

# Método Robusto de Marca de Agua para Audio con Transformada Haar\*

## *Robust Audio Watermaking Using Haar Transform*

Marco A. Acevedo M, José Luis López Bonilla y Roberto Linares y Miranda, ,

Sección de Estudios de Posgrado e Investigación ESIME-IPN

UPALM Edificio Z-4 3er Piso, Col. Lindavista, México D.F.

Tel 5 729-6000 ext, 54756

e-mail: macevedo@ipn.mx

*Artículo recibido en enero 14, 2003; aceptado en enero 21, 2005*

### Resumen

En este trabajo se describe un método robusto de marca de agua para introducirse en archivos de audio. La novedad de este método es la combinación de la transformada wavelet-Haar y el algoritmo Patchwork en el dominio de la frecuencia. La combinación propuesta hace una marca de agua robusta contra compresión, filtrado y remuestreo. Las pruebas de los archivos de audio se implementaron en un programa desarrollado en MATLAB.

**Palabras clave:** Wavelets, Marcas de agua, Patcwork.

### Abstract

This work shows a robust Watermarking method to be introduced in audio files. The new in this method is the use of the Wavelet Haar Transform in combination with Patcwork algorithm in the frequency domain. This combination makes the Watermarking robust against compression, filtering and resampling. We use MATLAB to implement the tests of audio files.

**Keywords:** Wavelets, Watermarking, Patcwork.

## 1 Introducción

En la actualidad los medios digitales son más populares y se han abierto a la distribución compacta de información con una gran flexibilidad y disminución de costos. Sin embargo se presenta el problema de que la información puede ser duplicada y redistribuida virtualmente sin costo, lo que hace que se pierda el derecho de autor. La protección del derecho de autor y el uso autorizado de la información es indispensable por lo que los retos de desarrollo para la información digital ya sea: texto, imágenes, audio y video requieren de una identificación de autenticidad para evitar copias ilegales y pérdida del derecho de autor.

La marca de agua se está utilizando como un mecanismo de protección de la información digital para las copias ilegales, incrustando un identificador del autor en los datos. Al respecto una gran cantidad de publicaciones se han presentado [1]-[4], enfocadas al desarrollo de marcas de agua robustas que protejan contra ataques de reducción de tamaño, eliminación y filtraje. Recientemente para soportar esos ataques, algunos de los esquemas de marca de agua emplean el modelo del “espectro disperso” [5]-[8], que transforma el documento al dominio de la frecuencia y perturba los coeficientes de la perceptibilidad de funciones más significantes, pero su implementación requiere de un procesamiento complicado.

En este trabajo se presenta el desarrollo de un método de marca de agua digital en un archivo de audio, cumpliendo con las principales características que se requieren para la protección de una obra. El método propuesto es bastante flexible, tal que se implementa en MATLAB con un archivo wav.

\* Financiado por el Proyecto CGPI20040349

\*\*Editorial correction: to the list of authors is added Juan Carlos Caravantes missing in previos version.

La marca de agua se introduce en el dominio de la frecuencia, al igual que el modelo de espectro–disperso y de la propuesta presentada por Hong Oh Kim *et al.* [9], quienes usan la Transformada Wavelet . En este trabajo se usa la Transformada Wavelet-Haar, a la cual se le modifica la tercera cifra significativa de cada muestra para introducir un bit de la información de la marca de agua. La Transformada Wavelet-Haar combinada con el algoritmo Patchwork, producen una marca de agua bastante robusta y fácil de implantar, que básicamente corresponde a la aportación de este trabajo que tiene la siguiente distribución: después de la introducción, en la sección II se explica brevemente la transformada Wavelet-Haar y el algoritmo Patchwork para la introducción de la marca de agua en archivos de audio; en la parte III se muestra la implementación en MatLab y finalmente se reportan las pruebas y conclusiones.

## 2 Conceptos Básicos

### A. Transformada Wavelet Haar Discreta (TWHD)

El archivo de audio es muestreado en MATLAB. La secuencia resultante  $x[n]$  se pasa a través de un banco de filtros espejo en cuadratura [9]. Posteriormente la señal de salida de cada filtro se decima por un factor de 2. La idea principal es la descomposición de  $x[n]$  en sus componentes de bajas y altas frecuencias.

La resolución de la señal, la cual es una medida de la cantidad detallada de la información dentro de la señal, se modifica por la acción de filtrado y la escala se modifica por la operación de decimación. La decimación de una señal corresponde a la reducción de la velocidad de muestreo, o a la eliminación de algunas de las muestras de la señal. A este proceso de filtrado y decimación sucesiva se le conoce como codificación en subbandas como se muestra en la figura 1.

De acuerdo a la velocidad de muestreo de Nyquist, la frecuencia de muestreo de la señal es igual a  $2\pi$ . Por lo tanto la componente de frecuencia más alta que existe en la secuencia  $x[n]$  será de  $\pi$  radianes.

El filtro pasa bajas elimina la mitad de las frecuencias, lo cual puede ser interpretado como la pérdida de la mitad de la información. Así, la secuencia  $v[n]$  tiene la frecuencia más alta en  $\pi/2$  y la resolución se reduce a la mitad después de la operación de filtrado, pero deja la escala sin cambios. Después de pasar  $x[n]$  a través de un filtro pasa bajas, la mitad del número de muestras de la secuencia  $v[n]$  se hacen redundantes y pueden ser eliminadas sin ninguna pérdida de información.

Por lo que la señal puede decimarse por un factor de 2, y  $y_{bajo}[n]$  tendrá la mitad del número de muestras. Sólo el proceso de decimación modifica la escala, que en este caso se ha duplicado [2].

Este procedimiento puede expresarse como:

$$y_{bajo}[n] = \sum_{k=-\infty}^{\infty} h[k] \cdot x[2n - k] \quad (1)$$

De este modo opera la Transformada Wavelet-Haar Discreta (TWHD), esta acción analiza la señal en diferentes bandas de frecuencia con diferentes resoluciones mediante la descomposición de la señal en componentes de mayor y menor energía.

La TWHD emplea dos tipos de funciones, llamadas funciones de escalamiento y funciones wavelet, las cuales están asociadas con los filtros pasa bajas y pasa altas, respectivamente. La descomposición de la señal en diferentes bandas de frecuencia se obtiene mediante el filtrado sucesivo de la señal en el dominio del tiempo como se muestra en la figura 1.

La secuencia original  $x[n]$  se pasa por un filtro pasa altas  $g[n]$  y uno pasa bajas  $h[n]$ . Después del filtrado, la mitad de las muestras pueden eliminarse de acuerdo con el criterio de Nyquist, ya que la señal tiene ahora la frecuencia más alta en  $\pi/2$  radianes en lugar de  $\pi$ . La señal puede así decimarse por factor de 2, simplemente descartando una muestra sí y una no.

Esto constituye un nivel en la descomposición y puede expresarse matemáticamente como:

$$y_{alto}[k] = \sum_n x[n] \cdot g[2k - n] \quad (2)$$

$$y_{bajo}[k] = \sum_n x[n] \cdot h[2k - n] \quad (3)$$

Donde  $y_{alto}[k]$  y  $y_{bajo}[k]$  son las salidas de los filtros pasa altas y pasa bajas respectivamente, después de la decimación por 2.

La codificación en subbandas puede repetirse para mayor descomposición. Cada nivel de filtrado y decimación resultará en la mitad del número de muestras (y por lo tanto en la mitad de la resolución en el tiempo) y esto divide a la mitad la banda de frecuencias abarcada (y por lo mismo duplica la resolución en la frecuencia). La figura 1 ilustra este procedimiento, donde  $x[n]$  es la señal original que va a ser descompuesta,  $h[n]$  y  $g[n]$  son los filtros pasa altas y pasa bajas, respectivamente.

Supongamos que la señal original  $x[n]$  tiene 512 puntos de muestras, como se observa en la figura 2, con una banda de frecuencia de 0 a  $\pi$  rad/s. Para descomponer la secuencia  $x$  ( $x[n]$ ) se aplicarán la función de escalamiento y la función wavelet de la transformada Haar de primer nivel.

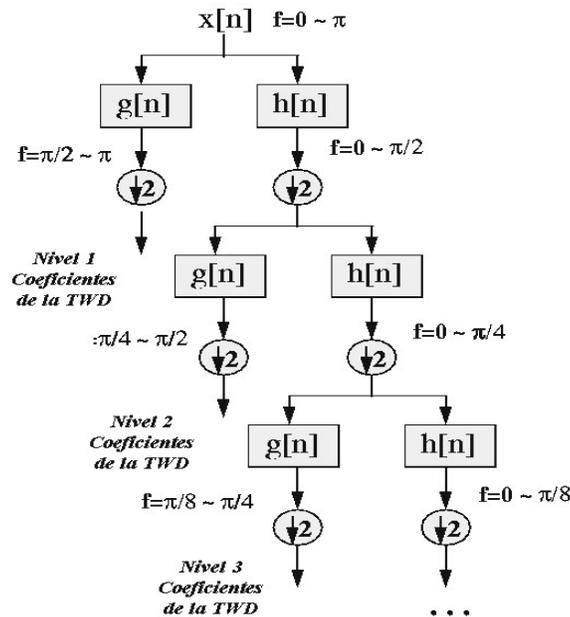


Fig. 1. Algoritmo de codificación en subbandas

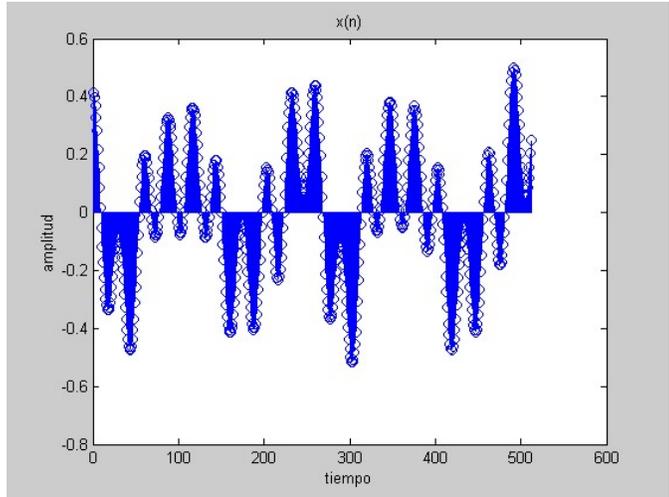


Fig. 2. 512 muestras de la señal de audio

Función wavelet de primer nivel.

$$\begin{aligned}
 \mathbf{W}_1^1 &= \left( \frac{1}{\sqrt{2}}, \frac{-1}{\sqrt{2}}, 0, 0, \dots, 0 \right) \\
 \mathbf{W}_2^1 &= \left( 0, 0, \frac{1}{\sqrt{2}}, \frac{-1}{\sqrt{2}}, 0, 0, \dots, 0 \right) \\
 &\vdots \\
 \mathbf{W}_{N/2}^1 &= \left( 0, 0, \dots, 0, 0, \frac{1}{\sqrt{2}}, \frac{-1}{\sqrt{2}} \right)
 \end{aligned} \quad (4)$$

Función de escalamiento de primer nivel [3].

$$\begin{aligned}
 \mathbf{V}_1^1 &= \left( \frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}, 0, 0, \dots, 0 \right) \\
 \mathbf{V}_2^1 &= \left( 0, 0, \frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}, 0, 0, \dots, 0 \right) \\
 &\vdots \\
 \mathbf{V}_{N/2}^1 &= \left( 0, 0, \dots, 0, 0, \frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}} \right)
 \end{aligned} \quad (5)$$

Para eliminar las frecuencias bajas de la señal original se realiza el siguiente producto escalar entre la señal  $x$  y la función wavelet  $\mathbf{W}$ :

$$d_m = \mathbf{f} \cdot \mathbf{W}_m^1 \quad (6)$$

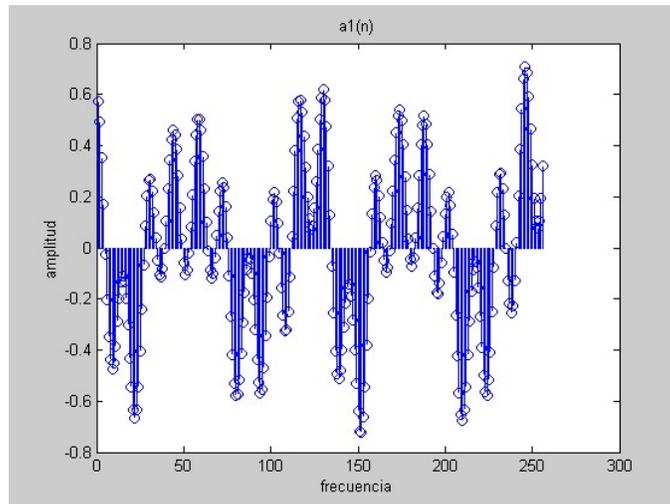
para toda  $m = 1, 2, \dots, N/2$ .

A la subseñal  $d_m$  resultante del producto escalar se le llama *fluctuación*. Del mismo modo, para eliminar las componentes de altas frecuencias de la señal original se realiza el producto escalar entre la señal  $x$  y la función de escalamiento  $\mathbf{V}$ :

$$a_m = \mathbf{f} \cdot \mathbf{V}_m^1 \quad (7)$$

para toda  $m = 1, 2, \dots, N/2$ .

A la subseñal  $a_m$  resultante del producto escalar se le llama  $c$ . En el primer nivel de descomposición la subseñal *promedio*  $a1[n]$  tiene 256 puntos (la mitad de la resolución en el tiempo) como se muestra en la figura 3, pero sólo abarca las frecuencias 0 a  $\pi/2$  rad/s (el doble de la resolución en la frecuencia).



**Fig. 3.** Subseñal promedio de 256 muestras de la señal

Estas 256 muestras constituyen el primer nivel de los coeficientes de la TWD. La subseñal *fluctuación*  $d1[n]$  tiene también 256 muestras, como se exhibe en la figura 4, pero abarca la otra mitad de la banda de frecuencia, es decir, las frecuencias de  $\pi/2$  a  $\pi$  rad/s.

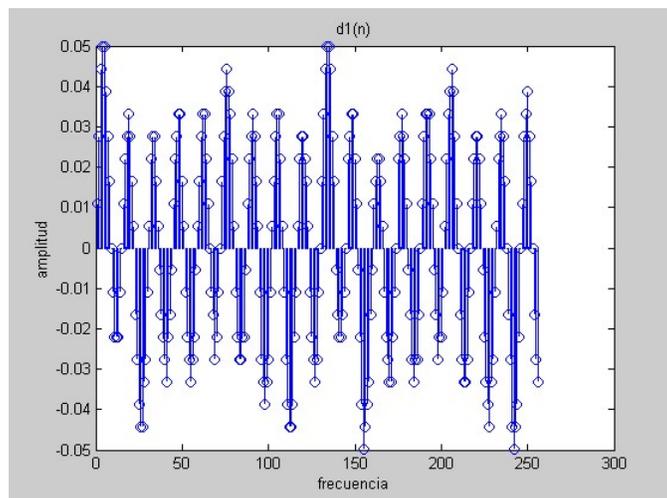


Fig. 4. Subseñal fluctuación de 256 muestras de la señal

Se puede observar que las amplitudes de la subseñal promedio  $a1[n]$  son mucho mayores que las de la subseñal  $d1[n]$ . Además, la subseñal  $a1[n]$  tiene las componentes de mayor energía donde se introduce posteriormente la marca de agua de acuerdo al algoritmo *Patchwork*.

Para una mayor descomposición a la secuencia  $a1[n]$  se le vuelve a aplicar el mismo proceso para obtener una subseñal promedio  $a2[n]$  de 128 muestras que abarcan una banda de frecuencia de 0 a  $\pi/4$  rad/s, y la subseñal  $d2[n]$  también tendrá una secuencia de 128 muestras que abarcan una banda de frecuencia de  $\pi/4$  a  $\pi/2$  rad/s. Esta señal tiene la mitad de la resolución en el tiempo, pero el doble de la resolución en la frecuencia respecto a la señal del primer nivel. En otras palabras, la resolución en el tiempo ha disminuido en un factor de 4, y la resolución en la frecuencia ha aumentado en un factor de 4 comparada con la señal original. Este podría continuar hasta que existieran sólo dos muestras. Para este ejemplo específico habría 8 niveles de descomposición, teniendo cada uno la mitad del número de muestras del nivel anterior. La TWHD de la señal original es luego obtenida concatenando todos los coeficientes comenzando del último nivel de descomposición. La TWHD tendrá entonces el mismo número de coeficientes que la señal original.

## B. Algoritmo Patchwork

Este algoritmo usa una codificación de tipo redundante para esparcir información oculta dentro de las muestras de audio haciendo una pequeña modificación. En este método se usa una clave secreta (contraseña) para un proceso pseudo-aleatorio que escogerá pares de muestras. Para cada muestra elegida, la muestra de mayor amplitud se hace mayor y la muestra de amplitud menor se hace menor y el contraste entre las dos amplitudes codifica un bit. Para ilustrar esto,  $n$  pares de muestras del archivo de audio ( $I_1, I_2$ ) son elegidas pseudo-aleatoriamente. La amplitud de  $I_1$  es disminuida en 1 y la de  $I_2$  es incrementada. Para un archivo de audio con información cifrada, el valor esperado de la suma de las diferencias de  $n$  pares de puntos es entonces  $2n$  como se observa en las siguientes ecuaciones:

Asumiendo que:

$$S = \sum_{k=1}^n (I_1[k] - I_2[k]) \approx 0 \quad (8)$$

Introducción de la marca de agua:

$$\begin{aligned} I'_1[k] &= I_1[k] + 1 \\ I'_2[k] &= I_2[k] - 1 \end{aligned} \quad (9)$$

Detección de la marca de agua:

$$S' = \sum_{k=1}^n (I'_1[k] - I'_2[k]) \approx 2n \quad (10)$$

La marca de agua está presente si  $S' \approx 2n$ . El proceso de extracción opera encontrando los mismos pares de píxeles o muestras de sonido que fueron escogidos en el proceso de inserción y analizando la diferencia de contraste para cada par.

La principal ventaja de esta técnica es que sin la clave para el generador de índices pseudo-aleatorios la marca es muy difícil de quitar completamente sin dañar la señal anfitriona.

Para el caso particular del programa de marca de agua realizado en este trabajo, se utiliza una contraseña que puede variar de un mínimo de 2 a un máximo de 10 caracteres. Esta contraseña se usa para realizar el proceso pseudo-aleatorio que genera los índices que escogen los pares de muestras de audio en los que va a ir codificado un bit.

### 3 Implementación en MATLAB

Se realizaron dos programas que corresponden a la introducción y extracción de una marca de agua en un archivo wav. A continuación se da una breve explicación de los programas.

En este programa se aplica el método propuesto por Hong Oh Kim et al [1], quienes hacen mención a la no necesidad de la señal original de audio al momento de la detección de la marca de agua, ya que la información puede ser introducida bajo el dominio de una wavelet. Además, se introduce una cadena de bits de sincronización para una rápida introducción y extracción de la marca da agua.

En el programa se usaron dos métodos para cifrar la información. Para introducir la contraseña y la longitud de la marca de agua se utilizó la inserción en el bit menos significativo. El otro método fue el algoritmo *Patchwork* para introducir la marca de agua. Cabe destacar que la longitud de la marca de agua y la contraseña se introducen también en el dominio de la transformada wavelet. La longitud de la marca de agua y la contraseña se manejan a nivel de bits, de modo que en cada muestra de la subseñal promedio  $aI[n]$  se introduce un bit. Se debe evitar comenzar a escribir los bits de la marca de agua al principio del archivo wav, ya que regularmente estos archivos comienzan con silencio y la marca de agua podría ser descubierta rápidamente. En este trabajo la primera posición se determina de acuerdo a la siguiente expresión:

$$P_1 = 2 \sum_{i=1}^m c_i \quad (11)$$

donde  $P_1$  es la primera posición donde se comienza a introducir la información,  $c_i$  es el valor ASCII de cada letra de la contraseña y  $m$  la longitud de la contraseña.

Por ejemplo, supongamos un archivo wav con 5,000 muestras, que al aplicarle una TWHHD tipo Haar se divide en 2500 muestras para la subseñal promedio  $aI[n]$  y 2500 muestras para la subseñal fluctuación  $dI[n]$ , en donde se desea introducir la palabra “marca” con la contraseña “agua”, la primera posición en donde se comienza a colocar la información en  $aI[n]$  será de acuerdo a la ecuación (11):

$$P_1 = 2 \sum_{i=1}^4 c_i = 2(97 + 103 + 117 + 97) = 828$$

Ya que se tiene la posición de la primera muestra, se procede a introducir el valor binario de la longitud de la marca de agua, alterando el bit menos significativo de cada muestra. Para este ejemplo, si la información de la marca de agua es “marca”, entonces tiene una longitud de 5 caracteres. El valor binario de 5 es 00000101. Estos bits se colocan a partir de la muestra  $aI[828]$  hasta  $aI[835]$ .

Inmediatamente después se procede a almacenar los bits del código ASCII de cada caracter que conforma la contraseña. En este caso  $m \times 8 = 32$  bits. Los caracteres de la contraseña se introducen desde  $aI[836]$  hasta  $aI[867]$ . Esta información sirve para autenticar la contraseña cuando se quiera extraer la marca de agua. Además, se verifica el orden de los caracteres, por lo que al introducir las letras “aagu” no se podrá extraer ninguna información.

La marca de agua se introduce inmediatamente después. La longitud total de la marca de agua  $L_{ma}$  se define como:

$$L_{ma} = 8 \cdot n_m \cdot 3 \quad (12)$$

donde 8 es el número de bits de cada caracter,  $n_m$  es la longitud de la marca de agua y 3 es la redundancia de la información. Aplicando la ecuación (12) al ejemplo se tiene:

$$L_{ma} = 8 \cdot 5 \cdot 3 = 120 \text{ bits}$$

La marca de agua comenzará a introducirse desde la muestra:

$$I_1[1] = P_1 + (8 + (mx8)) \quad (13)$$

De acuerdo al ejemplo, la primera muestra donde se comenzará a introducir la marca de agua es  $aI[I_1[1]] = aI[868]$ .

Al utilizar el algoritmo *Patchwork*, el cual dice que para almacenar un bit es necesario utilizar dos muestras, la cantidad de muestras necesarias para almacenar la marca será  $2 \times L_{ma} = 240$  muestras. Se debe evitar la adyacencia de las muestras, para esto se utiliza el número de caracteres de la contraseña para separar las muestras del índice uno, como se indica en la siguiente expresión:

$$I_1[n] = I_1[1] + (m \times n) \quad (14)$$

desde  $n=2$  hasta  $n = L_{ma}$

El arreglo será  $aI[I_1[n]] = [aI[868] aI[872] aI[876] \dots aI[1348]]$ . Se debe crear otro índice para almacenar cada bit, esto se hace para aumentar la complejidad de la detección de la marca de agua. Además, esto dispersa la información a través de todo el dominio espectral. El otro índice necesario se crea en base al índice  $I_1[n]$  del siguiente modo: primero se crea un índice auxiliar  $I_{11}$  aumentando en 1 los valores de  $I_1$ .

$$I_{11}[n] = I_1[1] + (m \times n) + 1 \quad (15)$$

desde  $n=2$  hasta  $n = L_{ma}$

Luego, se crea el índice  $I_2[n]$  invirtiendo todos los valores de  $I_{11}[n]$ , esto es, que el último valor del índice  $I_{11}[n]$  será el primer valor del índice  $I_2[n]$  y así sucesivamente de acuerdo al siguiente modelo:

$$I_2[n] = I_{11}[L_{ma} - (n - 1)] \quad (16)$$

desde  $n=1$  hasta  $n = L_{ma}$

El arreglo del segundo índice será de acuerdo a la ecuación (16) y al ejemplo,  $aI[I_2[n]] = [aI[1349] aI[1345] aI[1341] \dots aI[869]]$ . El algoritmo *Patchwork* para introducir un bit de la información de la marca de agua, modifica la tercera cifra significativa de cada valor de la subseñal  $aI[n]$  en la posición indicada por los arreglos  $I_1[n]$  e  $I_2[n]$  de acuerdo al siguiente criterio. Para un mejor entendimiento se realizará un ejemplo. Supongamos que  $aI(I_1[1]) = aI(868) = 0.35732$  y  $aI(I_2[1]) = aI(1349) = 0.95713$ , entonces  $cf_1 = 7$  y  $cf_2 = 7$ . Luego se toma como base el valor promedio de ambas cifras significativas, es decir:

$$base = (cf_1 + cf_2) / 2 = (7 + 7) / 2 = 7$$

Ya que se tiene definida esta base, entonces se observa si el bit que se quiere introducir tiene valor 0 ó 1.

Si se quisiera introducir un 1 entonces  $cf_1$  debe ser mayor a  $cf_2$  y se hace:

$$cf_1 = base + 1 \quad \text{y} \quad cf_2 = base - 1$$

que para este ejemplo corresponden a los valores  $cf_1 = 8$  y  $cf_2 = 6$ , los cuales sustituyen a los valores originales en un nuevo arreglo  $am[n]$  quedando:

$$am(I_1[1]) = 0.35832 \text{ y } am(I_2[1]) = 0.95613.$$

Para el caso en que el bit a introducir sea 0, el proceso se invierte, o sea que  $cf_1$  debe ser menor a  $cf_2$  y se hace

$$cf_1 = base - 1 \text{ y } cf_2 = base + 1$$

que en este caso corresponde a  $cf_1 = 6$  y  $cf_2 = 8$  de modo que

$$am(I_1[1]) = 0.35632 \text{ y } am(I_2[1]) = 0.95813.$$

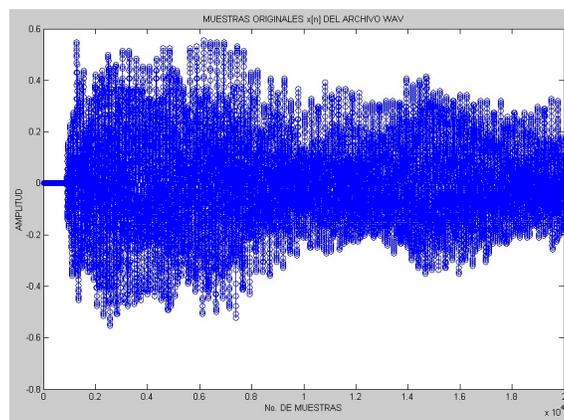
De esta forma se introduce la marca de agua en archivos de audio en el dominio de la frecuencia.

Para el caso de la extracción de dicha marca de agua, se sigue el mismo proceso que para la introducción de la misma, es decir, como primer paso se aplica una TWHF (igual a la usada en la introducción de la marca) al archivo wav con la marca de agua para obtener la subseñal promedio  $aI[n]$ . Luego, se introduce la contraseña pedida al programa para la extracción de la marca. Se busca la secuencia de sincronización para encontrar el comienzo de la información. Se extraen los bits de sincronización, para después autenticar la contraseña comparando la contraseña pedida con la contraseña introducida. Si estas contraseñas son iguales, el programa comienza a extraer la marca de agua por medio del algoritmo Patchwork y usando los mismos parámetros utilizados en la introducción de la misma.

## 4 Pruebas

Para las pruebas realizadas solo se toman las primeras 20,000 muestras del archivo de audio wav. En la figura 5 se muestra la secuencia  $x[n]$  original.

La marca de agua se introduce en la subseñal promedio  $a[n]$  con el procedimiento descrito. La gráfica de  $a[n]$  se muestra en la figura 6.



**Fig. 5.** Secuencia  $x[n]$ , primeras 20,000 muestras

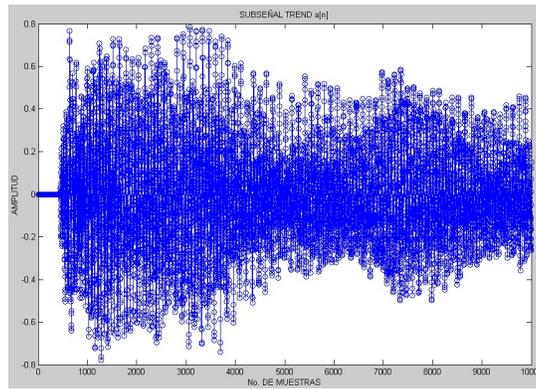


Fig. 6. Gráfica de la subseñal promedio  $a[n]$  de la transformada wavelet Haar

Para la introducción de la marca de agua se utiliza el texto “esime” y como contraseña “ipn”. La secuencia resultante  $am[n]$  se muestra en la figura 7.

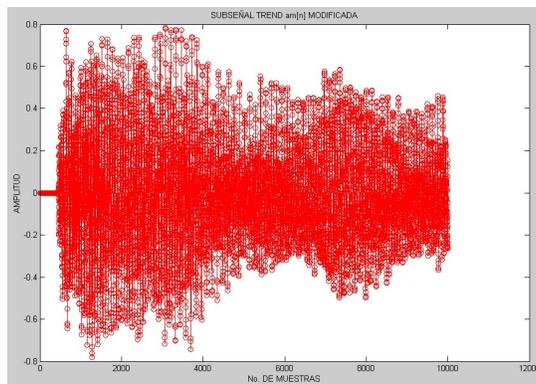
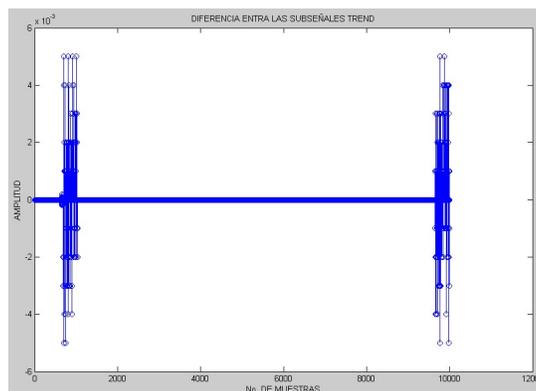


Fig. 7. Gráfica de la subseñal promedio  $am[n]$  después de introducir la marca de agua

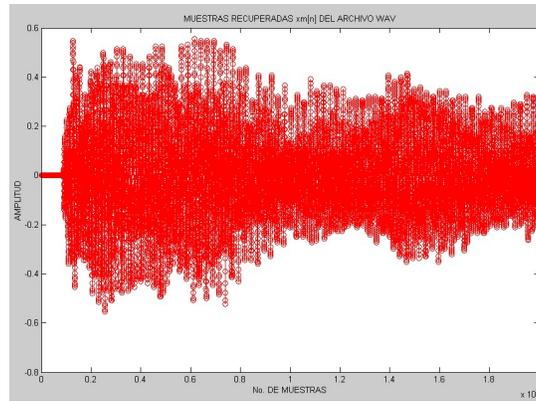
Se puede observar que las figuras 6 y 7 son muy semejantes. Ambas señales tienen el mismo número de muestras y casi tienen la misma amplitud. Para encontrar qué tan diferentes son se realiza la resta entre la subseñal promedio original  $a[n]$  y  $am[n]$ . El resultado se muestra en la figura 8.



**Fig. 8.** Diferencia de las subseñales promedio

De la figura 8 se observa que la mayoría de los valores son iguales a cero. Sin embargo, la amplitud de los valores diferentes de cero es del orden de  $10^{-3}$ . En estas muestras se han introducido la secuencia de sincronización, la contraseña y el texto a cifrar.

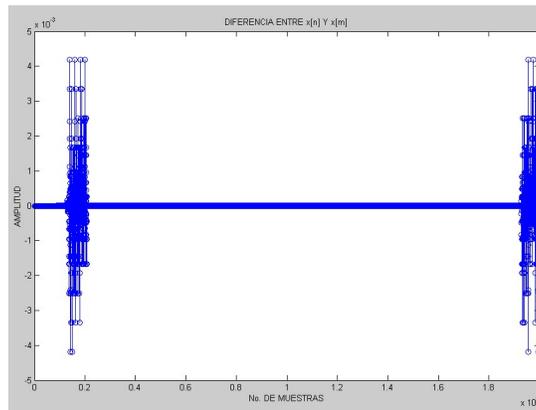
Para encontrar la señal “original”  $x[n]$  se aplica la transformada wavelet inversa utilizando la nueva subseñal promedio  $am[n]$  y dejando la subseñal fluctuación  $d[n]$  sin cambios. La señal recuperada  $xm[n]$  se muestra en la figura 9.



**Fig. 9.** Transformada inversa  $xm[n]$  del archivo wav

Al igual que en el caso de las subseñales promedio, la señal original de la figura 5,  $x[n]$ , tiene el mismo número de muestras de la transformada wavelet inversa  $xm[n]$ . También sus amplitudes son muy similares.

Debe recordarse que el método propuesto en este trabajo no requiere de la señal original para extraer la marca de agua. Sólo como demostración la figura 10, muestra la diferencia entre la señal original  $x[n]$  y  $xm[n]$ .



**Fig. 10.** Resta entre la señal  $x[n]$  y  $xm[n]$ .

Puede observarse que la mayoría de las muestras tiene el valor de cero y sólo aquellas donde se encuentra la marca de agua son diferentes de cero. Debe notarse que la amplitud de los valores diferentes de cero es muy pequeña (del orden de  $10^{-3}$ ).

Al reproducir el archivo de audio wav modificado no se percibe ningún cambio con respecto al archivo original.

## 5. Pruebas de Robustez

En esta sección se presentan los resultados obtenidos cuando se expuso el archivo wav a diversas pruebas como: alteración de las muestras, compresión y remuestreo. Esto se hace aplicando diferentes transformadas wavelet al archivo de audio con la marca de agua.

Primero se aplica la transformada Haar de primer orden a la señal original  $x[n]$  para obtener la señal promedio  $a[n]$ . Después, se aplican otras transformadas wavelet a la señal promedio  $a[n]$ , con esto se logra la compresión, remuestreo y alteración de las muestras. Al tratar de recuperar la marca de agua se obtuvo como resultado que a pesar de la eliminación de varias componentes la marca de agua se conservó, lo que comprueba la efectividad de este método como forma de protección.

Se aplicaron las transformadas wavelet tipo daubechies (daub2, daub4, daub8) y coiflet (coif1), se observaron los siguientes resultados:

- daub2 → se pudo recuperar la marca satisfactoriamente
- daub4 → se pudo recuperar la marca satisfactoriamente
- daub8 → se pudo recuperar la marca satisfactoriamente
- coif1 → se pudo recuperar la marca satisfactoriamente

Otra prueba que se realizó fue hacer cero la subseñal de fluctuación  $d[n]$  y posteriormente reconstruir el archivo sólo con la subseñal promedio  $a[n]$  y observar cómo afectaba a la marca de agua. La marca de agua se pudo recuperar satisfactoriamente.

### A. Prueba de Correlación de Señales

Existen muchas aplicaciones donde es necesario comparar una señal de referencia con una o más señales para determinar la similitud entre ellas y para obtener información adicional basada en la similitud.

Una medida de similitud entre un par de señales,  $x[n]$  y  $y[n]$ , está dada por la secuencia de correlación cruzada  $r_{xy}[l]$  definida por [6]

$$r_{xy}[l] = \sum_{n=-\infty}^{\infty} x[n]y[n-l], \quad l = 0, \pm 1, \pm 2, \dots \quad (17)$$

El parámetro  $l$  llamado *lag* (*retardo*), indica la variación en tiempo entre el par de señales. La secuencia  $y[n]$  es desplazada por  $l$  muestras a la derecha con respecto a la secuencia de referencia  $x[n]$  para valores positivos de  $l$ , y cambiada por  $l$  muestras a la izquierda para valores negativos de  $l$ .

La secuencia de *autocorrelación* de  $x[n]$  está dada por [6]

$$r_{xx}[l] = \sum_{n=-\infty}^{\infty} x[n]x[n-l] \quad (18)$$

para cuando  $y[n] = x[n]$ . Nótese en la ecuación (18) que  $r_{xx}[0] = \sum_{n=-\infty}^{\infty} x^2[n] = \mathcal{E}_x$ , es decir, la energía de la señal  $x[n]$ .

Por conveniencia para la comparación y visualización, se utilizan a menudo las siguientes formas normalizadas de autocorrelación y correlación cruzada:

$$\rho_{xx}[l] = \frac{r_{xx}[l]}{r_{xx}[0]}, \quad (19)$$

$$\rho_{xy}[l] = \frac{r_{xy}[l]}{\sqrt{r_{xx}[0]r_{yy}[0]}} \quad (20)$$

cabe hacer notar que  $\rho_{xx}[l] \leq 1$  y  $\rho_{xy}[l] \leq 1$  independientemente de los valores de  $x[n]$  y  $y[n]$ .

Para el caso particular de este trabajo, se obtuvo la forma normalizada de la correlación cruzada para observar la similitud entre la subseñal promedio del archivo de audio original  $a[n]$  y la subseñal promedio con la marca de agua  $am[n]$ . Asimismo se hizo la comparación del archivo de audio original  $x[n]$  con el archivo de audio  $xm[n]$  el cual contenía la marca de agua. Los datos utilizados en la marca de agua son los mismos que en las pruebas anteriores, es decir, el texto “*esime*” con la contraseña “*ipn*”. El procedimiento es el siguiente:

Primero, se realiza la autocorrelación de la señal original  $a[n]$ , obteniéndose

$$r_{aa}[0] = 411.8040$$

después, se obtiene la autocorrelación de  $am[n]$ :

$$r_{aam}[0] = 411.7950$$

Luego, tomando como señal de referencia a  $a[n]$  y  $am[n]$  como la segunda señal, y obteniendo la correlación cruzada entre ambas señales, se tiene

$$r_{aam}[0] = 411.7995$$

sustituyendo estos valores en la ecuación (20):

$$\rho_{aam}[0] = \frac{r_{aam}[0]}{\sqrt{r_{aa}[0]r_{aam}[0]}} = \frac{411.7995}{\sqrt{(411.8040)(411.7950)}}$$

$$\rho_{aam}[0] = 1$$

De los resultados anteriores se puede observar que la energía de la señal de audio original  $a[n]$  y la energía de la subseñal promedio con la marca de agua  $am[n]$  es prácticamente la misma. De igual manera, al observar el resultado de la correlación cruzada normalizada se comprueba que las dos señales son iguales.

Realizando el mismo procedimiento para  $x[n]$  y  $xm[n]$  se obtiene:

$$\rho_{xxm}[0] = 1$$

Con esto se demuestra que la señal con la marca de agua es prácticamente igual a la señal original de audio.

## 6 Conclusiones

El método descrito en este trabajo demostró ser resistente contra ataques de compresión, remuestreo y alteración de las muestras, por lo que puede considerarse como un método robusto. Además, la introducción de la marca de agua en archivos de audio es imperceptible, lo que resulta muy conveniente para autenticar los derechos de autor y evitar copias ilegales. Al introducir el valor ASCII de los caracteres de la marca de agua en el dominio de la frecuencia, se modifica todo el archivo de audio en el dominio del tiempo, haciendo imposible tratar de eliminar la marca de agua en dicho dominio. Del resultado obtenido de la correlación cruzada normalizada entre la secuencia original y la modificada demuestra que al introducir la marca de agua utilizando el método descrito hace indetectable la información oculta en el archivo de audio.

## Referencias

1. **Hong Oh Kim , Bae Keun Lee and Nam-Yong Lee** “Wavelet-based Audio Watermarking Techniques: Robustness and Fast Synchronization”. Division of Applied Mathematics, KAIST, Noviembre 2001
2. <http://engineering.rowan.edu/~polikar/WAVELETS/WTtutorial.html>. ENERO 12 DEL 2001
3. **J. S. Walker** “A primer on wavelets and their scientific applications”, Chapman & Hall/CRC, USA (1999).
4. **C. I. Podilchuk and E. J. Delp** “Digital Watermarking: Algorithms and Applications”, *IEEE signal processing*, pp. 33-46, July 2001.
5. **A. I. Saichev and W. A. Woyczyński** “Distributions in the physical and engineering sciences. Vol. I: Distributional and fractal calculus, integral transforms and wavelets”, Ed. Birkhäuser, Boston (1997).
6. **Sanjit K. Mitra** “Digital Signal Processing: a computer based approach”, Ed. McGraw-Hill International, second edition 2001, pp. 88-94.



**Marco Antonio Acevedo Mosqueda.** *Nació en México D.F. el 19 de julio de 1968. Obtuvo el título de ingeniero en Comunicaciones y Electrónica en 1992 y en 1996 obtiene el grado de Maestro en Ciencias en Ingeniería Electrónica en la Escuela Superior de Ingeniería en Electrónica del Instituto Politécnico Nacional. Actualmente es el coordinador de la maestría en ingeniería de Telecomunicaciones de la SEPI-ESIME-IPN.*



**José Luis López Bonilla.** *Nació en Cd. Madero, México el 15 de enero de 1956. Obtuvo el título de Licenciado en Física en 1976, en 1979 obtiene el grado de Maestro en Ciencias en Física y en 1982 obtiene el grado de Doctor en Física en la Escuela Superior de Física y Matemáticas del Instituto Politécnico Nacional. Actualmente es profesor investigador de la maestría en ingeniería de Telecomunicaciones de la SEPI-ESIME-IPN.*



**Roberto Linares y Miranda.** *Obtuvo su título de Licenciatura en Ingeniería Eléctrica y Comunicaciones en la ESIME del Instituto Politécnico Nacional; su grado de Maestro y Doctor en Ciencias en Ingeniería Eléctrica en el CINVESTAV del IPN. Actualmente es profesor de teoría de circuitos y teoría electromagnética del Departamento de Electrónica en la Escuela Superior de Ingeniería Eléctrica y Mecánica (ESIME) del IPN. Sus áreas de interés son en el campo de compatibilidad electromagnética. Estudia la interacción de radiación electromagnética con sistemas eléctricos, especialmente radiocomunicación.*