento de Imagen en 3D

Desenvolvimiento de Fase

El desenvolvimiento de fase es un proceso computacional en donde una superficie llamada φ es reconstruida de su llamada forma envuelta ψ . En la ausencia de ruido $\psi(x)$ es igual $\varphi(x) + 2\pi k(x)$, en donde $k(x)$ es una función entera tal que $-\pi < \psi \leq \pi$. El problema del desenvolvimiento de fase ha ganado considerablemente la atención en los últimos años. Es una parte esencial de muchas aplicaciones de procesamiento de señales coherentes como por ejemplo Radar Interferómetro de Apertura Sintética (ISAR siglas en inglés “Interferometric Synthetic Aperture Radar”), Interferometría Óptica y Perfilometría por Transformada de Fourier. Sin embargo, los valores reales de fase no pueden ser extraídos directamente de la señal física debido a que dichos algoritmos modifican la señal de forma en que sus valores se encuentran en un rango de $\pm\pi$ radianes. Lo único que se puede obtener de esos métodos es la fase envuelta. (OCAÑA, 2017)

Como se mencionó anteriormente, la fase se encuentra limitada a valores entre $-\pi$ y π , pero cuando esta exceda el valor, la fase brincara a $-\pi$ y π . Por cada salto negativo de 2π se necesitara añadir π , mientras que por cada salto positivo se necesitaran restar otros 2π .

El proceso de desenvolvimiento intenta proveer una estimación de la función de fase real φ dada la función ψ , algo que se convierte en un problema muy difícil [6]. Para empezar, la relación entre φ y ψ es no lineal. Además, las aplicaciones interferométricas o las de triangulación activa procesan información del mundo real, por lo tanto, el ruido en la fase medida es inevitable y eso

complica aún más la ya complicada tarea. Además, otro de los problemas que se presentan es el de poder distinguir entre envolturas de fase genuinas producidas por la función arco-tangente y saltos en la fase causados por discontinuidades del objeto.

De una manera más general, el algoritmo básico de desenvolvimiento de fase 1D es como sigue: Se inicia con el primer punto, y se compara con el segundo, si la diferencia entre el primero es igual o mayor a 2π entonces se le suma o se le resta dependiendo del signo de la primera diferencia 2π a todos los puntos a partir del segundo, y se repite dicha operación hasta terminar con la línea de información. Si no existe salto se continúa hasta encontrar un salto y se realiza el desenvolvimiento.

En el caso de estar trabajando con mapa de fase, el desenvolvimiento en 1D no es suficiente, y en la mayoría de las veces es incorrecto, así que se tiene que encontrar algún método que trabaje en 2D. Dado el desarrollo de aplicaciones que trabajan con mapas de fase como lo son algunos microscopios electrónicos y los sistemas de radar de apertura sintética, se han creado muchos algoritmos bidimensionales de desenvolvimientos de fase tanto complejos como simples.

Como se ha comentado anteriormente sobre lo que es el desenvolvimiento de fase este también tiene aplicaciones militares, en medicina e industriales entre otras, sobre el procesamiento digital de imágenes, que como parte de su operación fundamental depende de la extracción de una señal de fase de la imagen de entrada.

Desenvolver fases es una de las tareas más activas de la investigación en procesamiento digital de imágenes y ha sido ampliamente investigado. Como bien se decía hay muchos algoritmos propuestos para el desenvolvimiento de fase, pero pueden producir una variedad de resultados diferentes y no una perfecta solución. (Cedeño, 2006)

Machine Learning

Machine Learning es una disciplina científica del ámbito de la Inteligencia Artificial que crea sistemas que aprenden automáticamente. Aprender en este contexto quiere decir identificar patrones complejos en millones de datos. La máquina que realmente aprende es un algoritmo que

revisa los datos y es capaz de predecir comportamientos futuros. Automáticamente, también en este contexto, implica que estos sistemas se mejoran de forma autónoma con el tiempo, sin intervención humana. (Foundation, 2020)

¿Cómo funciona el Machine Learning?

El objetivo principal es aprender a desarrollar capacidades para asociar y generalizar, replicando las facultades cognitivas del ser humano, formando modelos que generalicen la información que se les presenta para realizar sus predicciones. El ingrediente clave son los datos. El machine learning asimila una amplia gama de datos lo que se conoce como big data que no los percibe como datos, sino como una enorme lista de ejemplos prácticos con los que aprende.

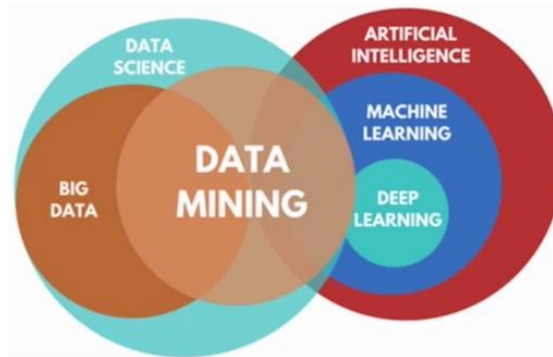


Figura 9: Funcionamiento del Machine Learning

Tipos de algoritmos:

Hay algoritmos para casi todo, desde los que buscan los patrones de cambio, los que simplemente se dedican a clasificar y hasta los que aplican las experiencias.

Los algoritmos se dividen en 3 principales categorías:

- **Supervised Learning (aprendizaje supervisado):** Este algoritmo nos permite predecir cuál es la propiedad desconocida de un elemento en concreto cuando conocemos esa misma propiedad en un conjunto de elementos.
- **Unsupervised learning (aprendizaje no supervisado):** Este algoritmo permite conocer los elementos que pertenecen a un mismo grupo o a diferentes mediante el estudio de sus características.

- Reinforcement learning (aprendizaje por refuerzo): Une los dos tipos anteriores. El algoritmo de aprendizaje recibe una valoración sobre la relevancia de la respuesta dada. Si la respuesta es correcta, el aprendizaje por refuerzo actúa como el aprendizaje supervisado, en ambos casos el aprendiz recibe información de lo que es apropiado. Sin embargo, ante las respuestas erróneas ambas aproximaciones difieren significativamente cuando el aprendiz responde de forma inadecuada.

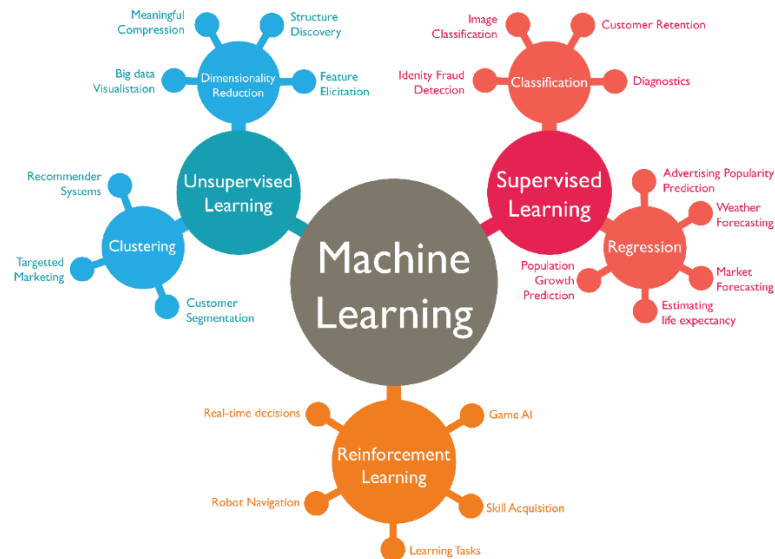


Figura 10: Tipos de Algoritmos Machine Learning

El aprendizaje supervisado le dice exactamente al aprendiz qué debería haber respondido, mientras que el aprendizaje por refuerzo solo le informa acerca de que el comportamiento ha sido inapropiado y (normalmente) cuánto error se ha cometido. La aproximación del aprendizaje por refuerzo, es más habitual en la naturaleza que en el aprendizaje supervisado. (SOLUTIONS, 2021)

El Machine Learning reconoce dentro de millones de imágenes aquellas que se le han programado para aprender a conocerlas. Modificar una app móvil basándose en los comportamientos de los usuarios, para para adaptarse a las costumbres y necesidades de este.

La ventaja de machine learning es que es posible aprovechar algoritmos y modelos para predecir resultados. Es importante asegurarse de que los científicos de datos que realizan el trabajo

usen los algoritmos correctos, asimilando los datos más apropiados (que sean precisos y limpios) y utilizando los mejores modelos de rendimiento. Si todos estos elementos se reúnen, es posible entrenar continuamente al modelo y explotar los resultados aprendiendo de los datos. La automatización de este proceso de modelado, entrenamiento del modelo y pruebas conduce a predicciones precisas para dar soporte a los cambios empresariales. (SOLUTIONS, 2021)

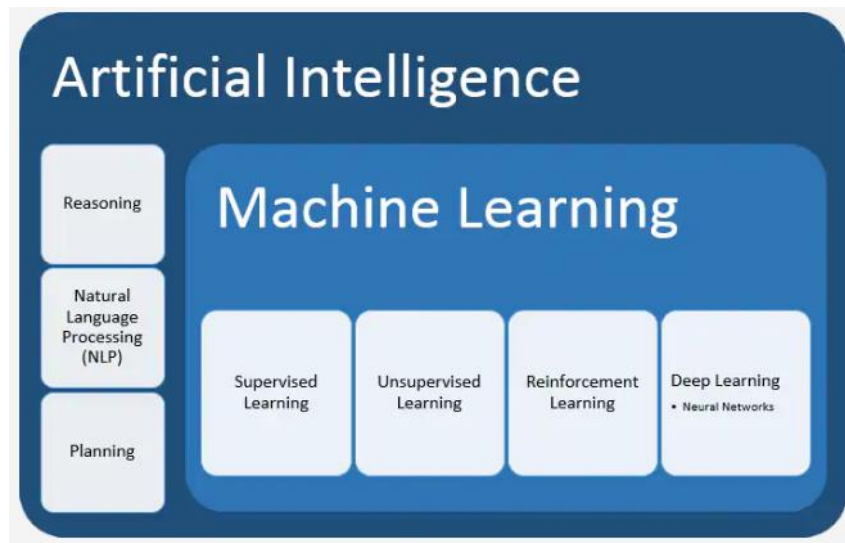


Figura 11: Modelado de un proceso Inteligencia Artificial

Python

Es un lenguaje de alto nivel de programación interpretado cuya filosofía hace hincapié en la legibilidad de su código, se utiliza para desarrollar aplicaciones de todo tipo, ejemplos: Instagram, Netflix, Spotify, Panda 3D, entre otros. Se trata de un lenguaje de programación multiparadigma, ya que soporta parcialmente la orientación a objetos, programación imperativa y, en menor medida, programación funcional. Es un lenguaje interpretado, dinámico y multiplataforma. (Python, 2022)

Posee una licencia de código abierto, denominada Python Software Foundation License.



Figura 12: Lenguaje de código abierto Python

Dispositivos externos

Proyector de video:

(Consumoteca, 2020) Un proyector de vídeo, vídeo proyector, cañón proyector o data show es un aparato óptico que recibe una señal de vídeo y proyecta la imagen correspondiente en una pantalla de proyección usando un sistema de lentes, permitiendo así mostrar imágenes fijas o en movimiento. Todos los proyectores de vídeo utilizan una luz muy brillante para proyectar la imagen, y los más modernos pueden corregir curvas, borrones y otras inconsistencias a través de los ajustes manuales.

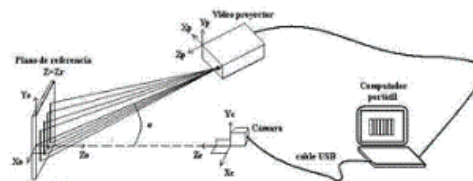


Figura 13: Funcionamiento del proyector de video para proyección de Franjas

Cámara CCD:

Un dispositivo de carga acoplada (charge-coupled device) es un circuito integrado que contiene un número determinado de condensadores enlazados o acoplados. Bajo el control de un circuito interno, cada condensador puede transferir su carga eléctrica a uno o a varios de los condensadores que estén a su lado en el circuito impreso. (Spiegato, 2022)

La alternativa digital a los CCD son los dispositivos CMOS (complementary metal oxide semiconductor) utilizados en algunas cámaras digitales y en numerosas cámaras web. En la actualidad los CCD son mucho más populares en aplicaciones profesionales y en cámaras digitales.



Figura 14: Cámara CCD

Matlab:

La plataforma de MATLAB está optimizada para resolver problemas científicos y de ingeniería. El lenguaje de MATLAB, basado en matrices, es la forma más natural del mundo para expresar las matemáticas computacionales. Las gráficas integradas facilitan la visualización de los datos y la obtención de información a partir de ellos. Una vasta biblioteca de herramientas (Toolboxes) integradas le permite empezar a trabajar inmediatamente con algoritmos esenciales para su dominio. El entorno de escritorio invita a experimentar, explorar y descubrir. Todas estas herramientas y funciones de MATLAB están probadas rigurosamente y diseñadas para trabajar juntas. (Carlos, 2015)



Figura 15: Software para las expresiones matemáticas

Capítulo III

Estado del Arte

Presenta los resultados de una investigación teórico-experimental de la técnica de proyección de franjas. El trabajo se realiza para evaluar el desempeño de la técnica en procedimientos de metrología facial. El montaje experimental consta de un video proyector y una cámara CCD; para recuperar la fase discontinua del patrón de franjas se usa el algoritmo "Phase

Shifting” con cuatro imágenes y se aplica un procedimiento típico de “Unwrapping” para obtener la fase continua relativa. Para llegar a la fase continua absoluta, se incluye una etapa de triangulación lineal en la calibración del sistema. En los resultados se muestra cómo, con la información que se obtiene de la topografía superficial de la piel, se puede determinar la longitud y la profundidad media de las estrías cutáneas, lo cual evidencia el potencial de la técnica en aplicaciones biomédicas. (González, Técnica en procedimiento de metrología facial, 2012)

Presentan una comparación entre estas dos técnicas en términos de su desempeño como insumo potencial para la generación de archivos CAM de objetos con dimensiones máximas laterales del orden de 15 mm, la industria dental en el desarrollo de sistemas que permitan rápidamente producir prototipos de piezas dentales, la técnica de escaneo láser perfila como la más apropiada para la generación de archivos CAM de piezas dentales con diferente morfología. (N. Múnera, 2010)

Presenta técnicas de metrología óptica que ofrecen una solución que permiten realizar un levantamiento digital de la topografía y contorno de la superficie del objeto. Entre las técnicas de reconstrucción 3D por métodos ópticos se encuentran la proyección de franjas y el escaneo láser. Por utilizar como sonda de medición radiación electromagnética, esas metodologías captan particular interés debido a que no destruye las muestras en el momento del análisis, lo que asegura que la superficie del objeto no se verá intervenida por la técnica de medición. (Hugo I. Medellín Castillo, 2011)

Desarrollo la metodología que se llevó a cabo para la realización de un sistema semiautomático de captura de imágenes y desenvolvimiento de fase que podrá realizar una reconstrucción de objetos en 3D, así como la obtención del perfil de dichos objetos, para lograr un objetivo, Para controlar el sistema se creó una interfaz gráfica de usuario (GUIDE) implementada mediante MATLAB, y que permitirá el análisis semiautomático para ambas técnicas de reconstrucción tridimensional. (Canchola, 2012)

Capítulo IV

Metodología

La técnica de proyección de franjas consiste en proyectar sobre el objeto a reconstruir franjas paralelas blancas y negras, que se deforman por la topografía superficial del objeto.

Para crear nuestra aplicación se aplicó la transformada de Fourier para el procesamiento de imágenes, con el objetivo de mejorar la calidad o facilitar la búsqueda de información. En el proceso de filtrado usando el conjunto de técnicas englobadas dentro del preprocesamiento de imágenes cuyo objetivo es obtener, a partir de una imagen origen, otra final cuyo resultado sea más adecuado para una aplicación específica mejorando ciertas características de la misma que posibilite efectuar operaciones del procesado sobre ella. Se consideran los filtros como operaciones que se aplican a los píxeles de una imagen digital para optimizarla, enfatizar cierta información o conseguir un efecto especial en ella. El proceso de filtrado puede llevarse a cabo sobre los dominios de frecuencia y/o espacio. Los filtros de frecuencia procesan una imagen trabajando sobre el dominio de la frecuencia en la Transformada de Fourier de la imagen.

- Se aplica la Transformada de Fourier
- Se multiplica posteriormente por la función del filtro que ha sido escogido
- Se re-transforma al dominio espacial empleando la Transformada Inversa de Fourier.

$$f(t) = \frac{1}{2\pi} V.P. C \int_{\gamma-i\infty}^{\gamma+i\infty} e^{st} F(S) ds$$

Teorema de la convolución: frecuencia

$$G(u, v) = F(u, v) * H(u, v)$$

F(u,v): transformada de Fourier de la imagen original y H(u,v): filtro atenuador de frecuencias. (Otarola, 2015)

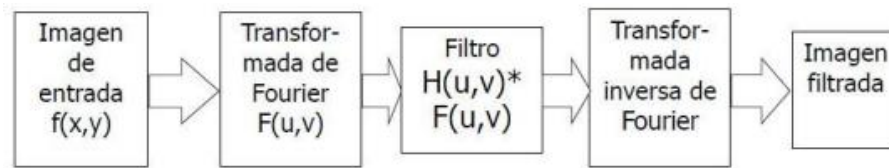


Figura 16: Línea de Procesamiento de imágenes

Al aplicar el aprendizaje profundo para el procesamiento de imágenes en nuestra aplicación, lo que va a realizar es un procesamiento de imágenes, como eliminar el ruido de la imagen y filtrado de la imagen, utilizando redes neuronales convolucionales, para aprender representaciones útiles de características directamente a partir de datos. Por ejemplo, puede usar una red neuronal previamente entrenada para identificar y eliminar artefactos como el ruido de las imágenes.

El procesamiento de imágenes se conoce comúnmente como procesamiento de imágenes digitales y el dominio más utilizado es visión por computadora. Los algoritmos de procesamiento de imágenes y los algoritmos de visión por computadora (CV) aceptan imágenes como entrada; sin embargo, en el procesamiento de imágenes, la salida también es una imagen, pero en la visión por computadora, la salida puede contener algunas características / información sobre la imagen. Los datos que recopilamos o generamos son en gran parte datos sin procesar, es decir, no son adecuados para su uso en aplicaciones directamente por varias razones posibles. Por lo tanto, primero debemos analizarlo, realizar el pre procesamiento necesario y luego utilizarlo.

Cada imagen está representada por un conjunto de píxeles, es decir, una matriz de valores de píxeles. Para una imagen en escala de grises, los valores de píxeles están entre 0 y 255 e indican la intensidad de esos píxeles. Si se trata de una imagen en color, hay tres formas: rojo, verde y azul (RGB). Por tanto, habría tres matrices de este tipo para una sola imagen.

Dado que se utilizó las siguientes librerías OpenCV a través de Python, es un requisito implícito que ya tenga Python \$pip install opencv-python, import cv2. Cambiaremos el tamaño de la imagen a escala de grises y para cargar la imagen con la imread () función, podemos recuperar algunas propiedades simples sobre ella, como el número de píxeles y las dimensiones, eliminación de ruido de la imagen como el ruido gaussiano y para recuperar la imagen original de una imagen de ruido, podemos pasar a la siguiente aplicación, la detección de bordes usando Canny Edge

Detector, para obtener las fases desenvuelta y envuelta ocuparemos la librería Unwrapping, una función que es utilizada para desenvolver una matriz determinada transformándola en deltas y en $2*\pi$ complement.

Capítulo V

Diseño de la interfaz gráfica (GUI)

Diseño Tratamiento de Imágenes

El desarrollo de una Interfaz Gráfica de Usuario la cual tiene la finalidad de que el sistema pueda ser utilizado por usuarios que no tengan conocimientos avanzados en el área y programación generando un entorno multimedia agradable y sencillo de utilizar. Una aplicación GUIDE consta de cargar una imagen en formato jpg. o png., esto se realiza a través de un botón de control abrir creado en la interfaz, su función será buscar la imagen que deseamos aplicar el tratamiento.

En la figura 17, muestra el diseño de la ventana principal del software, en el cual se visualiza el menú para el tratamiento de la imagen en el botón inicio.



Figura 17: Inicio de la interfaz Gráfica de Proyección de Franjas

En la figura 18, el botón Inicio, se observan cada una de las funciones que puede cumplir el software para el tratamiento de imágenes.



Figura 18: Menú de la interfaz Gráfica Proyección de franjas

Al elegir la opción de tratamiento de imágenes, en la pantalla se visualiza que tipo de tratamiento se puede aplicar a una imagen.

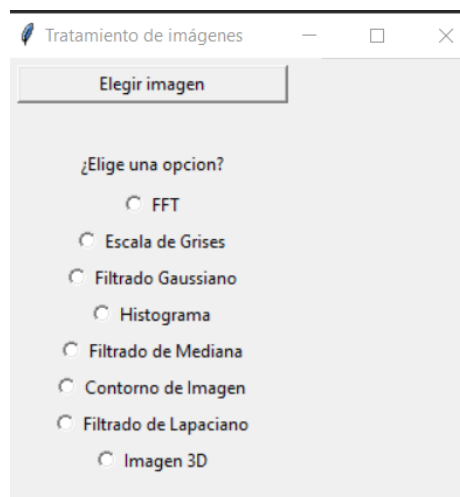


Figura 19: Menú de los Tratamiento que se pueden dar a una Imagen

En el menú de la transformada de Fourier de la imagen 20, como objetivo es obtener las frecuencias de una función durante el cual existió la señal; se obtiene un solo espectro de frecuencias para toda la función.

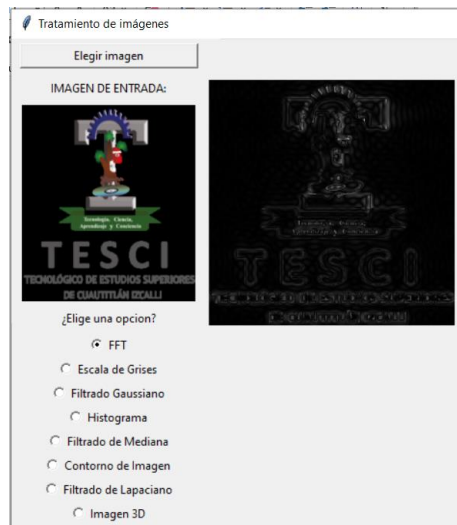


Figura 20: Resultado del tratamiento de Transformada de Fourier

En la figura 21, el menú escala de grises el objetivo es ayudar a medir la intensidad de la luz en cada píxel según una combinación ponderada particular de frecuencias o longitudes de onda.



Figura 21: Resultado del Tratamiento a Escala de Grises

El filtro gaussiano que se visualiza en la figura 22, se usa para emborronar imágenes y eliminar ruido de dicha imagen.

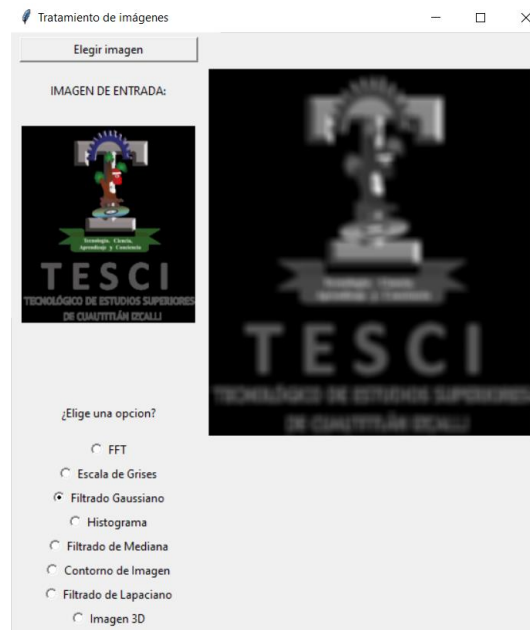


Figura 22: Resultado de Tratamiento de Filtrado de Gaussiano

La figura 23, menú de histograma, ayuda a medir la presencia de picos, o modos, que representan los valores de píxeles hallados en las características de la imagen.

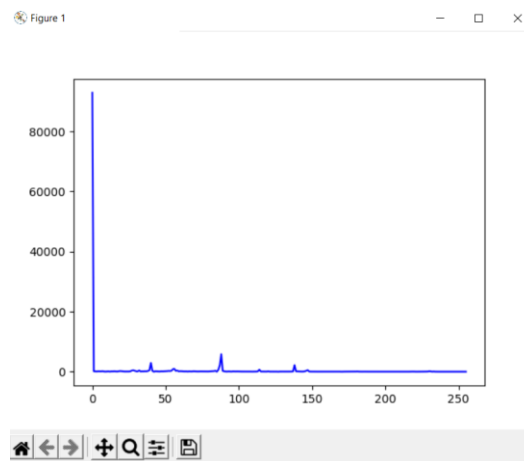


Figura 23: Resultado de un tratamiento de imagen y su Histograma

En la figura 24, menú mediano, su objetivo es que cada píxel de la imagen, se reemplaza por la mediana de los píxeles vecinos. La mediana se calcula ordenando los valores de los píxeles vecinos en orden y seleccionado el que queda en medio.



Figura 24: Resultado de tratamiento de Filtrado de Mediana

En la figura 25, El menú de Contorno de imagen, ayuda a detectar el aislamiento que consiste en la identificación de objetos dentro de una imagen.



Figura 25: Resultado de tratamiento de Contorno de Imagen

La figura 26, el menú Imagen 3D, su objetivo es dar como resultado un diagrama que indique los componentes clave de un flujo de trabajo de procesamiento de imágenes 3D.

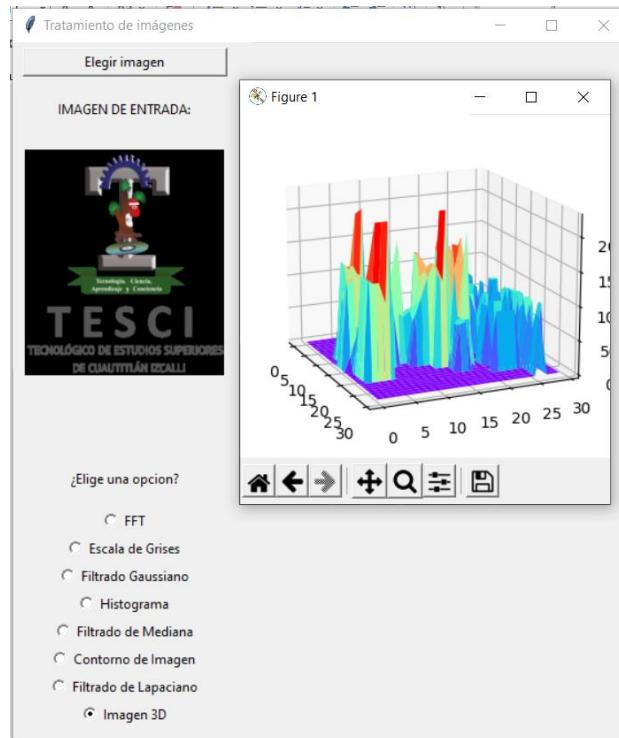


Figura 26: Resultado de tratamiento en una imagen en 3D

Capítulo VI

Funcionamiento de la interfaz gráfica (GUI)

Una aplicación GUIDE consta de cargar una imagen en formato jpg. o png., esto se realiza a través de un botón de control elegir imagen, creado en la interfaz, su función será buscar la imagen que deseamos aplicar el tratamiento. Como se muestra en la Figura 27 y Figura 28.

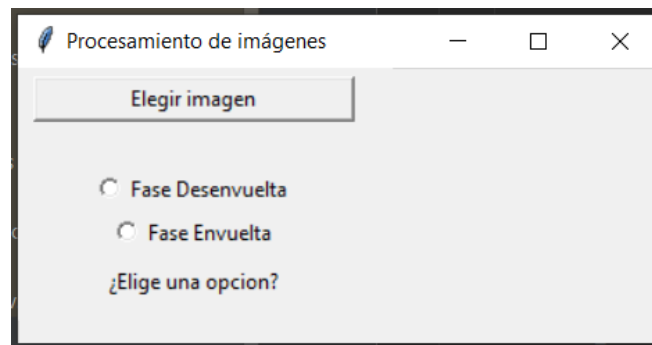


Figura 27: Interfaz desarrollada para procesamiento de imágenes.

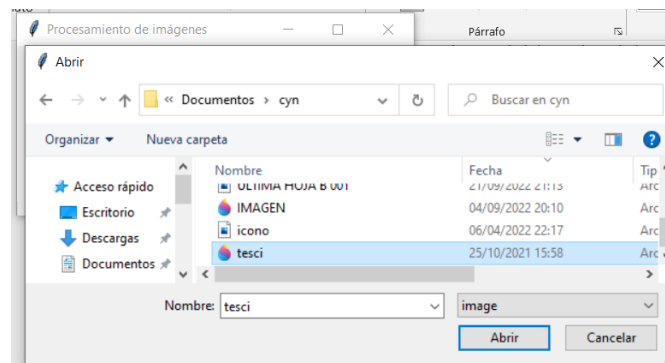


Figura 28: Selección de imágenes para el procesamiento

Una vez que se encuentra cargada la imagen, en la parte superior izquierda de la aplicación, se encontrara un menú con los filtros necesarios para el tratamiento de la imagen como se muestra en la Figura 29.



Figura 29: Selección de imagen para su tratamiento

El primer filtro aplicado, es la fase Envuelta Figura 30, el objetivo es quitar el mayor posible de ruido a la imagen, eliminar aquellos píxeles cuyo nivel de intensidad es muy diferente a la imagen origen puede estar tanto en el proceso de adquisición de la imagen como en el de transmisión, además de destacar los bordes que se localizan en la imagen. es necesario desenvolver la fase para recuperar la señal subyacente no envuelta, demostraremos un algoritmo implementado en skimag, todas las imágenes de una, dos y tres dimensiones se pueden desenvolver usando skimage.

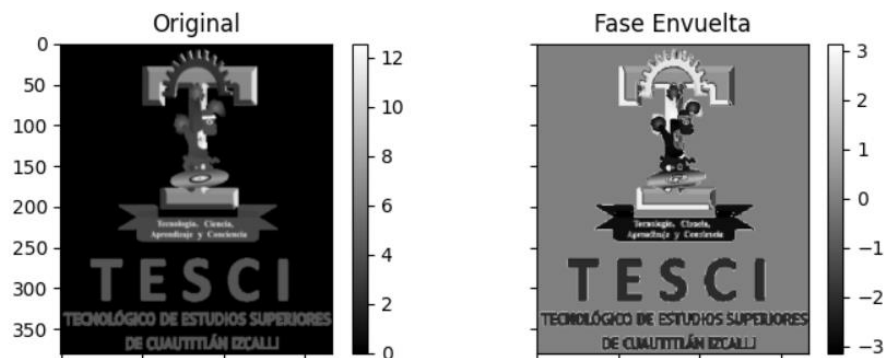


Figura 30: Aplicación de la fase Envuelta

El siguiente procedimiento es aplicar el desenvolvimiento de la fase, la cual acepta matrices enmascaradas y también los límites cíclicos para conectar los bordes de una imagen.

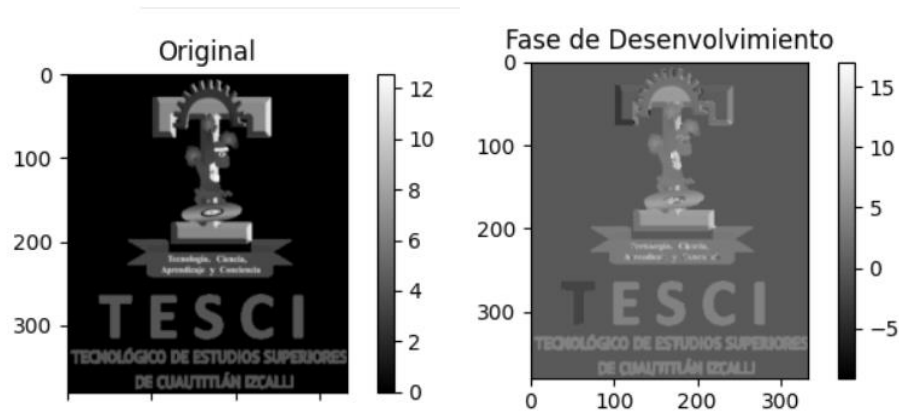


Figura 31: Aplicación de la fase desenvuelta

Por último, al tener la imagen de la fase desenvuelto menos la imagen origen, en la que se visualizará la imagen original aplicando un corrimiento a la fase por medio del teorema de 4 pasos para poder obtener la topografía del objeto y después por medio de las líneas rectas, se obtendrá una imagen con la fase desenvuelta como se observa en la Figura 6.

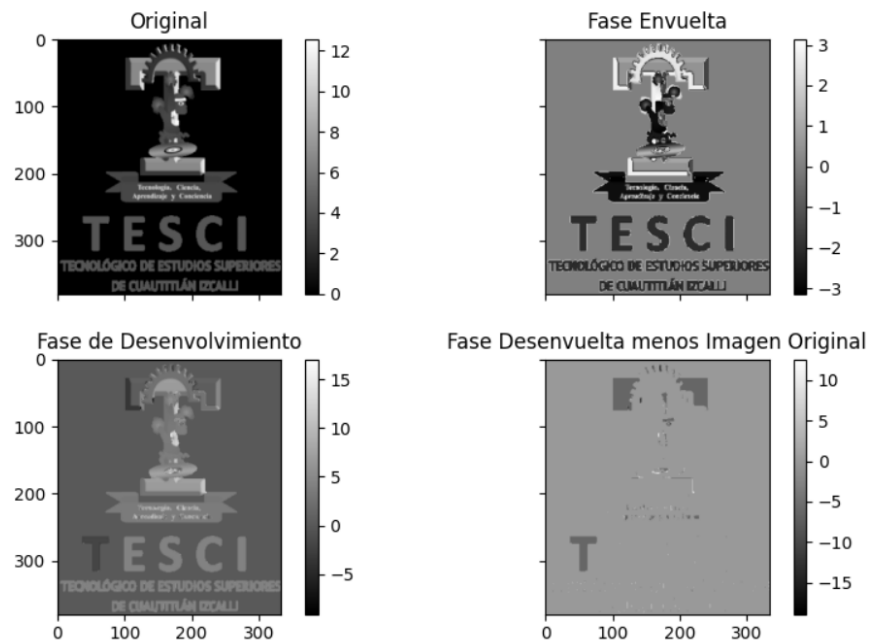


Figura 32: Envolvimiento y desenvolvimiento de la fase

Finalmente, con la imagen filtrada y con la obtención de la fase desenvuelta se obtiene el objeto reconstruido en 3D, generando una imagen con la fase desenvuelta, que se encuentra ligada a la forma en 3D del objeto lo que permite generar la nube de puntos del mismo, como se observa en la Figura 33.

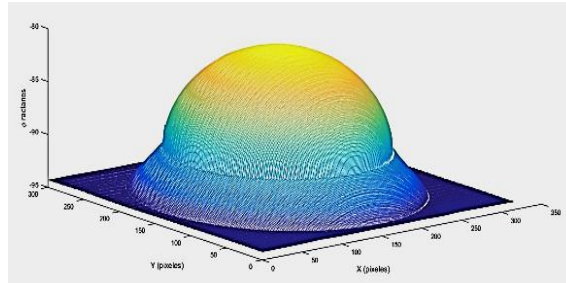


Figura 33: Imagen de la fase desenvuelta asociada a la forma 3D del objeto

Conclusiones

Con la elaboración de un software, un proyector de video que captura las imágenes de un objeto a través del método de proyección de franjas se comprueba la efectividad que tiene el uso de Machine Learning y el aprendizaje profundo para el procesamiento de imágenes, lo que le facilita el tratamiento de las mismas de una forma muy funcional.

Se verifico la efectividad de la técnica de proyección de franjas para reconstruir objetos en tres dimensiones y se hizo uso de los algoritmos de phase shifting que permite recuperar la fase de las imágenes y lograr un objeto final en 3d.

La técnica de proyección de franjas puede ser implementada en cualquier aplicación relacionada al procesamiento de imágenes con el uso de librerías que aporta machine learning, nos da la oportunidad de crear un tratamiento de imagen con un aprendizaje rápido. Una vez definidas las técnicas de tratamiento de imagen a usarse durante este trabajo, se realizó el análisis matemático de cada una de las técnicas para la obtención de las imágenes de cada fase.

En este proyecto se hicieron experimentos considerando imágenes de forma aleatoria, como resultado se demostró que la mayoría de las imágenes tuvieron un procesamiento exitoso, pero también hubo algunas imágenes donde se presentaron resultados de procesamiento no exitoso, en el cual se marcan las diferencias en la información de sus patrones y que, a pesar de ello, se llegó a los resultados propuestos en la metodología.

Bibliografía

1. CANCHOLA, M. A. (2012). MEDICIÓN DE PERFILES UTILIZANDO TÉCNICAS DE PROYECCIÓN DE FRANJAS.
2. CARLOS, J. (2015). *WEEBLY.COM*. WEEBLY.COM:
<HTTPS://JUANCARLOSUSOMATLAB2015.WEEBLY.COM/DEFINICION-MATLAB.HTML>
3. CEDEÑO, O. D. (AGOSTO DE 2006). *ALGORITMO RÀPIDO PARA RECUPERAR FASE DE UN SOLO INTERFEROGRAMA CON FRANJAS ABIERTAS Y CERRADAS*. CENTRO DE INVESTIGACIONES EN MATEMÀTICASB A.C.:
<HTTPS://CIMAT.REPOSITORIOINSTITUCIONAL.MX/JSPUI/BITSTREAM/1008/48/2/TE20191.PDF>
4. CONSUMOTECA. (2020).
<HTTPS://WWW.CONSUMOTECA.COM/ELECTRONICA/PROYECTOR-DE-VIDEO/>
5. DANIEL JARAMILLO CHAMBA, L. C. (JUNIO DE 2022). TRANSFORMADA DE FOURIERDISCRETA PARA EL ANÁLISIS DE SEÑALES DIGITALES. LA LIBERTAD, PENÍNSULA DE SANTA ELENA, UPSE, ECUADOR: REVISTA CIENTÍFICA Y TECNOLÓGICA UPSE.
6. FOUNDATION, P. S. (2020). *IBM*. <HTTPS://WWW.IBM.COM/MX-ES/ANALYTICS/MACHINE-LEARNING>
7. GONZÁLEZ, A. L. (09 DE 2012). PROYECCIÓN DE FRANJAS EN METROLOGÍA ÓPTICA FACIAL1. COLOMBIA.
8. GONZÁLEZ, A. L. (2012). *TECNICA EN PROCEDIMIENTO DE METROLOGIA FACIAL*.
9. GONZALEZ, J. P. (SEPTIEMBRE DE 2006). *ECUACIONES DIFERENCIALES,SERIES DE FOURIER, TRANSFORMADA DE FOURIRER Y LAPLACE*. UNIVERSIDAD DE GRANADA:
HTTPS://WWW.UGR.ES/~FJPerez/TEXTOS/EEDD_LAPLACE_FOURIER.PDF
10. HUGO I. MEDELLÍN CASTILLO, M. A. (2011). DESARROLLO DE UN SISTEMA DE VISUALIZACIÓN Y RECONSTRUCCIÓN 3D DE MODELOS ANATÓMICOS A PARTIR DE IMÁGENES MÉDICAS.
11. MATHWORKS. (2022). *¿QUÉ ES EL PROCESAMIENTO DE IMÁGENES 3D?* MATLAB:
<HTTPS://LA.MATHWORKS.COM/SOLUTIONS/IMAGE-VIDEO-PROCESSING/3D-IMAGE-PROCESSING.HTML>
12. N. MÚNERA, G. J.-S. (2010). *TÉCNICAS DE PROYECCIÓN DE FRANJAS Y DE ESCANEEO LÁSER PARA LA RECONSTRUCCIÓN 3D Y GENERACIÓN DE ARCHIVOS CAM DE PIEZAS DENTALES*.

13. OCAÑA, A. L. (2017). DESENVOLVIMIENTO DE FASE. *TESIS DESENVOLVIMIENTO DE FASE*. VALLE DE CHALCO SOLIDARIDAD, MÉXICO.
14. OLAVARRÍA, S. (2002). *TELEVISION DIGITAL*.
[HTTP://WWW2.ELO.UTFSM.CL/~ELO340/APUNTES/TVD/CAP4.PDF](http://www2.elo.utfsm.cl/~elo340/apuntes/tvd/cap4.pdf)
15. OTAROLA, L. A. (AGOSTO DE 2015). TRANSFORMADA DE FOURIER PARA EL PROCESAMIENTO DIGITAL DE IMÁGENES.
[HTTP://LCR.UNS.EDU.AR/FVC/NOTASDEAPLICACION/FVC-LUCAOTAROLA.PDF](http://lcr.uns.edu.ar/fvc/notasdeaplicacion/fvc-lucaotarola.pdf)
16. PYTHON. (2022). [HTTPS://WWW.PYTHON.ORG/](https://www.python.org/)
17. SOLUTIONS, L. D. (2021). *MACHINE LEARNING*.
[HTTPS://WWW.LISDATASOLUTIONS.COM/BLOG/MACHINE-LEARNING-SATELITAL/](https://www.lisdatasolutions.com/blog/machine-learning-satelital/)
18. SPIEGATO. (2022). [HTTPS://SPIEGATO.COM/ES/QUE-ES-UNA-CAMARA-CCD](https://spiegato.com/es/que-es-una-camara-ccd)
19. VÁSQUEZ, M. J. (2014). OPTIMIZACIÓN EN EL ANÁLISIS DE FOURIER CON PATRONES DE FRANJAS NORMALIZADAS. PUEBLA, PUEBLA, MEXICO.

Anexos

Códigos en Python. Para el tratamiento de imágenes, en el que se emplea las librerías y de machine learning, el menú inicio y la forma de cargar una imagen a la interfaz.

```
import tkinter
from tkinter import *
from tkinter import filedialog
import cv2
import numpy as np
from matplotlib import pyplot as plt
import numpy as np
from mpl_toolkits.mplot3d import Axes3D
from PIL import Image, ImageTk
import imutils

def deteccion_color():
    global image

def elegir_imagen():
    # Especificar los tipos de archivos, para elegir solo a las imágenes
    path_image = filedialog.askopenfilename(filetypes = [
        ("image", ".jpeg"),
        ("image", ".png"),
        ("image", ".jpg")])

    if len(path_image) > 0:
        global image

        # Leer la imagen de entrada y la redimensionamos
        image = cv2.imread(path_image)
        image= imutils.resize(image, height=380)

        # Para visualizar la imagen de entrada en la GUI
        imageToShow= imutils.resize(image, width=180)
        imageToShow = cv2.cvtColor(imageToShow, cv2.COLOR_BGR2RGB)
        im = Image.fromarray(imageToShow )
        img = ImageTk.PhotoImage(image=im)
```

```
lblInputImage.configure(image=img)
lblInputImage.image = img

# Label IMAGEN DE ENTRADA
lblInfo1 = Label(root, text="IMAGEN DE ENTRADA:")
lblInfo1.grid(column=0, row=1, padx=5, pady=5)

# Al momento que leemos la imagen de entrada, vaciamos
# la imagen de salida y se limpia la selección de los
# radiobutton
lblOutputImage.image = ""
selected.set(0)

# Creamos la ventana principal
root = Tk()
#root.geometry("500x300")
root.title("Tratamiento de imágenes")

# Label donde se presentará la imagen de entrada
lblInputImage = Label(root)
lblInputImage.grid(column=0, row=2)

# Label donde se presentará la imagen de salida
lblOutputImage = Label(root)
lblOutputImage.grid(column=1, row=1, rowspan=6)

# Label ¿Qué color te gustaría resaltar?
lblInfo2 = Label(root, text="¿Elige una opción?", width=25)
lblInfo2.grid(column=0, row=6, padx=5, pady=5)

# Creamos los radio buttons y la ubicación que estos ocuparán
selected = IntVar()
rad1 = Radiobutton(root, text='FFT', width=25,value=1, variable=selected, command=
deteccion_color)
rad2 = Radiobutton(root, text='Escala de Grises',width=25, value=2, variable=selected, command=
deteccion_color)
rad3 = Radiobutton(root, text='Filtrado Gaussiano',width=25, value=3, variable=selected,
command= deteccion_color)
rad4 = Radiobutton(root, text='Histograma',width=25, value=4, variable=selected, command=
deteccion_color)
rad5 = Radiobutton(root, text='Filtrado de Mediana',width=25, value=5, variable=selected,
```



```
command= deteccion_color)
rad6 = Radiobutton(root, text='Contorno de Imagen',width=25, value=6, variable=selected,
command= deteccion_color)
rad7 = Radiobutton(root, text='Filtrado de Lapaciano',width=25, value=7, variable=selected,
command= deteccion_color)
rad8 = Radiobutton(root, text='Imagen 3D',width=25, value=8, variable=selected, command=
deteccion_color)
rad1.grid(column=0, row=9)
rad2.grid(column=0, row=10)
    rad3.grid(column=0, row=11)
rad4.grid(column=0, row=12)
rad5.grid(column=0, row=13)
rad6.grid(column=0, row=14)
rad7.grid(column=0, row=15)
rad8.grid(column=0, row=16)

# Creamos el botón para elegir la imagen de entrada
btn = Button(root, text="Elegir imagen", width=25, command=elegir_imagen)
btn.grid(column=0, row=0, padx=5, pady=5)

def menu1():
    ven_trata1 = Toplevel()
    ven_trata1.title("Tratamiento de Imagen")

def menu2():
    ven_trata2 = Toplevel()
    ven_trata2.title("Método de Takaeda")

def menu3():
    ven_trata3 = Toplevel()
    ven_trata3.title("Desplazamiento de Fase")

def menu4():
    ven_trata4 = Toplevel()
    ven_trata4.title("Procesamiento de Imagen")

def menu5():
    ven_trata5 = Toplevel()
    ven_trata5.title("Desenvolvimiento de Fase")
```

```

# creamos la pantalla principal
root = Tk()
root.title("Proyección de Franjas")
root.geometry("500x300")

Label(root, text="Tecnológico de Estudios Superiores de Cuautitlán Izcalli").pack()

imagen = PhotoImage(file="tesci.png")
Label(root, image=imagen, bd=0).pack()

Label(root, text="Proyecto Proyección de Franjas :").pack()

Label(root, text="Elaborado por :").pack()
Label(root, text="Cynthia Walles Gutiérrez").pack()

# creamos el menu
menubar = Menu(root)

filemenu = Menu(menubar, tearoff=0)
filemenu.add_command(label="Tratamiento de Imagenes", command=menu1)
filemenu.add_command(label="Método de Takaeda", command=menu2)
filemenu.add_command(label="Desplazamiento de Fase", command=menu3)
filemenu.add_command(label="Procesamiento de Imagen", command=menu4)

filemenu.add_command(label="Desarrollo de Fase", command=menu5)
filemenu.add_separator()
filemenu.add_command(label="Exit", command=root.quit)

menubar.add_cascade(label="Inicio", menu=filemenu)

root.config(menu=menubar)

root.mainloop()

Códigos en Python. Para el tratamiento de imágenes, en el que se emplea la transformada
de Fourier de una imagen y sus algoritmos correspondientes.

if selected.get() == 1: #FFT
#IMAGEN ORIGEN
Nf = 255#380
Nc = 255#560

```

```
image = cv2.cvtColor(image, cv2.COLOR_RGB2BGR)
img = cv2.resize(image,(Nc,Nf))
# PASA ESCALA grises
i_gray = cv2.cvtColor(img, cv2.COLOR_RGB2GRAY)
i_grayd = np.float64(i_gray)
#transformada
Fuv = np.fft.fft2(i_grayd)
Fuv = np.fft.fftshift(Fuv)

Fuv_abs=np.abs(Fuv)
#espectro logaritmico
Fuv_log = 20 * np.log10(Fuv_abs)

# Para visualizar la imagen en lblOutputImage en la GUI
img = Image.fromarray(np.uint8(255 * Fuv_log / np.max(Fuv_log)))
img = ImageTk.PhotoImage(img)
lblOutputImage.configure(image=img)
lblOutputImage.image = img

#diseño del paso alto
f1 = np.arange(-Nf/2+1,Nf/2+1,1)
f2 = np.arange(-Nc/2+1,Nc/2+1,1)
[X,Y]=np.meshgrid(f1,f2)
D=np.sqrt(X**2+Y**2)
D=D/np.max(D)
#DEFINIR RADIO
D1=0.12
#CREACION DEL FILTRO EN 2D
Huv=np.zeros((Nf, Nc))
for i in range(Nf):
    for j in range(Nc):
        if(D[i,j]<D1):
            Huv[i,j]=1
#convertir paso alto
Huv=1-Huv

# Para visualizar la imagen en lblOutputImage en la GUI
img2 = Image.fromarray(np.uint8(255 * Huv))
img2 = ImageTk.PhotoImage(img2)
lblOutputImage.configure(image=img2)
```

```

lblOutputImage.image = img2

#filtrado en frecuencia
Guv = Huv * Fuv
#Magnitud
Guv_abs=np.abs(Guv)
Guv_abs=np.uint8(255*Guv_abs/np.max(Guv_abs))
# Para visualizar la imagen en lblOutputImage en la GUI
img3 = Image.fromarray(np.uint8(255 * Guv_abs))
img3 = ImageTk.PhotoImage(img3)
lblOutputImage.configure(image=img3)
lblOutputImage.image = img3

#ifft2
gxy=np.fft.ifft2(Guv)
gxy=np.abs(gxy)
gxy=np.uint8(gxy)

# Para visualizar la imagen en lblOutputImage en la GUI
img4 = Image.fromarray(gxy)
img4 = ImageTk.PhotoImage(img4)
lblOutputImage.configure(image=img4)
lblOutputImage.image = img4
#guardar
cv2.imwrite('FFT_image.png', img4)
cv2.waitKey(0)
cv2.destroyAllWindows()

```

Códigos en Python. Para el tratamiento de imágenes, en el que se emplea la escala de grises de una imagen y sus algoritmos correspondientes.

```

if selected.get() == 2: #Escala de grises
gray_img = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)

# Para visualizar la imagen en lblOutputImage en la GUI
gray_img1 = Image.fromarray(gray_img)
img = ImageTk.PhotoImage(image=gray_img1)
lblOutputImage.configure(image=img)
lblOutputImage.image = img

```

Códigos en Python. Para el tratamiento de imágenes, en el que se emplea la filtro de Gaussiano de una imagen y sus algoritmos correspondientes.

```
if selected.get() == 3: #filtrado Gaussiano
aux = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY) # Convertir a escala de grises
blur = cv2.GaussianBlur(aux, (9, 9), 9)

blur_img = Image.fromarray(blur)
blur_img_tk_global = ImageTk.PhotoImage(image=blur_img)
lblOutputImage.configure(image=blur_img_tk_global)
lblOutputImage.image = blur_img_tk_global
```

Códigos en Python. Para el tratamiento de imágenes, en el que se emplea la filtro de Mediana de una imagen y sus algoritmos correspondientes.

```
if selected.get() == 5: #mediana
media_gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
median = cv2.medianBlur(media_gray, 5)
median1 = Image.fromarray(median)
img = ImageTk.PhotoImage(image=median1)
lblOutputImage.configure(image=img)
lblOutputImage.image = img
```

Códigos en Python. Para el tratamiento de imágenes, en el que se emplea el filto de un histograma de una imagen y sus algoritmos correspondientes.

```
if selected.get() == 4: #Histograma
#LEE IMAGEN0
eg1 = cv2.cvtColor(image, cv2.COLOR_RGB2BGR)

#CALCULA HISTOGRAMA
h1 = cv2.calcHist([eg1],[0],None,[256],[0,250])
plt.plot(h1, color='blue')
fig = np.uint8(h1)
plt.show()
```

Códigos en Python. Para el tratamiento de imágenes, en el que se emplea el filtro del contorno de una imagen y sus algoritmos correspondientes.

```
if selected.get() == 6: #contorno y 3d
#imagen escala de grises
```

```
gray = cv2.cvtColor(image, cv2.COLOR_RGB2GRAY)
h=np.array([[1,0,-1],[1,0,-1],[1,0,-1]])
```

```
forma = np.shape(gray)
img2 = np.zeros(forma)
for x in list(range(1,forma[0]-1)):
    for y in list(range(1,forma[1]-1)):
        suma = 0
        for i in list(range(-1,2)):
            for j in list(range(-1, 2)):
                suma=gray[x-i,y-j]*h[i+1,j+1]+suma
            img2[x,y]=suma
maxs=np.max(img2)
img2=img2*255/maxs
img2=img2.astype(np.uint8)
```

```
# Para visualizar la imagen en lblOutputImage en la GUI
fil = Image.fromarray(img2)
fil = ImageTk.PhotoImage(fil)
lblOutputImage.configure(image=fil)
lblOutputImage.image = fil
```

Códigos en Python. Para el tratamiento de imágenes, en el que se emplea el filtro de Laplaciano de una imagen y sus algoritmos correspondientes.

```
if selected.get() == 7: # filtrado LAPLACIANO
aux1 = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY) # Convertir a escala de grises
laplac = cv2.Laplacian(aux1,ddepth=cv2.CV_64F,ksize=11)
mg = np.max(laplac)
laplac=laplac*255/mg
bordes = laplac.astype(np.uint8)
```

```
lapl_img = Image.fromarray(bordes)
lapl_img = ImageTk.PhotoImage(image=lapl_img)
lblOutputImage.configure(image=lapl_img)
lblOutputImage.image = lapl_img
```

Códigos en Python. Para el tratamiento de imágenes, en el que se emplea el filtro de 3D de una imagen y sus algoritmos correspondientes.

```
if selected.get() == 8: # 3D

img = cv2.cvtColor(image, cv2.COLOR_RGB2BGR)

# resize it, make sure the pixel is not too large, it will consume a lot of resource
img = cv2.resize(img, (30, 30))

# convert the gray
gray_image = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
height, width = gray_image.shape

fig = plt.figure()
ax = Axes3D(fig)
X = np.arange(0, width, 1)
Y = np.arange(0, height, 1)
X, Y = np.meshgrid(X, Y)
R = gray_image.view()
ax.plot_surface(X, Y, R, rstride=1, cstride=1, cmap='rainbow')
plt.show()

# Label IMAGEN DE SALIDA
lblInfo3 = Label(root, text="IMAGEN DE SALIDA:", font="bold")
lblInfo3.grid(column=1, row=0, padx=5, pady=5)
```

