Trabajo Práctico Especial 3: Reconocimiento de audio mediante huellas digitales acústicas

1 Introducción

Imaginen la siguiente situación: se encuentran en un bar tomando algo, inmersos en esa interesante conversación cuando de repente... "pará pará, qué es eso que suena?" Entre el ruido general alcanzan a distinguir esa canción que no escuchaban en tanto tiempo, que habían obsesivamente intentado encontrar pero que, a pesar de ello, nunca llegaron a dar siquiera con el nombre del intérprete. Su corazón se estremece... ahora que la tienen ahí, sonando nuevamente, la situación es bien diferente a la de hace algunos años: toman su teléfono celular, abren alguna de las aplicaciones de reconocimiento de audio que tienen y, en cuestión de segundos, aparece la información completa del tema, la banda, e incluso se atreve a sugerir artistas similares!

Si el protagonista de esta ficción fuera estudiante de la FIUBA, no debería extrañarnos que surjan en él varios interrogantes: ¿cómo hizo el programa para reconocer la canción escuchando sólo 10 segundos?, ¿cómo puede funcionar con el ruido de un bar de fondo? y ¿cómo puede ser que además identifique tan rápido a esta banda que nadie conoce? Resulta que todo este proceso se basa en reconocer huellas digitales acústicas, un técnica robusta y eficiente que puede entenderse en términos de Señales y Sistemas.

2 Reconocimiento mediante huellas digitales acústicas

El reconocimiento de audio mediante huellas digitales acústicas es una técnica que busca identificar una pieza de audio, contrastando la información contenida en dicha pieza contra la almacenada en una base de datos de pistas conocidas.

Existen varias formas de abordar el problema del reconocimiento, pero todas persiguen la misma finalidad:

1. Simplicidad computacional: el reconocimiento debe realizarse en forma rápida.
2. Ser eficiente en el uso de memoria y poseer buena capacidad de discriminación (existen aproximadamente 100 millones de canciones grabadas).
3. Ser robusto ante distorsiones: ruido de fondo, degradación por compresión digital del audio, ecualización por respuesta en frecuencia no plana del lugar y/o parlantes, etc.
4. Granularidad: capacidad de reconocimiento utilizando solo un segmento del audio.
5. Alta tasa de aciertos, baja tasa de falsos positivos y de rechazos.

El procedimiento básico consiste en analizar el segmento de audio, buscando extraer características particulares en el espacio tiempo-frecuencia (es decir, en su espectrograma) que sirvan para identificarlo luego. Una característica podría ser, por ejemplo, la potencia que posean las diferentes bandas de frecuencias. Al conjunto de estas características se las denomina huella digital acústica, ya que es análogo a como una huella dactilar se identifica por las convergencias, desviaciones, interrupciones y demás particularidades de las crestas papilares.

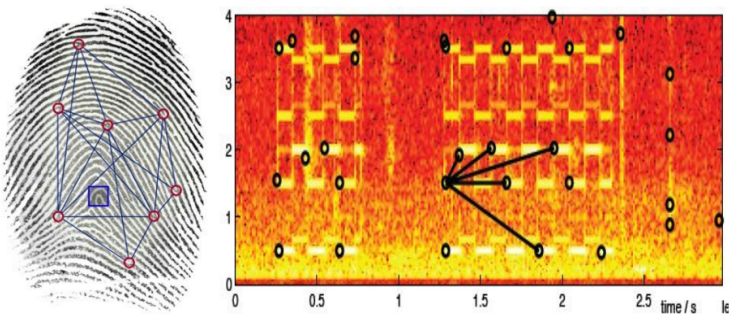

Figure 1: Analogía entre una huella digital tradicional y una huella digital acústica

Este procedimiento de extracción de huellas se utiliza tanto para confeccionar la base de datos de canciones conocidas como para posteriormente reconocer segmentos de audio sin identificar, consultando la base de datos. La elección de las características es la base del funcionamiento del método, ya que serán las claves que se utilizan luego para realizar la búsqueda en la base de datos. Por lo tanto, deben ser lo suficientemente específicas como para identificar a cada canción unívocamente, pero a la vez ocupar poca memoria para permitir realizar la búsqueda en forma rápida y eficiente. También deberán ser relativamente inmunes ante ruidos y distorsiones del audio, de manera de lograr un sistema de reconocimiento robusto.

Diferentes características han sido propuestas con el fin de lograr las especificaciones antes nombradas. Algunos ejemplos que se extraen del dominio tiempo-frecuencia son los MFCC (Mel-Frequency Cepstrum Coefficients, se utilizan para la representación del habla), SFM (Spectral Flatness Measure, es una estimación de cuánto se aproxima una banda espectral a ser un tono o ruido), los ubicación de los picos de potencia del espectrograma (es la base del algoritmo de Shazam), la energía de las bandas, y la lista continúa. Además del espectrograma, otras técnicas se han utilizado como wavelets y algoritmos de visión, algoritmos de machine learning y meta-características, útiles en grabaciones muy distorsionadas.

En este trabajo desarrollaremos una implementación que utiliza como características el signo de la diferencia de energía entre bandas, también conocido como algoritmo Philips. Cabe destacar que este y los algoritmos antes mencionados sólo reconocen las mismas versiones de las canciones que se almacenan en la base de datos es decir, no están diseñados para reconocer versiones en vivo o interpretaciones diferentes a la original.

3 Descripción del sistema de reconocimiento

El sistema de reconocimiento consta de dos bloques principales:

1. El algoritmo para extraer las huellas digitales acústicas.
2. La base de datos que almacena las huellas acústicas de las canciones conocidas.

El algoritmo para extraer las huellas acústicas toma como entrada el archivo con el audio y, luego de un pre-procesamiento de la señal, extrae las características y entrega como resultado la huella acústica. El proceso se esquematiza a continuación:

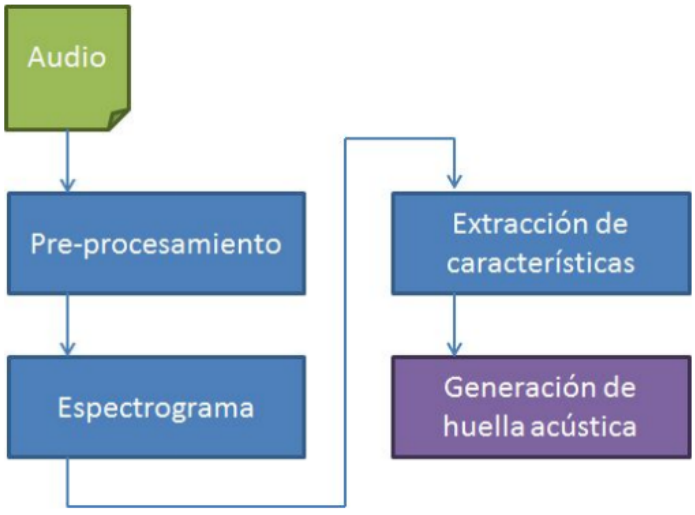

Figure 2: Esquema del algoritmo para extraer la huella digital acústica

La base de datos contiene las huellas digitales acústicas de las canciones conocidas. Tiene además un sistema de búsqueda (en inglés query) tal que al realizar un query con una huella acústica dada nos devuelve la canción más probable a la cual corresponde.

El esquema de este sistema se presenta en la figura 3.

4 Algoritmo de extracción de huellas digitales acústicas

El algoritmo de extracción de huellas digitales acústicas extrae características a partir del espectrograma del audio. El conjunto de estas características conforman la huella.

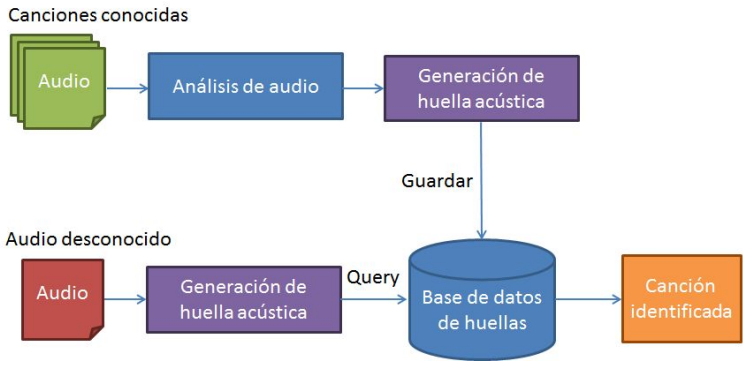

Figure 3: Esquema representando la generación de la base de datos de huellas acústicas y la consulta

Primeramente, se acondiciona el audio con un pre-procesamiento. Luego se realiza un espectrograma del audio pre-procesado y se extraen las características para cada intervalode tiempo.

Para nuestro algoritmo, las características se obtienen mediante una función signo F(m,n) que opera sobre la energía de las bandas E(m,n) según:

$$F(m,n) = \begin{cases} 1 & E(m,n) - E(n,m+1) > E(n-1,m) - E(n-1,m+1) \\ 0 & \text{en otro caso} \end{cases}$$

donde n es el tiempo y m la banda de frecuencia. Compruebe que F(m,n) es 1 si la diferencia de energía entre las bandas m y m+1 para el frame actual n es mayor a la del frame anterior n-1.

Experimentalmente, se verificó que mediante la función F(m,n) se obtienen características robustas ante varios tipos de procesamientos y distorsiones del audio.

5 Pre-procesamiento del audio

El pre-procesamiento del audio consiste en pasarlo a canal mono (si está en estéreo) y luego sub-muestrearlo, debido a que la información útil para la extracción de características se encuentra en bajas frecuencias. En general, el audio está muestreado a 44100 Hz, pero para este algoritmo basta con tenerlo muestreado a 1/8 de su frecuencia original, es decir 5512.5 Hz. Esto permite trabajar con menos muestras, aliviando la carga computacional.

Ejercicio 1: Implemente un sistema para reducir la frecuencia de muestreo del audio de 44100 Hz a 5512.5 Hz. Muestre diagrama en bloques y justifique su diseño. El mismo debe contener un pasabajos y un submuestreador. Indicar el tipo (FIR o IIR) y orden del filtro obtenido, diagrama de polos y ceros, respuesta en frecuencia y respuesta al impulso. Realice espetrogramas de la señal original y la filtrada por el pasabajos.

---

6 Extracción de características del espectrograma

Los parámetros con los que se realice el espectrograma tienen una influencia directa sobre la extracción de las características. La elección de la ventana (tipo y longitud) es el parámetro principal, ya que debería lograr una resolución razonable tanto en tiempo como en frecuencia, tal que permita distinguir las características espectrales del audio con claridad.

Ejercicio 2. Mediante la función fft, obtenga el espectro de una ventana rectangular y una de Hamming de igual duración y grafique su potencia en escala logarítmica en un mismo gráfico para compararlos. Cuál es la resolución en frecuencia de cada ventana? Al realizar un espectrograma, qué ventaja tiene utilizar la ventana de Hamming frente a la rectangular?

Ejercicio 3. Realice un espectrograma de la señal sub-muestreada y muestre una imagen (con zoom) de alguna región donde se vean las características espectrales de la señal dada la ventana utilizada. Muestre y

justifique cómo cambia la visualización de las características con 3 diferentes longitudes de ventana y comente qué longitud utilizará en el algoritmo. Realice las comparaciones con ventana rectangular y de Hamming.

A partir del espectrograma de la señal preprocesada, se procederá a aplicar la función F(m,n) a cada frame y extraer las características. El frame n se define como el vector que se forma al indexar la matriz del espectrograma en el índice n, es decir S(:,n). Las bandas de frecuencia se definirán con espaciamiento logarítmico (esto es similar a la escala psicoacústica Bark) debido a que el oído diferencia el espaciamiento entre tonos en forma logarítmica.

Ejercicio 4. Implemente el algoritmo de extracción de características. Divida al espectrograma en 21 bandas de frecuencia espaciadas logarítmicamente, con frecuencia inferior 300 Hz y frecuencia superior 2 kHz, y aplique la función F(m,n) sobre la potencia de cada banda para obtener la huella H(m,n). Ejecute el algoritmo sobre un segmento de audio y muestre una imagen de la huella digital acústica obtenida. Debido al efecto borde al aplicar F(n,m) la huella acústica debería ser una matriz de 20 filas.

---

7 Confección de la base de datos

Para confeccionar la base de datos, se utilizará como base el script de generar_DB que le fue suministrado. El mismo llamará secuencialmente al algoritmo que extrae las huellas digitales acústicas para ir completando la base de datos.

La base de datos es una tabla de 2^{20} filas y N columnas. Para cada frame de la huella acústica generamos un valor a guardar en la tabla. Cada valor es un entero de 32 bits, de los cuales los primeros 12 bits son el ID numérico de la canción que se está analizando, y los 20 restantes son el número del frame. La fila en la que se guarda cada valor se obtiene pasando a decimal el número binario conformado por las características del frame en cuestión, y la columna en la que se guarda se elige buscando la última columna vacía dentro de esa fila. Esto se esquematiza en la siguiente figura:

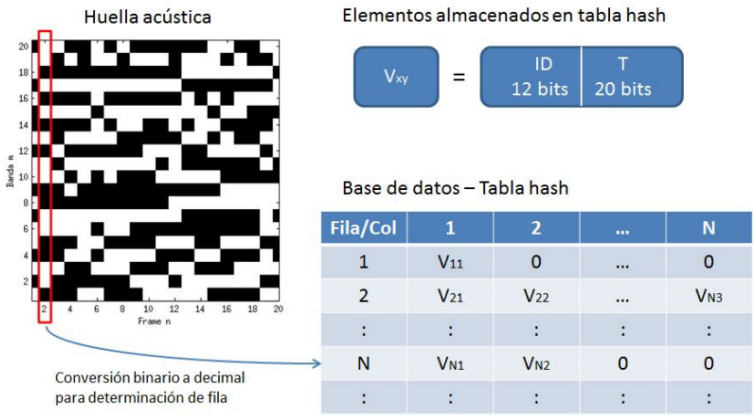

Figure 4: Indexación de tabla hash mediante huella acústica y representación de elementos almacenados. Para el frame número 2 (recuadrado en rojo) la característica en binario es 11010101011001100110. Pasando este número a decimal obtenemos la fila de la tabla en la que deberá guardarse el valor

Como una misma fila de la tabla puede contener valores asociados a distintos temas, o incluso a un mismo tema (la característica podría repetirse en frames distintos) la cantidad de columnas N deberá ser tal que la tabla posea espacio suficiente para la cantidad de canciones que se deseen almacenar. Para este trabajo, N=20 es suficiente. Si alguna fila de la tabla se completa y se desea ingresar un nuevo valor, se sobre-escribe algún valor anterior elegido al azar.

Ejercicio 5. Ejecute el script generar_DB para confeccionar la base de datos completa de su lista de canciones. Utilice al menos 40 canciones de su elección para llenar la base de datos. Observe que este script hace uso de la función guardar_huella para almacenar las huellas en la tabla hash tal como se explicó en esta sección.

8 Test del algoritmo

En esta etapa final, pondremos a prueba la eficacia del algoritmo completo de reconocimiento. El procedimiento consistirá en obtener el porcentaje de aciertos del método al reconocer segmentos de audio, los cuales serán sometidos a distintas distorsiones.

Para esto se suministra la función query_dB, la cual recibe como entradas la base de datos y la huella acústica del segmento de audio a reconocer, y devuelve los primeros 5 IDs de las canciones que mejor coinciden en orden de prioridad descendente.

Ejercicio 6. Evalúe la tasa de aciertos del algoritmo identificando segmentos de duración T con tiempo inicial elegido al azar de canciones elegidas al azar. Las canciones deberán ser las mismas que utilizó para confeccionar la base de datos. Realice la evaluación para 50 segmentos con duración T entre 5, 10 y 20 segundos, obteniendo la tasa de aciertos en cada caso.

Ejercicio 7. Reproduzca y grabe con la PC o el celular un segmento de alguna de las canciones e intente reconocerla con su algoritmo. Comente los resultados obtenidos así como las condiciones de ruido de fondo y los dispositivos utilizados.

Ejercicio 8. Repita el ejercicio 6 sumando ruido a los segmentos de audio. Evalúe tasa de aciertos para SNR 0dB, 10dB y 20dB, mostrando sus resultados en una tabla para las 9 combinaciones de longitud temporal y ruido.

$$SNR[dB] = 10log10(Px/Pn)$$

donde Px es la potencia de la señal y Pn la potencia del ruido.
