FILTRADO DE IMÁGENES

PRÁCTICA nO. 2

María Camila Restrepo Duque **e-mail:** mariarestrepo164618@correo.itm.edu.co

Jorge Alexander David Rodríguez **e-mail:**

jorgedavid248961@correo.itm.edu.co

**31/03/2022**

**RESUMEN:** *En esta práctica buscamos potenciar las habilidades adquiridas con el lenguaje de programación basado en Python a través del entorno de desarrollo Visual Studio Code. Además de realizar un filtro o “limpieza” a una serie de imagenes establecidas, para mejorar su calidad*

**PALABRAS CLAVE**: *Programación, Histograma, Imagen digítal, VS Code, Python.*

# OBJETIVO General

En esta práctica, nos basaremos en potenciar habilidades adquiridas con el lenguaje de programación basado en Python por medio del entorno de desarrollo de Visual Studio Code. Además de llevar a cabo una secuencia de instrucciones respecto al filtrado de imágenes, reconociendo los diferentes ruidos que existen y dándoles su respectiva limpieza y solución. También se busca la creación de nuevas máscaras y matrices de convolución por medio de comandos de OpenCV.

## OBJETIVOS ESPECÍFICOS

# Desarrollar un algoritmo que por medio de video en tiempo real detecte un intruso y de una señal de alerta al usuario

# Diseñar máscaras de convolución, de paso bajo y otro paso alto.

# Identificar oportunamente las clases de ruidos en las imágenes para darles una solución.

# fundamento teorico

**IMAGEN DIGITAL**

Antes de conocer sobre qué es y las aplicaciones de la transformada de Fourier o incluso de algún otro proceso, de limpieza o filtrado de imágenes; vamos a hablar un poco sobre el concepto de Imágen digital.

Una imagen digital es “Una representación bidimensional de una imagen a partir de una matriz numérica, frecuentemente en binario (unos y ceros). Dependiendo de si la resolución de la imagen es estática o dinámica, puede tratarse de una imagen matricial (o mapa de bits) o de un gráfico vectorial.” *(Imagen Digital, May. 2011,* [*https://www.definicionabc.com/tecnologia/imagen-digital.php*](https://www.definicionabc.com/tecnologia/imagen-digital.php)*)*

Ahora bien, teniendo un breve concepto de lo que es imagen digital, se definirá el término de transformada de Fourier, principalmente aplicado al procesamiento de imágenes.

**TRANSFORMADA DE FOURIER**

La transformada de Fourier es una transformación matemática usada para transformar señales entre el dominio del tiempo o espacio al dominio de la frecuencia, y viceversa. El concepto de ‘transformada de Fourier’ se refiere a varios conceptos de forma simultánea:

* Operación de transformación de una función.
* Función resultado de la operación.
* Espectro de frecuencias de una función.

La función original suele recibir el nombre de x(t), siendo muy común que ‘t’ sea el tiempo, mientras que la función transformada suele recibir el nombre de X(f), en mayúscula, siendo ‘f’ la frecuencia.

Imagen que contiene Texto

Descripción generada automáticamente

**Figura 1. Ecuaciones de la Transformada de Fourier, Fuente: (** [**FVC-LucaOtarola.pdf (uns.edu.ar)**](http://lcr.uns.edu.ar/fvc/NotasDeAplicacion/FVC-LucaOtarola.pdf) **)**

La imagen digital es una función bidimensional, finita y discreta. Su transformada de Fourier 𝜑 es una

función generalmente compleja que puede describirse así:

Texto

Descripción generada automáticamente

**Figura 2. Ecuación para el procesamiento de Imágenes con T. Fourier Fuente: (** [**FVC-LucaOtarola.pdf (uns.edu.ar)**](http://lcr.uns.edu.ar/fvc/NotasDeAplicacion/FVC-LucaOtarola.pdf) **)**

De la Figura 2, podemos denotar que r y s son frecuencias espaciales, i y j las líneas y columnas de la imagen original de dimensiones KxK.

Según los teoremas de convolución, se puede establecer que bajo algunas circunstancias la transformada de Fourier es un producto punto; es decir, el dominio es equivalente al producto punto en el otro dominio (De espacio temporal a espacio espectral)

Diagrama, Esquemático

Descripción generada automáticamente

**Figura 3. Convolución de T. Fourier en imágenes Fuente: (** [**FVC-LucaOtarola.pdf (uns.edu.ar)**](http://lcr.uns.edu.ar/fvc/NotasDeAplicacion/FVC-LucaOtarola.pdf) **)**

**PARÁMETROS PARA SU REALIZACIÓN**

Para hacer un respectivo filtrado con la Transformada de Fourier, se debe de tener en cuenta:

1. Se aplica la Transformada de Fourier,

2. Se multiplica posteriormente por la función del filtro que ha sido escogido,

3. Para concluir re-transformándola al dominio espacial empleando la Transformada Inversa de

Fourier donde se tiene:

Teorema de Convolución (

F(u,v): Transformada de Fourier de la imagen original.

H(u,v): Filtro atenuador de frecuencias.

Como la multiplicación en el espacio de Fourier es idéntica a la convolución en el dominio espacial,

todos los filtros podrían, en teoría, ser implementados como un filtro espacial *(F. Vela, 2013,* [*http://lcr.uns.edu.ar/fvc/notasdeaplicacion/fvc-vela%20franco.pdf*](http://lcr.uns.edu.ar/fvc/notasdeaplicacion/fvc-vela%20franco.pdf) *)*

**TIPOS DE FRECUENCIAS**

La TF se puede aplicar a tres tipos de filtros, que buscaran limpiar los niveles o “señales” de intensidad correspondientes a niveles de grises de las diferentes filas o columnas de la matriz.

* Filtro paso bajo: Atenúa las frecuencias altas y mantiene sin variaciones las bajas. El resultado en el dominio espacial es equivalente al de un filtro de suavizado, donde las altas frecuencias que son filtradas se corresponden con los cambios fuertes de intensidad. Consigue reducir el ruido suavizando las transiciones existentes.
* Filtro paso alto: Atenúa las frecuencias bajas manteniendo invariables las frecuencias altas. Puesto que las altas frecuencias corresponden en las imágenes a cambios bruscos de densidad, este tipo de filtros es usado, porque entre otras ventajas, ofrece mejoras en la detección de bordes en el dominio espacial, ya que estos contienen gran cantidad de dichas frecuencias. Refuerza los contrastes que se encuentran en la imagen.
* Filtro paso banda: Atenúa frecuencias muy altas o bajas manteniendo una banda de rango medio.

Interfaz de usuario gráfica, Sitio web

Descripción generada automáticamente

**Figura 4. Ejemplo de tipos de filtros con TF con imagen “Lena” Fuente: (** [**http://lcr.uns.edu.ar/fvc/NotasDeAplicacion/FVC-LucaOtarola.pdf**](http://lcr.uns.edu.ar/fvc/NotasDeAplicacion/FVC-LucaOtarola.pdf) **)**

Las aplicaciones de la Transformada de Fourier son muy amplias y utilizadas para pasar al dominio de la frecuencia una señal para así obtener información que no es evidente en el dominio temporal en diferentes campos cómo se había indicado anteriormente, también sirve para resolver las ecuaciones de Fourier con mayor agilidad, implementándola así en otros campos como lo son el diseño electrónico de controladores, la orientación de vehículos y en términos de procesamiento de imágenes, su debida compresión.

Las funciones correspondientes en OpenCV son cv2.dft () y cv2.idft ().

Donde se debe tener en cuenta:

src：La imagen de entrada debe estar en formato np.float32.

dst：Imagen de salida, doble canal (parte real y parte imaginaria), su tamaño y tipo dependen de los indicadores de tercer parámetro

flags：El valor predeterminado es 0, y los valores posibles son los siguientes:

DFT\_INVERSE: Reemplace la transformación directa predeterminada con transformación inversa unidimensional o bidimensional

DFT\_SCALE: Identificador de relación de escala, basado en el número promedio de elementos de datos para encontrar su resultado de escala, si hay N elementos, el resultado de salida es 1 / N de escala, a menudo usado junto con DFT\_INVERSE. UNA

DFT\_ROWS: Realice una transformación de Fourier directa o inversa en cada fila de la matriz de entrada; este identificador se puede utilizar para reducir el costo de los recursos al procesar una variedad de cantidades apropiadas. Estos procesos a menudo son complejos, como las transformaciones tridimensionales o de alta dimensión. operando. UNA

DFT\_COMPLEX\_OUTPUT: Realice una transformación hacia adelante en una matriz real unidimensional o bidimensional. Aunque este resultado es una matriz compleja, tiene un par conjugado de números complejos (CCS), que se puede llenar con una matriz real del mismo tamaño que la matriz original. Esta es la opción más rápida y el método predeterminado de la función. Es posible que desee obtener una matriz compleja de tamaño completo (como un análisis espectral simple, etc.) Al establecer el bit indicador, la función puede generar una matriz de salida compleja de tamaño completo. UNA

DFT\_REAL\_OUTPUT: Transforme a la inversa una matriz compleja bidimensional unidimensional, tal resultado suele ser una matriz compleja del mismo tamaño, pero si la matriz de entrada tiene una simetría conjugada de números complejos (como un resultado de transformación positivo con un identificador DFT\_COMPLEX\_OUTPUT), Se generará la matriz real. UNA

nonzeroRows：El valor predeterminado es 0. Cuando este parámetro no es 0, la función supone que solo la primera fila de la matriz de entrada (sin el conjunto DFT\_INVERSE) o la primera matriz de salida (con el conjunto DFT\_INVERSE) contiene valores distintos de cero. De esta manera, la función puede procesar otras filas de manera más eficiente y ahorrar algo de tiempo. Esta técnica es muy efectiva especialmente cuando se usa DFT para calcular la convolución de la matriz.

**PASOS PARA EL USO DE LA TF CON OPEN CV**

**Paso 1: Cargar la imagen**

**Paso 2: use np.float32 para la conversión de formatos**

**Paso 3: use cv2.dft para el cambio de Fourier**

**Paso 4: Use np.fft.shiftfft para cambiar la frecuencia baja a la posición media**

**Paso 5: Use cv2.magnitude para proyectar las partes reales e imaginarias en el dominio espacial**

**Paso 6: realizar la operación**

**Paso 7: use np.fft.ifftshift para transferir la parte de baja frecuencia a la posición original de la imagen**

**Paso 8: Use cv2.idft para invertir la transformada de Fourier**

# procedimiento

Para la realización de esta práctica se tomaron en cuenta ciertos parámetros para llevar un paso a paso.

Se aplica un filtro Gaussiano de 35x35 donde da una imagen resultante. Posterior a esto se realiza una imagen diferente a la cual se le aplica una ecuación establecida.

**Punto A**

Imagen que contiene edificio, foto, espejo, hombre

Descripción generada automáticamente

**Figura 5. Imagen original A y su debido ruido. Fuente: Propia**

Posterior a esto, se realiza una nueva imagen donde se le aplica el filtro Gaussiano.

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

**Figura 6. Imagen B con filtro Gaussiano. Fuente: Propia**

Al aplicar la ecuación dada Imagen C = 0.3xImagenA + 0.7xImagenB – 34 se puede apreciar la siguiente imagen

Imagen que contiene espejo, aparato, foto, reflejo

Descripción generada automáticamente

**Figura 7. Imagen C. Fuente: Propia**

Podemos concluir de este primer punto que la imagen resultante B en comparación con la imagen resultante C es más borrosa es un poco más complejo poder mirar los detalles para un mejor análisis; mientras que en la imagen C aunque quedó mas nítida que la anterior, esta es un poco más “oscura” con un efecto más de “sombra”. Incluso en comparación con la original se pueden notar grandes diferencias.

**PUNTO B**

Se diseñaron dos máscaras de convolución (Paso bajo y paso alto) con matrices desarrolladas por nosotros ya que fueron las que más se acomodaron al resultado que se quería obtener obteniendo así su debido filtro

Foto en blanco y negro de un niño

Descripción generada automáticamente con confianza media

**Figura 8. Imagen con filtro paso bajo y paso alto. Fuente: Propia**

**PUNTO C**

Se desarrolla un algoritmo que detecta en tiempo real de un video la entrada de un intruso y este nos envía una señal de alerta. Se utilizan diversas herramientas, como la captura de pantalla para realizar la comparación de dos imágenes para proceder a restarlas y realizar la creación de una imagen binaria, detectando así el movimiento en tiempo real.

Captura de pantalla de un celular con la foto de una persona

Descripción generada automáticamente

**Figura 9. Detección de Intruso. Fuente: Propia**

**PUNTO D**

Para este punto se debía de tener en cuenta varios factores que se explicarán a continuación.

Imagen en blanco y negro de una persona

Descripción generada automáticamente con confianza baja

**Figura 10. Imagen de lena con ruido Gaussiano y su respectivo filtro. Fuente: Propia**

En la figura anterior se puede apreciar una imagen con ruido Gaussiano que afecta notablemente su calidad, como en su nitidez, se le aplican cuatro diferentes tipos de filtros para apreciar una mejor limpieza de esta, (Filtro paso Bajo, Paso Alto, Mediana, Gaussiana y Sharpen) del cual, el mejor resultado en calidad de imagen y nitidez es el Gaussiano ya que se pueden observar mejor los detalles pequeños sin verse tan borroso o difuminado.

Imagen en blanco y negro de una pareja

Descripción generada automáticamente

**Figura 11. Imagen ruido Sal. Fuente: Propia**

De la figura anterior se puede concluir que el mejor filtro para usar es el de la mediana debido a que al momento de hacer el barrido del ruido sustituye los píxeles negros y blancos por un píxel que contenga información de la imagen.

Imagen que contiene foto, frente, mujer, jugando

Descripción generada automáticamente

**Figura 12. Imagen ruido Sal y pimienta y su respectivo filtro. Fuente: Propia**

De la figura anterior se puede concluir que el mejor filtro para usar es el de la mediana debido a que al momento de hacer el barrido del ruido sustituye los píxeles negros y blancos por un píxel que contenga información de la imagen.

**PUNTO E**

Se desarrolla un filtro con una matriz de convolución adecuada que tenga la capacidad de realizar un barrido manual de estos y obtener un resultado establecido para la imagen.

**Interfaz de usuario gráfica

Descripción generada automáticamente con confianza baja**

**Figura 13. Imagen Lena con ruido. Fuente: Propia**

De la figura anterior se puede concluir que el mejor filtro para usar en esta imagen es el Gaussiano, debido a que al momento de hacer el barrido del ruido encontrado, que en este caso fueron dos Gaussiano y Sal y pimienta, se sustituye manualmente los píxeles negros, blancos y con el daño por el ruido gaussiano por un píxel que contenga información de la imagen.

**PUNTO F**

Interfaz de usuario gráfica

Descripción generada automáticamente

**Figura 14. Imagen implementación de la Transformada de Fourier. Fuente: Propia**

En la anterior figura podemos apreciar el resultado del uso de la Transformada de Fourier sobre la imagen Lena, donde podemos apreciar las diferencias en los espectros o frecuencias que proporcionan el brillo y nitidez de la imagen.

# conclusiones

1. Se puede concluir que el uso de filtros en imágenes es de gran importancia para obtener una información deseada, al igual que recuperar datos perdidos simplemente apreciar una buena calidad en está.
2. La aplicación de la transformada de Fourier es muy importante por que nos ayuda a ver que es un método muy efectivo y simple que se puede utilizar en cualquier programa realizado
3. A nivel de ventajas y desventajas, los filtros tienen una mejor optimización en algunas imágenes dependiendo del ruido que se posea, es decir, para la implementación de la TF se puede apreciar que es un filtro que proporciona flexibilidad y rapidez en el momento de usar teoremas de convolución, pero este filtro no nos elimina el ruido completamente.

# Bibliografía

|  |
| --- |
| Imagen Digital. (2011, mayo). Definición ABC. Recuperado 28 de marzo de 2022, de https://www.definicionabc.com/tecnologia/imagen-digital.php |
| L. Otarola. (2015, agosto). Transformada de Fourier para el procesamiento digital de imágenes. http://lcr.uns.edu.ar/fvc/NotasDeAplicacion/FVC-LucaOtarola.pdf  F. Vela. (2013, marzo). Transformada de Fourier en procesamiento digital de Imágenes Funciones de Variable Compleja. http://lcr.uns.edu.ar/fvc/notasdeaplicacion/fvc-vela%20franco.pdf  Fourier2D. (s. f.). Transformada de Fourier. Aplicación al procesamiento de imágenes. Recuperado 30 de marzo de 2022, de https://www.unioviedo.es/compnum/laboratorios\_py/Fourier/Fourier2D.html#compr  OpenCV-Python-Capítulo 19: Transformada de Fourier - programador clic. (s. f.). Transformada de Fourier - programador clic. Recuperado 30 de marzo de 2022, de https://programmerclick.com/article/1211697827/ |