

Esteganografía digital utilizando la Transformada Discreta de Fourier

INTEGRANTES:

Edwin Alexander Bohorquez
Fernando Muñoz Sánchez
Gustavo Adolfo Castañeda
Jose Deinober Perdomo

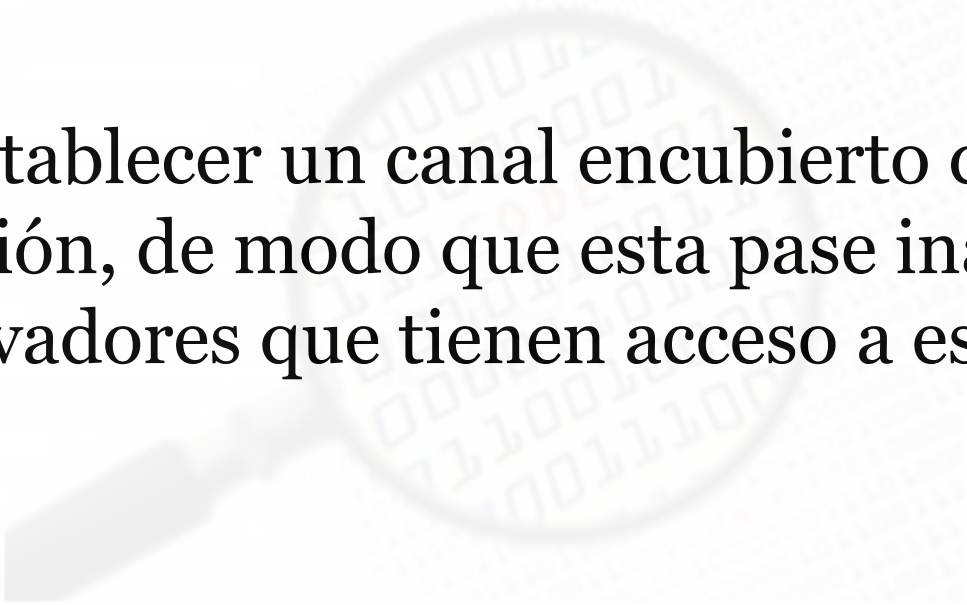
Objetivos

- Aplicar la Transformada Discreta de Fourier a la Esteganografía digital de imágenes.
- Aprender sobre la Transformada de Fourier, la Transformada Rápida de Fourier y la Transformada Discreta de Fourier
- 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.

¿Qué es la esteganografía?

La esteganografía trata el estudio y aplicación de técnicas que permiten ocultar mensajes, dentro de otros de modo que no se perciba su existencia.

Trata de establecer un canal encubierto de comunicación, de modo que esta pase inadvertida para observadores que tienen acceso a ese canal.



Un poco de historia...

- Antigüedad:
 - Herodoto (480 a.C) - The History
 - Tintas Invisibles
- Segunda guerra mundial:
 - Doll Woman - Espía japonesa
 - Jeremiah Denton - Prisionero de guerra

Esteganografía moderna

- Imágenes ocultas entre los fotogramas de un video
- Hacer el color del texto igual al color de fondo.
- Ocultar mensajes en los bits menores de archivos de audio o imágenes (LSB)

Conceptos básicos

- Portador
- Mensaje
- Canal auxiliar
- Estego-imagen

Imagen a nivel computacional

- Una imagen es un arreglo de enteros a nivel computacional.
- Una imagen puede verse como un dominio discreto.
- Los algoritmos esteganográficos no son robustos ante compresión. [jpg, png]
- Se puede trabajar sobre el resultado que da la transformada de una imagen para procesarla.

Aplicaciones de la TDF a las imágenes

- La transformada de Fourier nos permite pasar una imagen al dominio de la frecuencia.
- Los algoritmos de procesamiento de imágenes suelen ser más fáciles de aplicar sobre el resultado de la transformada que sobre las imágenes en sí.

¿Qué es la Transformada Discreta de Fourier (TDF) ?

La transformada discreta de Fourier es una transformación lineal e invertible:

$$F = \mathbb{C}^N \rightarrow \mathbb{C}^N$$

donde \mathbb{C} denota el cuerpo de los números complejos.

Esta transformada recibe como entrada una secuencia de N números complejos y los transforma en otra secuencia de números complejos.

¿Qué es la Transformada Discreta de Fourier (TDF) ?

Esta transformación está dada por la siguiente sumatoria:

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

donde:

- i es la unidad imaginaria : $i = \sqrt{-1}$
- N es la tamaño de la muestra
- x_n representa el n -esimo término de la secuencia de números complejos
- $e^{\frac{-2\pi kni}{N}}$ se conoce como el núcleo de la transformación, que al aplicar la ecuación de euler :

$$e^{i\theta} = \cos(\theta) + i\sin(\theta) \text{ queda de la siguiente forma: } \cos\left(\frac{-2\pi kn}{N}\right) + i\sin\left(\frac{-2\pi kn}{N}\right)$$

Transformada discreta de fourier inversa

La transformada discreta nos permite hallar los coeficientes X_k . Ahora bien, si se quiere recuperar la muestra original se aplica la transformada inversa dada por:

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

donde ahora la entrada de esta transformación serán los coeficientes (números complejos) obtenidos de la TDF.

Cálculo de la TDF

Ejemplo: Hallar la transformada de:

$$X[n] = \{1, 0, 6\} \quad \text{Donde} \quad X[0] = 1, \quad X[1] = 0, \quad X[2] = 6$$

Primero planteamos la sumatoria:

$$x_k = \sum_{n=0}^2 x[n] e^{\frac{-2\pi kni}{3}}$$

Ahora hallamos cada coeficiente:

$$X_0 = 1e^{\frac{-2\pi i(0)(0)}{3}} + 0e^{\frac{-2\pi i(0)(1)}{3}} + 6e^{\frac{-2\pi i(0)(2)}{3}}$$

$$X_0 = 1 + 0 + 6 = 7$$

$$X_1 = 1e^{\frac{-2\pi i(1)(0)}{3}} + 0e^{\frac{-2\pi i(1)(1)}{3}} + 6e^{\frac{-2\pi i(1)(2)}{3}}$$

$$X_1 = 1 + 6e^{\frac{-4\pi i}{3}}$$

Cálculo de la TDF

Ahora usamos la identidad de Euler:

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

$$e^{\frac{-4\pi i}{3}} = \cos\left(\frac{-4\pi}{3}\right) + i \operatorname{sen}\left(\frac{-4\pi}{3}\right)$$

$$X_1 = 1 + 6\left(\cos\left(\frac{-4\pi}{3}\right) + i \operatorname{sen}\left(\frac{-4\pi}{3}\right)\right) = -2 + 5.2i$$

$$X_2 = 1e^{\frac{-2\pi i(2)(0)}{3}} + 0e^{\frac{-2\pi i(2)(1)}{3}} + 6e^{\frac{-2\pi i(2)(2)}{3}}$$

$$X_2 = 1 + 6e^{\frac{-8\pi i}{3}} = 1 + 6\left(\cos\left(\frac{-8\pi}{3}\right) + i \operatorname{sen}\left(\frac{-8\pi}{3}\right)\right) = -2 - 5.2i$$

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

$$R[k] = \{ 7, -2 + 5.2i, -2 - 5.2i \}$$

$$\text{Dónde: } R[0] = 7, \quad R[1] = -2 + 5.2i, \quad R[2] = -2 - 5.2i$$

Complejidad de la TDF

La evaluación directa de la transformada discreta de Fourier requiere $O(n^2)$ operaciones aritméticas.

Es decir si tenemos una muestra de 78 números, tendremos que hacer 6084 operaciones aritméticas, lo cual resulta muy costoso.

¿Como se puede resolver esto ?

FFT

- Es un algoritmo recursivo basado en la periodicidad de la Transformada Discreta de Fourier.
- El algoritmo Cooley-Tukey es una implementación de esta idea.
- Se basa en dividir la DFT en dos sumatorias, tal que una de ellas sea la suma de los términos pares y la otra de los términos impares.

FFT

Con sólo dividir una vez la sumatoria en dos sumatorias equivalentes, obtendremos una reducción del número de operaciones necesarias para el cálculo de la DFT.

Pero si a su vez seguimos dividiendo las partes resultantes, en sumatorias pares e impares sucesivamente hasta que estas ya no se puedan dividir más, lograremos pasar de tener N^2 operaciones, a solo $N \log_2 N$ operaciones

FFT

DFT (N) = N^2 operaciones complejas

FFT(N) = DFN(N/2) pares + DFN(N/2)
impares

FFT(N) = $(N/2)^2 + (N/2)^2 = N^2/2$
operaciones complejas
con solo una división.

FFT

$\text{FFT}(N) = 2\text{FFT}(N/2) + N$ al aplicar la primera división se reduce a la mitad las operaciones y si seguimos el algoritmo recursivo vemos la siguiente secuencia

$$\text{FFT}(N) = 2\text{FFT}(N/2) + N$$

$$\text{FFT}(N) = 2(2\text{FFT}(N/4) + N/2) + N$$

$$\text{FFT}(N) = 4\text{FFT}(N/4) + 2N$$

$$\text{FFT}(N) = 8\text{FFT}(N/8) + 3N$$

$$\text{FFT}(N) = 2^k \text{FFT}(N/2^k) + kN \quad \text{DFT}(1) = 0$$

$$\text{FFT}(N/2^k) = 0$$

$$N/2^k = 1 \quad k = \log_2 N$$

$$\text{FFT}(N) = 2^k(0) + kN \text{ si reemplazamos } k \text{ obtenemos } \text{FFT}(N) = N \log_2 N$$

Algoritmo para ocultar información en una imagen

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



Matriz de imagen

12	3	4	13	34	45
5	6	67	76	78	89
89	1	2	65	4	53
3	4	6	90	78	46
12	25	10	45	6	8
89	67	41	29	37	0

Lista asociada a la imagen:

12,5,89,3,12,89,3,6,1,4,25,67,4,67,2,6,10,41,13,76,65,90,45,29,34,78,4,78,6,37,45,89,53,46,8,0

Algoritmo para ocultar información en una 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.

Algoritmo para ocultar información en una imagen

De forma general, lo que vamos a tener es:

Sean k, l, m, n , 4 componentes de la imagen donde $k, l, m, n \in \mathbb{Z}^+$ y están en el intervalo $[0 - 255]$. Al hallar la TDF de esta muestra de tamaño 4 vamos a obtener el siguiente resultado:

$$\text{coeficiente}[0] = a \quad \text{donde } a, b, c, d \in \mathbb{Z}$$

$$\text{coeficiente}[1] = b + ci$$

$$\text{coeficiente}[2] = d$$

$$\text{coeficiente}[3] = b - ci$$

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

Algoritmo para ocultar información en una imagen

Para hacer lo anteriormente dicho razonamos de la siguiente forma:

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

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

Algoritmo para ocultar información en una imagen

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.

Conclusiones

- Al proponer este algoritmo, el punto clave fue escoger muestras de tamaño 4, para hallarle la TDF, porque de esta forma solo trabajamos 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.
- Fue mucho mejor trabajar con los números enteros que con los números reales, ya que estos nos proporcionaron la seguridad de que los bits ocultos no se perdieran, en cambio con los reales la información se podría perder.

Conclusiones

- El usar la transformada de Fourier en esteganografía nos da un algoritmo de ocultamiento de información más robusto en comparación con otros de índole similar como el LSB.
- No es muy recomendable implementar la DFT directamente, debido al gran tiempo de ejecución que esta toma.
- Existen muchos algoritmos distintos para implementar la FFT
- Nos dimos cuenta, que el uso de la FFT, es una herramienta muy poderosa, no solo en esteganografía, sino también en muchas otras áreas.

Bibliografía

- José Miguel Sanchiz Martí, La Transformada Discreta de Fourier en Análisis de Imagen, Universidad Jaume.
- 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.
- J. F. James, A Students Guide to Fourier Transforms. CAMBRIDGE, 2011.
- Gustavus J. Simmons, Advances in Cryptology: Proceedings of CRYPTO '83, The Prisoners' Problem and the Subliminal Channel. Plenum, 1983.