



Instituto Politécnico Nacional



**Unidad Profesional Interdisciplinaria de Ingeniería Campus
Zacatecas**

Ingeniería en Sistemas Computacionales

Análisis de Imágenes

Transformada Discreta de Fourier en 2D

Profesor Roberto Oswaldo Cruz Leija

Grupo 3CM3

Angel Isaac Saldivar Delgado

03 / 06 / 2021

Introducción

La **transformada de Fourier**, denominada así por Joseph Fourier, es una transformación matemática empleada para transformar señales entre el dominio del tiempo (o espacial) y el dominio de la frecuencia, que tiene muchas aplicaciones en la física y la ingeniería. Es reversible, siendo capaz de transformarse en cualquiera de los dominios al otro. El propio término se refiere tanto a la operación de transformación como a la función que produce.

La **transformada discreta de Fourier** o **DFT** (del inglés, *discrete Fourier transform*) es un tipo de transformada discreta utilizada en el análisis de Fourier. Transforma una función matemática en otra, obteniendo una representación en el dominio de la frecuencia, siendo la función original una función en el dominio del tiempo. Pero la DFT requiere que la función de entrada sea una secuencia discreta y de duración finita. Dichas secuencias se suelen generar a partir del muestreo de una función continua, como puede ser la voz humana.

Desarrollo

Primero, necesitamos tener el algoritmo que calcule la transformada. El código es el siguiente:

```
public void operacion_TransformadaFourierD(){
    double parteImaginaria = 0;
    double parteReal = 0;
    double modulo_k = 0;
    double tam_Muestra = muestras_x.size();

    for(int valorK=0;valorK<tam_Muestra;valorK++){//k = 0,1,2,3...N-1
        parteImaginaria = 0;
        parteReal = 0;
        for(int valorN = 0;valorN<tam_Muestra;valorN++){//Aqui es el ciclo de la sumatoria
            parteReal += (Math.cos((double)((2*Math.PI*valorK*valorN)/tam_Muestra)))*muestras_x.get(valorN);
            parteImaginaria += (Math.sin((double)(2*Math.PI*valorK*valorN)/tam_Muestra))*muestras_x.get(valorN);
        }
        modulo_k = Math.sqrt((Math.pow(parteReal, 2))+Math.pow(parteImaginaria,2));//Obtenemos el modulo del numero imaginario
        if(modulo_k < 0.000001 && modulo_k > -0.000001){
            modulo_k = 0;
        }
        this.transformada_X.add(valorK, modulo_k);
    }
}
```

El código anterior se encarga de, a partir de una serie de muestras, calcular la transformada. Recorre todas las muestras y en cada ciclo aplica lo siguiente:

$$X_k = \sum_{n=0}^{N-1} x_n e^{-\frac{2\pi i}{N} kn} \quad k = 0, \dots, N-1$$

Transformada en 2D aplicada en una imagen

Para aplicar la transformada a una imagen, lo que tenemos que hacer es:

- Primero aplicar la transformada en forma vertical u horizontal.
- Después, aplicar de forma vertical u horizontal, dependiendo al caso anterior, si primero aplicamos vertical, después aplicaremos horizontal y viceversa.

Código en Java para lo anterior:

```
ArrayList<Double> puntos_muestra_red = new ArrayList<>();
ArrayList<Double> puntos_muestra_green = new ArrayList<>();
ArrayList<Double> puntos_muestra_blue = new ArrayList<>();

for(int h = 0; h < this.height; h++){
    color = new Color(this.buffer_cambiada.getRGB(h, w));
    puntos_muestra_red.add((double)color.getRed());
    puntos_muestra_green.add((double)color.getGreen());
    puntos_muestra_blue.add((double)color.getBlue());
}

//Con los puntos muestra, aplicamos la FFT
fft.setMuestrasTransformada(puntos_muestra_red);
fft.operacion_TransformadaFourierD();

//Obtenemos los puntos de la operacion
ArrayList<Double> fft_resultados_red = fft.getTransformadaDiscreta_X();

fft.setMuestrasTransformada(puntos_muestra_green);
fft.operacion_TransformadaFourierD();

//Obtenemos los puntos de la operacion
ArrayList<Double> fft_resultados_green = fft.getTransformadaDiscreta_X();

fft.setMuestrasTransformada(puntos_muestra_blue);
fft.operacion_TransformadaFourierD();

//Obtenemos los puntos de la operacion
ArrayList<Double> fft_resultados_blue = fft.getTransformadaDiscreta_X();

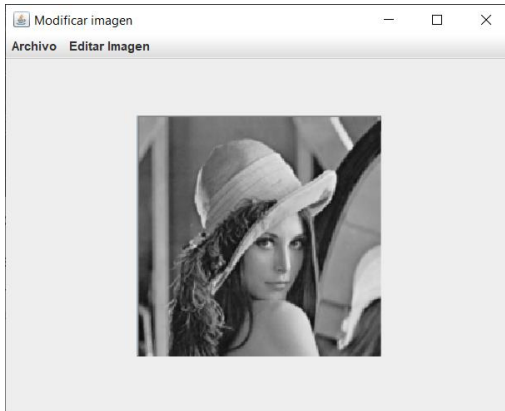
//Ahora esos valores, los seteamos de nuevo en la imagen
for(int numPixel = 0; numPixel < this.height; numPixel++){
    Color nuevoColor = new Color(checarColor(fft_resultados_red.get(numPixel).intValue()),
                                checarColor(fft_resultados_green.get(numPixel).intValue()),
                                checarColor(fft_resultados_blue.get(numPixel).intValue()));
    this.buffer_cambiada.setRGB(numPixel,w,nuevoColor.getRGB());
}
```

El código anterior se repite N veces, donde N es la cantidad de columnas que tiene la imagen, y a cada imagen se le aplica la transformada en su canal R, G y B, y antes de acabar el ciclo asignamos esos valores nuevos al pixel anterior.

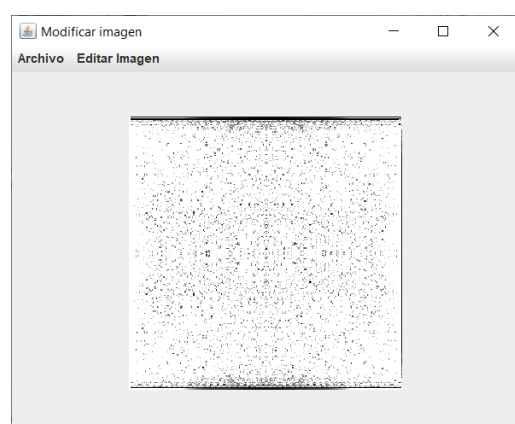
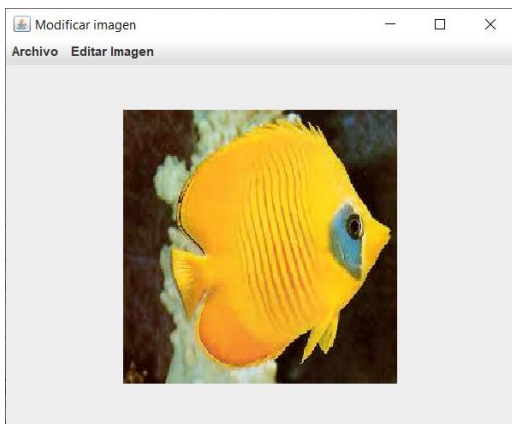
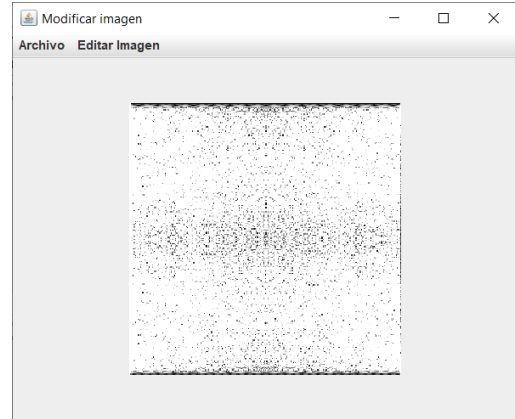
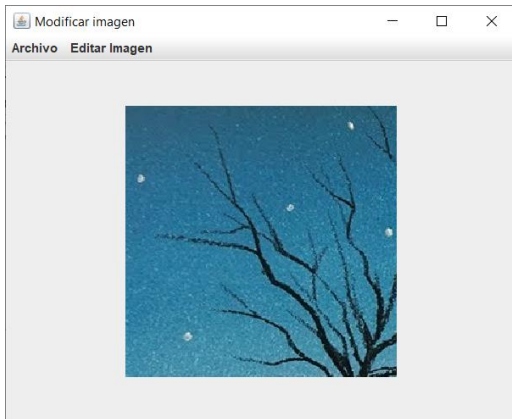
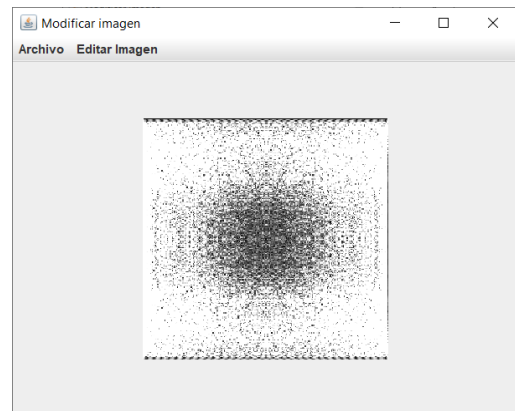
Para lo horizontal es igual, la única diferencia es que en vez de recorrer la imagen columna por columna, se recorre fila por fila.

Pruebas

Imagen original



Espectro de Frecuencias



Conclusiones

Al aplicar la transformada a una imagen, el resultado arrojado es distinto para cada imagen, por ejemplo, en la primera imagen de la mujer todo se ve mas concentrado en medio, mientras que en la del pez se ve más disperso el espectro.

Bibliografía

- colaboradores de Wikipedia. (2021a, marzo 24). *Transformada de Fourier*.

Wikipedia, la enciclopedia libre.

https://es.wikipedia.org/wiki/Transformada_de_Fourier

- colaboradores de Wikipedia. (2021a, enero 17). *Transformada de Fourier discreta*.

Wikipedia, la enciclopedia libre.

https://es.wikipedia.org/wiki/Transformada_de_Fourier_discreta