

# Detector de audio duplicado

Andrés Gilli, Paez Mariángeles

Trabajo práctico final de “Muestreo y procesamiento digital”, II-FICH-UNL.

**Resumen—** Actualmente tenemos acceso a una cantidad de archivos musicales, discos, canciones sueltas, grabaciones, que ocupan en nuestro disco mucho más espacio del que imaginamos.

Para poder ganar espacio y orden en nuestra pc, diseñamos este programa que basándose en procesamiento digital de señales de audio, para comparar unas canciones con otras, detectamos si tenemos archivos de audio duplicados en nuestra máquina.

**Palabras clave—**canciones repetidas, análisis de señales de audio similares, mp3 .

## I. INTRODUCCIÓN

**B**ásicamente se implementó un prototipo de software que por medio de su utilización permite realizar la búsqueda de archivos duplicados de audio que se encuentren en formato mp3. La aplicación cuenta con un desarrollo básico en Matlab y consta de cinco módulos: obtención de la señal de audio desde una ubicación específica, lectura de los archivos, extracción de características necesarias para el análisis utilizando el método de codificación por predicción lineal (LPC) , comparación de características y obtenemos el listado de las canciones duplicadas.

El documento que se presenta pretende exponer lo que se puede realizar al finalizar el cursado de la materia Procesamiento Digital de Señales de la Facultad de Ingeniería y Ciencias Hídricas de la Universidad Nacional del Litoral.

En la sección II y III se presentan los aspectos generales teóricos pertinentes a cada una de las etapas en la detección de audio duplicado y se exponen las generalidades sobre la implementación y puesta en marcha.

## II. ASPECTOS GENERALES

En el contexto del análisis de señales de audio, es necesario captar lo que corresponde para realizar una serie de etapas para adecuar la señal para que pueda ser comparada con otras señales.

Para llevar a cabo nuestro proyecto implementamos las funciones que se ven en los esquemas adjuntos presentados con el trabajo.

## III. IMPLEMENTACIÓN

La aplicación se desarrolló utilizando Matlab 7, y para poder analizar archivos mp3 necesitamos anexarle el SOX; que es una utilidad de línea de comandos que puede convertir varios formatos de archivos de audio de ordenador en otros formatos. También se puede aplicar varios efectos a estos archivos de sonido, y, como un bono adicional, SOX puede reproducir y grabar archivos de audio en la mayoría de las plataformas.

En nuestro caso al sox lo utilizamos para convertir cada archivo mp3 en wav, para poder leerlo desde Matlab con el comando wavread, también lo empleamos para bajar la frecuencia de la señal a 11khz, que es a la mínima frecuencia en que no se pierden datos de las canciones; para tener todas las señales en la misma frecuencia, para poder compararlas, y también nos permitió transformar a las canciones de dos canales (estéreo) en un solo canal (mono), para que la comparación sea con una señal por canción.

Al bajar la frecuencia no se observan pérdidas de información fundamental. Vemos un ejemplo:

En el dominio temporal

$F_m = 22 \text{ KHz}$

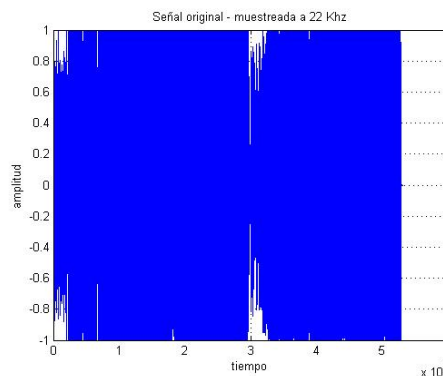


Fig. 1: Señal original de 22khz

$F_m = 11 \text{ kHz}$

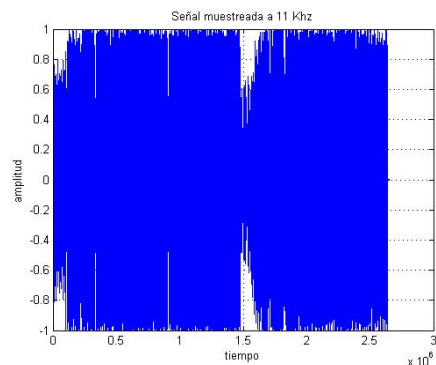


Fig. 2: Señal modificada a 11Khz

A continuación graficamos como es la respuesta en frecuencias de estas señales.

$F_m = 22 \text{ KHz}$

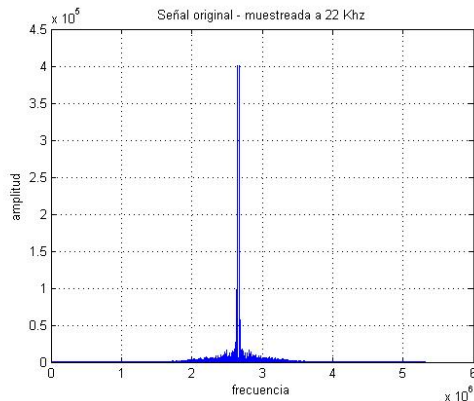


Fig. 3: Transformada de Fourier de la señal original

$F_m = 11 \text{ KHz}$

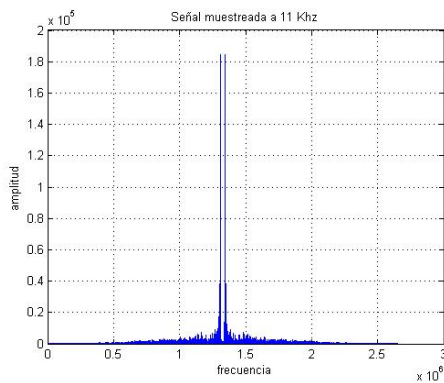


Fig. 4: Transformada de Fourier de la señal modificada.

Aquí se puede observar más claramente que no se pierde la información fundamental.

Tomamos los 2 minutos centrales de cada canción y calculamos las envolventes de las señales a analizar, en el dominio frecuencial y en el dominio temporal

La siguiente grafica es de la señal en los dos minutos utilizados para el análisis

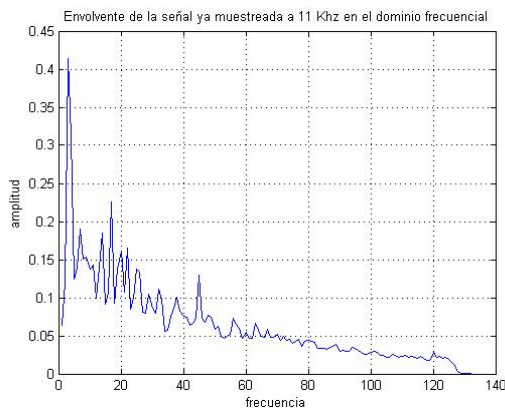


Fig. 5: Espectro frecuencial de la envolvente de la señal de 2 minutos

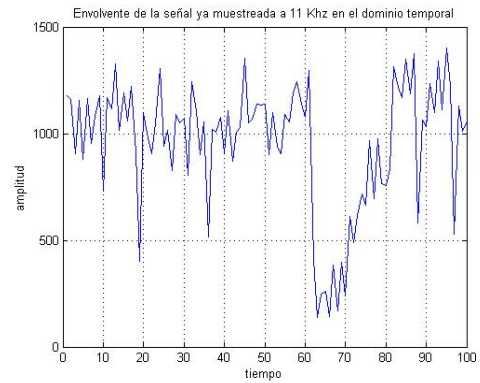


Fig. 6: Espectro temporal de la envolvente de la señal de 2 minutos

Podemos notar que hay poca variación entre una y otra por lo que nos permite reducir en gran parte la cantidad de muestras de las señales para poder analizarlas.

Para decidir si dos canciones son similares o no utilizamos dos índices que nos permitirán tomar la decisión final.

Para el primero necesitamos obtener la energía (1) de la señal por ventanas. De esta forma obtenemos una envolvente a la señal en el dominio temporal.

$$E = |x|_2^2 = \sum_{n=1}^N |x_n|^2 \quad (1)$$

Luego, calculamos la correlación cruzada (2) entre la envolvente en el dominio temporal normalizada del archivo que necesitamos comparar y las envolventes en el dominio temporal del resto de los archivos. Tomamos el valor máximo de las correlaciones para decidir si son parecidas o no.

$$R_{xy}(i) = \sum_{i=1}^N x(i) y(n+i)^* \quad (2)$$

Mediante múltiples pruebas, se definió que el valor de la correlación, cuando sean similares, estará entre un 0,9 y un 1,1 por ciento del valor de la norma del archivo con el cual estamos comparando. Estos índices incluyen un margen de error, ya que estamos buscando archivos parecidos.

Esto hace que en escasas oportunidades se detecten archivos como parecidos de manera errónea.

Sin embargo, este problema se soluciona con el uso de un segundo índice a la hora de decidir si las canciones son similares o no.

Para ello se utilizó la función LPC (Linear Predictor Coefficients)(3). Esta nos brinda los coeficientes de aproximación lineal que mejor representan la señal para un orden dado. Luego calculamos la respuesta en frecuencias de estos coeficientes. Esto nos genera una envolvente a la señal en el dominio frecuencial.

$$\bar{x} = -a(2)x(n-1) - a(3)x(n-2) - \dots - a(p+1)x(n-p) \quad (3)$$

Luego las comparamos mediante el producto interno (4). Si son similares el resultado estará muy próximo a 1 (ya que las normalizamos previamente).

$$\langle x, y \rangle = \sum_{i=1}^N x(i) y(i)^* \quad (4)$$

Una vez que obtenemos los resultados del producto interno y la correlación cruzada, empezamos a detectar si son duplicados o no los archivos.

Cuando tenemos todos los archivos detectados se genera un documento txt, con todos las rutas y los nombres de los archivos duplicados si hubiera.

#### IV. CONCLUSIONES

El detector de audio duplicado, pudo detectar los archivos que se encontraban repetidos, lo hemos probado con varias bases de datos, con diferentes canciones y cantidades de ellas. También hicimos el testeo en varias computadoras con memorias de 2gb, 3gb y 4gb y distintos micros, en la de 2gb con un micro i3, tuvo problemas de memoria, al leer una cantidad grande de archivos.

Comparamos nuestro programa con otros ya existentes en el mercado, y chequeamos que los otros, son más rápidos para analizar los datos, pero no detectan la totalidad de archivos duplicados, lo que sí hace el nuestro. Una prueba fue analizar aproximadamente 600 archivos, de los cuales 85 duplicados detectó el otro programa y 100 archivos el nuestro, según analizamos los otros programas se basaban en comparar solamente el nombre de las canciones y el nuestro compara la señal de audio.

Cómo mejora futura para nuestro programa, tendríamos que implementarlo en otro lenguaje para aumentar velocidad, ya que Matlab es un lenguaje interpretado y hace que sea lento operativamente.

Para visualizar alguno de los resultados obtenidos en las pruebas realizadas:

```

|Aguarde mientras se procesan los datos...
|Se analizaron 255 archivos en 2284.699 segundos
|Tasa de Análisis: 8.9596 segundos por archivo
|Se detectaron 22 Archivos duplicados
|
|Lista =
|
|C:\mp3_2\02 Bienvenidos (2).mp3
|C:\mp3_2\02 Bienvenidos.mp3
|
|C:\mp3_2\02-Fergie-cLumsy-1ncomp.mp3
|C:\mp3_2\02-FERGIE\02-Fergie-cLumsy.mp3
|
|C:\mp3_2\04 Hombre paranoico (2).mp3
|C:\mp3_2\04 Hombre paranoico.mp3
|
|C:\mp3_2\07 Todo eso.mp3
|C:\mp3_2\07 Todo eso2.mp3
|
|C:\mp3_2\besos besitos completo.mp3
|C:\mp3_2\Cumbia\Grupo Alegria\Toda una vida (2008)\13 - BESOS BESITOS.mp3
|
|C:\mp3_2\Canciones - copia.mp3
|C:\mp3_2\Canciones.mp3
|
|C:\mp3_2\Comida Chatarra (2).mp3
|C:\mp3_2\Comida Chatarra.mp3
|
|C:\mp3_2\Flor marchita.mp3
|C:\mp3_2\Cumbia\Calif\El poder de la cumbia\Flor marchita.mp3
|
|C:\mp3_2\Beyonce\Beyonce_-_crazy_in_love - copia.mp3
|C:\mp3_2\Beyonce\Beyonce_-_crazy_in_love.mp3
|
|C:\mp3_2\Beyonce\Beyonce_-_hip_hop_star - copia.mp3
|C:\mp3_2\Beyonce\Beyonce_-_hip_hop_star.mp3
|
|C:\mp3_2\Beyonce\Beyonce_-_irreplaceable - copia.mp3
|C:\mp3_2\Beyonce\Beyonce_-_irreplaceable.mp3

```

#### REFERENCIAS

- [1] Wickert, M.A.; , "High quality digital audio experiments for a signal processing first course," Digital Signal Processing Workshop and IEEE Signal Processing Education Workshop (DSP/SPE), 2011 IEEE , vol., no., pp.174-179, 4-7 Jan. 2011  
doi: 10.1109/DSP-SPE.2011.5739207  
URL: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5739207&isnumber=5739176>
- [2] Serra, J.; Gomez, E.; , "Audio cover song identification based on tonal sequence alignment," Acoustics, Speech and Signal

Processing, 2008. ICASSP 2008. IEEE International Conference on , vol., no., pp.61-64, March 31 2008-April 4 2008  
doi: 10.1109/ICASSP.2008.4517546  
URL: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4517546&isnumber=4517521>

- [3] Yi Yu; Downie, J.S.; Moerchen, F.; Lei Chen; Joe, K.; , "Using Exact Locality Sensitive Mapping to Group and Detect Audio-Based Cover Songs," Multimedia, 2008. ISM 2008. Tenth IEEE International Symposium on , vol., no., pp.302-309, 15-17 Dec. 2008  
doi: 10.1109/ISM.2008.18  
URL: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4741184&isnumber=4741128>
- [4] Burges, C.J.C.; Plastina, D.; Platt, J.C.; Renshaw, E.; Malvar, H.S.; , "Using audio fingerprinting for duplicate detection and thumbnail generation," Acoustics, Speech, and Signal Processing, 2005. Proceedings. (ICASSP '05). IEEE International Conference on , vol.3, no., pp. iii/9- iii/12 Vol. 3, 18-23 March 2005  
doi: 10.1109/ICASSP.2005.1415633  
URL: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1415633&isnumber=30652>
- [5] Sinitsyn, A.; , "Duplicate Song Detection using Audio Fingerprinting for Consumer Electronics Devices," Consumer Electronics, 2006. ISCE '06. 2006 IEEE Tenth International Symposium on , vol., no., pp.1-6, 0-0 0  
doi: 10.1109/ISCE.2006.1689403  
URL: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1689403&isnumber=35627>