suma+ 57
Febrero 2008, pp. 23-29

Intercambio de información secreta con la Transformada Discreta de Fourier

Existen ocasiones donde se desea mantener secreta alguna información: transacciones bancarias, secretos militares, comercio electrónico, etc. Pero esta información viajará probablemente por canales inseguros como Internet y podrá ser interceptada por algún intruso no autorizado. Por lo tanto, es muy importante estudiar formas de intercambiar información de forma segura entre dos usuarios. En este trabajo se describe una forma de transmitir información de forma segura que usa la transformada discreta de Fourier.

Sometimes, it is necessary to keep certain information hidden: bank transactions, military secrets, electronic commerce, etc. This information may be sent by means of insecure channels, such as the Internet, and it can be intercepted by some unauthorized person. Hence, it is worth studying a way to interchange information between two users in a safe way. In this article a method to transmit information safely is described, using the Fourier's discrete transformed.

Desde que comenzaron las experiencias sobre implantación experimental del sistema ECTS en nuestra Escuela, nos planteamos qué tipo de actividades académicas dirigidas íbamos a proponer a nuestros alumnos en clase de Matemáticas de primer y segundo curso de Ingeniería Técnica Informática. No queríamos que estas actividades se redujeran a la realización de colecciones de problemas, sino que existieran otro tipo de actividades. Éstas no debían tener una dificultad matemática excesiva para que el alumno pudiese trabajar de forma autónoma y, por otro lado, debían ser atractivas y útiles para el alumno. Con esto perseguimos que los alumnos aprendan Matemáticas sabiendo para qué sirven en la vida real y, concretamente, en qué áreas relacionadas con sus estudios se utilizan, para que así valoren más los conocimientos que están adquiriendo. Con este tipo de actividades pretendemos conseguir los siguientes objetivos:

- Un proceso de enseñanza-aprendizaje más motivador para los alumnos.
- Integrar los contenidos matemáticos de nuestras asignaturas en áreas de interés para la titulación.
- Incentivar la búsqueda de información y la investigación.

Las actividades propuestas han abordado los siguientes temas: métodos de esteganografía digital, técnicas de compresión de imágenes digitales, métodos criptográficos, curvas de Bézier, diseño de fractales, introducción a los códigos detectores y correctores de errores, algoritmo PageRank de Google, etc. En este artículo se desarrolla una de estas actividades: Intercambio de información secreta con la transformada discreta de Fourier.

En el momento actual viaja por Internet una gran cantidad de información que, en ocasiones, debe ser secreta por motivos de seguridad. Por lo tanto, está claro que es muy importante la codificación de esta información pensando siempre en que ésta puede ser interceptada por un intruso malicioso.

En el momento actual viaja por Internet una gran cantidad de información que, en ocasiones, debe ser secreta por motivos de seguridad.

Supongamos que tenemos una imagen digital que representa un punto estratégico dentro de un mapa y queremos enviarla por un canal inseguro como Internet. Podemos enviarla codificada de modo que un intruso sólo vea una imagen "extraña"

Ángela Rojas Matas
Sociedad Andaluza de Educación Matemática Thales
Universidad de Córdoba

y sólo el verdadero receptor sea capaz de decodificar dicha imagen y recuperar la imagen original tal como era en principio. Una situación similar podemos trasladarla a un fichero de audio, por ejemplo una grabación de voz con unas instrucciones secretas se puede codificar de forma que un intruso sólo oiga algo ininteligible.

A continuación se define la transformada discreta de Fourier y un par de propiedades elementales de dicha transformada. Veremos en este mismo apartado cómo usar esta transformada para codificar una imagen. La misma idea se aplicará a un fichero de audio para codificarlo.

Sin embargo puede resultar mucho más interesante que la información que se desea mantener secreta viaje oculta en un fichero digital de cobertura totalmente "inocente" (una imagen de unas vacaciones familiares en la playa, por ejemplo) que no levante sospechas. Si un intruso intercepta una imagen de este tipo probablemente no sospeche nada y, sin embargo, oculta en la imagen puede haber información secreta. De esto se ocupa la esteganografía digital. En el siguiente apartado, se hace una introducción al método más simple y utilizado de la esteganografía digital: el método del bit menos significativo. Posteriormente se describe otro método de esteganografía digital más sofisticado que el anterior, que usa la transformada discreta de Fourier.

La transformada de Fourier es una herramienta matemática compleja y con una gran cantidad de aplicaciones en el tratamiento de señales digitales. Sin embargo, para poder realizar la actividad aquí propuesta, sólo es necesario una breve introducción a dicha transformada.

Transformada Discreta de Fourier: codificación de un fichero de audio o una imagen

Se llama Transformada Discreta de Fourier de un vector

$$Y = (y_0, y_1, \ldots, y_{N-1})$$

de N componentes al vector:

$$\beta = (\beta_0, \beta_1, \ldots, \beta_{N-1})$$

obtenido de la siguiente forma:

$$TDF[Y] = \beta \Rightarrow \beta_n = \sum_{k=0}^{N-1} \omega^{-n\,k}\, y_k \qquad n = 0, 1, \ldots, N-1$$

siendo $\omega = e^{\frac{2\,\pi\,i}{N}}$

Esta transformación se puede llevar a cabo mediante un producto matricial:

$$F = \begin{pmatrix} \vdots & & \\ \cdots & \omega^{-n\,k} & \cdots \\ \vdots & & \end{pmatrix}_{n,\,k\,=\,0,\,1,\,\ldots,\,N-1} \Rightarrow \qquad \begin{pmatrix} \beta_0 \\ \vdots \\ \beta_{N-1} \end{pmatrix} = F \begin{pmatrix} y_0 \\ \vdots \\ y_{N-1} \end{pmatrix}$$

Esta matriz F es inversible y además:

$$F^{-1} = \frac{1}{N}\,\bar{F}$$

siendo $\bar{F}$ la matriz conjugada de F.

Por lo tanto, esta transformación se puede invertir y esto nos conduce a la definición de la transformada inversa:

$$\beta = F\,Y \;\Rightarrow Y = F^{-1}\,\beta \Rightarrow Y = \frac{1}{N}\;\bar{F}\;\beta$$

siendo:

$$\begin{pmatrix} y_0 \\ \vdots \\ y_{N-1} \end{pmatrix} = F^{-1} \begin{pmatrix} \beta_0 \\ \vdots \\ \beta_{N-1} \end{pmatrix}$$

siendo:

$$F^{-1} = \frac{1}{N} \begin{pmatrix} \vdots & & \\ \cdots & \omega^{n\,k} & \cdots \\ \vdots & & \end{pmatrix}_{n,\,k\,=\,0,\,1,\,\ldots,\,N-1}$$

La *transformada discreta inversa de Fourier* vendrá dada por:

$$ITDF[\beta] = Y \Rightarrow y_k = \frac{1}{N}\sum_{n=0}^{N-1} \beta_n\, \omega^{nk} \qquad k = 0, 1, \ldots, N-1$$

Si Y es un vector real con N par, puede demostrarse fácilmente que:

1) β_0 y β_M son números reales, siendo:

$$M = N/2$$

2) Los coeficientes:

$$\{\beta_{M+1}, \beta_{M+2}, \ldots, \beta_{N-1}\}$$

no tienen mucho interés ya que son los complejos conjugados y en orden inverso de los coeficientes:

$$\{\beta_1, \beta_2, \ldots, \beta_{M-1}\}$$

ya que: $\beta_{M+1} = \overline{\beta_{M-1}}$, $\beta_{M+2} = \overline{\beta_{M-2}}$, ..., $\beta_{N-1} = \overline{\beta_1}$

Como hemos dicho antes, la TDF se puede calcular mediante un producto matricial, pero si el vector es de gran tamaño, serán muchos los cálculos necesarios. El comando *Fourier* de Mathematica nos permite calcular rápidamente la TDF ya que aplica la llamada *transformada rápida de Fourier*, que es simplemente una forma de disponer los cálculos de modo que se reducen drásticamente el número de operaciones que hay que efectuar. El comando *InverseFourier* nos devuelve la transformada inversa de Fourier.

Ahora vamos a ver cómo podemos conseguir una codificación de un fichero digital de un modo muy simple. Supongamos, por ejemplo, que tenemos una grabación de audio con una información secreta. Queremos codificar el fichero de audio de modo que si alguien no autorizado lo escucha sólo oiga cosas ininteligibles y sólo el receptor autorizado sea capaz de decodificar el fichero.

Una grabación de audio digital es un vector:

$$Y=(y_0, y_1, \ldots, y_{N-1})$$

de datos reales (supondremos que tiene un número par de componentes, si no es así añadiremos un cero). Con la orden ListPlay de Mathematica podemos oír la señal original almacenada en dicho vector. La codificación se puede conseguir de muchísimas formas, por ejemplo la siguiente:

• Se escriben los datos en orden inverso: .

$$(y_{N-1}, \ldots, y_1, y_0)$$

• Calculamos la transformada discreta de Fourier de la lista anterior, obteniendo el vector β .

• Las componentes $\{\beta_1, \beta_2, \beta_3, \ldots, \beta_{M-1}\}$ se dividen en cuatro bloques a, b, c y d que se intercambian entre sí, por ejemplo d, c, b y a, obteniendo la lista:

$$\left\{\beta'_1, \beta'_2, \cdots, \beta'_{M-1}\right\}$$

• Se construye el vector:

$$\beta' = \left\{\beta_0, \beta'_1, \beta'_2, \cdots, \beta'_{M-1}, \beta_M, \overline{\beta'_{M-1}}, \overline{\beta'_{M-2}}, \overline{\beta'_1}\right\}$$

• Se calcula la transformada inversa del vector anterior, obteniendo el vector Y'.

Si oímos el vector Y' con la orden ListPlay, el resultado será ininteligible. Sin embargo, el receptor autorizado que recibe el vector Y' podrá seguir los pasos inversos de los descritos anteriormente para lograr recolocar los coeficientes en su sitio original. En definitiva, será capaz de recuperar el vector original . Lógicamente, no podemos proporcionar en estas páginas un ejemplo de codificación de un fichero de audio.

Sin embargo, esta misma idea podríamos aplicarla a una imagen que queremos enviar codificada. En el ejemplo que presentamos a continuación, una imagen de tamaño 256X256 fue convertida a un vector de 65536 componentes, escribiendo los niveles de gris de los píxeles uno detrás de otro. A este vector se le aplicaron todos los pasos descritos anteriormente, obteniendo un vector Y' de 65536 componentes que volvió a convertirse en matriz de tamaño 256X256. El resultado es la imagen codificada que se muestra en la figura 1.

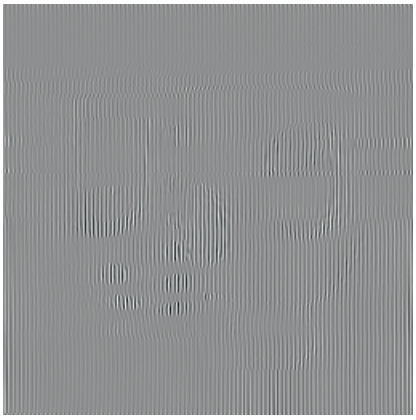

Figura 1

Esta imagen viajará probablemente por un canal inseguro a un usuario autorizado que, conocedor de los pasos seguidos en su codificación, será capaz de recuperar la imagen original. La imagen que se recupera es la representada en la figura 2.

Figura 2

Introducción a la esteganografía digital

En la sección anterior hemos visto cómo codificar una imagen digital (o un fichero de audio). Pero si ésta es interceptada por un intruso malicioso, consciente de que se trata de una imagen codificada, podría intentar decodificarla para ver qué información se intenta ocultar. Existe otra forma de ocultar información de modo que no levante sospechas, de esto se ocupa la esteganografía digital.

La esteganografía digital (Provos, 2003) se ocupa de la ocultación de información que se desea mantener secreta en un objeto de cobertura "inocente" como puede ser una imagen digital o un fichero de audio. La información a ocultar puede ser de distinto tipo:

- un mensaje de texto: un usuario envía a otro unas instrucciones secretas en un mensaje de texto.
- una imagen: un usuario puede enviar a otro una fotografía secreta con la localización concreta de un punto estratégico dentro de un mapa.
- una grabación de audio: donde se hace una grabación de voz con instrucciones secretas.

En los tres casos, la información secreta a ocultar se convertirá en una secuencia de bits: ceros y unos. Esta secuencia de bits será almacenada en un fichero digital que nos va servir de cobertura.

Nos vamos a ocupar de cómo ocultar información concretamente en una imagen digital. Para ocultar la información secreta, la imagen original será ligeramente modificada por el algoritmo de ocultación utilizado obteniendo una nueva imagen que se conoce con el nombre de estego-imagen. Comenzamos comentando en qué consiste el método esteganográfico más usado: el método LSB (**L**east **S**ignificative **B**it), o método del bit menos significativo.

Una imagen digital en escala de grises (o como decimos vulgarmente, una imagen en blanco y negro) es una matriz de números, donde cada número indica el nivel de gris de cada píxel, habitualmente entre 0 (negro) y 255 (blanco). Para almacenar un nivel de gris necesitamos 8 bits, siendo el último bit el menos significativo. Por ejemplo, una imagen 256×256 se compone de 65536 niveles de gris. Si usamos solamente el último bit, es decir, el bit menos significativo, el número máximo de bits que se pueden ocultar en dicha imagen sería: 256×256 = 65536.

Vamos a tomar una imagen en blanco y negro de tamaño como imagen de cobertura. Por otro lado, supongamos que tenemos un mensaje de texto a ocultar: mensaje = " La criptografía…" que convertiremos a código ANSI: {76, 97, 32, …} y que después pasaremos a binario (8 bits), obteniendo una ristra de bits a ocultar. En nuestro ejemplo el mensaje estaba compuesto por 6584 caracteres , lo que nos proporciona una ristra de 6584×8 = 52672 bits a ocultar.

La esteganografía digital (Provos, 2003) se ocupa de la ocultación de información que se desea mantener secreta en un objeto de cobertura "inocente" como puede ser una imagen digital o un fichero de audio.

Supongamos que deseamos almacenar un bit secreto en el bit menos significativo del nivel de gris de un pixel. Por ejemplo, si el nivel de gris es 112, que en binario es 01110000, entonces la forma de proceder sería como sigue:

- si el bit a ocultar es 0, no hacer nada ⇒ nivel_gris_modificado = $01110000_{2)} = 112$

- si el bit a ocultar es 1, modificar bit menos significativo ⇒ nivel_gris_modificado = $01110001_{2)} = 113$

Esto se puede implementar de una forma muy sencilla en la práctica:

Imagen Original

Estego-Imagen

Figura 3

• Si el bit a ocultar es 0 y el nivel de gris del píxel es impar entonces decrementar el nivel de gris en 1.
• Si el bit a ocultar es 1 y el nivel de gris del píxel es par entonces aumentar el nivel de gris en 1.

Razonando de esta forma se obtiene el resultado mostrado en la Figura 3.

En este caso hemos ido ocultando un bit en cada pixel de forma secuencial a la largo de la imagen de cobertura. Como vemos, no es apreciable a simple vista ninguna modificación. Ése es el objetivo de la esteganografía, la ocultación de un mensaje secreto en una fichero digital pero de tal forma que no se haga patente la manipulación a la que ha sido sometido dicho fichero. Para recuperar el mensaje oculto bastará con leer los píxeles de la estego-imagen de forma secuencial: si el nivel de gris es par, el bit oculto es un cero, en caso contrario, el bit oculto es un uno.

Si usamos los dos últimos bits (los dos bits menos significativos) para ocultar información en esa misma imagen, el número máximo de bits que podremos ocultar será: 256×256×2=131072. No se recomienda usar más de dos bits, por el deterioro que se produce en la estego-imagen.
Una imagen en color en formato RGB (**R**ed **G**reen **B**lue), se compone de tres capas o planos de color: rojo, verde y azul. Cada capa es una imagen del mismo tamaño que la original donde cada dato se interpreta como la cantidad de rojo, verde o azul presente en el píxel de la imagen en color. Los datos de cada capa de color ocupan un byte (8 bits): desde el 0 hasta el 255. Una imagen en color va a tener el triple de capacidad para ocultar información que una imagen en blanco y negro. Así, por ejemplo para una imagen en color de tamaño 256×256 tendríamos: 256×256×3=196608 si usamos sólo 1 bit de cada píxel para ocultar información secreta.

En el ejemplo mostrado en la Figura 4 se oculta un mensaje de 79008 caracteres que equivale a 79008×8=632064 bits en los tres planos de color usando el método LSB y el bit menos significativo de cada píxel en las tres capas de color.

Imagen Original

Estego-Imagen

Figura 4

La información secreta a ocultar en el caso anterior era un mensaje de texto. Pero el tipo de información puede ser de cualquier tipo: en todos los casos dará lugar a una cadena de bits, ceros y unos, que se almacenarán en la imagen de cobertura. En la Figura 5, tenemos a la izquierda una estego-imagen de un cuadro de Vincent Van Gogh, una imagen en color de 500×396 píxeles donde se ha ocultado una imagen en blanco y negro de tamaño 249×266 píxeles usando el bit menos significativo. La imagen oculta extraída se presenta a la derecha de la Figura 5.

Estego-Imagen

Imagen secreta oculta

Figura 5

no es resistente a una compresión JPEG. En la siguiente sección se describe un método esteganográfico que se basa en la transformada discreta de Fourier y que tiene la ventaja de ser resistente a ligeras compresiones JPEG.

Esteganografía digital con la Transformada Discreta de Fourier

Ahora vamos a usar una idea diferente, vamos a aplicar una transformada discreta, por ejemplo la transformada de Fourier (Alturki, 2001), y a continuación vamos a manipular los coeficientes de la transformada para ocultar la información secreta. Resumiendo, los cambios se van a efectuar en el dominio de la transformada. Por supuesto, se pueden usar otras transformadas: la transformada discreta del coseno, la transformada wavelet de Haar, etc.. La ventaja de este nuevo método es que la estego-imagen es resistente a ligeras compresiones JPEG, es decir, que a pesar de la pérdida de información que supone una compresión JPEG, la información secreta se puede recuperar.

Para ello, la imagen de entrada se convertirá en una lista 1D. El método LSB es sencillo de implementar y muy usado en la práctica. Además admite múltiples variaciones: los píxeles se pueden escoger de forma pseudoaleatoria, se puede cifrar antes la información secreta a ocultar, etc.. Sin embargo, tiene un inconveniente importante: la estego-imagen no se debe alterar en absoluto porque si ésta se modifica se destruye el mensaje oculto. Si cogemos la estego-imagen y la comprimimos usando el formato JPEG, aunque sea ligeramente, se destruye totalmente el mensaje oculto. En resumen, este método

Si la imagen es 256×256, esta lista unidimensional tendrá 65536 datos. Este vector lo indicaremos como *Y*. A continuación calcularemos la TDF de dicho vector, usando el comando *Fourier* de Mathematica, obteniendo un vector β también con 65536 componentes complejas. Usaremos tanto la parte real como la parte imaginaria de algunos coeficientes del vector β para ocultar los bits de la información secreta de una forma similar a la comentada en el método LSB.

Supongamos que una de las componentes de β es el número complejo *a+bi*, tanto en la parte real como en la imaginaria ocultaremos un bit. Nos fijaremos en la parte real (igual se razona con la parte imaginaria) que hemos llamado *a*. Se calculan los números:

$$\text{signo} = \begin{cases} 1 & a >= 0 \\ -1 & a < 0 \end{cases} \qquad q = \text{Abs}\left[\text{Round}\left[\frac{a}{\Delta}\right]\right]$$

donde Δ es una constante fija, *Abs* es la función valor absoluto y *Round* es la función que nos devuelve el entero más cercano. En la paridad de q es donde se guarda el bit secreto,

igual que con el método LSB, es decir, lo haremos par si el bit que toca ocultar es un cero y lo haremos impar si el bit que toca ocultar es un uno. Después de esto el valor de q modificado, lo indicamos por q' . Entonces calculamos:

$$a' = signo \cdot q' \cdot \Delta$$

Lógicamente no se obtiene exactamente el número original a. De la misma forma se modifica la parte imaginaria b para ocultar otro bit de información obteniendo b'. Después de la ocultación de los dos bits, la componente de Fourier que originalmente era $a+bi$ será sustituida por $a'+b'i$.

La ventaja de este nuevo método es que la estego-imagen es resistente a ligeras compresiones JPEG, es decir, que a pesar de la pérdida de información que supone una compresión JPEG, la información secreta se puede recuperar.

Razonando de esta forma, se irán modificando los coeficientes del vector β, obteniendo otro vector distinto β', donde se habrán ocultado los bits secretos. No hemos usado todas las componentes de para ocultar bits sino sólo los coeficientes

$$\{\beta_1, \beta_2, ..., \beta_{M-1}\}$$

siendo M = N/2.

A continuación, se calcula la transformada inversa de Fourier. Lógicamente no se recupera el vector Y original, sino otro vector Y'. Este vector Y' se volverá a escribir en formato matricial, obteniendo la estego-imagen.

En la Figura 6 mostramos la imagen original y la estego-imagen creada usando el método de la transformada discreta de Fourier. Hemos usado una imagen en blanco y negro de tamaño 256×256 como imagen de cobertura donde hemos ocultado un mensaje compuesto por 6584 caracteres. En este caso, la ventaja frente al método LSB es que, si la estego-imagen se comprime ligeramente, se puede seguir recuperando el mensaje oculto. Por supuesto, el método descrito en esta sección admite multitud de variaciones. ■

Imagen Original

Estego-Imagen

Figura 6

REFERENCIAS BIBLIOGRÁFICAS

ALTURKI, F., MERSEREAU, R. (2001): "Secure blind image steganographic technique using discrete Fourier transformation". Proceedings of 2001 International Conference on Image Processing. Greece, pp. 542-545.

PROVOS, N. (2003): "Hide and Seek: An Introduction to Steganography". IEEE Security & Privacy. IEEE Computer Society, June 2003, pp. 32-44.