MAURICIO ENRIQUE RUIZ FONT

MONITOREO AUTOMÁTICO DE EMISIONES RADIOFÓNICAS

Vo.Bo. Dr. Rogelio Hasimoto Beltrán

MONITOREO AUTOMÁTICO DE EMISIONES RADIOFÓNICAS

TESIS QUE PARA OBTENER EL GRADO DE

MAESTRÍA EN CIENCIAS CON ESPECIALIDAD EN COMPUTACIÓN Y MATEMÁTICAS INDUSTRIALES

PRESENTA

MAURICIO ENRIQUE RUIZ FONT

TESIS REALIZADA BAJO LA DIRECCIÓN DE:

Dr. Rogelio Hasimoto Beltrán

Dr. Edgar Leonel Chávez González

Centro de Investigaciones en Matemáticas A.C.
Guanajuato, Guanajuato

Agosto 2007

Mauricio Enrique Ruiz Font: *Monitoreo automático de emisiones radiofónicas,* Maestria en Computación, CIMAT A.C., © Agosto 2007

A mi madre

A mis hermanas Maria, Angélica, Alejandra y Adriana y mi hermano Mario

A Jabneelita

RESUMEN

En esta tesis se presentamos el problema de monitoreo de medios y proponemos una solución basada en huellas de audio (*audio fingerprint*). Las huellas de audio son pequeños resúmenes basados en el contenido de la señal de audio, sus características principales son ser pequeñas en comparación con los segmentos de audio que representan, y ser robustas a las degradaciones que el audio presente.

Para su uso en aplicaciones de tiempo real es necesario que el tiempo de extracción de las huellas sea el mínimo posible, por ello las características que son extraídas directamente sobre los datos de la señal son preferidas, en este sentido, a las que requieren alguna transformación adicional. La TES (*Time Entropy Signature*) es una huella de audio basada en la extracción de la entropía instantánea en segmentos de audio de longitud fija.

En el presente trabajo hacemos un afinamiento de los parámetros de dicha huella. Con este afinamiento construimos un sistema basado en huellas de audio que es robusto a diferentes degradaciones y nos permite resolver en un 100 % el problema de monitoreo de medios.

El sistema obtenido se puede implementar fácilmente en dispositivos portátiles debido a su sencillez.

We have seen that computer programming is an art,
because it applies accumulated knowledge to the world,
because it requires skill and ingenuity, and especially
because it produces objects of beauty.

— Donald E. Knuth

AGRADECIMIENTOS

Quiero agradecer al Dr. Edgar Chávez por la invitación que culminó en la elaboración de este trabajo. Al candidato a Doctor Jose Antonio Camarena Ibarrola por sus consejos y guia. Al equipo colaborador del Dr. Chávez de la Universidad Michoacana de San Nicolás de Hidalgo por todas las facilidades otorgadas durante mi visita a Morelia: Profesora Azucena Chávez, Edgardo Morales y Dra. Karina Figueroa Mora.

A Jabneel le agradezco el apoyo que me brindo durante la escritura y revisión de esta tesis.

Al personal de CIMAT por las facilidades otorgadas para la realización de mis estudios.

Agradezco a CONACYT los apoyos otorgados para realizar la maestría y a CONCYTEG por la ayuda para la realización de esta tesis.

ÍNDICE GENERAL

ÍNDICE DE FIGURAS

ÍNDICE DE CUADROS

ACRÓNIMOS

dB *Decibel*.- Unidad de medida relativa a una cantidad.

HMM (*Hidden Markov Model*) Modelo Oculto de Markov

MFCC Mel Frecuency Cepstral Coefficients

MPEG Movie Picture Experts Group

SFM Spectral Flatness Measure

Parte I

INTRODUCCIÓN

1 INTRODUCCIÓN

1.1 PROBLEMAS DE MONITOREO DE MEDIOS

Cuando una compañía contrata los servicios de una televisora o radiodifusora para transmitir sus campañas publicitarias (comerciales), frecuentemente contrata a una tercera compañía de monitoreo de medios, para asegurarse de que sus comerciales sean transmitidos. La compañía de monitoreo elabora una agenda del contenido de las transmisiones de televisión y radio, a fin de verificar que se haya cumplido la transmisión de propaganda, y entrega a sus clientes dichas agendas.

En la Fig. 1 observamos un ejemplo de agenda. Una persona, mientras escucha la transmisión de una radiodifusora hace las anotaciones necesarias en una tarjeta de programación o en un programa de captura. Las columnas indican: la fecha de transmisión, canal, siglas de la estación, nombre de la estación, hora de transmisión del bloque de comerciales, observaciones y posición del comercial dentro del bloque de anuncios, este formato es el que las compañías de monitoreo suelen entregar a sus clientes.

La necesidad de este servicio va mas allá del simple hecho de comprobar si las campañas publicitarias fueron transmitidas o no. Con la información de dichas agendas se elaboran diferentes estudios de mercadeo, que permiten a las compañías comparar sus campañas publicitarias con los competidores, saber que impacto tiene en la ventas el horario de transmisión de los comerciales o analizar la evolución de una campaña en un lapso de tiempo. Aún las grandes televisoras o radiodifusoras contratan servicios de monitoreo para comprobar la calidad de sus servicios. Las casas disqueras también están interesadas en el monitoreo de sus estrenos y en cuestiones relacionadas a los derechos de autor.

Según un estudio realizado por la Universidad de Berkeley[1] la cantidad de información generada por la radio y la televisión por año es enorme. En el Cuadro 1 se mencionan los detalles sobre la cantidad de información producida en el año 2003 en la radio y televisión a nivel mundial, la cantidad de datos justifica por sí misma la necesidad de métodos automáticos de administración.

Actualmente gran parte del monitoreo se hace manualmente, ya sea en papel o con programas de captura, varias personas escuchan todo el día la radio o ven la televisión y van creando agendas como la de la Fig. 1. Este método de resolver el problema es costoso y esta propenso a errores.

Hay compañías que ofrecen soluciones para el problema de monitoreo; de acuerdo al tipo de solución podemos dividirlas en tres tipos: las que ofrecen soluciones a nivel de hardware, las

1 http://www2.sims.berkeley.edu/research/projects/how-much-info-2003/execsum.htm.

REPORTE SCAN
DEL LUNES 19 AL MARTES 20 DE FEBRERO DE 2007
VARIAS MARCAS

FEHA	CANAL	SIGLAS	NOMBRE	HORA	VERSION	OBSERVACIONES	POSICION
19/feb	FM	XHGDA	MAXIMA	07:07	ANDATTI CAFE/OXXO	NORMAL	1/9
19/feb	FM	XHGDA	MAXIMA	07:07	ANDATTI CAFE/OXXO	NORMAL	9/9
19/feb	FM	XHGDA	MAXIMA	07:24	ANDATTI CAFE/OXXO	NORMAL	1/16
19/feb	FM	XHGDA	MAXIMA	07:24	ANDATTI CAFE/OXXO	NORMAL	16/16
19/feb	FM	XHGDA	MAXIMA	07:44	ANDATTI CAFE/OXXO	NORMAL	1/16
19/feb	FM	XHGDA	MAXIMA	07:44	ANDATTI CAFE/OXXO	NORMAL	16/16
19/feb	FM	XHGDA	MAXIMA	08:13	ANDATTI CAFE/OXXO	NORMAL	11/11
19/feb	FM	XHGDA	MAXIMA	08:27	ANDATTI CAFE/OXXO	NORMAL	3/15
19/feb	FM	XHGDA	MAXIMA	08:27	ANDATTI CAFE/OXXO	NORMAL	15/15
19/feb	FM	XHGDA	MAXIMA	08:41	ANDATTI CAFE/OXXO	NORMAL	1/9
19/feb	FM	XHGDA	MAXIMA	08:41	ANDATTI CAFE/OXXO	NORMAL	9/9
19/feb	FM	XHGDA	MAXIMA	08:54	ANDATTI CAFE/OXXO	NORMAL	1/9
19/feb	FM	XHGDA	MAXIMA	09:07	ANDATTI CAFE/OXXO	NORMAL	1/8
19/feb	FM	XHGDA	MAXIMA	09:07	ANDATTI CAFE/OXXO	NORMAL	8/8

Figura 1. Ejemplo de agenda.

Cuadro 1. Producción anual de programación en medios de emisión 2003

Medio	Número de Estaciones	Items únicos	Factor	Mínimo (TBytes)	Máximo (TBytes)
Radio	47,776	7×10^7 horas	0.05 GB/hora	3,488	3,488
Televisión	21,264	3.1×10^7 horas	$1.3 - 2.25$ GB/hora	39,841	68,955

Figura 2. Sistema OBSERVER de Volicon

que ofrecen software que trabaja en equipo de escritorio normal, y las que tienen instalaciones para el monitoreo y rentan tiempo de procesamiento en ellas.

Volicon (www.volicon.com) es una compañía que ofrece hardware para el monitoreo de medios, en la Fig. 2 se muestra su dispositivo OBSERVER, esta es una solución muy completa al problema de monitoreo, genera agendas en tiempo real y de varios canales, puede hacer respaldos de las emisiones recibidas en varios dispositivos de almacenamiento y generar avisos automáticos sobre la emisión de cierto comercial, sin embargo es muy costosa. Además de que requiere software propietario.

AudibleMagic (www.audiblemagic.com) vende software para monitoreo que trabaja en computadoras de escritorio normales, aún esta solución menos compleja es cara. Esta tecnología esta basada en huellas de audio, el método que utilizan esta basado en medidas sobre las siguientes características acústicas: sonoridad(*loudness*), bass (esta es una característica relacionada con la frecuencia más baja), altura(*pitch*), brillantez (*brightness*) y coeficientes de Mel Cepstrum (patente U.S.5, 918, 223).

CyberAlert es otra compañía que vende servicios de monitoreo con instalaciones y servicios de telecomunicaciones que permiten monitorear varios puntos del planeta. CyberAlert presta servicios a través de sus instalaciones. Sus soluciones están basadas en técnicas de huellas de audio al igual que AudibleMagic.

El monitoreo de medios es una de las aplicaciones de huellas de audio (*audio fingerprinting*) que está tomando gran relevancia en la literatura [12] [13] [16][19][20].

1.2 ANTECEDENTES

Las huellas de audio son codificaciones de descriptores que obtenemos basándonos en el contenido de un archivo multimedia, sus principales escenarios de aplicación son los siguientes:

- Gracias a los programas para compartir archivos y al bajo costo de los medios de almacenamiento, el promedio de información en archivos de audio por usuario es del orden de gigabytes, manejar este tipo y cantidad de información es complicado debido a su naturaleza y es deseable contar con esquemas parecidos a los disponibles para la búsqueda en archivos de texto, que nos faciliten su manejo.

  Una posible aplicación de las huellas de audio es diseñar algún método automático para relacionar archivos de audio con meta-datos (autor, álbum, cantante, género). Estos esquemas deben tomar en cuenta la degradación que tiene el audio debido a las diferentes técnicas de compresión con perdida de datos usadas (MP3, OGG, MPEG, etc.).

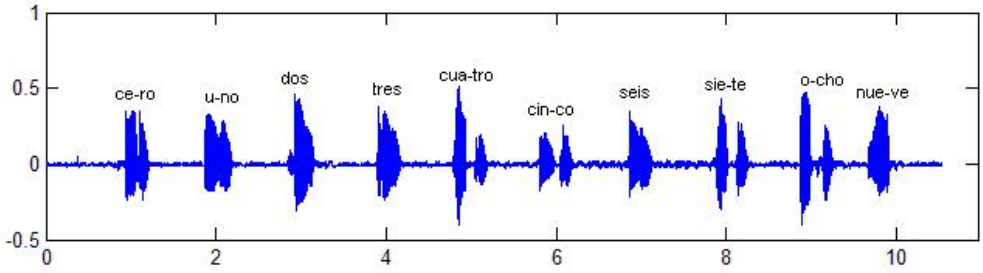
- El escenario clásico para usar huellas de audio es el problema del celular [16], en donde mientras escuchamos la radio, transmiten una canción que nos gusta pero de la cual desconocemos su título o interprete. En tal caso nos gustaría tener un sistema al cual enviarle unos cuantos segundos de grabación usando el celular y que nos devolviera en un mensaje SMS los datos de la canción. Esta aplicación suena algo banal, pero ha sido un problema centra porque es uno de los escenarios más complejos dentro de la aplicación de sistemas de huellas de audio, debido a los problemas que involucra: utilizar pocos segundos de audio para identificar la canción, ecualizaciones extremas, ruido ambiental, compresión por la codificación GSM y regrabación, distorsiones en la linea de transmisión.

- En las redes P2P hay interés por vigilar material de audio que tenga derechos de autor. Un ejemplo de esto es la desaparecida red P2P de la compañía Napster, usando un programa del mismo nombre los usuarios podían compartir archivos de audio. Esto fue hasta Marzo de 2001 cuando una corte emitió un falló para prohibir la descarga de archivos con copyright. Napster instaló un sistema de filtrado basado en el nombre de los archivos, pero los usuarios comenzaron a evadir el sistema poniendo mal los nombres de los mismos. Napster instaló entonces un sistema de detección basado en huellas de audio, este sistema resultó tan efectivo que Napster cerro sus actividades unos meses después, hasta la fecha se desconocen las características de dicho sistema de huellas [16].

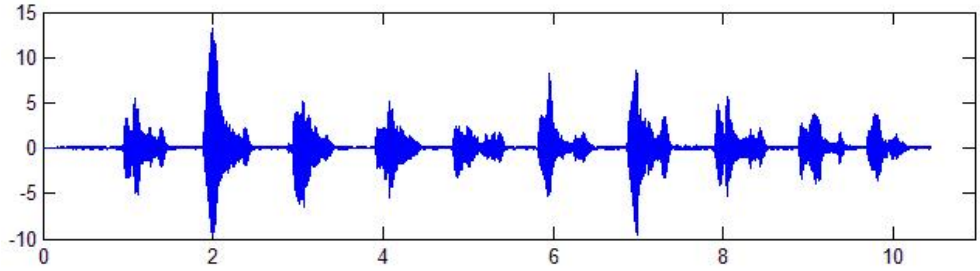
- En un sentido más positivo para los usuario de programas de descarga, en el uso de redes P2P las huellas de audio pueden usarse para garantizar que el contenido de los archivos sea el mencionado en sus metadatos [15].

Para resolver estos problemas se puede pensar en los métodos clásicos de procesamiento de señales. Por ejemplo, para buscar un segmento de audio contenido en otro, podemos usar una correlación cruzada. Esta es una forma muy pesada de hacer esta tarea y frecuentemente poco efectiva, aún cuando el audio esté muestreado a una frecuencia baja. Por ejemplo, tomemos la señal de la Fig. 3, en la parte (a) se muestra la gráfica de la voz de una persona contando del cero al nueve, supongamos que deseamos encontrar el número seis, tomamos las muestras que

pertenecen a la pronunciación de este número y al hacer la correlación cruzada entre ambas señales obtenemos la función presentada en (b), que nos indica que la pronunciación del uno se asemeja más a la pronunciación del seis, que a la pronunciación del seis mismo.



(a) Señal de voz contando



(b) Señal con correlación cruzada con el segmento del 6

Figura 3. Resultado de usar correlación cruzada para identificación de eventos de audio.

Pensemos ahora en el problema de encontrar una canción en una colección de archivos de audio, si tenemos dicha canción podríamos tratar de usar algún esquema de hashing, como las llaves MD5. Con este método por cada archivo se obtiene una combinación de 32 dígitos hexadecimales. Este tipo de claves son usadas para tener certeza sobre la integridad de un archivo, pero son muy frágiles, basta invertir un bit del archivo para modificar la clave completamente.

Las huellas de audio sirven para resolver los problemas antes mencionados y además son robustas, esto debido a que se basan en las características perceptuales de los archivos de audio y no solo en la información de los bytes.

1.3 CIENCIA DEL MONITOREO DE MEDIOS

El monitoreo automático de medios está tomando un auge importante en la literatura científica:

En [19] se propone un sistema de monitoreo de música para resolver el problema del celular. Este sistema esta planteado para funcionar en tiempo real: se obtienen unos segundos de grabación de música de alguna estación, se mandan a un sistema que mantiene un buffer de varias estaciones de radio y utilizando agendas de programación se obtienen los datos de la canción. Este método aunque se presume extensible para monitoreo de comerciales, es poco robusto debido a la extracción de características que utiliza.

En [23] y [15] los autores proponen enfrentar el problema incrustando marcas de agua (*watermarking*) en los comerciales y canciones que se desean monitorear. Esta técnica consiste en insertar información dentro del audio original de manera que sea inaudible, esta será posteriormente extraída y procesada cuando el audio sea monitoreado. Estos métodos deben insertar la información de tal manera que no se pierda en el momento de transmitirse. El problema que tiene este enfoque, es que no es posible trabajar con audio de emisiones sin marcas de agua, y es un tarea común que las empresas quieran hacer estudios de mercado basados en periodos largos de emisiones que probablemente no tengan las marcas de agua.

En [4] proponen un sistema para monitoreo basado en esquemas clásicos de reconocimiento de voz, usan Mel Frecuency Cepstral Coefficients (MFCC) para extraer características y utilizan distancias de Modelos Ocultos de Markov (HMM) para comparar patrones.

La mayoría de sistemas de audio trabaja con características en la frecuencia, lo cual requiere cálculo adicional invertido en transformadas de Fourier, si usaramos alguna característica en el dominio del tiempo obtendríamos un sistema más eficiente.

1.4 OBJETIVOS DE LA TESIS

El rendimiento y la escalabilidad de un sistema de monitoreo de emisiones radiofónicas esta fuertemente asociado al extractor de características que utilice. En base a los requerimientos de tiempo de cómputo que necesite la extracción de características, se determina la cantidad estaciones que puede estar monitoreando un equipo con ciertas características.

En [17] se propone una huella de audio que usa como extractor de características la entropía instantánea, la cual es una característica que se obtiene en el dominio del tiempo. Usando la entropía instantanea queremos hacer un sistema para resolver el problema de monitoreo de medios.

Los objetivos de esta tesis son:

- Hacer una revisión de los sistemas de huellas de audio, revisar en especial la extracción de características de cada sistema y el método de búsqueda planteado. Nos enfocaremos a los sistemas que mencionen ser usados en monitoreo de medios.
- Comprobar los tiempos de TES para hacer una conclusión sobre su escalabilidad.

- Proponer un algoritmo basado en la TES que nos permita hacer monitoreo de medios de manera rápida y efectiva en equipo común de cómputo.
- Usando emisiones radiofónicas grabadas de un radio común comprobar el desempeño del sistema realizado.

1.5 DESCRIPCIÓN DE LOS CAPÍTULOS

Esta tesis esta dividida en 6 capítulos y 3 apéndices.

En el capítulo 2 se presenta una revisión de los conceptos básicos que usaremos.

En el capítulo 3 haremos una introducción a los sistemas de huellas de audio, revisaremos cual es el esquema general de estos, cuales son las características usadas en sistemas que son el estado del arte.

En el capítulo 4 realizaremos las pruebas para determinar el desempeño de la huella creada en [17]. Haremos pruebas de robustez y mostraremos su desempeño sobre el problema de monitoreo usando la base de audio proporcionada.

En el capítulo 5 daremos algunas conclusiones sobre el desempeño del sistema y trabajo futuro.

Los Apéndices estan integrados por 4 partes: en el apéndice A se describe el formato de archivos de audio WAVE que usamos en toda la tesis para hacer las pruebas de rendimiento, en el apéndice B mencionamos como usar el sistema de audio de Java, en el apéndice C hacemos un listado de las herramientas de edición de audio usadas en la tesis, por último en el Apéndice D se encuentra la agenda proporcionada por Contacto Media para hacer pruebas.

Parte II

PRELIMINARES

2 HERRAMIENTAS DE ANÁLISIS DE SEÑALES

El paso más importante en el proceso de creación de huellas de audio es la extracción de características robustas a degradaciones como ruido aditivo, cambios de escala, ecualización, etc. Muchas de las características usadas son obtenidas en el dominio de la frecuencia, por ello haremos un repaso de la transformada de Fourier, veremos algunos conceptos de psicoacústica para comprender como trabaja el sistema auditivo humano y la percepción y terminaremos con la revisión de algunas herramientas del área de Recuperación de Información (*Information Retrieval*).

2.1 TRANSFORMADA DE FOURIER

Se define la *transformada de Fourier* $F(u) \triangleq \mathfrak{F}\{f\}(u)$ de una señal compleja $f(x)$ continua como :

$$\mathfrak{F}\{f\}(u) = \int_{-\infty}^{\infty} f(x) e^{-i2\pi ux} dx \tag{2.1}$$

donde $i = \sqrt{-1}$. Dada la transformación $F(u)$ podemos obtener la función original $f(x)$ usando la *transformada inversa de Fourier* $\mathfrak{F}^{-1}\{F\}$:

$$\mathfrak{F}^{-1}\{F\} = f(x) = \frac{1}{2\pi} \int_{-\infty}^{\infty} F(u) e^{i2\pi ux} du \tag{2.2}$$

Las ecuaciones 2.1 y 2.2 existen siempre que $f(x)$ sea continua y $F(u)$ sea integrable.

La transformada de Fourier de una función real es una función compleja, es decir:

$$F(u) = R(u) + iI(u) \tag{2.3}$$

donde $R(u)$ y $I(u)$ se llaman parte real e imaginaria de $F(u)$ respectivamente,y son funciones reales. Además $F(u)$ es *hermitiana*, $F(-u) = \overline{F(u)}$, donde $\overline{F(u)}$ denota el conjugado de $F(u)$. También podemos escribir $F(u)$ en forma polar:

$$F(u) = |F(u)| e^{i\phi(u)} \tag{2.4}$$

donde :

$$|F(u)| = \sqrt{R^2(u) + I^2(u)} \tag{2.5}$$

$$\phi(u) = \tan^{-1}\left(\frac{I(u)}{R(u)}\right) \tag{2.6}$$

La función módulo $|F(u)|$ recibe el nombre de *espectro de Fourier* de $f(x)$ y $\phi(u)$ es su *ángulo de fase*. El cuadrado del espectro se conoce como la *energía de la señal* ó *espectro de potencia*.

Recordando la igualdad de Euler:

$$e^{-i2\pi ux} = \cos(2\pi ux) - i\,\mathrm{sen}(2\pi ux) \tag{2.7}$$

Tomando en cuenta esta ecuación en 2.1, vemos que la transformada de Fourier nos da una descomposición de f en términos de funciones periódicas seno y coseno, donde $F(u)$ nos da el coeficiente de la componente $e^{i2\pi ux}$. Por esta razón cuando usamos la señal f decimos que estamos trabajando en el dominio del tiempo o del espacio, dependiendo de la naturaleza de la variable independiente, y cuando usamos la representación $F(u)$ decimos que trabajamos en el dominio de la frecuencia.

En la práctica no trabajamos sobre una señal análoga, lo hacemos sobre una señal discretizada que es el resultado de tomar el valor de la señal original a espacios uniformes de tiempo, entonces obtenemos una nueva señal $\hat{f}$ que cumple:

$$\hat{f}(n) = f(x_0 + n\Delta x) \tag{2.8}$$

donde Δx representa la cantidad de tiempo entre la lectura de cada muestra. Podemos ahora manejar los valores de f usando n en vez de x y daremos por hecho que nos referimos a la señal muestreada.

Para analizar nuestra señal discreta usamos la *transformada discreta de Fourier* (TDF), esta se obtiene de la definición 2.1 aplicando el esquema de integración numérica del trapecio, más detalles en el libro [3].

$$F(u) = \sum_{n=0}^{N-1} f(n)e^{-i2\pi un/N} \tag{2.9}$$

y su inversa es:

$$f(n) = \frac{1}{N}\sum_{u=0}^{N-1} F(u)e^{i2\pi un/N} \tag{2.10}$$

Los valores $u = 0, 1, 2, \ldots, N-1$ de la TDF en 2.9, corresponden a las muestras de la transformación continua en los valores $0, \Delta u, 2\Delta u, \ldots, (N-1)\Delta u$. En otras palabras, $F(u)$ representa $F(u\Delta u)$. Esta notación es similar a la usada para la función discreta de $f(x)$, excepto que ahora las muestras de $F(u)$ empiezan en el origen del eje de frecuencias. Los términos Δu y Δx están relacionados por la expresión:

$$\Delta u = \frac{1}{N\Delta x} \tag{2.11}$$

Esta ecuación define lo que se llama *el principio de incertidumbre* : no se puede tener una resolución arbitraria en el dominio del espacio y en el dominio de la frecuencia simultáneamente.

Si bien es posible calcular la TDF usando la fórmula 2.9, este método es un algoritmo de orden $O(n^2)$.

Hagamos la siguiente descomposición a la TDF:

$$F(u) = \sum_{n=0}^{N-1} f(n)e^{-i2\pi un/N} \tag{2.12}$$

$$= \sum_{n=0}^{N/2-1} f(2n)e^{-i2\pi u(2n)/N} + \sum_{n=0}^{N/2-1} f(2n+1)e^{-i2\pi u(2n+1)/N} \tag{2.13}$$

$$= \sum_{n=0}^{N/2-1} f(2n)e^{-i2\pi un/(N/2)} + e^{-i2\pi u/N} \sum_{n=0}^{N/2-1} f(2n+1)e^{-i2\pi un/(N/2)} \tag{2.14}$$

$$= F^p(u) + e^{-i2\pi u/N} F^i(u) \tag{2.15}$$

En esta descomposición observamos como podemos calcular la TDF de una señal de longitud N haciendo dos TDF's en las señales formadas por los elementos pares (F^p) e impares (F^i) de la señal original. Este proceso se repite con cada TDF hasta llegar a TDF's de señales de longitud 1, este método conocido como transformada rápida de Fourier (FFT por sus siglas en ingles), es preferido al anterior por ser de orden $O(n \log n)$, sin embargo tiene la desventaja de que requiere que la longitud de la señal sea potencia de 2. El código completo puede verse en [26].

2.2 TRANSFORMADA DE FOURIER DE SEÑALES LARGAS

Como en nuestro caso, a veces es necesario calcular la transformada de Fourier de cierta longitud a lo largo de una señal cuyo fin no conocemos *a priori*. Normalmente los segmentos en los que será calculada la TDF están sobrelapados y podemos ahorrarnos cálculos usando el siguiente truco que aparece en [9].

Sea f nuestra señal de entrada, tomemos las primeras N muestras para formar el primer segmento, al que aplicaremos la TDF, denotaremos con $F(k)|_0^{N-1}$ la TDF del segmento formado con las muestras $0, \ldots, N-1$. Entonces de acuerdo a la sección anterior:

$$F(k)|_0^{N-1} = \sum_{n=0}^{N-1} f(n)e^{-i2\pi kn/N} \tag{2.16}$$

Ahora, si deseamos conocer la TDF del segmento recorrido una muestra hacia adelante, i.e. de las muestras $1, \ldots, N$:

$$F(k)|_1^N = \sum_{n=1}^{N} f(n)e^{-i2\pi k(n-1)/N} \tag{2.17}$$

$$= e^{i2\pi k/N} \sum_{n=1}^{N} f(n)e^{-i2\pi kn/N} \tag{2.18}$$

$$= e^{i2\pi k/N} \left[\sum_{n=1}^{N} f(n)e^{-i2\pi k(n-1)/N} + f(0)e^{-i2\pi k0/N} - f(0)e^{-i2\pi k0/N} \right] \tag{2.19}$$

$$= e^{i2\pi k/N} \left[\sum_{n=1}^{N} f(n)e^{-i2\pi k(n-1)/N} + f(0) - f(0) \right] \tag{2.20}$$

$$= e^{i2\pi k/N} \left[\sum_{n=0}^{N-1} f(n)e^{-i2\pi k(n-1)/N} + f(N)e^{-i2\pi kN/N} - f(0) \right] \tag{2.21}$$

$$= e^{i2\pi k/N} \left[\sum_{n=0}^{N-1} f(n)e^{-i2\pi k(n-1)/N} + f(N) - f(0) \right] \tag{2.22}$$

$$= e^{i2\pi k/N} \left[F(k)|_0^{N-1} + f(N) - f(0) \right] \tag{2.23}$$

Esto ya nos da un método para calcular la nueva TDF sin tener que recalcular todo. Si deseamos desplazar el segmento más de una muestra podemos usar este mismo truco recorriendo una muestra a la vez, también se puede generalizar las operaciones anteriores, supongamos que ahora deseamos deaplazar el segmento m muestras, supondremos que el desplazamiento es tal que hay un sobrelape entre segmentos, i.e. $m < N$.

$$F(k)|_1^N = \sum_{n=m}^{N+m-1} f(n)e^{-i2\pi k(n-m)/N} \tag{2.24}$$

$$= e^{i2\pi km/N} \left[\sum_{n=m}^{N+m-1} f(n)e^{-i2\pi kn/N} + \sum_{n=0}^{m-1} f(n)e^{-i2\pi kn/N} - \sum_{n=0}^{m-1} f(n)e^{-i2\pi kn/N} \right] \tag{2.25}$$

$$= e^{i2\pi km/N} \left[\sum_{n=0}^{N-1} f(n)e^{-i2\pi kn/N} + \sum_{n=N}^{N+m-1} f(n)e^{-i2\pi kn/N} - \sum_{n=0}^{m-1} f(n)e^{-i2\pi kn/N} \right] \tag{2.26}$$

$$= e^{i2\pi km/N} \left[\sum_{n=0}^{N-1} f(n)e^{-i2\pi kn/N} + \sum_{n=0}^{m-1} (f(N+n) - f(n))e^{-i2\pi kn/N} \right] \tag{2.27}$$

$$= e^{i2\pi km/N} \left[F(k)|_0^{N-1} + \sum_{n=0}^{m-1} (f(N+n) - f(n))e^{-i2\pi kn/N} \right] \tag{2.28}$$

la conversión de 2.26 a 2.27 se hace por la periodicidad que tiene el factor $e^{-i2\pi kn/N}$.

Ya con esto tenemos el cálculo de $F(k)|_m^{N+m-1}$ es términos de $F(k)|_0^{N-1}$. El sumando de la última expresión puede calcularse de manera eficiente usando el esquema de la FFT, notemos que no podemos aplicar exactamente el mismo procedimiento, debido a que el factor de la exponencial no esta divido por m si no por N. Usar este esquema nos permite calcular el vector para actualizar la TDF ya calculada en $O(N \log m)$.

Ninguna de los tips comentados en esta sección puede usarse de manera general, ya que dependiendo de la longitud del segmento y la cantidad de sobrelape usado y la implementación usada la actualización puede tardar más que recalcular toda la TDF usando la FFT. Nótese además que si se requiere usar los operadores de ventana que discutiremos en la siguiente sección, esta técnica no puede usarse.

2.3 OPERADORES DE VENTANA

Cuando deseamos procesar una señal en el dominio de la frecuencia normalmente no aplicamos la TDF sobre la señal completa, en el caso del audio podemos incluso desconocer en que momento terminará la señal. Para tener una apreciación correcta de las frecuencias que forman la señal a intervalos cortos de tiempo se usa la transformada de Fourier de tiempo corto. Lo que hacemos es segmentar la señal utilizando operadores de ventana antes de aplicar la TDF[27].

Un operador de ventana es una señal que fuera de cierto intervalo es cero. Segmentamos la señal multiplicándola por el operador ventana.

El operador ventana más sencillo es la ventana rectangular, esta equivale a tomar un segmento de la señal de longitud fija, el problema es que al aplicar la TDF obtenemos valores en frecuencias que no están contenidas en la señal original, fenómeno conocido como *leakage*, esto debido a los posibles saltos en los valores al principio y final del segmento, en la Fig. 4 podemos ver este efecto.

Existen otros operadores de ventana, cada uno atenúa de diferente modo el efecto de *leakage*. En el Cuadro 2 vemos las descripciones de algunos de ellos, donde M es la longitud del segmento que extraeremos de la señal original, y en la Fig. 5 están sus gráficas.

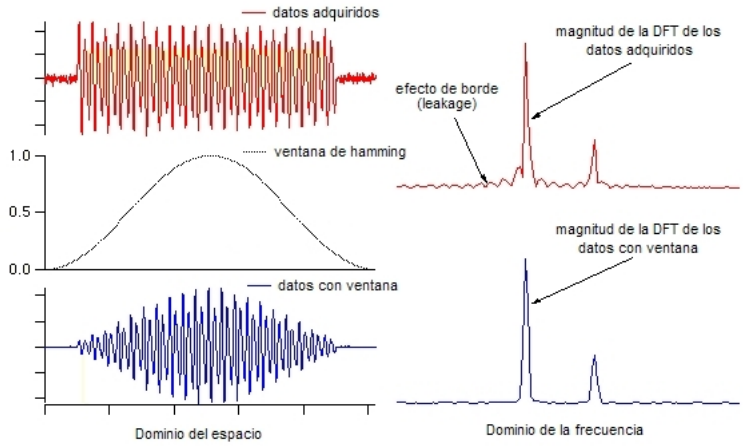


Figura 4. Efecto de *leakage* y atenuación usando operadores de ventana.

Nombre de la Ventana	Fórmula, $0 \leqslant n \leqslant M-1$
Bartlett(Triangular)	$1 - \dfrac{2\left\lvert n - \dfrac{M-1}{2}\right\rvert}{M-1}$
Blackman	$0.42 - 0.5\cos\left(\dfrac{2\pi n}{M-1}\right) + 0.08\cos\left(\dfrac{4\pi n}{M-1}\right)$
Hamming	$0.54 - 0.46\cos\left(\dfrac{2\pi n}{M-1}\right)$
Hann	$0.5 - 0.5\cos\left(\dfrac{2\pi n}{M-1}\right)$

Cuadro 2. Operadores ventana

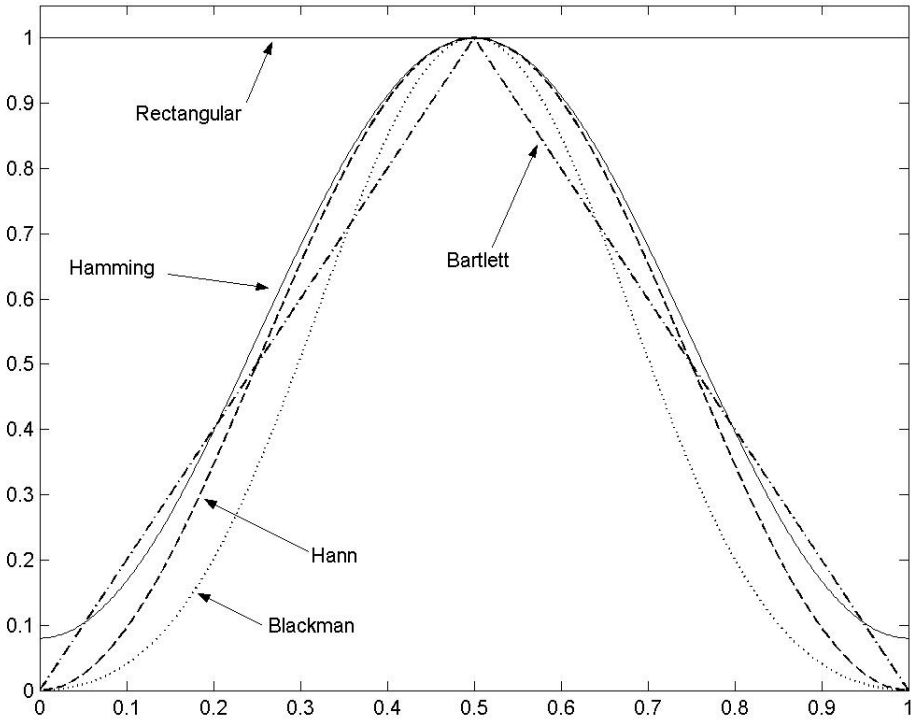


Figura 5. Operadores de ventana.

2.4 RELACIÓN SEÑAL RUIDO (*signal to noise ratio*)

La relación señal ruido (SNR por sus iniciales en inglés), es una forma de medir la contaminación que tiene una señal. El SNR se obtiene al medir la razón entre la energía de la señal ($P_{señal}$) y la energía del ruido contenido en la señal (P_{ruido}):

$$SNR = \frac{P_{señal}}{P_{ruido}} = \left(\frac{A_{señal}}{A_{ruido}}\right)^2 \tag{2.29}$$

Dada una señal f y una señal contaminada $\hat{f}$, el ruido es la señal $\hat{f} - f$.

El poder de una señal se obtiene sumando los cuadrados de sus términos. $A_{señal}$ representa la raíz del error promedio (RMS por sus iniciales en ingles, *Root Mean Square Error*) y se define como:

$$A_{señal} = \sqrt{\frac{1}{N}\sum_{u=0}^{N} f^2(u)} \tag{2.30}$$

Es mas cómodo medir el SNR en *decibeles* (dB)[1], que es una unidad de medida relativa. Para hacer comparaciones en decibeles se toma una señal base W_0, que será nuestra señal de referencia, y calculamos la cantidad de L decibeles de una señal W_1 de la siguiente forma:

$$L(dB) = 10 \log_{10} \frac{W_1}{W_0} \tag{2.31}$$

Una medida de 3 decibeles significa que W_1 es el doble de señal W_0,una medida de 10 decibeles quiere decir que la señal W_1 es 10 veces la señal W_0, si W_1 fuera $10,000$ veces más grande que W_0 entonces $L = 40dB$, por ello es muy cómodo usar decibeles, ya que las grandes diferencias quedan en números más manejables.

El SNR medido en decibeles está dado por:

$$SNR(dB) = 10 \log_{10}\left(\frac{P_{señal}}{P_{ruido}}\right) = 20 \log_{10}\left(\frac{A_{señal}}{A_{ruido}}\right) \tag{2.32}$$

Entre más pequeño sea el SNR más información será aportada por el ruido de la señal que por la señal misma, un SNR=0 indica que el ruido esta aportando la misma cantidad de información que la señal.

Si suponemos alguna distribución para el ruido podemos calcular el RMS directamente en término de sus parámetros, por ejemplo, supongamos que el ruido tiene una distribución Gaussiana $N(\mu, \sigma^2)$, entonces el RMS es $\sqrt{\mu^2 + \sigma^2}$.

1 Un decibel es la décima parte de un bel (B). Desarrollada por los ingenieros de la Bell Telephone Laboratory para medir la reducción de el audio en 1 milla (1.6 km) de cable para teléfono.

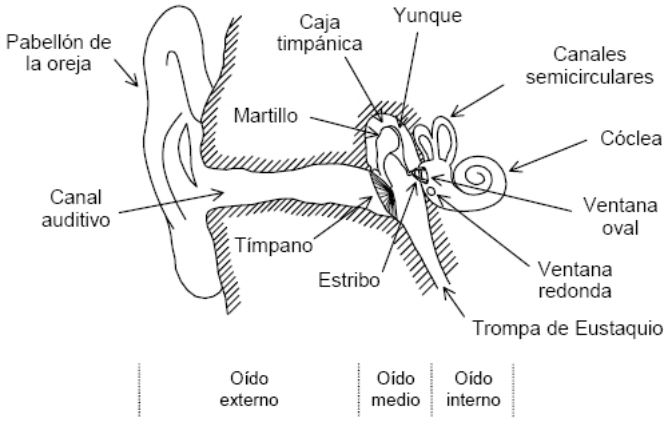


Figura 6. Esquema del oido humano

2.5 SISTEMA AUDITIVO Y PSICOACÚSTICA

En esta sección queremos presentar la manera en que trabaja el sistema auditivo humano. Comenzaremos por describir brevemente su anatomía, daremos algunas definiciones de psicoacústica y luego hablaremos de la escala de Bark que es el objetivo principal de esta sección.

2.5.1 *Anatomía del sistema auditivo humano*

El sistema auditivo humano se compone de 3 partes :

- El oído externo formado por la oreja y el canal auditivo.
- El oído medio formado por el tímpano, los huesecillos u oscículos (martillo, yunque y estribo).
- El oído interno formado por los canales semicirculares, el vestíbulo y el caracol.

En la Fig. 6 se muestra un esquema de esta división.

La función del oído externo es recolectar las ondas sonoras del exterior y llevarlas al tímpano, que es el comienzo del oído medio. El oído medio esta ubicado en la caja timpánica. Cuando una onda sonora hace vibrar al tímpano, este hace vibrar al martillo, que basándose en una estructura de palanca, amplifica la fuerza que recibe, misma que transmite al yunque y este a su vez a la ventana oval, que es parte del caracol.

El caracol contiene el órgano principal de la audición: la *cóclea*, que es un tubo enrollado en espiral de dos vueltas y media, la Fig. 7 muestra un corte transversal de este tubo. La cóclea

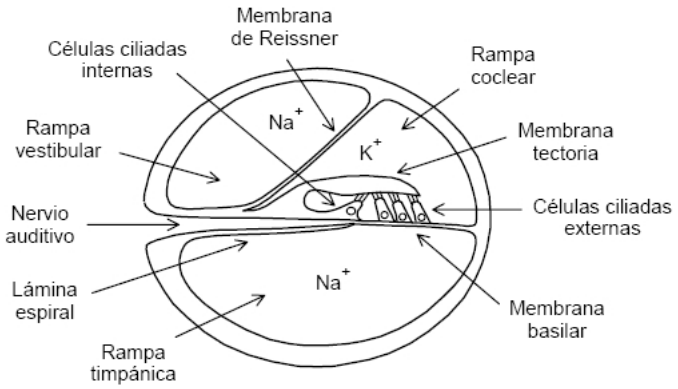


Figura 7. Corte transversal de la cóclea

esta formada por tres secciones: rampa timpánica, rampa vestibular y rampa coclear, las dos primeras se conectan a través de un pequeño orificio, el helicotrema, ubicado hacia el vértice (ápex) del caracol. La rampa vestibular se comunica con el oído medio a través de la ventana oval, y la rampa timpánica lo hace a través de la ventana redonda. La rampa coclear contiene la membrana basilar, una membrana elástica donde se encuentra el órgano de Corti que es una estructura que contiene las células ciliadas o pilosas. Las células ciliadas se comportan como pequeños micrófonos, generando pulsos eléctricos, estos pulsos son enviados al cerebro a través de una serie de células nerviosas (neuronas) reunidas en el nervio auditivo.

La membrana basilar mide alrededor de 35 mm de longitud y tiene unos 0.04 mm de ancho en su zona basal (la más próxima a la base del caracol) y unos 0.5 mm en la zona apical (próxima al vértice o ápex), además la zona más angosta es también la más rígida, lo cual será importante para la capacidad discriminatoria de frecuencias del oído interno. En la Fig. 8 se muestran dos vistas de la membrana con la cóclea hipotéticamente estirada desde su forma helicoidal hasta una forma rectilínea.

Cuando llega una perturbación a la ventana oval, el líquido de la sección superior se encuentra inicialmente a mayor presión que el de la sección inferior, lo cual provoca una deformación de la membrana basilar, que se propaga en forma de onda (denominada onda viajera) desde la región basal hasta la región apical, tendiendo a aumentar la amplitud conforme la rigidez de la membrana va disminuyendo.

Si la perturbación es periódica, tal como sucede con una vibración sonora, la membrana comienza a vibrar con una envolvente, Fig. 9, cuyo máximo se produce en cierta posición que depende de la frecuencia del sonido, como se muestra en la Fig. 10. Resulta así, que existe una localización del pico de resonancia de la membrana basilar en función de la frecuencia, que se

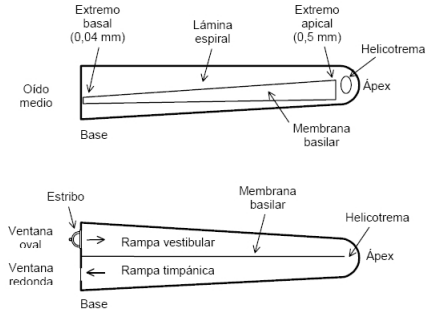


Figura 8. Dos vistas de la cóclea hipotéticamente "desenrollada". Arriba, vista superior. Abajo, vista lateral.

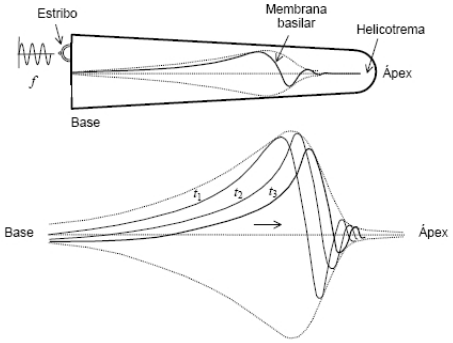


Figura 9. Arriba, onda viajera en la membrana basilar en un instante dado. Abajo, posición de la onda en tres instantes de tiempo $t1$, $t2$ y $t3$. Las líneas indican el lugar geométrico de los picos de la onda conforme ésta va avanzando a lo largo de la membrana.

ha representado gráficamente en la Fig. 11. Esto confiere al oído interno una cualidad analítica que es de fundamental importancia en la discriminación tonal del sonido, especialmente para los sonidos de frecuencias superiores a los 1000 Hz. El descubrimiento de la mecánica de la membrana basilar se debe a Georg Békésy.

Como ya se mencionó, el movimiento de la membrana basilar ocasiona que las células ciliadas emitan un pulso eléctrico. Debido a que las membranas basilar y tectoria tienen ejes diferentes, el movimiento relativo provoca un pandeo de los cilios que fuerza la apertura de unas diminutas compuertas iónicas. El intercambio iónico genera una diferencia de potencial electroquímico que se manifiesta como un pulso de unos 90 mV de amplitud o potencial de acción.

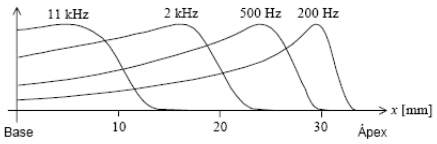


Figura 10. Envolvente espacial de las ondas viajeras sobre la membrana basilar para cuatro frecuencias diferentes.

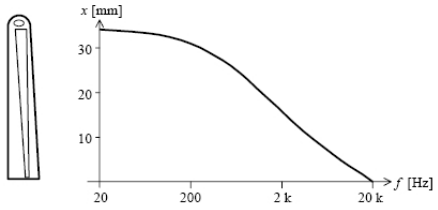


Figura 11. Ubicación de la resonancia a lo largo de la membrana basilar en función de la frecuencia

El potencial de acción generado por cada célula ciliada debe ser comunicado al cerebro. Ello se realiza a través de las neuronas. Donde se procesa y se da sentido a cada sonido.

2.5.2 *Psicoacústica y escala de Bark*

La *psicoacústica* es una rama de la psicología que se encarga de estudiar la percepción subjetiva de las cualidades (características) del sonido. Como vimos en la sección anterior, el sistema auditivo es muy complejo, su funcionamiento no sigue el principio de superposición (no es lineal). Las respuestas a los estímulos son igualmente complejas.

Las características psicoacústicas básicas del sonido son:

- Sonoridad(*loudness*).- Percepción subjetiva de la intensidad (amplitud) sonora. Es decir, la sonoridad es el atributo que nos permite ordenar sonidos en una escala del más fuerte al más débil.
- Altura(*pitch*).- Está ligada a la percepción del tono, en concreto, con la frecuencia fundamental de la señal sonora. Como se percibe lo grave o agudo que es un sonido.
- Timbre(*timbre*).-Es la capacidad que nos permite diferenciar los sonidos. El timbre está caracterizado por la forma de la onda, es decir, por su componente armónico.

Como vimos en la sección 2.5.1, dependiendo de la frecuencia con la que es excitada la membrana basilar se determina un punto de máxima altura, pero normalmente un sonido no viene en una sola frecuencia, sino que esta formado por varias frecuencias cercanas. Existe un

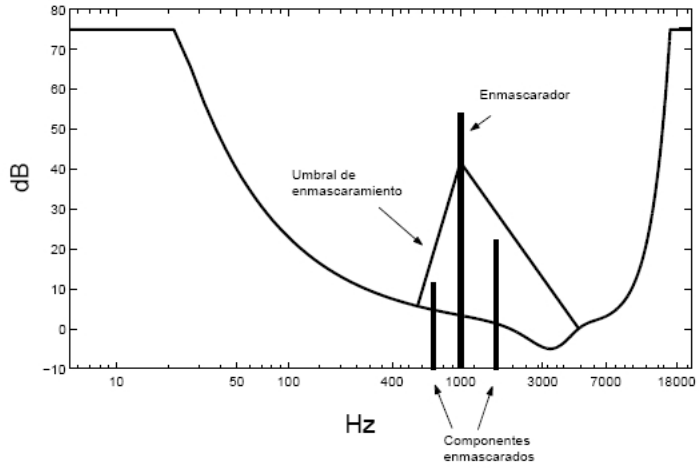


Figura 12. Enmascaramiento de frecuencias

fenómeno llamado enmascaramiento, se produce cuando una frecuencia crítica hace imperceptible a frecuencias vecinas debido a la estimulación que hace sobre el oído. En la Fig. 12 se muestra este efecto y el nivel mínimo de percepción de acuerdo a la frecuencia.

Debido a lo anterior, existen bandas de frecuencias críticas que tienen un lugar determinado en la cóclea. Estas bandas determinan la escala de Barks , obtenida experimentalmente por Eberhard Zwicker en 1961, y que debe su nombre a Heinrich Barkhausen quien propuso la primera medida de la altura(*loudness*). En la Fig. 13 se muestra la correspondencia de lugar de la cóclea con su frecuencia crítica.

La escala de Barks representa estos anchos de banda. Hay varias fórmulas para calcular los Barks de acuerdo a la frecuencia, la más usual es:

$$\text{Bark}(f) = 13\arctan(0.00076f) + 3.5\arctan((f/7500)^2) \tag{2.33}$$

aquí f esta en hertz. La fórmula anterior es solo una aproximación, la tabla con la distribución de los anchos de banda correspondientes esta en la Fig. 14.

Para finalizar hay que mencionar que toda medida relacionada con la psicoacústica no es 100 % válida para todos los individuos, ya que estas se obtuvieron de manera empírica y varían levemente de persona a persona y también cambian con la edad.

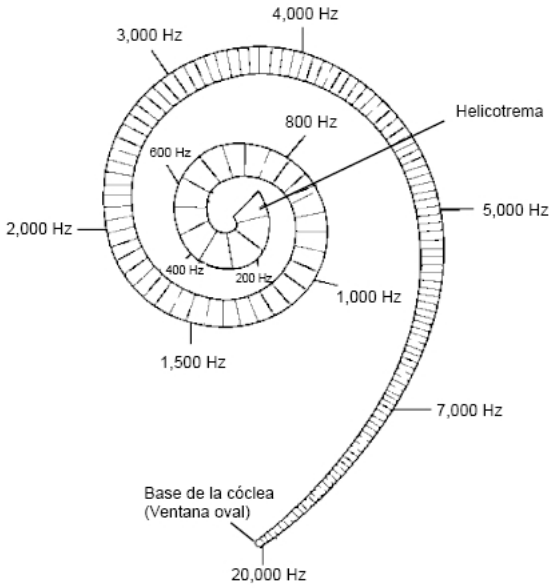


Figura 13. Posición en la cóclea de las frecuencias críticas

Banda	f_i [Hz]	f_o [Hz]	f_s [Hz]	Δf_{BC} [Hz]
1	20	50	100	80
2	100	150	200	100
3	200	250	300	100
4	300	350	400	100
5	400	450	510	110
6	510	570	630	120
7	630	700	770	140
8	770	840	920	150
9	920	1000	1080	160
10	1080	1170	1270	190
11	1270	1370	1480	210
12	1480	1600	1720	240
13	1720	1850	2000	280
14	2000	2150	2320	320
15	2320	2500	2700	380
16	2700	2900	3150	450
17	3150	3400	3700	550
18	3700	4000	4400	700
19	4400	4800	5300	900
20	5300	5800	6400	1100
21	6400	7000	7700	1300
22	7700	8500	9500	1800
23	9500	10500	12000	2500
24	12000	13500	15500	3500

Figura 14. Distribucion de frecuencias en la escala de Bark, frecuencia inferior (f_i), central (f_o) y superior (f_s) y ancho de banda (Δf)

2.6 HERRAMIENTAS DE RECUPERACIÓN DE INFORMACIÓN.

En esta sección hablaremos sobre algunas herramientas del área de recuperación de información (*Information Retrieval*) que utilizaremos para comprobar el rendimiento de los programas de clasificación que implementemos.

2.6.1 *Curvas ROC (*Receiver operating characteristic*)*

Las curvas ROC son herramientas gráficas que nos permiten comparar diferentes métodos de clasificación y afinar parámetros de configuración.

Veamos algunas definiciones. Supongamos que tenemos un clasificador que a cada entrada le asigna una etiqueta del conjunto $\{p, n\}$, que representan positivo y negativo respectivamente, este tipo de clasificadores binarios abundan, por ejemplo en el diagnóstico médico. Para cada entrada se pueden obtener uno de cuatro posibles resultados:

- si la entrada es positiva
  - y el clasificador asigna positivo entonces se le llama *verdadero positivo*(VP)
  - si asigna negativo se le llama *falso negativo* (FN)
- si la entrada es negativa
  - y asigna positivo se le llama *falso positivo*(FP)
  - si asigna negativo *verdadero negativo*(VN)

Dado un conjunto finito de entradas para el clasificador, denotemos por P la cantidad de positivos, N la de los negativos y por VP, FP, VN, FN las cantidades correspondientes a los conjuntos descritos anteriormente después de que el clasificador procesó todas las entradas.

- Porcentaje de verdaderos positivos (*true positive rate o recall*).- $TPR = \frac{VP}{P}$.
- Porcentaje de falsos positivos (*false positive rate*).- $FPR = \frac{FP}{N}$.
- Precisión (*precision*).- $Pr = \frac{VP}{VP + FP}$.
- Exactitud (*accuracy*) .- $E=\frac{VP + VN}{P + N}$
- Medida-F (*F-measure*).- $\frac{2}{\frac{1}{Pr} + \frac{1}{TPR}}$

Una gráfica ROC, es una gráfica bidimensional en la que cada punto representa una configuración para el clasificador, la coordenada esta formada en el eje X por el FPR y en el eje Y por el TPR. El objetivo de esta gráfica es analizar el intercambio entre los elementos correctamente clasificados contra los incorrectamente clasificados. En la Fig. 15 observamos una gráfica ROC

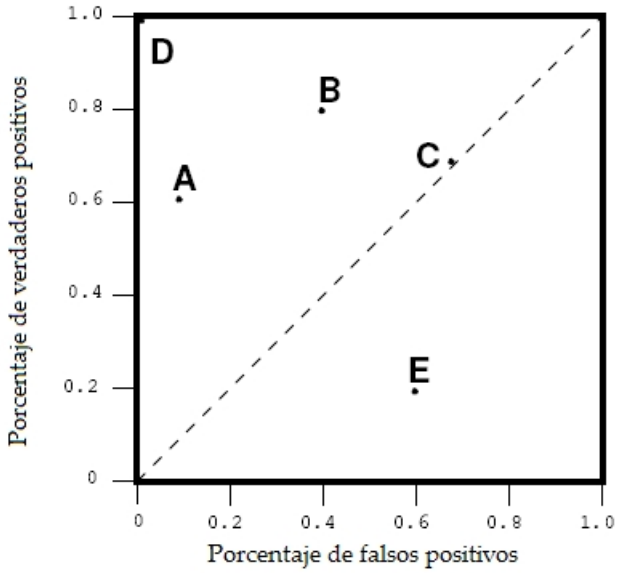


Figura 15. Gráfica ROC con 5 clasificadores

que representa el resultado de 5 clasificadores.

Existen varios puntos en el espacio de la curva ROC que es importante hacer notar. El punto $(0, 0)$ corresponde al clasificador que etiqueta todas las entradas como negativas, por esta razón no comete falsos positivos, ni verdaderos positivos, la estrategia opuesta esta representada por el punto $(1, 1)$, todas las entradas son etiquetadas como positivas, por lo tanto todos los positivos son correctamente etiquetados y todos los negativos erróneamente etiquetados. El punto que representa el clasificador perfecto es el $(0, 1)$. En el caso de la Fig. 15 el punto D representa un clasificador perfecto.

La recta identidad representa los clasificadores que escogen una fracción del tiempo la misma etiqueta, por ejemplo el punto $(0.25, 0.25)$ representa al clasificador que escoge el 25 % de las veces la etiqueta positiva, se espera que le atine al 25 % de las entradas verdaderamente positivas y por lo tanto, también al 25 % de los negativos les será dado una etiqueta positiva, lo cual hace un FPR de 0.25. En la Fig. 15 el punto C representa un clasificador que etiqueta con positivo el 70 % del tiempo.

El triángulo debajo de la recta identidad marca clasificadores que se comportan peor que los clasificadores aleatorios. Esta parte suele estar vacía, ya que si un clasificador cae en esta zona al negar sus etiquetas obtendremos un clasificador simétrico con respecto a la recta identidad.

Conforme un clasificador esté más a la izquierda en el triángulo superior se dice que es más conservador, prefiere no tener demasiado alto el FPR, en cambio si esta más a la derecha se

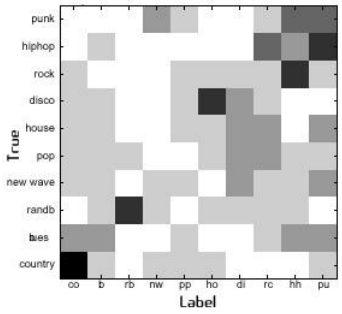


Figura 16. Ejemplo de matriz de confusión

dice que es más liberal,no importa tener un alto FPR si es que tenemos alto TPR.

Cuando tenemos un clasificador cuya salida es un valor real podemos definir un umbral para obtener un clasificador binario. Usamos las curvas ROC para conocer que umbral nos conviene más, de acuerdo al tipo de TPR que busquemos y el FPR que estemos dispuestos a pagar.

2.6.2 *Matriz de confusión*

Cuando tenemos patrones y deseamos medir la calidad de la métrica con la que son comparados y/o el método con que fueron obtenidos utilizamos una matriz de confusión. En una matriz se colocan las distancias que son el resultado de comparar todos los patrones dentro de un grupo contra sí mismo, el orden en que están los patrones en las filas deberá ser el mismo que sobre las columnas, así lo que se espera ver, si es que la métrica es útil, es que las distancias de la diagonal principal sean las mas bajas.

La Fig. 16 es un ejemplo de una matriz de confusión, que muestra que la métrica escogida no es muy buena. La Fig. 17 es, por el contrario, indicador de una buena métrica.

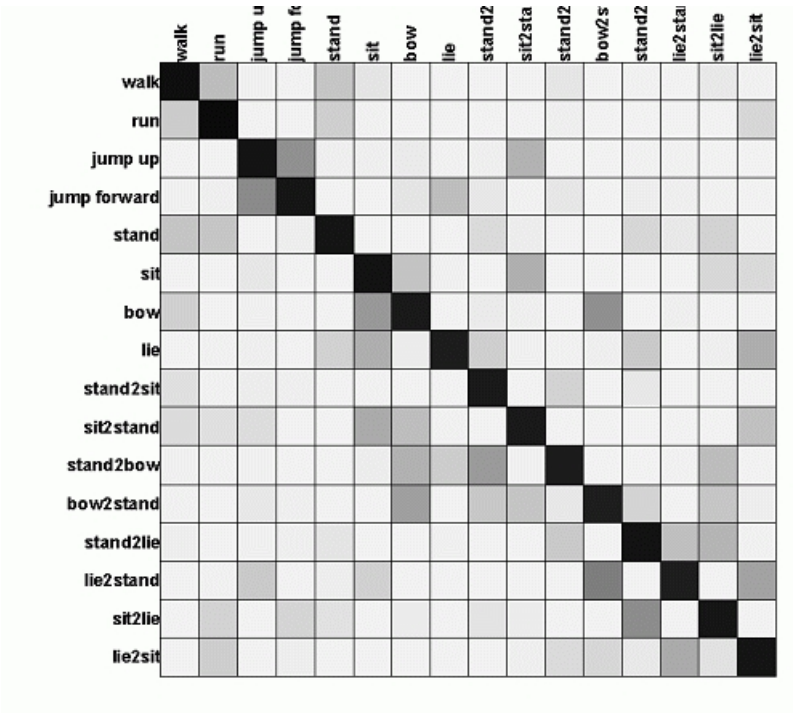


Figura 17. Ejemplo de matriz de confusión

3 SISTEMAS DE HUELLAS DE AUDIO

En esta sección presentamos los sistemas de huellas de audio, hacemos además una actualización a [25] con algunas citas recientes.

3.1 INTRODUCCIÓN A LOS SISTEMAS DE HUELLAS DE AUDIO

Las huellas de audio[1] representan una solución factible y efectiva a los problemas de identificación de audio presentados en el Capítulo 1.

La finalidad de un Sistema de Huellas de Audio (SHA) es proporcionar una firma compacta y única a cada archivo de audio, esta firma se conoce como huella de audio y se calcula basándose en las características perceptuales del audio contenido en el archivo. Generalmente un SHA consta de dos partes : una responsable de la extracción de las huellas de audio y otra que realiza la búsqueda.

Durante una etapa de entrenamiento, el SHA crea una huella de audio para cada archivo o segmento de audio pertenecientes a una colección, y las almacena en una base de datos. Posteriormente durante la búsqueda, dado un archivo perteneciente a la colección pero desconocido, el SHA debe ser capaz de identificarlo calculando su huella de audio y comparandola con las contenidas en la bases de datos, este proceso debe ser posible incluso si el archivo dado presenta distorsiones o esta fragmentado.

Los SHA presentan importantes ventajas sobre los sistemas que hacen comparaciones usando las formas de onda directamente (audio crudo) debido a sus bajos requerimientos de memoria y almacenamiento. Dado que las irrelevancias perceptuales han sido removidas de las huellas, los SHA son más robustos.

3.1.1 *Requisitos de un sistema de huellas de audio*

Existen requisitos básicos tanto para los SHA's como para las huellas en sí.

Un SHA debe cumplir con:

- Robustez(Robustness).- Para que un sistema sea considerado robusto debe ser capaz de identificar correctamente un archivo, sin importar que estén presentes degradaciones tales como:
  - Compresión (MP3, GSM, OGG, etc.).

1 Las huellas de audio también se conocen en la literatura como: identificadores basados en el contenido, búsqueda robusta (*robust matching*), hashing perceptual, marcas de agua pasiva, reconocimiento automático de música y firmas digitales basadas en el contenido [25].

  - Distorsión debido al canal de transmisión.
  - Cambios de frecuencia (pitch shifting).
  - Ecualización.
  - Ruidos ambientales.
  - Conversiones A/D - D/A.
  - Compresiones/Expansiones en el tiempo.

  Para que un sistema logre un alto grado de robustez, la huella de audio debe basarse en características fuertemente invariantes con respecto a la degradación de la señal. El porcentaje de Falsos Negativos (huellas de audio significativamente distintas que corresponden a archivos de audio perceptualmente similares) se utiliza como medida de robustez.

- Confiabilidad(Reliability) La confiabilidad de un SHA es inversamente proporcional al porcentaje de falsos positivos (la taza con la que se identifica un archivo incorrectamente). Así entonces, un SHA confiable debe cometer muy pocos de estos errores y es preferible que marque un archivo como indeterminado cuando su valor de confianza de identificación es muy bajo o bien se encuentra debajo de un umbral [24].

- Granularidad (Granularity).- Se refiere a la capacidad de un SHA de identificar archivos correctamente utilizando como datos de entrada únicamente fragmentos de unos cuantos segundos de duración.

- Eficiencia (Efficiency).- Los principales factores que determinan la eficiencia computacional de un SHA son: el tamaño de la huella, la complejidad del algoritmo utilizado para generarla y la velocidad del algoritmo de búsqueda.

- Escalabilidad (Scalability).- Los algoritmos utilizados en las distintas fases de un SHA deben ser capaces de mantener los parámetros de robustez, confiabilidad y eficiencia conforme se agreguen elementos a la base de datos.

Existe una interdependencia entre los requisitos mencionados anteriormente, esto es, cuando mejoramos un parámetro esto implica un decremento en los otros [25, 16].

Por otro lado, una huella de audio debe cumplir:

- *Poder de discriminación entre un gran número de huellas*.- La característica acústica elegida para obtener la huella de audio, debe ser lo suficientemente representativa para obtener una buena discriminación entre un gran número de huellas. Esto puede interferir con la eficiencia computacional.

- *Invarianza bajo distorsión*.- Se deriva del requisito de robustez del sistema, esta condición puede ser relajada para preservar deliberadamente distorsiones que permitan reconocer manipulaciones no deseadas del archivo.

- *Compacidad*.- Dado que un número importante de huellas (tal vez millones) deberán ser almacenadas y comparadas, será deseable contar con un tamaño de huella pequeño. Sin embargo una huella demasiado chica puede no ser útil para las búsquedas, lo que afecta la robustez y confiabilidad del sistema.

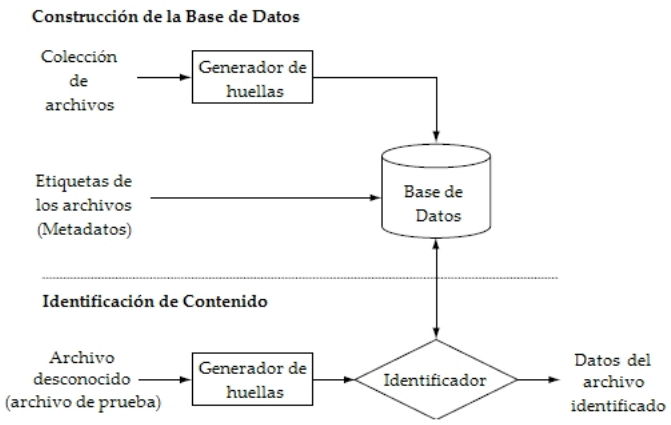


Figura 18. Esquema de identificación basada en contenido.

- *Complejidad computacional simple*.- El algoritmo de extracción de huella deberá ser lo suficientemente eficiente de acuerdo a la tarea en que quiera aplicarse.

3.1.2 *Estructura general de un Sistema de Huellas de Audio*

Cuando utilizamos los SHAs como sistemas de identificación de audio, surge una arquitectura común, independiente del método que se utilice para la extracción de las huellas, del indexamiento de las bases de datos o del algoritmo implementado para la búsqueda. Esta arquitectura puede verse en la Fig. 18 y se divide en dos etapas:

- *Construcción de la base de datos*.- Dada una colección de archivos de audio, el generador de huellas procesa cada archivo para calcular una huella. Dicha huella esta basada en las características perceptuales del audio contenido en el archivo y por lo tanto es única. Cada huella generada es entonces almacenada en una base de datos donde se le asocia una etiqueta o algún otro metadato de interés.
- *Identificación de contenido*.- Dados datos de entrada obtenidos en linea o mediante un archivo, el generador de huellas extrae su correspondiente huella de audio, misma que será utilizada para realizar una búsqueda en la base de datos del sistema, si se encuentra una coincidencia se presenta la información asociada correspondiente, también es posible proporcionar un nivel de confiabilidad de la identificación.

3.1.3 *Extracción de huellas*

El generador de huellas de audio obtiene a partir de un archivo una huella basada en las características perceptuales relevantes. A continuación se describen los dos bloques en que se divide un generador de huellas, tal como se muestra en la Fig. 19.

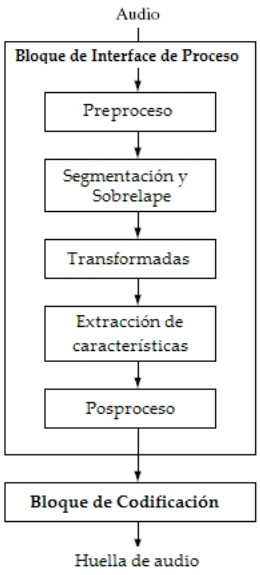


Figura 19. Generador de huellas: interface de proceso y modelado de huellas.

Bloque de Interface de proceso

La finalidad de la interface de proceso es convertir la señal de audio en una secuencia de características relevantes que se pasarán al bloque de codificación. El diseño de este bloque debe satisfacer los siguientes requisitos: reducción de dimensionalidad, selección y extracción de parámetros significativos, y búsqueda de invarianza y robustez a posibles degradaciones. Como se aprecia en la Fig. 19 la interface de proceso esta formada por cinco subbloques.

- Preproceso.- La mayoría de las interfaces de proceso inician esta etapa digitalizando la señal de audio (en caso de ser necesario) y convirtiéndola en un formato general (ej. 16 bit PCM, $5 - 44.1$ khz, mono). Dependiendo de la aplicación final la señal también puede someterse a otro tipo de procesos tales como: codificación/decodificación GSM para telefonía celular, preenfasís(aplicación de un filtro que atenúa los valores de las bajas frecuencias frente a las altas, $\hat{f}(n) = f(n+1) - \alpha f(n)$), cambio del rango de la amplitud, etc.. En la fase de entrenamiento (cuando se añade una nueva huella a la base), la huella es extraída de una fuente de audio de la mejor calidad posible, tratando de minimizar las degradaciones.

- Segmentación y sobrelape.- La señal de audio es dividida en segmentos, cuyo tamaño debe ser determinado de tal manera que a cada segmento de señal se le considere estacionario. A continuación se aplican operadores de ventana para minimizar discontinuidades (si es que aplicaremos FFT), es necesario que los segmentos seleccionados presenten cierta cantidad de sobrelape para asegurar robustez a los deplazamientos.

Existe una relación intrínseca entre el tamaño de segmento, el tipo de ventana y la cantidad de sobrelape, con el desempeño general del sistema. Los valores más usados para el tamaño del segmento son $0.01 - 1.5$ segundos y la cantidad de sobrelape de $50\,\% - 95\,\%$.

- Transformadas.- Para lograr la disminución deseada de redundancia, los segmentos calculados en el paso anterior pasan por una transformación. La transformación más usada es la FFT, otras transformaciones propuestas son: la transformada discreta coseno (TDC), la transformada de Haar y la transformada Walsh-Hadamard.

  Existen transformadas óptimas en el sentido de compresión de la información y la descorrelación, como la Karhunen-Loève (KL) o la descomposición en valores singulares (SVD), sin embargo debido a su dependencia de la señal y a su complejidad computacional se suelen preferir otras transformadas para mejorar la eficiencia en la compresión, en el tratamiento de ruido y otros procesos subsecuentes [14].

- Extracción de características.- Una vez obtenida la representación en el tiempo o en la frecuencia del audio, se aplican transformadas adicionales para obtener los vectores con las características acústicas. Numerosos algoritmos han sido propuestos para la obtención de este vector final de características, siempre buscando reducir su dimensionalidad e incrementar la invarianza a distorsiones, muchos de estos algoritmos aprovechan la forma en que trabaja el sistema auditivo humano utilizando el análisis de las bandas críticas del espectro (escala de Bark) para obtener parámetros perceptualmente significativos.

  En [9], se usan los coeficientes Mel-Cepstrum (MFCC). En [2] la característica utilizada es la medida de planaridad del espectro (SFM), la cual es una estimación de las características de ruido y tono por banda en el espectro. Papaodysseus y su grupo proponen una solución basada en vectores de representantes de banda, en los cuales está contenido que banda es más representativa que la demás [14]. Kimura [1] usa la energía por banda, Haitsma y Kalker [16] proponen el uso de 33 bandas espaciadas según la escala de barks para obtener una secuencia de bits, la cual es el signo de los cambios de energía, tanto en tiempo como en frecuencia.

  Burges [6, 7] menciona que todos los métodos usados hasta el momento estan basados en heurísticas y por lo tanto no son óptimos, por ello utiliza una variante del análisis de componentes principales (PCA), este es conocido como Análisis de Componentes Principales Orientado (OPCA) y se utiliza para encontrar las características óptimas de un modo no supervisado. El PCA encuentra un conjunto de direcciones que maximizán la varianza, el OPCA obtiene un conjunto de direcciones posiblemente no ortogonales que toman en cuenta las distorsiones que tienen las señales.

  En [28, 18], los autores usan los centroides de la señal en el espacio de las frecuencias. Esta característica esta relacionada con una propiedad denominada brillantez del sonido, que es una medida perceptual.

- Posproceso.- La mayoría de las medidas mencionadas son medidas absolutas. Para caracterizar mejor las variaciones temporales que pueden presentarse, se suelen utilizar derivadas de varios ordenes de las características extraídas. En [25] los autores mencionan un sistema basado en MFCC y que agrega a estos valores la derivada y la doble derivada, así como la derivada de la energía y su doble derivada, con el fin de decorrelacionar los datos se usa la PCA. Algunos otros sistemas usan solamente las derivadas de las características, sin tomar en cuenta sus valores absolutos [8, 17, 2, 14]. Es común aplicar una cuantización de baja resolución a las características: ternaria o binaria. Esto permite obtener robustez en el momento de comparar firmas, ya que hacemos comparaciones sobre elementos de la firma más globales [8]. Cuantizar nos permite hacer comparaciones entre elementos que están dentro de la misma categoría y además la cuantización en términos binarios nos permite almacenar mas fácilmente las huellas en memoria.

Bloque de codificación

El bloque de codificación normalmente recibe una secuencia de vectores de características calculados frame a frame, de forma tal que se explote la redundancia que hay sobre los bloques vecinos. El tipo de codificación usada determinará el tipo de métrica que sea posible utilizar y a la vez el tipo de indexado.

Una forma concisa de huella es representar todo el archivo o segmento de audio en un solo vector. Por ejemplo, aquel formado por la medias y varianzas de todos los vectores obtenidos.

FreeEtantrum es una librería para identificación de música, que comenzó como un sólo proyecto libre y despues se dividió en una parte comercial y otra parte libre. La huella que utilizan es calculada usando las medias y las varianzas de las energías obtenidas de un banco de 16 filtros, aplicado a 30 segundos de audio, codificando la firma en una huella de hasta 512 bits.

MusicBrainz es otro proyecto libre que sirve para automatizar el etiquetado de librerías musicales. La huella que utilizan se basa en el promedio de cruces por cero, la cantidad de notas por minuto (*beats per minute*) que se obtiene de una representación de espectro y algunas otras características de el archivo de audio.

Las huellas de audio pueden ser también simples secuencias de las características. En [16, 9, 8, 17] la huella consiste en secuencias de vectores de características de cada frame, que son codificados de forma binaria para mejorar el manejo y ahorro de memoria.

Se ha intentado ver el problema de identificación de contenido desde el punto de vista del problema de reconocimiento de voz. En [10] se toma un archivo de audio como una sucesión de eventos acústicos, entonces se forma un corpus de eventos acústicos (que en reconocimiento de habla es un archivo que contiene los fonemas y la características que lo representan) y por cada archivo se crea un modelo oculto de Markov(HMM), entonces para cada archivo se obtienen sus características y son evaluadas con el HMM de cada canción.

En [18], los autores proponen agregar a su método basado en centroides espectrales un paso

intermedio antes de la codificación binaria. Apuntando que la codificación binaria de datos reales puede provocar cierta degradación en el rendimiento del sistema, los autores proponen usar un método de clasificación llamado AdaBoost, que combina varios clasificadores débiles para obtener uno solo más robusto.

3.1.4 *Métricas y métodos de búsqueda*

Después de que una base de huellas de audio ha sido indexada, se pueden ingresar consultas al SHA para saber si un segmento de audio esta o no en la base de datos(Fig. 18). El resultado de la comparación de las huellas almacenadas en la base de datos con la extraída de la consulta es una lista de distancias. La decisión para una identificación puede ser hecha basándose en un umbral. Calcular este umbral no es trivial y debe hacerse en base a pruebas, por ejemplo usando curvas ROC (2.6.1). Aparte del método del umbral se puede usar el método descrito en [19], el cual en lugar de buscar puntos que superen un umbral, que es una característica global, usa un método donde se calcula la varianza de las distancias que se obtienen al restar el máximo de un segmento con los elementos de este, así una varianza grande nos indica la presencia de un máximo local que puede pasar desapercibido si lo buscamos usando un método de umbral.

Para obtener las distancias, el método de la correlación es muy usado, tanto para representaciones a nivel de bits como para representaciones de vectores reales. La distancia euclideana o alguna variación de esta también son muy utilizadas. Cuando tenemos vectores de características cuantizadas o binarias, las distancias de Manhattan y Hamming son las más usadas.

Los esquemas que buscan una identificación exacta de las huellas no son muy usados, esto debido a que rara vez la firma de la consulta corresponderá exactamente a algún elemento de la base de huellas, por ello se prefiere usar métodos de búsqueda aproximada, por ejemplo el esquema de LSH (*Local Sensitive Hashing*) [31].

Otro método de búsqueda aproximada muy citado es el mencionado en [22], en este los autores toman la huella de Haitsman [16] (Haitsma forma bloques de bits de 32 por 256, como firma de segmentos de 3 segundos) y proponen formar un árbol 256-ario, en cada nodo se encuentra un bloque de 8 bits de la firma, hacen un descenso en profundidad, en cada nodo calculan la distancia de Hamming de la subfirma definida por el camino recorrido en el subárbol y en base a esto calculan una probabilidad, con la cual deciden si es viable seguir buscando en el subárbol debajo del nodo actual.

Un método referido por muchos en la literatura [12, 11, 25] consiste en aplicar k-medias, dividiendo las huellas en clusters para reducir el tamaño de la búsqueda.

En la búsqueda en colecciones de huellas binarias puede hacerse un muestreo de las firmas para no hacer el proceso de búsqueda con la huella completa. Esto lo hacen en [29] para demostrar que la firma de Haitsma puede utilizarse sin problemas en aparatos electrónicos personales.

3.2 HUELLA BASADA EN LA ENTROPÍA EN EL TIEMPO

La huella desarrollada por Ibarrola e Chávez [17], usa como característica la entropía. Esta huella sobresale por ser una de las pocas huellas mencionadas en la literatura cuya característica se calcula en el dominio del tiempo.

La entropía de una señal f cuantizada de n niveles es :

$$H(f) = -\sum_{i=1}^{n} p_i \log(p_i) \tag{3.1}$$

donde p_i respresenta la probabilidad de ocurrencia del nivel i, para esta definición $\log(0) = 0$. Nótese que no hemos aclarado ninguna base en particular para el logaritmo, y es porque cualquier base puede usarse como definición.

La entropía de una señal indica que tan impredecible es su comportamiento, la entropía mínima en una señal cuantizada de n niveles se alcanza cuando la señal es constante ($p_j = 1, p_i = 0 \forall i \neq j$) y es máxima cuando la distribución de la señal es uniforme ($p_i = \frac{1}{n}$):

$$H_{min} = -\sum_{i=1}^{n} p_i \log(p_i) = -\log(0) = 0 \tag{3.2}$$

$$H_{max} = -\sum_{i=1}^{n} p_i \log(p_i) = -\sum_{i=1}^{n} \frac{1}{n} \log(\frac{1}{n}) = \log(n) \tag{3.3}$$

Por ejemplo si cada muestra de la señal ocupara un byte, entonces la probabilidad máxima sería $\log_{10}(2^8) = 2.4$.

Para calcular la entropía de la señal de audio podemos usar la Eq. 3.1, esta ecuación implica obtener la entropía usando el histograma de la señal, pero puede darse el caso en que deseemos procesar audio en tiempos muy cortos, por ejemplo si queremos frames de 30 ms y estamos usando muestras de 16 bits a 8 khz de frecuencia de muestreo obtendríamos 240 valores para definir un histograma con un soporte de 65536 valores lo que no brindaría una buena estimación.

En la literatura [5] se justifica que puede aproximarse la distribución de los valores de la señal de audio como Gaussiana o Laplaciana. Entonces podemos usar la definición de la entropía de una señal continua para hacer una estimación paramétrica de la entropía.

$$H(f) = \int_{-\infty}^{\infty} p(x) \ln(x) dx \tag{3.4}$$

aquí es conveniente utilizar el logaritmo natural para simplificar los cálculos. Suponiendo que los datos tienen una distribución gaussiana, $p(x) = \frac{1}{\sqrt{2\pi\sigma}} e^{-(x-\mu)/2\sigma^2}$, obtenemos:

$$
\begin{aligned}
H(f) =& -\int_{-\infty}^{\infty} p(x)\ln(x)dx && (3.5)\\
=& -\frac{1}{\sqrt{2\pi}\sigma}\int_{-\infty}^{\infty} e^{-(x-\mu)^2/2\sigma^2}\ln\left(\frac{1}{\sqrt{2\pi}\sigma}e^{-(x-\mu)^2/2\sigma^2}\right)dx && (3.6)\\
=& -k\ln(k)\int_{-\infty}^{\infty} e^{-u^2/2\sigma^2}du + k\int_{-\infty}^{\infty} e^{-u^2/2\sigma^2}\frac{u^2}{2\sigma^2}du && (3.7)\\
=& -k\ln(k)\sigma\sqrt{2\pi} + k\sqrt{\frac{\pi}{2}}\sigma && (3.8)\\
=& -\ln\left(\frac{1}{\sqrt{2\pi}\sigma}\right) + \frac{1}{2} && (3.9)\\
=& \frac{1}{2}(\ln(2\pi) + \ln(\sigma^2) + 1) && (3.10)
\end{aligned}
$$

en los cálculos anteriores $k = \frac{1}{\sqrt{2\pi}\sigma}$, del segundo al tercer paso hicimos el cambio de variable $u = x - \mu$.

Si las muestras tienen una distribución Laplaciana, $p(x) = \frac{1}{2b}e^{-|x-\mu|/b}$, entonces:

$$
\begin{aligned}
H(f) =& -\frac{1}{2b}\int_{-\infty}^{\infty} e^{-|x-\mu|/b}\ln\left(\frac{1}{2b}e^{-|x-\mu|/b}\right)dx && (3.11)\\
=& -\frac{1}{2b}\int_{-\infty}^{\infty} e^{-|u|/b}\ln\left(\frac{1}{2b}e^{-|u|/b}\right)du && (3.12)\\
=& -\frac{1}{2b}\int_{-\infty}^{0} e^{u/b}\ln\left(\frac{1}{2b}e^{u/b}\right) - \frac{1}{2b}\int_{0}^{\infty} e^{-u/b}\ln\left(\frac{1}{2b}e^{-u/b}\right)du && (3.13)\\
=& \frac{1}{2} + \frac{1}{2}\ln(2b) + \frac{1}{2} + \frac{1}{2}\ln(2b) && (3.14)\\
=& 1 + \ln(es2b) && (3.15)
\end{aligned}
$$

Otra medida de la dispersión de una densidad de distribución es el índice de Gini, esta es la varianza obtenida de los valores p_i, denotemos por $\bar{p}$ el promedio de los valores p_i, entonces $\bar{p} = \frac{1}{N}\sum_{i=1}^{N} p_i = \frac{1}{N}$:

$$\begin{aligned}\mathrm{Var}(p_i) &= \frac{1}{N}\sum_{i=1}^{N}(p_i-\bar{p})^2 \\ &= \frac{1}{N}\sum_{i=1}^{N}p_i^2 - 2p_i\bar{p}+\bar{p}^2 \\ &= \frac{1}{N}\sum_{i=1}^{N}p_i^2 - 2\bar{p}^2+\bar{p}^2 \\ &= \frac{1}{N}\left(\sum_{i=1}^{N}p_i^2 - \frac{1}{N}\right)\end{aligned}$$

La parte importante de la última igualdad es: $\sum_{i=1}^{N} p_i^2$, esta medida resulta también ser muy eficiente ya que es fácil de actualizar al mover la ventana sobre la señal y no requiere ningún computo pesado.

En [17] describen el siguiente esquema para obtener la huella de audio:

1. Verificamos que la señal x_n de entrada esté muestreada a una frecuencia de 16 khz, con una definición de 8 bits por muestra y tenga un solo canal.
2. Segmentamos la entrada en bloques de 2 segundos de longitud con un 50 % de sobrelape, asi obtenemos una secuencia de segmentos f_n.
3. A cada segmento f_n le calculamos su entropía $h_n = H(f_n)$, con los parámetros mencionados cada frame tendrá 32000 muestras y por el tamaño de cada muestra se puede usar el método de histograma para calcular la entropía.
4. Obtenemos la firma binaria s_n codificando la derivada de la entropía:

$$s_n = \begin{cases} 1 & \text{Si } \Delta s_n > 0 \\ 0 & \text{Otro caso} \end{cases} \tag{3.16}$$

La mayoría de sistemas de huellas de audio calculan las características usando frames con cierto traslape, por ello es siempre deseable contar con algún método que nos permita actualizar la característica extraída de los frames sin tener que recalcularla utilizando el frame completo. Si calculamos la entropía suponiendo gaussianidad, entonces es conveniente usar la siguiente relación para mantener actualizada la varianza en cada frame:

$$\sigma^2 = E(X^2) - \mu^2 \tag{3.17}$$

La fórmula 3.1 implica muchos cálculos de logaritmos, en [17] se menciona que todas las probabilidades p_i son de la forma $i/n, i = 0, \ldots, N$, donde N es la cantidad de valores diferentes en una muestra de audio, entonces se calcula una tabla T_l tal que almacena los valores

Algoritmo 1 Calculo de la entropía usando histograma

```
1: T_l = computeLogTable(sizeFrame)
2: frame = readData(streamAudio, sizeFrame)
3: histogram = computeHistogram(frame)
4: entropy = 0
5: for i = 0 to 255 do
6:     entropy += T_l[histogram[i]]
7: end for
8: idx = 0, star = 0
9: entropies[idx++] = entropy
10: while dataAvailable do
11:     subFrame = readData(streamAudio, sizeSubFrame)
12:     for j = 0 to sizeSubFrame − 1 do
13:         entropy− = T_l[histogram[frame[star]]]
14:         entropy− = T_l[histogram[subFrame[j]]]
15:         histogram[frame[star]]−−
16:         histogram[subFrame[j]]++
17:         entropy+ = T_l[histogram[frame[star]]]
18:         entropy+ = T_l[histogram[subFrame[j]]]
19:         frame[star] = subFrame[j]
20:         star = (star + 1) mod sizeFrame
21:     end for
22:     entropies[idx++] = entropy
23: end while
24: return entropy
```

$T_l[i] = \log(\frac{i}{N})$.

El programa utilizado para el cálculo de la de la entropía se muestra a continuación.

Este modo de codificar la huella fue usado en [17], pero los parámetros no son buenos para algunos ejemplos que encontramos en una colección de emisiones radiofónicas. Además, los autores mencionan que la firma no es robusta a las degradaciones producidas por la ecualización. En el siguiente capítulo haremos algunas pruebas para obtener los parámetros que nos den un reconocimiento óptimo y demostraremos que esta huella no es tan débil a la ecualización dentro de cierto contexto.

Parte III

RESULTADOS EXPERIMENTALES

4 EXPERIMENTOS

En el presente capítulo se muestran los resultados obtenidos al aplicar TES a secuencias de audio con las diferentes degradaciones que la literatura de huellas de audio menciona, hacemos énfasis en la degradación producida por la ecualización y probamos la robustez del método a la compresión del esquema GSM. A continuación describimos el conjunto de datos sobre los que se haran las pruebas.

4.1 INTRODUCCIÓN

Para realizar las pruebas contamos con colecciones de segmentos de audio, la primera consta de una emisión de 14 horas, tomada de una estación de radio que fue sintonizada con la mayor claridad posible, con las siguientes características: formato WAVE con codificación PCM, 16 khz, un canal y 8 bits por muestra, de esta colección se han extraído aleatoriamente 100 segmentos de audio de 30 segundos de duración, dichos segmentos no se traslapan y aparecen una sola vez en la emisión.

La segunda colección consta de la emisión de un día de 5 estaciones diferentes grabadas en la ciudad de Guadalajara, esta fue proporcionada por la compañía de monitoreo de medios CONTACTO M.R.S MÉXICO, con formato MP3 a 16 khz, un canal y a compresión 64 kbps. La cantidad de horas grabadas en cada estación varia entre 16.8 y 18 horas, esto es aproximadamente el tiempo que define una jornada laboral en una estación, acumulando un total 87 horas de audio. El día que se obtuvieron las grabaciones fue el 20 de Febrero de 2007, las estaciones son:

1. 1027 FM 40 Principales
2. 971 FM Ke Buena
3. 891 FM Máxima
4. 931 FM Nueva Amor
5. 1003 FM Super RMX

Tenemos un conjunto de 10 comerciales diferentes y sus tiempos de emisión en cada una de las estaciones, el total de repeticiones es de 226. Cabe mencionar que la agenda fue también proporcionada por CONTACTO M.R.S MÉXICO dado que estaba incompleta para el conjunto de comerciales se corrigió manualmente, para asegurar que los comerciales se emitieran al momento que marca la agenda y que no hubiera repeticiones de dichos comerciales no marcadas. Puede encontrarse una copia de esta agenda en el Apéndice D.

A diferencia de la primera colección estas grabaciones son de menor calidad, ya que fueron tomadas directamente del equipo con el que trabajan en la estación de monitoreo, sin embargo,

estos datos presentan un buen conjunto de prueba, ya que aunque desconocemos la degradación que contienen, si los resultados son favorables sabremos que tenemos un método eficaz para resolver el problema de monitoreo.

Todos los experimentos se realizaron en una computadora laptop con Procesador Pentium IV, 2.66 Ghz y 512 MB RAM. Se utilizó el compilador C++ Builder Ver. 5.0 de Borland.

4.2 DESCRIPCIÓN DE LA IMPLEMENTACIÓN

En una primera etapa de experimentación con la TES usamos el valor de la entropía de cada segmento extraído del audio para crear la firma de un comercial. Para encontrar el momento de repetición del comercial dentro de la emisión radiofónica obteníamos primero la firma de la emisión completa y desplazábamos la firma del comercial sobre de esta, para calcular la distancia usamos como métrica la integral del valor absoluto de las derivadas de las distancias entre cada entrada de las firmas, está métrica es diferente a la usada en el paper.

En la Fig 20 se muestra la firma del comercial "IFE Actualizate", la Fig. 21 muestra las distancias que se obtienen al buscar la firma del comercial dentro de la firma de la emisión. En la Fig. 22 se encuentra el segmento de la firma de la emisión que presenta menor distancia, podemos ver las similitudes en la forma de los valores de la entropía entre el comercial buscado y el segmento encontrado en la emisión.

Este esquema de trabajo no resulto muy efectivo, debido a que no es posible aplicar una técnica de discriminación sencilla, por ejemplo un umbral, esto puede verse en la gráfica de distancias obtenidas al buscar el comercial de "Cerveza Indio" en la Fig. 23. Podemos usar una técnica como la descrita en [19], en la que se usan la varianza de la distancias respecto a la distancias mínima dentro de una secuencia de características extraídas para tener un estimador del lugar donde hay una ocurrencia.

Sin embargo el problema radica en tener que hacer las cuentas con vectores flotantes. Procesar la emisión de un solo día de esta forma toma aproximadamente 20 minutos por comercial, y aunque esto nos indique que es viable para usarlo en tiempo real los métodos descritos en la literatura tienen un tiempo de ejecución del orden de segundos.

Basados en lo anterior decidimos usar la codificación binaria de la TES como fue descrita en el Capítulo 3. Para hacer esta codificación binaria es necesario definir un tamaño de segmento y traslape adecuados, como vemos en las imágenes de la Fig. 24 hacer los segmentos más chicos hace que aumente la cantidad de detalle contenido en la firma, el efecto del traslape se refleja en la suavidad, entre más pequeño sea, hace una firma con un perfil más suave. Al final encontramos que los parámetros que dan mejores resultados son longitud de segmento de 0.25 segundos y de traslape del 95 %.

Otra forma de discriminar los parámetros de longitud de segmento y traslape es usando las matrices de confusión, en la Fig. 25 podemos ver esto, en el contraste que se presenta en las matrices de confusión obtenidas para diferentes parámetros.

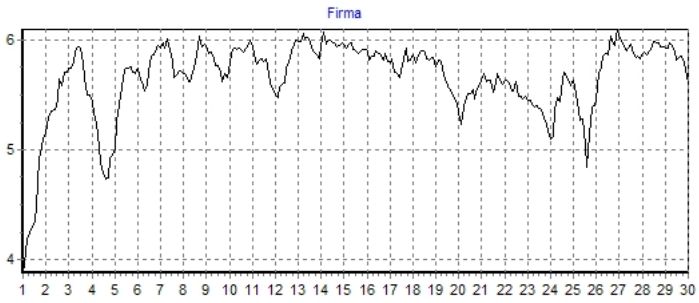


Figura 20. Firma del comercial IFE Actualizate. El eje X representa el tiempo en segundos y el eje Y la entropía.

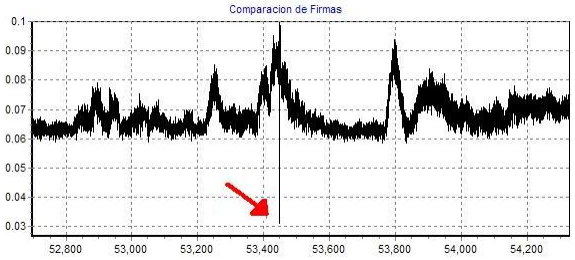


Figura 21. Distancias producidas al buscar la firma del comercial IFE Actualízate. El eje X indica el tiempo y el Y la distancia. Se indica el valor mínimo que marca la ocurrencia.

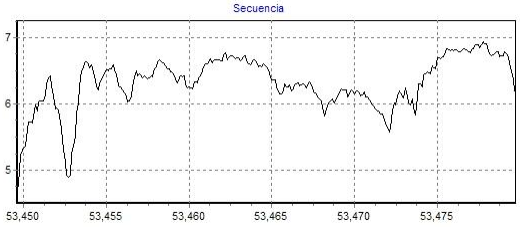


Figura 22. Firma del comercial IFE Actualizate encontrado en la emisión. El eje X representa el tiempo en segundos y el eje Y la entropía.

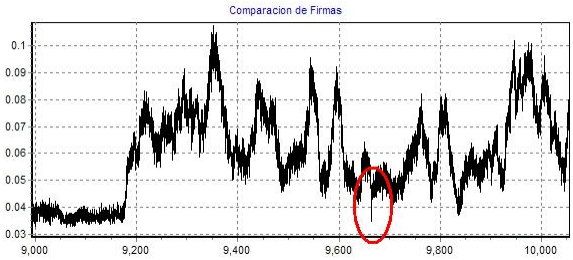
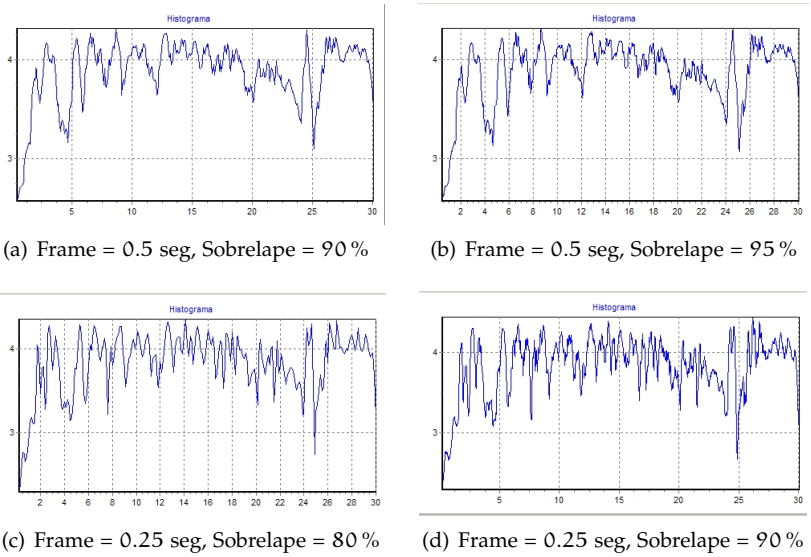


Figura 23. Distancias producidas al buscar la firma del comercial "Cerveza Indio", codificada de forma real.



(a) Frame = 0.5 seg, Sobrelape = 90 %

(b) Frame = 0.5 seg, Sobrelape = 95 %

(c) Frame = 0.25 seg, Sobrelape = 80 %

(d) Frame = 0.25 seg, Sobrelape = 90 %

Figura 24. Firmas del comercial "IFE Actualízate" con diferentes tamaños de frame y sobrelape.

[16] sugiere que la búsqueda puede acelerarse realizando un submuestreo de la huellas de audio, esto fue comprobado e implementado para dispositivos portátiles en [29].

En la Fig 26 vemos las huellas de audio codificadas según TES, en las figuras marcadas como original y encontrada las zonas blancas son 1 y las negras 0, en las marcadas como diferencias las zonas blancas marcan las partes que son iguales y las negras las partes que son diferentes. Podemos apreciar como los errores están repartidos a los largo de toda la firma. Este detalle sera usado mas adelante para saber si ha habido algún recorte en la transmisión del comercial.

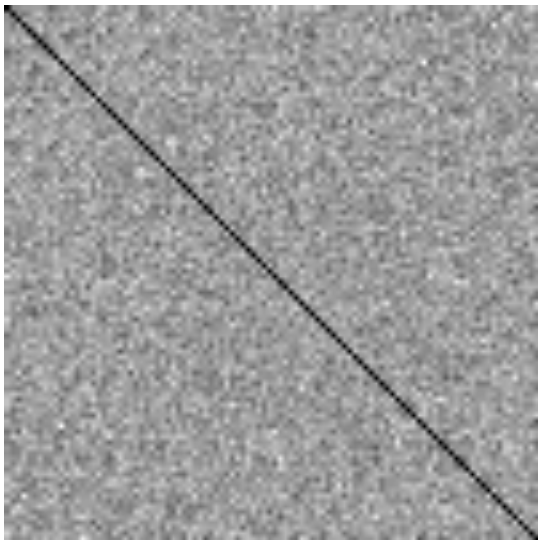
Para realizar la búsqueda de un comercial dentro de la emisión completa procederemos de la misma forma que con la codificación real: primero obtendremos la firma de toda la emisión y a continuación buscaremos la huella del comercial dentro de la huella de la emisión desplazándola bit a bit, la métrica usada es la distancia de Hamming. Este paso debe ser implementado de manera eficiente ya que después de la extracción de la huella es el método más pesado en todo el proceso.

Podemos implementar una cola circular que mantenga una secuencia de los últimos bits, la cual sera usada para hacer la comparación con el comercial que buscaremos. Pero hacer la comparación con esta estructura puede ser pesado, notemos ademas que las distancias de Hamming entre bits serán calculadas muchas veces.

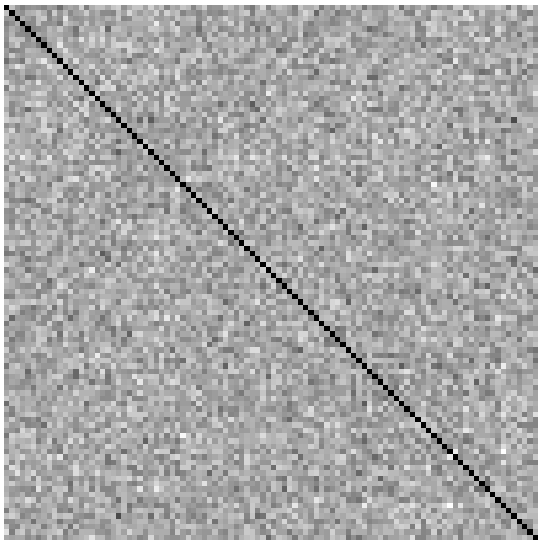
Para hacer esto de manera eficiente se construyo una tabla *lookup* que recibe 3 entradas, las primeras dos indican el valor de dos variables de tamaño un byte sin signo, la tercera indica un desplazamiento en bits de la segunda variable respecto a la primera y contiene la distancia de Hamming. Por ejemplo, `tabla[120][122][3]` contiene como valor 2, la Fig. 27 aclarará este proceso.

Esta forma de resolver el problema es muy eficiente. En la primera propuesta necesitariamos hacer una prueba bit a bit, mientras que en la segunda estamos obteniendo la distancia total haciendo operaciones en bloques de bytes.

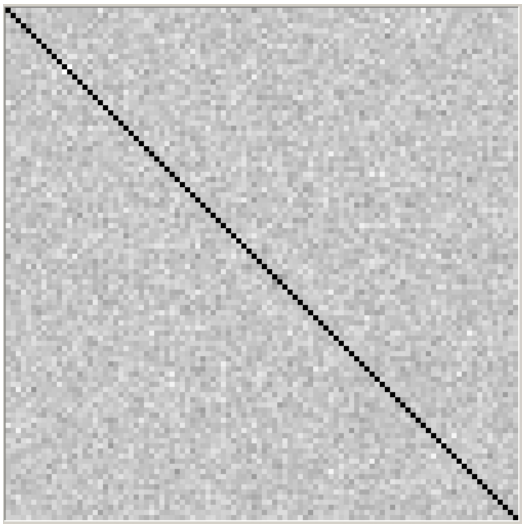
A continuación mostraremos la robustez de la TES para algunas degradaciones mencionadas en la literatura.



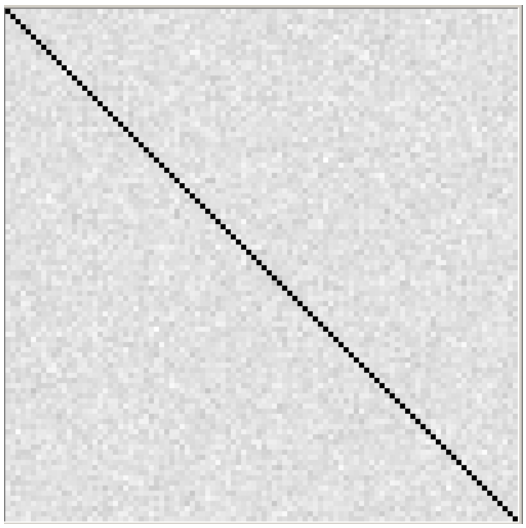
(a) Frame = 1.5 seg, Sobrelape = 30 %



(b) Frame = 1 seg, Sobrelape = 50 %



(c) Frame = 0.25 seg, Sobrelape = 50 %



(d) Frame = 0.25 seg, Sobrelape = 95 %

Figura 25. Matrices de confusión formadas por las distancias de un grupo de 100 segmentos a si mismos, entre mayor contraste presente la diagonal con el fondo mejor será la discriminación.

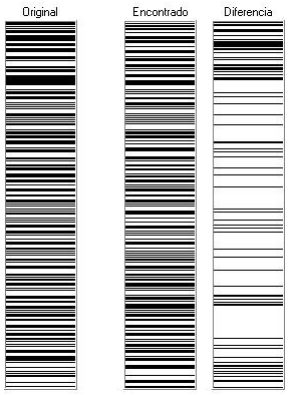


Figura 26. Distancias producidas al buscar la firma del comercial "IFE Actualizate".

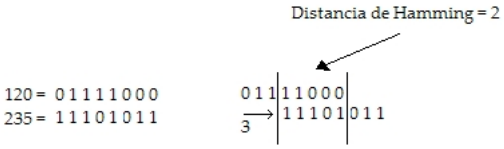


Figura 27. Tabla *lookup* usada para calcular eficientemente distancias de Hamming.

4.3 PRUEBA DE ROBUSTEZ

En esta sección realizaremos algunas pruebas para comprobar la robustez de la TES a las siguiente degradaciones: compresión MP3 a distintos niveles, compresión GSM, ruido aditivo y distintos grados de ecualización.

4.3.1 *Compresión con pérdida*

Tomamos para el siguiente experimento nuestra colección de audio formada por una emisión radiofónica de 14 horas, procedemos a aplicarle compresión del esquema MP3 a varios niveles, como el archivo original esta muestreado a 8 khz, no tiene sentido intentar usar compresión MP3 de más de 64 kbps, ya que los archivos comprimidos quedan más grandes que el original, por lo tanto usaremos compresión a $16, 32, 48$ y 64 kbps.

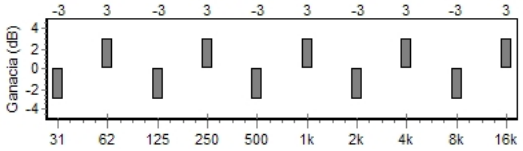
Los resultados están clasificados de acuerdo al siguiente criterio: de los datos de audio que tenemos podemos tener su ubicación en la emisión a nivel de muestra, pero realmente esa precisión tan extrema no nos es útil, nos interesa más bien la precisión del orden de segundos, para entonces proporcionar en la agenda el tiempo en segundos en que comienza un segmento, entonces decimos que el método encontró el segmento si da un tiempo con a lo más dos segundos de diferencia, un segmento que no se encuentra se marca como "no encontrado" y por último si el sistema detecta un evento fuera de los agendados lo marcaremos como falso positivo. Los resultados fueron los siguientes:

Compresión	Verdaderos P	Falsos P	No Encontrados
16 kbps	100	0	0
24 kbps	100	0	0
32 kbps	100	0	0
48 kbps	100	0	0
64 kbps	100	0	0

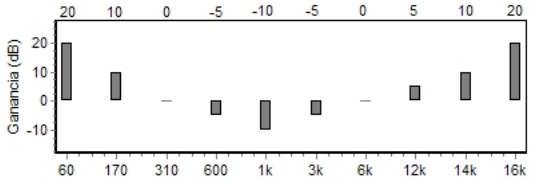
Como podemos ver el método no tiene problemas para el manejo de audio comprimido con pérdida, lo cual es muy bueno, ya que en una implementación grande esto puede ahorrarnos bastante espacio de almacenamiento.

4.3.2 *Ecualización*

En [17] los autores reportan que la TES es muy debil a distorsiones debido a la ecualización, cabe mencionar que las distorsiones hechas por los autores son muy extremas con respecto a las mencionadas en la literatura, por ejemplo, [16] Haitsma menciona que las huellas de audio deben resistir una ecualización con la siguiente configuración:

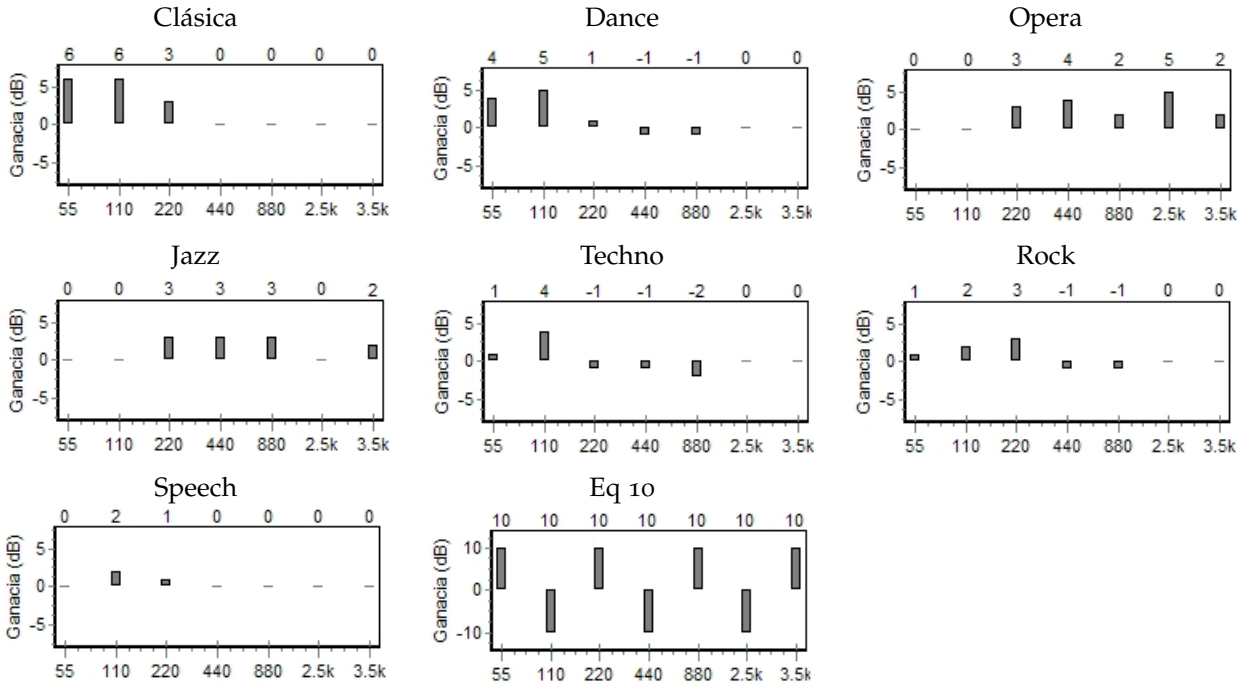


Ibarrola y Chávez proponen el siguiente modo de ecualización en [8]:



La primera ecualización es en efecto débil, en el sentido de que no se puede hacer gran diferencia entre el degradado y el original. La segunda es más fuerte en este sentido, pero en efecto es muy extrema, en la literatura de huellas de audio aplicado al monitoreo [13] se comenta que la radiodifusoras suelen ecualizar el audio para hacerlo mas llamativo, pero no lo hacen de manera extrema.

Para comprobar la resistencia de la TES a la ecualización buscamos parámetros de ecualización predefinidos en algunos aparatos de música, estos parámetros estan hechos para resaltar las propiedades acústicas de acuerdo al ritmo.



la siguiente son los resultados para la ecualización:

Ecualización	Verdaderos P	Falsos P	No Encontrados
Haitsman	100	0	0
Clásica	100	0	0
Dance	100	0	0
Opera	100	0	0
Jazz	100	0	0
Techno	100	0	0
Rock	100	0	0
Speech	100	0	0
Eq 10	83	0	17
SES eq	29	0	71

Como vemos, en efecto el reconocimiento se degrada bastante cuando la ecualización es extrema. Una forma de ver como a que grado la señal cambia respecto al original es obteniendo el SNR. Para las ecualizaciones arriba mencionadas, excepto las últimas dos, está entre 5 y 10 dB para los parámetros de Eq 10 obtenemos un SNR de 2.1 dB y para SES eq es de -2.3 dB.

Concluimos que en condiciones normales, la TES es resistente a la ecualización.

4.3.3 *Codificación GSM*

El esquema de codificación GSM es el usado para telefonía celular, toma extractos de voz de 20 ms y los codifica en 260 bits, haciendo una taza de transferencia de 13 kbits, es de esperarse que muchos de los detalles del sonido se pierdan, mas que nada por que no es un codec diseñado para uso general, esta enfocado a voz. Utilizando la herramienta de audio sox descrita en el Apéndice C, tomamos la emisión de radio y la codificamos a GSM, los resultados son los siguientes.

Codec	Verdaderos P	Falsos P	No Encontrados
GSM	100	0	0

4.3.4 *Ruido Aditivo*

Probaremos la robustez de la TES al ruido agregando a la señal de entrada ruido Gaussiano. Este es otro aspecto en que la literatura los autores tienden a modificar muy poco, por ejemplo [16] menciona que el requisito a superar son $23 - 25$ dB pero eso es mu poco, para el estandar de las estaciones de radio sería el equivalente a una pequeña desintonización de la estación.

Los resultados que obtuvimos son los siguientes:

σ	SNR	Verdaderos P	Falsos P	No Encontrados
5	13.8	100	0	0
10	7.8	100	0	0
15	4.2	99	0	1
18	2.6	99	0	1
20	1.7	99	0	1

4.3.5 *Cropping*

Para esta prueba se procedio a recortar la parte final de cada segmento extraido, asi en la parte de busqueda la firma obtenida solo representa un fragmento del principio de cada segmento, la tabla siguiente muestra los resultados obtenidos.

Duración(seg)	Verdaderos P	Falsos P	No Encontrados
5	98	0	2
10	100	0	0
15	100	0	0
20	100	0	0

4.3.6 *Desempeño con datos reales*

Ahora haremos una prueba de tiempo en la base de datos de audio reales. Cabe mencionar que en muchos sistemas descritos en la unidad 3 utilizaban algún método de clustering o de búsqueda de similaridad, para nuestro problema no usaremos ninguno de esos métodos, usaremos la firma completa del comercial para la búsqueda dentro de la firma de la secuencia.

En la Fig. 28 vemos la gráfica de distancias formada al buscar el comercial IFE Actualízate dentro de la transmisión de 18 horas de la estación de "Las 40 Principapales.". Los valores más bajos indican las apariciones del comercial, estos "picos" no están formados por una sola muestra sino por varias, en la Fig. 29 mostramos un acercamiento a uno de estos descensos, los puntos en la imagen marcan el punto donde se alinea la firma del comercial para hacer la comparación, además notamos que el valor de la distancia es muy sensible, una muestra de diferencia en la posición y la distancia cambia hasta en un 10 % sobre el tamaño de total de la firma del comercial. Esto muestra que si deseamos usar un esquema de umbral para detectar las emisiones de comerciales debemos revisar en cada muestra. Se aprecia también que cerca de la firma hacia ambos lados hay dos puntos de valor máximo, estos puntos son debidos a que las firmas no tienen una estructura regular a bloques de 1's y 0's. En la Fig. 31 una ilustración de la situación mencionada.

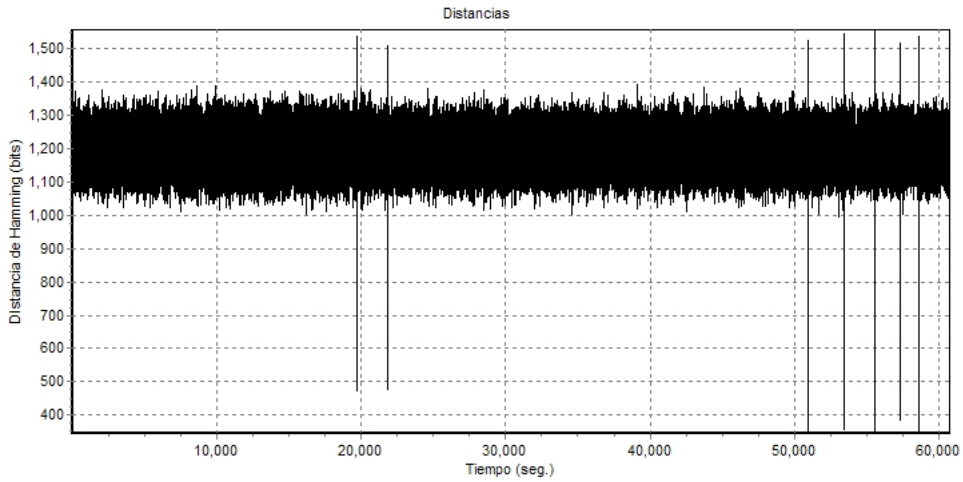


Figura 28. Gráfica de distancias obtenidas al buscar el comercial "IFE Actualízate" en la emisión de la estación los 40 Principales. La longitud de frame es de 0.2 segundos y un traslape de 95 %.

En toda la literatura se concuerda en que una forma de obtener mejor taza de reconocimiento durante el proceso de búsqueda es usando un tamaño de frame más pequeño y/o traslapes más grandes, el tamaño de traslape determina cuanto tiempo tardara el método en extraer la huella de una secuencia de audio, por lo tanto no puede usarse muy pequeño. En las pruebas realizadas empíricamente, ver Fig. 25, encontramos que los valores buenos para la longitud de frame y de traslape son del 0.25 segundos y traslape del 95 %, formalizaremos este resultado haciendo una prueba con una curva ROC.

Tomamos el conjunto de 100 segmentos usados en esta sección, con ellos formamos una colección en la cual realizaremos búsquedas con las firmas de audio, a estos 100 segmentos originales agregamos otros 100 segmentos y les hacemos degradaciones para formar un grupo de consultas, hemos realizado 4 tipo de degradaciones: ruido con SNR de 4 dB, compresión gsm, y dos ecualizaciones, la de SES y la Clásica. Hemos dejado fija la cantidad de traslape (95 %) usada, y movimos el tamaño de frame usado (0.25 seg., 0.5 seg.,1 seg. y 1.5 seg.), vemos que no existe una gran diferencia entre los resultados obtenidos, el mejor resultado es el que usamos: longitud de frame de 0.25 seg. y traslape del 95 %.

Ya que tenemos los parámetros ideales usaremos la colección de audio formada por la emisión de 5 estaciones y la base de diez comerciales, en esta experimento haremos mas énfasis a los tiempos. Se obtuvieron los siguientes resultados: el tiempo total de extracción de la huella de audio para cada emisión en el equipo mencionado fue de 30 segundos para la emisión mas larga (18 horas), el tiempo de revisión de un comercial sobre la firma de audio fue de 20 segundos.

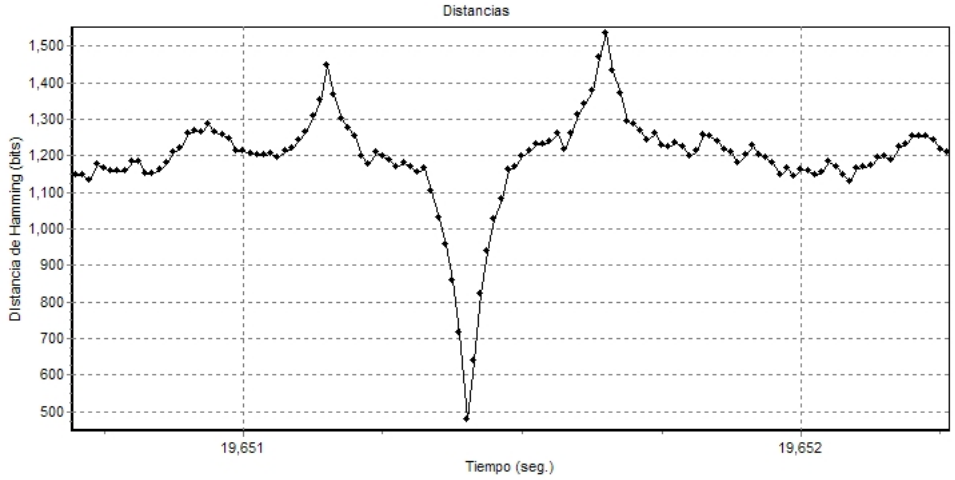


Figura 29. Zoom a uno de los puntos de mínima distancia, notemos que no esta conformado por una sola toma, pero si que es muy sensible el valor de la distancia.

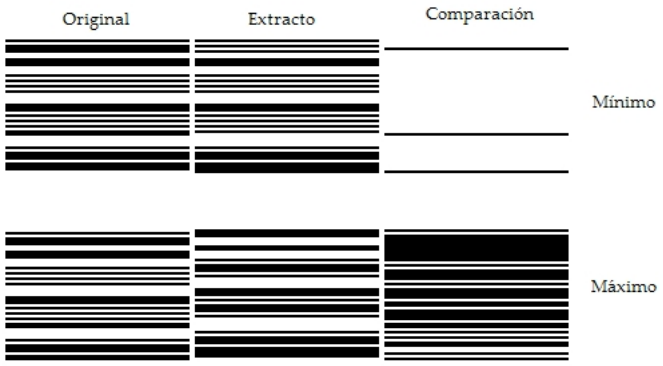


Figura 30. Los bloques de arriba muestran aproximadamente 240 bits de las huella de audio original, la extraída de la secuencia y la diferencia de estas, en el punto de máximo y mínima distancia.

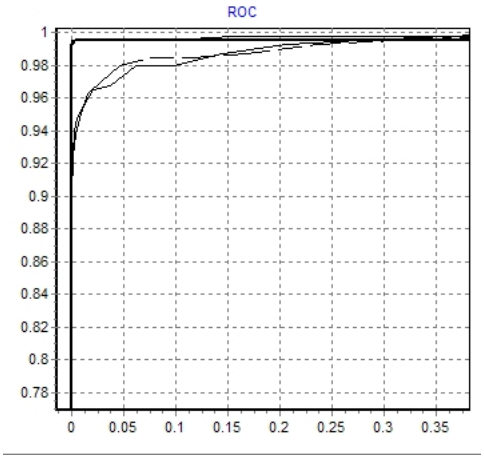


Figura 31. Resultados obtenidos para diferentes longitudes de frame. La linea negra indica la curva obtenida por la longitd de frame de 0.25 seg y 95 % de traslape.

Haciendo cuentas esto nos dice que usamos menos de la milésima parte del tiempo real para revisar si en la transmisión actúal esta un comercial, sería fácil pensar que entonces un solo equipo podría monitorear toda una ciudad, pero hay muchos detalles que necesitan ser tomados en cuenta, un sistema así necesita una implementación multihilos, que dependiendo del lenguaje usado puede llegar a consumir muchos recursos, aparte de la forma en que se le alimentara del flujo de audio a la computadora, ya que las tarjeta con mas entradas de audio tienen a los más unas 24. La cantidad en promedio de estaciones en una ciudad como Guadalajara, Monterrey o el D.F. es de 60, y la cantidad de objetivos que se desean monitorear son del orden de unos 100, entonces usando esta técnica una maquina con las características mencionadas al principio de esta unidad puede monitorear alrededor de una docena de estaciones.

Los resultados obtenidos para la base de datos por comercial y por estación son:

Comercial Estación	40 Prin.	Nuev. Amor	Super RMX	Ke Buena	Maxima
IFE_A	100 %	100 %	100 %	100 %	100 %
IFE_Q	100 %	-	100 %	100 %	100 %
PAP	100 %	100 %	-	-	-
CER	100 %	-	100 %	-	100 %
IBE_N	100 %	100 %	100 %	-	-
IBE_D	-	-	-	-	100 %
IBE_A	-	-	-	-	100 %
LEM	100 %	100 %	-	100 %	-
TEL	100 %	-	-	100 %	-
PEP	-	100 %	100 %	100 %	100 %

Como vemos en la tabla el método encuentra todas las emisiones de los comerciales que buscamos. El umbral para esta búsqueda fue puesto en 35 % como lo marco el análisis de la curva ROC.

5 CONCLUSIONES Y TRABAJO FUTURO

5.1 CONTRIBUCIONES

En el trabajo presentado en esta tesis obtuvimos las siguientes contribuciones:

1. Mejoramos los tiempos de búsqueda con la tabla *lookup* que diseñamos.
2. Con el afinamiento de parámetros realizado conseguimos una identificación correcta bajo las distorsiones de ruido, compresión con perdida, mp3 y gsm, además demostramos en particular que si es robusta bajo la ecualización no extrema.
3. Con la TES con parámetros afinados construimos un prototipo de un sistema de monitoreo de audio

5.2 CONCLUSIONES

Después de las pruebas realizadas llegamos a las siguientes conclusiones:

1. Presentamos una solución al problema de monitoreo de audio.
2. El espacio ocupado por una huella de audio es del orden de una milésima parte de su original bajo la codificación de 8 khz con una resolución de 1 byte por muestra.
3. Encontramos el 100 % de los comerciales.
4. El tiempo de búsqueda es del orden de un milésimo de tiempo real.
5. El sistema obtenido es escalable.
6. Mejoramos la huella TES presentada originalmente.
7. Es posible pensar en la implementación de un sistema comercial altamente competitivo con los productos mostrados en la introducción.

5.3 TRABAJO FUTURO

Para hacer el sistema aún más eficiente es necesaria la creación de un indice, dicho indice no sería como el comentado en la mayor parte de la literatura donde se almacenan solo extractos de cada objeto a identificar, si no más bien se almacenarían las huellas de los comerciales completos. Para hacer pruebas de este tipo necesitamos una base de emisiones más grande.

Ahora hay un interés por extender la técnologia de huellas de audio a video [1]. Se puede pensar en extender la idea no solo procesando el audio, si no también el video en si. Esto

1 Como referencia lease esta noticia

debido a que existen páginas de internet como YouTube en las que los usuarios tienen facilidad de subir videos, que en algunas ocasiones contienen material sujeto a derechos de autor o bien suben los archivos completos de películas, por ello compañías como Warner Bros están interesados en tecnologías de huellas de video, actualmente el problema radica en que no existe una huella que les permita manejar eficientemente la cantidad de información que manejan, y que trabaje a la velocidad que requieren debido a la complejidad computacional que se manejan las huellas actuales. La idea seria buscar un esquema que permita usar la entropía para obtener una firma, sin la necesidad de recurrir a operaciones pesadas sobre la secuencia de video.

Parte IV

APÉNDICE

ESTRUCTURA ARCHIVOS WAVE

A

Los programas descritos en esta tesis procesan el audio de archivos en formato WAVE canónico(*.wav). Las principales ventajas de este formato son que no requiere de ninguna librería especial para leer los datos que forman la señal de audio, es muy flexible en cuanto a las características del audio (frecuencia, canales, tamaño de la muestra, etc.) y puede ser reproducido por cualquier programa de multimedia.

La estructura de una archivo WAVE es una variante del formato RIFF (*Resource Interchange File Format*, formato de fichero para intercambio de recursos), almacena la información en "bloques", y es relativamente parecido al IFF y al formato AIFF usado por Macintosh.

El formato WAVE toma en cuenta algunas peculiaridades del CPU Intel, y es el formato principal usado por Windows. Por ser un formato sin perdida es muy utilizado para tareas de alta calidad, pero su uso no es estandar debido a la cantidad de espacio que ocupa, por ejemplo un minuto de audio a 44.1 Khz, estéreo con muestras de 16-bits (calidad CD), ocupa aproximadamente 5 megabytes de almacenamiento.

El formato WAVE es flexible con respecto a los características del audio que puede almacenar, pero requiere mucho espacio para almacenamiento.

A.1 ESTRUCTURA DE UN ARCHIVO WAVE

En la Fig. 32 se muestra el encabezado de un archivo en formato WAVE, a esta configuración en particular se le conoce como formato WAVE canónico. Esta conformado por 3 bloques (*Chunks*), el primero determina que tipo de archivo RIFF que estamos procesando, el segundo que características tiene la señal de audio (codificación, frecuencia, número de canales y tamaño de muestra) y el último indica cuantas muestras hay y contiene las muestras de la señal. El Cuadro 4 indica el significado de los campos del encabezado y su tamaño.

El siguiente es un ejemplo de los primeros 44 bytes de un archivo WAVE de 30 segundos, con un canal, 8 khz y muestras de 1 byte. Los primero 4 bytes (`52 49 46 46`) son los códigos ascii de la palabra "RIFF". Haciendo cuentas sabemos que la cantidad de datos de audio son $240,000$ bytes (0x0003A980), mas 36 es 0x0003A9A4, que son los bloques de bits $4-7$ y $40-43$ respectivamente. Es fácil corroborar las otras características.

```
52 49 46 46    A4 A9 03 00    57 41 56 45    66 6D 74 20
10 00 00 00    01 00 01 00    40 1F 00 00    40 1F 00 00
01 00 08 00    64 61 74 61    80 A9 03 00
```

Es importante recordar que algunos datos están guardados en formato *little endian* (los bytes menos significativos están almacenados en las direcciones más bajas) y otros en *big endian* (los bytes menos significativos en las direcciones más altas).

Para conocer la informacíon de un archivo WAVE podemos usar la siguiente estructura declarada en la librería `wavCab.h`:

```
typedef struct {
               char ChunkID[4];
               unsigned int ChunkSize;
               char Format[4];
               char Subchunk1ID[4];
               unsigned int Subchunk1Size;
               unsigned short int AudioFormat;
               unsigned short int NumChannels;
               unsigned int SampleRate,ByteRate;
               unsigned short int BlockAlign,BitsPerSample;
               char Subchunk2ID[4];
               unsigned int Subchunk2Size;
} CabeceraWav;
```

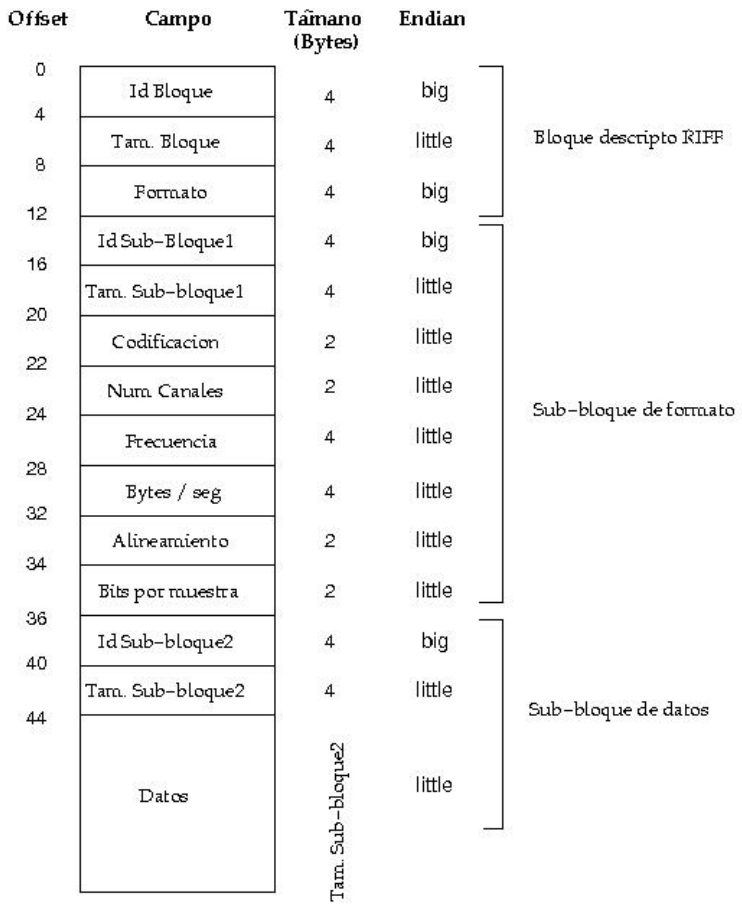


Figura 32. Mapa de un archivo en formato WAVE.

OFFSET	TAMAÑO	NOMBRE	DESCRIPCIÓN
0	4	Id Bloque	Contiene las letras "RIFF" en ASCII (0x52494646).
4	4	Tam. Bloque	$=36 + sc_2$ ó $4 + (8 + sc_1) + (8 + sc_2)$. Este es el tamaño del resto del archivo, a partir del siguiente campo.
8	4	Formato	Contiene las letras "WAVE" en ASCII (0x57415645).
12	4	Id Sub-bloque1	Contiene las letras "fmt" en ASCII (0x666d7420).
16	4	Tam. Sub-bloque1(sc_1)	16 para codificación PCM.
20	2	Codificación	PCM = 1, cualquier valor diferente de 1 indica algún otro formato de compresión.
22	2	Num Canales(n_c)	Mono = 1, Stereo = 2, etc.
24	4	Frecuencia(f)	8000, 16000, 44100 Hz etc.
28	4	Bytes/seg(b_s)	$= f * n_c * b_r/8$
32	2	Alineamiento	$= n_c * b_m/8$ El numero de bytes en una muestra en todos los canales.
34	2	Bits por muestra(b_m)	8 bits = 8, 16 bits = 16, etc.
	2	ExtraParamSize	Solo si no es codificación PCM
	X	ExtraParams	Parámetros extra.
36	4	Id Sub-bloque2	Contiene las letras "data" en ASCII (0x64617461).
40	4	Tam. Sub-bloque2 (sc_2)	= Num. muestras $*n_c * b_m/8$ Este es el numero de bytes que siguen, y repesentan la señal de audio.
44	*	Datos	Los datos de la señal de audio.

Cuadro 4. Formato WAVE canónico.

B

MANEJO DE ARCHIVOS Y DISPOSITIVOS DE AUDIO CON JAVA

La interfaz de programación de sonido para Java (Java Sound API) esta incorporada al JDK desde la versión 1.3. Esta interface es muy versátil ya que nos permite obtener un flujo de datos desde un archivo de audio o directamente de un dispositivo de captura de audio, además es multiplataforma, usa la misma sintaxis independientemente del sistema operativo.

En este apéndice describiremos la forma de leer los datos de audio desde un archivo o desde un dispositivo de audio, no trataremos la reproducción de estos. Para más información sobre la Java Sound API puede consultar [21].

B.1 INTRODUCCIÓN A LA INTERFACE DE SONIDO DE JAVA (JSA)

La JSA esta conformada de 4 paquetes:

- `java.sound.sampled`
  Este paquete especifica las interfaces para captura, mezcla y reproducción de audio, es el único paquete que necesitamos de la JSA.
- `java.sound.midi`
  Este paquete permite accesar las interfaces MIDI.
- `java.sound.sampled.spi,java.sound.midi.spi`
  Estos paquetes son usados para crear extensiones a la JSA.

La JSA no supone ninguna arquitectura en particular, dependiendo de la plataforma utiliza los recursos disponibles para accesar los dispositivos de audio, por ejemplo, en Windows utiliza la librería Direct Sound y en Linux la ALSA.

B.2 COMO OBTENER LOS DATOS DE AUDIO

Para reproducir o capturar sonido usando la JSA se necesitan al menos 3 elementos: el formato del audio, un mezclador (`mixer`) y una linea (`line`) de entrada.

El formato del audio de entrada se establece mediante un objeto de la clase `AudioFormat`, el cual incluye los siguientes atributos:

- Codificación (usualmente PCM)
- Número de canales
- Frecuencia de muestreo (Número de muestras por segundo, por canal)
- Bits por muestra por canal
- Frecuencia de frames

- Tamaño de frame en bytes
- Tipo de alineación (Big-Endian, Little-Endian)

En lo anterior, por frame nos referimos al conjunto de muestras que fueron tomadas al mismo tiempo en todos los canales.

El JSA no soporta cualquier tipo de codificación, normalmente la JSA solo soporta los formatos PCM, Mu-law y a-law. Dependiendo del dispositivo que leeremos serán los formatos que podamos usar. Los formatos de archivos que podemos leer son: WAVE, AIFF y AU, pero podemos extender las capacidades de la JSA usando librerías como tritonus [30], que agregan capacidades a la JSA que viene por default en el JDK.

Java crea una capa intermedia para accesar los recursos de audio dependiendo del sistema operativo, y el mezclador (`mixer`) es la representación que le da a un dispositivo de captura o reproducción de audio.

Por último, una linea (`line`) es la representación que sirve para la lectura ó escritura de los datos de audio, según el esquema de la JSA cada mezclador puede tener varias lineas.

B.3 INSTRUCCIONES BÁSICAS

El modo más sencillo de empezar a grabar datos de la entrada de audio es creaando un objeto `AudioFormat` con el formato de audio deseado, utilizar el método `AudioSystem.isLineSupported (AudioFormat infoLinea)` para confirmar si se tiene disponible alguna linea que soporte dicho formato, en caso de ser asi, se solicita dicha linea con el método `AudioSystem.getLine (AudioFormat infoLinea)`. Este proceso se detalla en el siguiente código:

```
01    AudioFormat formatoAudio;
02    DataLine.Info infoLinea;
03    TargetDataLine linea;
04
05    formato = new AudioFormat(8000,8,1,true,false);
06    infoLinea = new DataLine.Info(TargetDataLine.class,formato);
07
08    if (!AudioSystem.isLineSupported(infoLinea)){
09        System.out.println("No hay soporte para este tipo de linea");
10        System.exit(2);
11    }
12
13    try{
14        linea = (TargetDataLine) AudioSystem.getLine(infoLinea);
15        linea.open(formato,linea.getBufferSize());
16    }catch(LineUnavailableException es){
17        System.err.println("Linea no disponible");
18    }catch(SecurityException ex){
19        System.err.println("Error de seguridad: "+ex.toString());
```

```
20    }catch(Exception ex){
21        System.err.println("Error al abrir la linea: "+ex.toString());
22    }
```

En la linea `05` inicializamos un objeto `AudioFormat`, el constructor que usamos tiene el siguiente formato:

```
AudioFormat(float sampleRate, int sampleSizeInBits, int channels,
            boolean signed, boolean bigEndian)
```

Una vez que tenemos el formato deseado, verificamos si existe una linea disponible para captura de datos (`TargetDataLine`) con esta configuración (linea `08`). Si tenemos alguna linea del tipo solicitado usamos los métodos `open()` (linea `15`) y `read(byte[] b, int off, int len)` (linea `15`) que sirven para dejar la linea lista para empezar a recibir el audio y para leer el audio en el formato solicitado.

C

HERRAMIENTAS PARA EL MANEJO DE ARCHIVOS DE AUDIO

Para editar y manejar los archivos de audio utilicé dos programas: Audacity y Foobar2000. Ambos son programas libres y pueden encontrarse en :

- Foobar `http://www.foobar2000.org`
- Audacity `http://audacity.sourceforge.net/`

C.1 AUDACITY

Audacity es un programa de edición de archivos de audio, permite realizar muchas tareas básicas tales como captura, edición, conversión entre varios formatos de audio, cambios en la frecuencia de muestreo y en la escala, etc.. Además de contar con un menú de efectos bastante completo, eculización, eco, filtrado, etc., otra ventaja es que la interface es muy intuitiva.

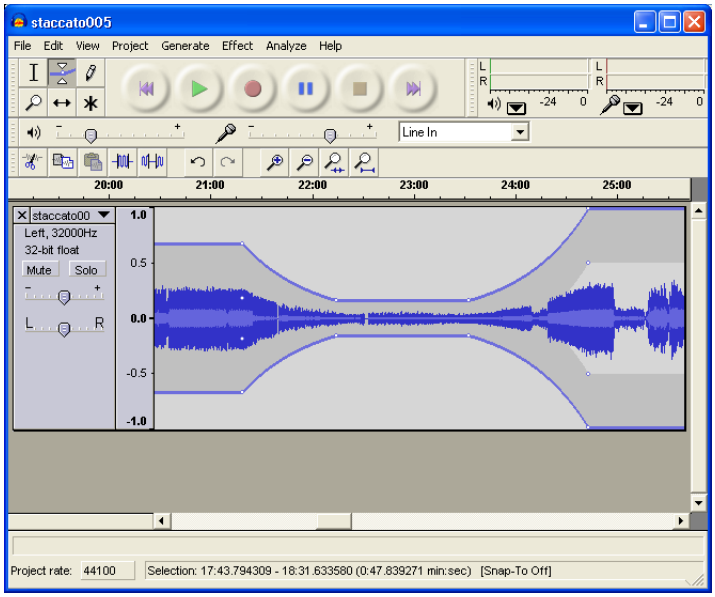


Figura 33. Audacity

C.2 FOOBAR2000

Foobar2000 es un programa que permite hacer varios tipos de procesamiento a una lista de archivos, basta agregar estos en la lista del reproducción del programa y seleccionarlos para hacerles conversiones. Tiene una interface de ecualización mas cómoda que el audacity pero menos flexible. Esta disponible en `http://audacity.sourceforge.net/`.

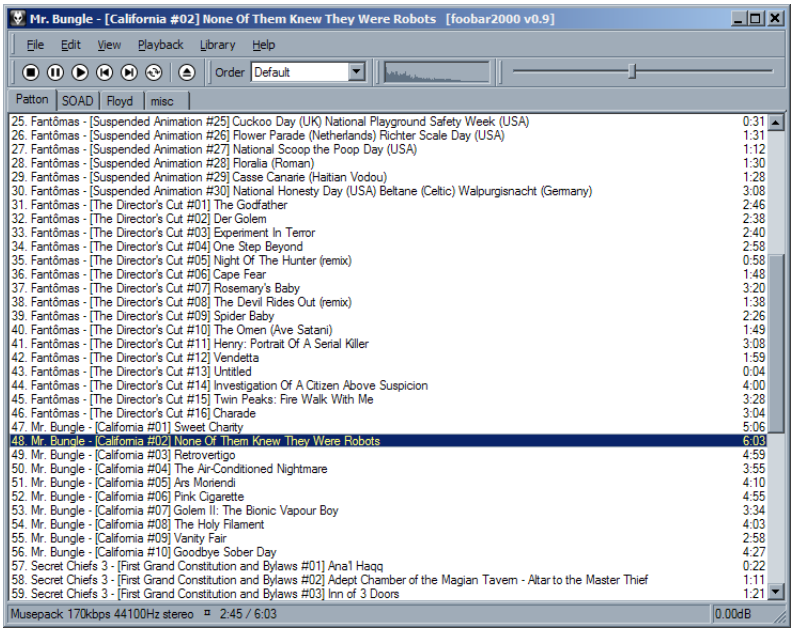


Figura 34. Foobar

C.3 SOX

Este programa es llamado la navaja suiza para manejar sonido, funciona en linea de comando. Con ella hicimos algunos procesamientos básicos como cambio de frecuencia, codificación gsm, y algunos efectos como promediar canales. Este se encuentra disponible en `http://sox.sourceforge.net/`.

D

AGENDA DE LA BASE DE AUDIO

A continuación presentamos la agenda corregida, hecha a partir de la agenda proporcionada por la empresa CONTACTO.

IFE_A = IFE Actualizate
IFE_Q = IFE Quiero Decirles
PAP = Papirolas UDG
CER = Cerveza Indio 1L $13.50
IBE_N = Ibedel Lic. Nutrición
IBE_D = Ibedel Maestria Admon.
LEM = Lemon One
PEP = Pepsi 1.5 L/ Papa
TEL = Telcel Nokia

Las 40 Principales (IFE_A, IFE_Q, PAP, CER, IBE_N, LEM, TEL)

IFE_A 7
IFE_Q 7
PAP 10
CER 16
IBE_N 1
LEM 12
TEL 4

Archivo	Comercial	Min:Sec	Sec	Sec. Com.
5-00-00				
6-00-00				
7-00-00	CER	24:13	1453	8654
	LEM	24:53	1493	8693
	TEL	25:14	1514	8714
	CER	41:02	2462	9663
	PAP	41:22	2482	9683
8-00-00	TEL	3:23	203	11004
	CER	26:47	1607	12409
	CER	39:17	2357	13158
	PAP	58:32	3512	14312
9-00-00	IFE_Q	31:51	1911	16313
	CER	52:20	3140	17541
10-00-00	CER	5:29	329	18331
	IFE_Q	9:23	563	18564
	CER	24:36	1476	19477
	IFE_A	27:33	1653	19651
	PAP	27:58	1678	19679
	CER	45:53	2753	20755
	IFE_Q	48:31	2911	20912
10-50-57	IFE_A	13:38	818	21828
11-16-54	CER	4:52	292	22486
	IFE_Q	8:56	536	22730
	PAP	9:45	585	22779
	CER	26:52	1612	23805
	IFE_Q	29:20	1760	23953
11-59-58	TEL	5:49	349	25126
	CER	25:15	1515	26292
	TEL	27:38	1658	26435
13-00-00	CER	2:35	155	28534
	CER	21:51	1311	29690
	CER	42:25	2545	30925
14-00-00	PAP	2:13	133	32113
	CER	40:57	2457	34437
	PAP	41:39	2499	34479
15-00-00	CER	0:58	58	35639
	LEM	3:01	181	35761
	LEM	22:42	1362	36943
	PAP	23:03	1383	36964
	LEM	42:54	2574	38154
16-00-00	IBE_N	1:27	87	39268
	LEM	2:08	128	39309
	LEM	21:44	1304	40484
	LEM	39:35	2375	41556
	LEM	40:38	2438	41619
	LEM	57:02	3422	42603
	PAP	59:17	3557	42738
17-00-00	LEM	17:19	1039	45043
	LEM	37:42	2262	46223
	LEM	57:22	3442	43820
18-00-00	PAP	0:08	8	46390
	IFE_Q	55:48	3348	49730
	PAP	56:47	3407	49790
19-00-00	IFE_A	15:34	934	50916
	IFE_Q	32:43	1963	51945
	IFE_A	57:46	3466	53448
20-00-00	IFE_A	33:27	2007	55590
21-00-00	IFE_A	1:56	116	57300
	IFE_A	24:38	1478	58662

IFE_A = IFE Actualizate
IFE_Q = IFE Quiero Decirles
PAP = Papirolas UDG
CER = Cerveza Indio 1L $13.50
IBE_N = Ibedel Lic. Nutrición
IBE_D = Ibedel Maestria Admon.
LEM = Lemon One
PEP = Pepsi 1.5 L/ Papa
TEL = Telcel Nokia

Ke Buena (IFE_A, IFE_Q, LEM, PEP, TEL)

IFE_A 7
IFE_Q 7
LEM 12
PEP 14
TEL 2

Archivo	Comercial	Min:Sec	Sec	Sec. Com.
5-58-52	IFE_A	17:15	1035	1034
	IFE_Q	50:03	3003	3003
6-59-59	IFE_A	0:49	49	3714
	LEM	17:07	1027	4693
	IFE_A	18:32	1112	4778
	PEP	30:19	1819	5485
	IFE_Q	32:03	1923	5589
	PEP	32:33	1953	5619
	TEL	45:39	2739	6405
	IFE_A	47:01	2821	6487
	IFE_Q	59:42	3582	7248
8-00-00				
8-12-28	PEP	19:05	1145	9091
	IFE_Q	21:26	1286	9232
	IFE_A	37:08	2228	10174
	PEP	45:07	2707	10653
8-59-59	LEM	7:12	432	11230
	IFE_Q	22:19	1339	12137
	PEP	23:08	1388	12188
	PEP	33:57	2037	12835
	IFE_Q	49:16	2956	13755
10-00-00	LEM	0:09	9	14407
	PEP	31:59	1919	16318
	IFE_A	32:19	1939	16337
	LEM	51:42	3102	17501
11-00-00	PEP	4:17	257	18256
	PEP	37:47	2267	20265
	LEM	39:58	2398	20397
	PEP	53:48	3228	21228
	PEP	56:37	3397	21396
12-00-00	PEP	21:18	1278	22877
	LEM	36:49	2209	23808
13-00-00	LEM	1:29	89	25288
	TEL	2:30	150	25350
	PEP	19:10	1150	26350
14-00-00	PEP	20:14	1214	30015
15-00-00	LEM	2:48	168	32568
	LEM	58:35	3515	35916
16-00-00	LEM	57:16	3436	39437
17-00-00	LEM	46:51	2811	42413
18-00-00	LEM	0:31	31	43233
19-00-00				
20-00-00				
21-00-00				
22-00-00				
23-00-00	IFE_A	17:53	1073	62277
	IFE_Q	41:51	2511	63716

IFE_A = IFE Actualizate
IFE_Q = IFE Quiero Decirles
PAP = Papirolas UDG
CER = Cerveza Indio 1L $13.50
IBE_N = Ibedel Lic. Nutrición
IBE_D = Ibedel Maestria Admon.
LEM = Lemon One
PEP = Pepsi 1.5 L/ Papa
TEL = Telcel Nokia

Super RMX (IFE_A, IFE_Q, CER, IBE_N, PEP)

IFE_A 6
IFE_Q 6
CER 15
IBE_N 2
PEP 13

Archivo	Comercial	Min:Sec	Sec	Sec. Com.
5-00-00				
6-00-00	IFE_Q	49:21	2961	6561
7-00-00	IFE_A	31:36	1896	9096
	PEP	42:49	2569	9770
8-00-00	CER	14:26	866	11668
	IBE_N	31:20	1880	12681
	IFE_Q	32:28	1948	12749
	CER	56:20	3380	14181
	PEP	56:40	3400	14201
9-00-00	CER	46:00	2760	17161
	PEP	53:47	3227	17628
	IFE_A	55:29	3329	17730
10-00-01	CER	9:54	594	18597
	PEP	10:14	614	18617
	IFE_Q	30:47	1847	19851
	CER	31:17	1877	19880
	CER	44:28	2668	20671
10-50-57	CER	3:06	186	21197
11-16-54	CER	15:39	939	23134
	CER	29:25	1765	23961
	IFE_A	40:50	2450	24645
11-59-58	CER	23:38	1418	26196
	CER	57:07	3427	28206
	PEP	57:28	3448	28226
13-00-00	CER	14:43	883	29263
	PEP	33:21	2001	30382
	CER	48:54	2934	31315
14-00-00	CER	27:22	1642	33623
	CER	40:10	2410	34391
	PEP	55:18	3318	35299
15-00-00	IBE_N	50:43	3043	38624
	IFE_Q	52:26	3146	38727
16-00-00	IFE_A	18:15	1095	40277
	PEP	51:10	3070	42251
17-00-00	IFE_Q	29:42	782	45381
	PEP	12:30	750	44565
18-00-00	PEP	12:30	750	47133
	IFE_A	13:10	790	47173
	PEP	27:34	1654	48037
19-00-00	PEP	13:51	831	50815
	PEP	29:45	1785	51768
	IFE_Q	43:49	2629	52613
20-00-00	IFE_A	14:59	899	54482
21-00-00				
22-00-00	IFE_Q	54:04	3244	64024

IFE_A = IFE Actualizate
IFE_Q = IFE Quiero Decirles
PAP = Papirolas UDG
CER = Cerveza Indio 1L $13.50
IBE_N = Ibedel Lic. Nutrición
IBE_D = Ibedel Maestria Admon.
LEM = Lemon One
PEP = Pepsi 1.5 L/ Papa
TEL = Telcel Nokia

Nueva Amor (IFE_A, PAP, IBE_N, LEM, PEP)

IFE_A 13
PAP 8
LEM 14
IBE_N 1
PEP 14

Archivo	Comercial	Min:Sec	Sec	Sec. Com.
5-00-00				
6-00-00	IFE_A	22:39	1359	4959
	PAP	45:50	2750	6350
7-00-00	IFE_A	24:13	1453	8773
	LEM	24:53	1493	9831
	PEP	25:14	1514	8714
8-00-00	PEP	4:16	256	11057
	PEP	23:03	1383	12184
	LEM	25:26	1526	12327
	IFE_A	25:47	1547	12348
9-00-00	PEP	24:09	1449	15850
	IFE_A	26:11	1571	15973
	LEM	40:37	2437	16838
10-00-00	PAP	2:36	156	18158
	IFE_A	23:10	1390	19393
	IBE_N	39:01	2341	20344
	PAP	40:02	2402	20405
	LEM	41:26	2486	20488
	PEP	42:06	2526	20528
10-50-57				
11-16-54	LEM	3:10	190	22385
	PEP	5:03	303	22498
	IFE_A	25:39	1539	23734
11-59-58	LEM	21:37	1297	26075
13-00-00	PEP	21:45	1305	29685
	LEM	23:57	1437	29816
14-00-00	PEP	23:25	1405	33385
	LEM	24:27	1467	33447
15-00-00				
16-00-00	IFE_A	2:30	150	39332
	PAP	22:20	1340	40522
	LEM	41:52	2512	41693
	PAP	43:23	2603	41784
	IFE_A	43:43	2623	41804
17-00-00	PEP	1:33	93	42875
	LEM	24:05	1445	44227
	IFE_A	26:07	1567	44349
18-00-00	LEM	3:06	186	46569
	PAP	4:23	263	46645
	PEP	4:44	284	46666
	PEP	42:24	2544	48926
	LEM	43:20	2600	48982
19-00-00	PEP	2:06	126	50109
	IFE_A	5:03	303	50287
	IFE_A	24:47	1487	51470
	PEP	41:03	2463	52446
20-00-00	LEM	19:04	1144	54727
	PEP	37:46	2266	55849
	LEM	39:46	2386	55970
	IFE_A	40:29	2429	56012
	PAP	58:56	3536	57119
21-00-00				
22-00-00	PAP	31:24	1884	62668
	IFE_A	34:03	2043	62828

IFE_A = IFE Actualizate
IFE_Q = IFE Quiero Decirles
PAP = Papirolas UDG
CER = Cerveza Indio 1L $13.50
IBE_N = Ibedel Lic. Nutrición
IBE_D = Ibedel Maestria Admon.
LEM = Lemon One
PEP = Pepsi 1.5 L/ Papa
TEL = Telcel Nokia

Maxima (IFE_A, IFE_Q, CER, IBE_D, IBE_A, PEP)

IFE_A 7
IFE_Q 7
CER 5
IBE_D 1
PEP 14
IBE_A 1

Archivo	Comercial	Min:Sec	Sec	Sec. Com.
5-58-52	IFE_A	44:05	2645	2645
	PEP	49:13	2953	2953
	IFE_Q	58:19	3499	3499
6-59-59	PEP	26:24	1584	5250
	PEP	47:04	2824	6489
8-00-00	IFE_A	3:14	194	7461
8-12-26				
9-00-05	IFE_Q	49:52	2992	13791
10-00-00	IBE_D	2:21	141	14538
	IFE_A	8:04	484	14880
11-00-00	IFE_A	8:40	520	18518
	IBE_A	10:42	642	18640
12-00-00				
13-00-00				
14-00-00	PEP	31:52	1912	30711
15-00-00	IFE_Q	30:42	1842	34241
	PEP	50:28	3028	35428
16-00-00	PEP	1:54	114	36113
	PEP	16:16	976	36975
	IFE_Q	26:33	1593	37593
	PEP	44:36	2676	38675
	PEP	54:06	3246	39245
17-00-00	PEP	0:57	57	39658
	IFE_A	13:28	808	40408
	PEP	13:58	838	40438
	PEP	35:27	2127	41727
	PEP	42:19	2539	42139
18-00-00	IFE_Q	48:06	2886	46086
19-00-00	CER	0:12	12	46812
	PEP	0:52	52	46854
	IFE_Q	13:22	802	47603
	CER	45:22	2722	49523
20-00-00	IFE_A	1:46	106	50507
	CER	31:57	1917	55893
21-00-00	CER	31:30	1890	52318
22-00-00	CER	17:08	1028	58630
	IFE_A	31:26	1886	59489
23-00-00	IFE_Q	11:27	687	61890

BIBLIOGRAFÍA

[1] Kimura A. and *et. al*. Very quick audio searching : Introducing global pruning to the time-series active search. In *ICASSP 2001*, 2001. (Cited on page 35.)

[2] E. Allamanche, J. Herre, O. Helmuth, B. Frba, T. Kasten, and M. Cremer. Content-based identification of audio material using mpeg-7 low level description. In *Proceedings of the International Symposium of Music Information Retrieval*, 2001. URL citeseer.ist.psu.edu/allamanche01contentbased.html. (Cited on pages 35 y 36.)

[3] A. Baggers and F.J. Narcotic. *A first Course in Wavelets with Fourier Analysis*. Prentice Hall, Upper Saddle River, New Jersey, USA, 1st edition, 2001. ISBN 0130228095. (Cited on page 14.)

[4] Eloi Batlle, Jaume Masip, and Pedro Cano. System analysis and performance tuning for broadcast audio fingerprinting. In *6th Int. Conference on Digital Audio Effects*, London, UK, September 8-11 2003. (Cited on page 8.)

[5] J.F. Bercher and C. Vignat. Estimating the entropy of a signal with applications. *IEEE Transactions on Signal Processing*, 48(12):1687–1694, June 2000. URL citeseer.ist.psu.edu/253788.html. (Cited on page 38.)

[6] C. Burges, J. Platt, and S. Jana. Distortion discriminant analysis for audio fingerprinting. *IEEE Transactions in Speech and Audio Processing*, 11:165–174, March 2003. URL citeseer.ist.psu.edu/burges01distortion.html. (Cited on page 35.)

[7] Christopher J.C. Burges, John C. Platt, and Soumya Jana. Extracting noise-robust features from audio data. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 1021–1024, May 2002. URL citeseer.ist.psu.edu/burges02extracting.html. (Cited on page 35.)

[8] Antonio Camarena-Ibarrola and Edgar Chavez. On musical performances identification, entropy and string matching. In *Fifth Mexican International Conference on Artificial Intelligence 2006 (MICAI2006)*, November 2006. (Cited on pages 36 y 53.)

[9] Fragoulis D. and *et. al.* On the automated recognition of seriously distorted musical recordings. (Cited on pages 15, 35 y 36.)

[10] Battle E., Masip J., Guaus E., and Cano P. Scalability issues in an hmm-based audio fingerprint. In *IEEE International Conference on Multimedia and Expo*, pages 735–738, 2004. (Cited on page 36.)

[11] B. Oliveira *et. al.* Proceedings of the 11th brazilian symposium on multimedia and the web. Minas Gerais, Brazil, 2005. (Cited on page 37.)

[12] E. Allamanche *et. al.* Audio-id: Towards content-based identification of audio material. In *Audio Engineering Society 110th Convention*, Amsterdam, The Netherlands, May 2001. (Cited on pages 5 y 37.)

[13] E. Allamanche *et. al.* Robust modelling for song detection in broadcast audio. In *Audio Engineering Society 112th Convention*, Munich, Germany, May 2002. (Cited on pages 5 y 53.)

[14] Kurth *et. al.* Identification of highly distorted audio material for querying large scale data bases. In *Audio Engineering Society 112th Convention*, Munich, Germany, May 2002. (Cited on pages 35 y 36.)

[15] Emilia Gómez, Pedro Cano, Leandro de C.T. Gomes, Eloi Batlle, and Madeleine Bonnet. Mixed watermarking-fingerprinting approach for integrity verification of audio recordings. In *Proceedings of IEEE International Telecommunications Symposium. Natal, Brazil*, 2002. URL citeseer.ist.psu.edu/541093.html. (Cited on pages 6 y 8.)

[16] J. Haitsma and T. Kalker. A highly robust audio fingerprinting system. In *International Conference on Music Information Retrieval (ISMIR)*, 2002. (Cited on pages 5, 6, 32, 35, 36, 37, 49, 52 y 54.)

[17] A.C. Ibarrola and E. Chávez. A robust entropy-based audio-fingerprint. *IEEE International Conference on Multimedia and Expo (ICME2006)*, July 2006. (Cited on pages 8, 9, 36, 38, 40, 42 y 52.)

[18] S. Kim and Chang D. Yoo. Boosted binary audio fingerprint based on spectral subband moments. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2007. (Cited on pages 35 y 36.)

[19] Frank Kurth and Roman Scherzer. Robust real time identification of pcm audio sources. In *Audio Engineering Society 114th Convention*, Amsterdam, The Netherlands, March 2003. (Cited on pages 5, 8, 37 y 46.)

[20] F. Mapelli, R. Pezzano, and R. Lancini. Robust audio fingerprinting for song identification. In *EURASIP 2004*, 2004. (Cited on page 5.)

[21] Sun MicroSystems. Java sound programmers guide, 2005. URL http://java.sun.com/j2se/1.5.0/docs/guide/sound/programmer_guide/contents.html. (Cited on page 69.)

[22] M. Miller, M. Rodriguez, and I. Cox. Audio fingerprinting: nearest neighbor search in high dimensional binary spaces. In *IEEE Workshop on Multimedia Signal Processing*, pages 182–185, 2002. (Cited on page 37.)

[23] Taiga Nakamura, Ryuki Tachinaba, and Seiji Kobayashi. Automatic music monitoring and boundary detection for broadcast using audio watermarking. In *Proceedings of Security and Watermarking of Multimedia Contents IV, SPIE*, pages 170–180, 2002 1996. URL citeseer.ist.psu.edu/684728.html. (Cited on page 8.)

[24] Cano P., Kaltenbrunner M., Mayor O., and Batlle E. Statistical significance in song-spotting in audio. In *International Symposium on Music Information Retrieval (MUSIC IR 2001)*, 2001. (Cited on page 32.)

[25] Cano P., Gómez E., Batlle E., Gomes L., and Bonnet M. Audio fingerprinting: Concepts and applications. In *International Conference on Fuzzy Systems Knowledge Discovery (FSKD 2002)*, 2002. (Cited on pages 31, 32, 36 y 37.)

[26] W. H. Press, S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery. *Numerical Recipes in C*. Cambridge, 2 edition, 1992. ISBN 0521431085. (Cited on page 15.)

[27] J. G. Proakis and D. G. Manolakis. *Digital Signal Processing: Principles, Algorithms and Applications*. Prentice Hall, 3 edition, 2005. ISBN 8120311299. (Cited on page 17.)

[28] J. Seo, M. Jin, D. Jang S. Lee, and C. Yoo S. Lee. Audio fingerprinting based on normalized spectral subband centroids. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2005. (Cited on page 35.)

[29] Alexander Sinitsyn. Duplicate song detection using audio fingerprint for consumer electronics devices. 2006. (Cited on pages 37 y 49.)

[30] Tritonus, 2005. URL `http://tritonus.org/`. (Cited on page 70.)

[31] Cheng Yang. Efficient acoustic index for music retrieval with various degrees of similarity. In *MULTIMEDIA 02: Proceedings of the tenth ACM international conference on Multimedia*, pages 584–591, New York, NY, USA, 2002. ACM Press. ISBN 1-58113-620-X. (Cited on page 37.)

COLOFÓN

Esta tesis fue escrita con el sistema LaTeX 2_ε usando las fuentes *Palatino* y *Euler* de Hermann Zapf. El estilo es la platilla *ClassicThesis* de Andre Miéde.