Esteganografía digital utilizando la Transformada Discreta de Fourier

UN-ESTEGANOS

Bohórquez, Edwin
eabohorquezg@unal.edu.co

D'Anetra, Gustavo
gadanetrac@unal.edu.co

Muñoz, Fernando
fmunozs@unal.edu.co

Perdomo, José
jdperdomos@unal.edu.co

25 de noviembre 2014

1. Introducción

Algunas veces necesitamos enviar un mensaje a través de Internet a otra persona, pero no queremos que nadie más lea este, es decir queremos ocultar la información, para evitar que alguien no autorizado tenga acceso a ella; pero al pensar en esto se nos viene una pregunta a la cabeza ¿es suficiente?, hoy en día existen muchos métodos para descifrar información, así que se hace necesario hacer algo más, si nosotros enviamos una imagen común y corriente a nuestro destinatario, en lugar de un mensaje codificado y esta llega a ser interceptada, tal vez no logre llamar la atención del atacante, al ver una simple imagen en el mensaje, como por ejemplo la foto de algún lugar de interés; entonces si nos aprovechamos de este hecho, podríamos enviar mensajes que no solo están cifrados, sino que además están ocultos en imágenes normales, evitando posibles ataques o accesos indeseados a la información enviada.

En este proyecto mostraremos una forma de proteger la información de la forma anteriormente descrita, utilizando los coeficientes de la Transformada discreta de Fourier.

2. Objetivos

- Lograr establecer un canal digital encubierto de comunicación, de modo que pase inadvertido para observadores que tienen acceso a ese canal (en este caso una imagen).
- Aplicar una técnica que permita ocultar un mensaje o imagen dentro de otra imagen, de modo que no se perciba su existencia.
- Aprender sobre la Transformada de Fourier, la Transformada Rápida de Fourier y la Transformada Discreta de Fourier.

- Aplicar la Transformada Discreta de Fourier a la Esteganografía digital de imágenes.
- Implementar una aplicación en el lenguaje de programación Java, en la que se pueda ver la utilidad de la Transformada Discreta de Fourier en la Esteganografía digital de imágenes.

3. Esteganografía

3.1. ¿Qué es?

La esteganografía es el arte de ocultar información de tal manera que se prevenga la detección del mensaje oculto. La palabra esteganografía tiene origen griego en las palabras *steganos* y *graphein*, las cuales significan "oculto o protegido" y "dibujar o escribir" respectivamente. Esta disciplina permite ocultar mensajes secretos en imágenes, audio o video para encubrir la información y prevenir la detección del mensaje oculto por usuarios no autorizados.

El Problema del Prisionero y el Canal Oculto, propuesto en 1983 por Gustavus Simmons describe un escenario en el cual la esteganografía brinda una solución, dos prisioneros cómplices en un crimen fueron arrestados y colocados en celdas aisladas.

El guardián de la cárcel, les permite intercambiar mensajes pero sólo si él transporta el mensaje de uno hacia el otro y siempre que al leer esas comunicaciones esté seguro de que los mensajes que se intercambian son inofensivos y no están intercambiando mensajes para poder escapar.

Esta forma de comunicación es la única que tienen disponible los prisioneros. Los prisioneros necesitan comunicarse para poder planear su escape, por lo cual deben aceptar las condiciones. El guardia finalizará la comunicación si descubre un canal secreto, por esta razón no pueden utilizar métodos de cifrado. La única posibilidad que tienen es establecer algún tipo de canal oculto en los mensajes que se intercambian. [1]

3.2. Conceptos básicos

Portador Básicamente es cualquier objeto o canal que puede ser alterado para ocultar el mensaje, en nuestro caso será una imagen.

Mensaje legítimo Se refiere al mensaje que está actualmente en el portador.

Mensaje Esteganográfico Se refiere al mensaje que queremos ocultar.

Estego-algoritmo Es el algoritmo que usamos para ocultar el mensaje, este nos dice cómo incorporar el mensaje en el portador.

Canal de selección Es un canal auxiliar, que permite seleccionar las posiciones del portador que van a ser usadas para ocultar el mensaje.

4. Transformada de Fourier

La palabra transformada indica que vamos a trabajar con una herramienta que transforma un tipo determinado de problema en otro. Para conocer bien este tema vamos a comenzar estudiando la Transformada Continua de Fourier.

4.1. Transformada Continua de Fourier

La Transformada continua de Fourier es una aplicación lineal que tiene una serie de propiedades de continuidad que garantizan que puede extenderse a espacios de funciones mayores e incluso a espacios de funciones generalizadas.

Un proceso físico se puede describir en el dominio del tiempo, $f(t)$, o en el dominio de la frecuencia angular $F(w)$. Donde $w = 2\pi f$ (frecuencia angular) se refiere a la frecuencia del movimiento circular expresada en proporción del cambio de ángulo.

En general, $f(t)$ y $F(w)$ son funciones complejas $f(t)$ y $F(w)$ son dos representaciones diferentes de la misma función:

$$F\{f(t)\} = \int_{-\infty}^{\infty} f(t)e^{-jwt}dw = F(W)$$

$$F^{-1}\{F(w)\} = \frac{1}{2\pi}\int_{-\infty}^{\infty} F(w)e^{jwt}dw = f(t)$$

Donde e^{-jwt} se conoce como el núcleo de la transformada de $f(t)$, y e^{jwt} es el núcleo de la transformada de $F(w)$.

4.2. Transformada Discreta de Fourier

En tratamiento digital de señal y por extensión en tratamiento digital de imagen, no disponemos de una señal continua, sino de una serie de muestras de la señal. Supongamos que tenemos N muestras:

$$h_k = h(t_k); t_k = k\Delta; k = 0, 1, 2, ..., N-1$$

Donde Δ es el intervalo de muestreo.

Con N números de entrada no podemos producir más que N números independientes entre sí en la salida.

Como nuestra señal es discreta estimaremos la transformada de Fourier solo para algunos valores discretos de frecuencia:

$$f_n = \frac{n}{N\Delta}; n = -\frac{N}{2}, ..., \frac{N}{2}$$

Para ello aproximamos la integral por una suma discreta, tenemos que:

$$H(w) = \int_{-\infty}^{+\infty} h(t)e^{-jwt}dt$$

Teniendo en cuenta que $w = 2\pi f$:

$$H(f_n) = \int_{-\infty}^{+\infty} h(t)e^{-j2\pi f_n t}dt \approx \sum_{k=0}^{N-1} h_k e^{-j2\pi f_n t_k}\Delta \approx \Delta \sum_{k=0}^{N-1} h_k e^{-j2\pi kn/N}$$

La suma final no depende del intervalo de muestreo, Δ y se conoce como transformada discreta de Fourier de la serie de N números de h_k:

$$H_n = \sum_{k=0}^{N-1} h_k e^{-j2\pi kn/N}$$

La transformada discreta mapea N números complejos, h_k, en N números complejos H_n. La relación entre la transformada de Fourier (continua) en las frecuencias f_n y la transformada discreta, es: $H(f_n) \approx \Delta H_n$.

Finalmente, la transformada discreta inversa de Fourier que recupera los números h_k a partir de los números H_n tiene la expresión:

$$h_k = \frac{1}{N} \sum_{k=0}^{N-1} H_n e^{j2\pi kn/N}$$

Donde j es la unidad imaginaria $j = \sqrt{-1}$.

Para ilustrar la teoría estudiada anteriormente, hallaremos la transformada discreta de Fourier de la siguiente secuencia:

$$X[n] = 1, 0, 1$$

Donde:

$$X[0] = 1, X[1] = 0, X[2] = 1$$

Sabemos que la TDF está definida como:

$$X_k = \sum_{n=0}^{N-1} X[n]e^{-\frac{2\pi i}{N}kn}$$

Donde $i = \sqrt{-1}$ y $n = 0, ..., N-1$.

Como sabemos la TDF transforma una secuencia de números complejos en otra secuencia de números complejos, en nuestro ejemplo tenemos una secuencia binaria. Entonces tenemos que la sumatoria con la que obtendremos los números complejos de nuestra secuencia resultado es:

$$X_k = \sum_{n=0}^{2} X[n]e^{-\frac{2\pi ikn}{3}}$$

$$X_0 = 1e^{\frac{-2\pi i(0)(0)}{3}} + 0e^{\frac{-2\pi i(0)(1)}{3}} + 1e^{\frac{-2\pi i(0)(2)}{3}}$$

$$X_0 = 1 + 0 + 1 = 2$$

$$X_1 = 1e^{\frac{-2\pi i(1)(0)}{3}} + 0e^{\frac{-2\pi i(1)(1)}{3}} + 1e^{\frac{-2\pi i(1)(2)}{3}}$$

$$X_1 = 1 + e^{\frac{-4\pi i}{3}}$$

Ahora usamos la identidad de Euler para continuar:

$$e^{i\theta} = cos(\theta) + isen(\theta)$$

$$e^{\frac{-4\pi i}{3}} = cos(\frac{-4\pi}{3}) + isen(\frac{-4\pi}{3})$$

$$X_1 = 1 + cos(\frac{-4\pi}{3}) + isen(\frac{-4\pi}{3})$$

$$X_1 = 1 + (-0{,}5) + 0{,}86i$$
$$X_1 = 0{,}5 + 0{,}86i$$

$$X_2 = 1e^{\frac{-2\pi i(2)(0)}{3}} + 0e^{\frac{-2\pi i(2)(1)}{3}} + 1e^{\frac{-2\pi i(2)(2)}{3}}$$

$$X_2 = 1 + e^{\frac{-8\pi i}{3}}$$

Usamos identidad de Euler:

$$X_2 = 1 + cos(\frac{-8\pi}{3}) + isen(\frac{-8\pi}{3})$$

$$X_2 = 1 + (-0{,}5) + (-0{,}86i)$$
$$X_2 = 0{,}5 - 0{,}86i$$

En conclusión tenemos que la transformada discreta de Fourier de nuestra secuencia $X[n]$ (secuencia binaria) es otra secuencia (de números complejos) que la llamaremos $R[k]$.

$$R[k] = 2, 0{,}5 + 0{,}86i, 0{,}5 - 0{,}86i$$

Donde:

$$R[0] = 2, R[1] = 0{,}5 + 0{,}86i, R[2] = 0{,}5 - 0{,}86i$$

4.3. Transformada Rápida de Fourier

Como vimos anteriormente, al realizar la Transformada Discreta de Fourier (DFT) sobre una muestra determinada es necesario realizar un total de N^2 multiplicaciones complejas donde N es el tamaño de la muestra, lo cual en términos computacionales no es muy eficiente, en respuesta a esto surgió el algoritmo de la Transformada Rápida de Fourier (FFT), este al contrario requiere un número de operaciones del orden $Nlog(N)$, esto significa que la DFT puede calcularse en un tiempo mucho menor, a continuación veremos en detalle cómo funciona este algoritmo y su estrecha relación con la DFT.

Como lo hicimos anteriormente, primero hallaremos la Transformada Discreta de Fourier de una secuencia dada:

$$X_{[n]} = [1, 1, 1, 1]$$

Donde $X_{[0]} = 1$, $X_{[1]} = 1$, $X_{[2]} = 1$, $X_{[3]} = 1$. Sabemos que la TDF está definida como:

$$X_k = \sum_{j=0}^{N-1} X_{[j]} \mathrm{e}^{\frac{-2\pi i}{N} kj}$$

$$\text{Con } k \in [0, ..., N-1]$$

Donde $i = (-1)^{1/2}$ y $n \in [0, ..., N-1]$. Reemplazamos:

$$X_k = \sum_{j=0}^{3} X_{[j]} \mathrm{e}^{\frac{-2\pi i}{4} jn}$$

$$\text{Para } X_{[0]} = 1(\mathrm{e}^{\frac{-\pi i}{2}(0)(0)}) + 1(\mathrm{e}^{\frac{-\pi i}{2}(0)(1)}) + 1(\mathrm{e}^{\frac{-\pi i}{2}(0)(2)}) + 1(\mathrm{e}^{\frac{-\pi i}{2}(0)(3)})$$

$$X_{[0]} = 1 + 1 + 1 + 1 = 4$$

$$\text{Para } X_{[1]} = 1(\mathrm{e}^{\frac{-\pi i}{2}(1)(0)}) + 1(\mathrm{e}^{\frac{-\pi i}{2}(1)(1)}) + 1(\mathrm{e}^{\frac{-\pi i}{2}(1)(2)}) + 1(\mathrm{e}^{\frac{-\pi i}{2}(1)(3)})$$

$$X_{[1]} = 1 + 1(\mathrm{e}^{\frac{-\pi i}{2}}) + 1(\mathrm{e}^{-\pi i}) + 1(\mathrm{e}^{\frac{-3\pi i}{2}})$$

Aplicando la identidad de Euler $\mathrm{e}^{i\theta} = cos(\theta) + iSin(\theta)$

$$X_{[1]} = 1 + 1(cos(-\pi/2) + iSin(-\pi/2)) + 1(cos(-\pi) + iSin(-\pi)) + 1(cos(-3\pi/2) + iSin(-3\pi/2))$$

$$X_{[1]} = 1 + (0 - i) + (-1) + (0 + i) = 0$$

$$\text{Para } X_{[2]} = 1(\mathrm{e}^{\frac{-\pi i}{2}(2)(0)}) + 1(\mathrm{e}^{\frac{-\pi i}{2}(2)(1)}) + 1(\mathrm{e}^{\frac{-\pi i}{2}(2)(2)}) + 1(\mathrm{e}^{\frac{-\pi i}{2}(2)(3)})$$

$$X_{[2]} = 1 + 1(\mathrm{e}^{-\pi i}) + 1(\mathrm{e}^{-2\pi i}) + 1(\mathrm{e}^{-3\pi i})$$

Aplicando la identidad de Euler

$$X_{[2]} = 1+1(cos(-\pi)+iSin(-\pi))+1(cos(-2\pi)+iSin(-2\pi))+1(cos(-3\pi)+iSin(-3\pi))$$

$$X_{[2]} = 1 + (-1) + 1 + (-1) = 0$$

Para $X_{[3]} = 1(\mathrm{e}^{\frac{-\pi i}{2}(3)(0)}) + 1(\mathrm{e}^{\frac{-\pi i}{2}(3)(1)}) + 1(\mathrm{e}^{\frac{-\pi i}{2}(3)(2)}) + 1(\mathrm{e}^{\frac{-\pi i}{2}(3)(3)})$

$$X_{[3]} = 1 + 1(\mathrm{e}^{-3\pi i/2}) + 1(\mathrm{e}^{-3\pi i}) + 1(\mathrm{e}^{-9\pi i/2})$$

Aplicando la identidad de Euler

$$X_{[3]} = 1 + 1(cos(-3\pi/2) + iSin(-3\pi/2)) + 1(cos(-3\pi) + iSin(-3\pi)) + 1(cos(-9\pi/2) + iSin(-9\pi/2))$$

$$X_{[3]} = 1 + (0 + i) + (-1) + (0 - i) = 0$$

Ahora obtenemos nuestra secuencia de números complejos:

$$R_{[k]} = [4, 0, 0, 0]$$

Donde $R_{[0]} = 4$, $R_{[1]} = 0$, $R_{[2]} = 0$, $R_{[3]} = 0$.

Ya sabemos que al aplicar la transformada inversa obtendremos nuestro arreglo original, así que no la aplicaremos en esta ocasión ya que nos centraremos en la FFT.

$$X_k = \sum_{j=0}^{N-1} X_{[j]} W_N^{jk}$$
$$\text{Con } W_N = \mathrm{e}^{-2\pi i/N}$$

Tenemos nuestra fórmula general de la transformada discreta de Fourier, ahora si dividimos nuestra sumatoria, en dos sumatorias equivalentes, tal que una de ellas sean las sumas de los términos que estén en las posiciones pares, mientras que en la otra la suma de los términos que están ubicados en las posiciones impares, no alteramos la ecuación original.

$$X_k = \sum_{j=0}^{N/2-1} X_{[2j]} W_N^{2jk} + \sum_{j=0}^{N/2-1} X_{[2j+1]} W_N^{(2j+1)k}$$

Si llamamos a las muestras $X_{[2j]}$ "pares" como $X_{1[j]}$ y a las muestras $X_{[2j+1]}$ "impares" $X_{2[j]}$. Obtenemos:

$$X_k = \sum_{j=0}^{N/2-1} X_{1[j]} W_N^{2jk} + \sum_{j=0}^{N/2-1} X_{2[j]} W_N^{2jk} W_N^k$$

Nótese que tenemos un factor común en ambas sumatorias, W_N^{2jk}, al desarrollarlo obtenemos:

$$W_N^{2jk} = \mathrm{e}^{-4jk\pi i/N}$$

Ahora, si multiplicamos y dividimos el exponente por $1/2$ no alteramos su magnitud.

$$-4jk\pi i/N = -4jk\pi i/N^{1/2}/1/2 = -2jk\pi i/N/2$$

Este resultado nos dará una expresión que llamaremos

$$W_{N/2}^{jk} = \mathrm{e}^{-2jk\pi i/N/2}$$

Sustituyendo en la ecuación general obtenemos

$$X_k = \sum_{j=0}^{N/2-1} X_{1[j]} W_{N/2}^{jk} + W_N^k \sum_{j=0}^{N/2-1} X_{2[j]} W_{N/2}^{jk}$$

Vemos que la expresión $\sum_{j=0}^{N/2-1} X_{1[j]} W_{N/2}^{jk}$ es la DFT definida para $N/2$ puntos. Entonces:

$$X_k = X_{1[k]} + W_N^k X_{2[k]} \tag{1}$$

Donde $X_{1[k]}$ y $X_{2[k]}$ son dos DFT definidas para $N/2$ puntos. Ahora, como nuestra señal X_k es periódica podemos decir que $X_k = X_{[k+N/2]}$, al resolver esta última expresión obtenemos:

$$X_{[k+N/2]} = X_{1[k+N/2]} + W_N^{k+N/2} X_{2[k+N/2]}$$

También sabemos que $W_N^{k+N/2} = -W_N^k$. Reemplazando obtenemos:

$$X_{[k+N/2]} = X_{1[k+N/2]} - W_N^k X_{2[k+N/2]}$$

Pero como $X_k = X_{[k+N/2]}$ podemos decir:

$$X_k = X_{1[k]} - W_N^k X_{2[k]} \tag{2}$$

En resumen tenemos que la DFT de una muestra, X_k,se puede expresar como:

$$X_k = F_1(k) - W_N^k F_2(k)$$

Donde $F_1(k)$ y $F_2(k)$ son la DFT definida para $N/2$ puntos.

Podemos tomar dos funciones $G_1 = F_1(k)$ y $G_2 = W_N^k F_2(k)$ y a partir de estas la DFT se puede definir como:

$$X_k = G_1 + G_2$$

$$X_{[k+N/2]} = G_1 - G_2$$

$$\text{Con } k \in [0, ..., N/2 - 1]$$

Habiendo realizado el proceso una vez, se puede repetir el proceso para cada una de las funciones G_1 y G_2. Podemos repetir este mismo proceso recursivamente, es decir, si por ejemplo hiciéramos este proceso para G_1 y G_2 obtendremos cuatro funciones H_1, H_2, H_3 y H_4 definidas para $N/4$ puntos. Siguiendo este proceso hasta su mínima expresión llegaremos a la DFT de un arreglo tamaño 2, el cual desarrollaremos a continuación.

$$R_{1[k]} = \sum_{j=0}^{1} X_{[j]} \mathrm{e}^{-jk\pi i}$$

Tendríamos entonces:

$$R_{1[k]} = R_{1[0]} + R_{1[1]} = (r_{[0]}\mathrm{e}^0 + r_{[1]}\mathrm{e}^0) + (r_{[0]}\mathrm{e}^0 - r_{[1]}\mathrm{e}^0)$$

Esto tiene sentido ya que nuestra función es periódica y por lo cual podemos decir que $k = k + {}^{N}/_{2}$, es decir, $1 = 1 + 0$. El algoritmo recursivo de la FFT, se basa sobre esta operación.

Examinemos el siguiente ejemplo, tomaremos el mismo arreglo $X_{[n]} = [1, 1, 1, 1]$

$$X_k = \sum_{j=0}^{{}^{N}/_{2}-1} X_{[2j]} W_N^{2jk} + W_N^k \sum_{j=0}^{{}^{N}/_{2}-1} X_{[2j+1]} W_N^{(2j+1)k}$$

$$X_k = Q_1(k) + W_N^k Q_2(k)$$

$$X_{k+{}^{N}/_{2}} = Q_1(k) - W_N^k Q_2(k)$$

Para $k = 0$ tenemos:

$$X_{[0]} = Q_1(0) + W_N^k Q_2(0) = \sum_{j=0}^{{}^{N}/_{2}-1} X_{[2j]} W_N^{2jk} + W_N^k \sum_{j=0}^{{}^{N}/_{2}-1} X_{[2j+1]} W_N^{(2j+1)k}$$

$$X_{[0]} = (X_{[0]} W_4^{2(0)(0)} + X_{[2]} W_4^{2(1)(0)}) + (X_{[1]} W_4^{2(0)(0)} + X_{[3]} W_4^{2(1)(0)})$$

$$X_{[0]} = (1(1) + 1(1)) + 1(1(1) + 1(1)) = 4$$

Para $k = 1$ tenemos:

$$X_{[1]} = Q_1(1) + W_N^k Q_2(1) = \sum_{j=0}^{{}^{N}/_{2}-1} X_{[2j]} W_N^{2jk} + W_N^k \sum_{j=0}^{{}^{N}/_{2}-1} X_{[2j+1]} W_N^{(2j+1)k}$$

$$X_{[1]} = (X_{[0]} W_4^{2(0)(1)} + X_{[2]} W_4^{2(1)(1)}) + (X_{[1]} W_4^{2(0)(1)} + X_{[3]} W_4^{2(1)(1)})$$

$$X_1 = (1(1) + (-1)) + W_4^1(1(1) + (-1))$$

Usando la identidad de Euler tenemos

$$X_{[1]} = (1(1)+(-1))+(1+cos({}^{-\pi}/_{2})+isin({}^{-\pi}/_{2}))(1(1)+(-1)) = 0+(0+i)(0) = 0$$

Para $k = 2$ tenemos:

$$X_{[2]} = Q_1(2) + W_N^k Q_2(2)$$

Pero ya que $X_{[k]} = X_{[k+{}^{N}/_{2}]}$ podemos usar el resultado obtenido en X_0 para hallar X_2. Ya que $X_0 = (X_{[0]} W_4^0 + X_{[2]} W_4^0) + W_4^0(X_{[1]} W_4^0 + X_{[3]} W_4^0)$, tenemos

$$X_{[2]} = (X_{[0]} W_4^0 - X_{[2]} W_4^0) + W_4^0(X_{[1]} W_4^0 - X_{[3]} W_4^0)$$

$$X_{[2]} = (1(1) + 1(1)) - 1(1(1) + 1(1)) = 2 - 2 = 0$$

Para $k = 3$ usaremos el resultado obtenido de $X_{[1]} = (X_{[0]}W_4^0 - X_{[2]}) + W_4^1(X_{[1]}W_4^0 - X_{[3]})$, tenemos:

$$X_{[3]} = (X_{[0]}W_4^0 - X_{[2]}) - W_4^1(X_{[1]}W_4^0 - X_{[3]})$$

Usando la identidad de Euler tenemos

$$X_{[3]} = (1(1) - 1) + (1 + cos(-\pi/2) + isin(-\pi/2))(1(1) - 1) = 0 + (0 + i)(0) = 0$$

Puede verse que el resultado coincide con el obtenido a través de la DFT. En primera instancia el algoritmo FFT se ve más complicado pero es mucho más sencillo ya que requiere menos operaciones. Sabemos que la DFT tiene un orden de N^2 operaciones. Por su parte la FFT para una muestra de tamaño N viene dada por:

$$FFT(N) = DFT(N/2) + DFT(N/2) = N^2/2$$

Puede verse que el número de operaciones requeridas ya es inferior al de la DFT, en forma general tendremos que el orden de operaciones será, $FFT(N) = 2FFT(N/2) + N$. Veamos que pasa si seguimos la recursión.

$$FFT(N) = 2(2FFT(N/4) + N/2) + N$$

Para el caso general podemos decir:

$$FFT(N) = 2^k FFT(N/2^k)kN$$

Cuando lleguemos a la última recursión tendremos que es la DFT para una pareja de elementos, es decir, cero operaciones. Podemos decir entonces que que el orden de operaciones de la DFT para 2 elementos es mínima, es decir $DFT(1) = 0$, entonces teniendo en cuenta que $(N/2^k = 1$ y despejando k tenemos, $K = Log_2 N$, con lo cual el orden de operaciones:

$$FFT(N) = 2^k FFT(N/2^k)kN = 2^k(0) + kN = NLog_2 N$$

En resumen para una muestra de tamaño N se requieren del orden de $NLogN$ operaciones, a nivel computacional siempre será más eficiente el algoritmo FFT que la definición de la DFT.

5. Funcionamiento de la aplicación

Primero leemos la imagen por columnas, y almacenamos sus componentes RGB en una lista (convertimos la imagen en una lista 1D).

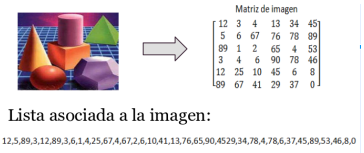

Figura 1: Procesar imagen.

Ahora tomamos el código ASCII asociado al carácter que queremos ocultar, por ejemplo si se quiere ocultar la letra E, su código ASCII es 69, este número lo convertimos en binario: 01000101

Ahora necesitamos ocultar 8 bits, para ello hallamos la TDF de cada 4 componentes de la lista asociada a la imagen.

De forma general, lo que vamos a tener es:
Sean k,l,m,n, 4 componentes de la imagen donde k,l,m,n pertecenen a los números enteros positivos y están en el intervalo [0 - 255]. Al hallar la TDF de esta muestra de tamaño 4 vamos a obtener el siguiente resultado:

$$coeficiente[0] = a, dondea, b, c, d \in Z_+$$
$$coeficiente[1] = b + ci$$
$$coeficiente[2] = d$$
$$coeficiente[3] = b - ci$$

Ahora lo que hacemos es guardar un bit en la parte real (b), y otro bit en la parte imaginaria (c).

Para hacer lo anteriormente dicho razonamos de la siguiente forma:

- Si bit = 0, y el número es par, el numero se deja igual.
- Si bit = 0, y el numero es impar, decrementar el numero en una unidad.
- Si bit = 1, y el numero es impar, el numero se deja igual.
- Si bit = 1 y el numero es par, incrementar el numero en una unidad.

Una vez terminado esto hallamos la transformada inversa, y asi obtenemos la estego-imagen, es decir la imagen que tiene el mensaje oculto.

Por último para obtener el mensaje oculto, solo se halla la TDF de cada 4 componentes de la estego-imagen, y se verifica si la parte real es par entonces el bit oculto es 0, sino entonces es 1, y de la misma forma con la parte imaginaria hasta que se construya todo el mensaje oculto.

Al proponer este algoritmo, el punto clave fue escoger muestras de tamaño 4, para hallarle la TDF, porque de esta forma solo trabajariamos con los números enteros, en cambio si se hubiera escogido muestras mayores a 4, el resultado de la TDF sería valores reales, y al trabajar con estos existe el riesgo de perder información.

La aplicación se encuentra dividida en dos funcionalidades principales, las cuales son:

5.1. Ocultar información

Ventana Principal:

Figura 2: Ventana principal.

Elegimos la imagen que nos servirá para ocultar el mensaje, por ejemplo:

Figura 3: Imagen portadora.

Hacemos clic en ocultar información, después cargamos la imagen portador e ingresamos el mensaje que vamos a ocultar:

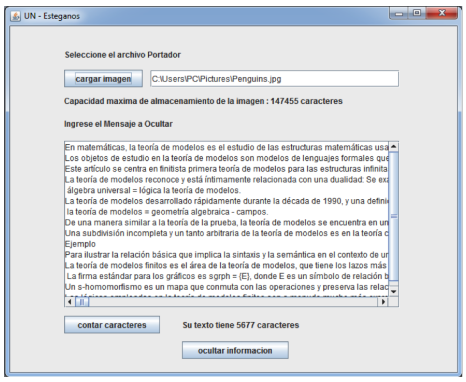

Figura 4: Ocultar mensaje en EstegoImagen.

Después hacemos clic en Ocultar Información y nos aparece la siguiente ventana que nos permitirá guardar la estego imagen:

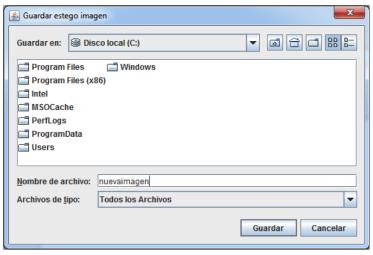

Figura 5: Mensaje al generar EstegoImagen.

Ahora vamos a la ruta donde guardamos la estego imagen, y vemos que esta no se altero en lo más mínimo, a pesar de que ocultamos en ella 5677 caracteres. De hecho hubiéramos podido guardar muchos más caracteres y la imagen no presentaría grandes cambios.

Figura 6: EstegoImagen resultante.

5.2. Revelar información

Ahora volvemos a la ventana principal, seleccionamos la opción revelar información, y cargamos la estego imagen. Después de esto hacemos clic en revelar información, y en el área de texto nos aparecerá el mensaje oculto, y en la parte inferior de la ventana nos mostrara el número de caracteres de dicho mensaje:

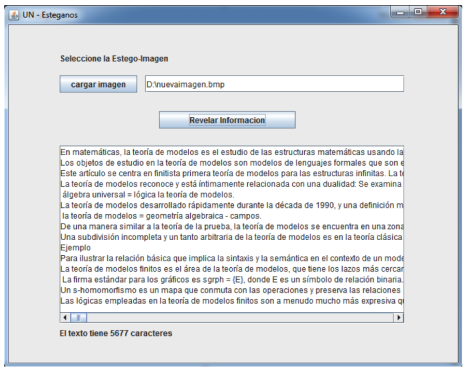

Figura 7: Extracción de mensaje de EstegoImagen.

Referencias

[1] Gustavus J. Simmons,
Advances in Cryptology: Proceedings of CRYPTO '83, The Prisoners' Problem and the Subliminal Channel Plenum, 1983.

[2] J. F. James,
A Students Guide to Fourier Transforms CAMBRIDGE, 2011.

[3] Pablo Roncagliolo,
Procesamiento Digital de Imágenes Universidad Técnica Federico Santa María, 2007.

[4] Ángela Rojas Matas,
Intercambio de información secreta con la Transformada Discreta de Fourier. Universidad de Córdoba, 2008.

[5] Carlos Velasco-Bautista, Julio López-Hernández, Mariko Nakano-Miyatake, Héctor Pérez-Meana,
Esteganografía en una imagen digital en el dominio DCT. Escuela Superior de Ingeniería Mecánica y Eléctrica, Unidad Culhuacán, Instituto Politécnico Nacional 2007.

[6] Mehdi Kharrazi, Husrev T. Sencar, Nasir Memon,
Image Steganography: Concepts and Practice. Polytechnic University 2004.

[7] José Miguel Sanchiz Martí,
La Transformada Discreta de Fourier en Análisis de Imagen. Universidad Jaume.

[8] Faisal Alturki, Russell Mersereau.
Secure blind image steganographic technique using discrete Fourier transformation. Proceedings of 2001 International Conference on Image Processing. Greece, 2001. pp. 542-545.