B E N E M É R I T A

UNIVERSIDAD AUTÓNOMA DE PUEBLA

FACULTAD DE CIENCIAS DE LA COMPUTACIÓN

EXTRACCIÓN DE CARACTERÍSTICAS DE AUDIO MEDIANTE HUELLA DIGITAL.

TESIS

QUE PARA OBTENER EL TÍTULO DE:

LICENCIADO EN INGENIERÍA EN CIENCIAS DE LA COMPUTACIÓN

PRESENTA:

DAVID CARCAÑO VENTURA

ASESOR:

DR. JOSÉ ARTURO OLVERA LÓPEZ

PUEBLA, PUEBLA.

AGOSTO 2019

AGRADECIMIENTOS

A Dios por su amor y la oportunidad de vivir esta vida, ya que muchas cosas que viví me ayudaron a terminar esta tesis y las demás poco a poco se unirán al rompecabezas.

A mi familia por su apoyo incondicional en todos los sentidos. A mi papá porque nunca dudo de mi, creyó en mí incluso cuando yo no lo hice; a mi mamá por su paciencia en todo este año y permitirme seguir practicando todo lo que me gusta y a mi hermana porque a pesar de la lejanía supo ayudarme de distintas maneras, y eternamente estaré agradecido con ella.

A mi asesor el Dr. Arturo Olvera por su paciencia porque llegué sin saber nada, pero aprendí mucho sobre lo que es y cómo hacer una tesis. A la Dra. Liliana Mantilla por sus conocimientos que me transmitió durante la carrera y también por su paciencia en el salón de clases, sin ella las materias de hardware no hubieran sido divertidas. Al Dr. Iván Olmos porque gracias a él conocí las diferentes aplicaciones de la derivada y otros temas de las matemáticas en el procesamiento de imágenes digitales; al ser las imágenes y los audios ondas, decidí realizar una tesis en el área procesamiento de audio digital.

Un especial agradecimiento a mi abuelita Anita, que después de un año y medio de que está en Casa, cumplí lo que en sus últimas palabras me dijo: “Termina la Universidad”.

A mi tía Rosi, mi tío Gil y sus hijos(Beto y Ricardo), por todo su apoyo y por recibirme 4 años y medio en su casa, gracias por aguantarme en mis desveladas, mi carácter y mis malos hábitos como dejar prendido el boiler; vivir conmigo no es fácil.

A mi tío Mario, tía Mary y sus hijos(Samuel, Mario y Cindy), por su apoyo y también por aguantarme con mi carácter. Gracias por permitirme jugar fútbol con ustedes, el haber ganado mi único torneo no es un gran logro como el poder haber jugado los domingos en la mañana.

A mis abuelitos Juanita y Ramón porque sus consejos me siguieron desde niño hasta el día de hoy, y porque por ustedes hoy estamos aquí, a mis tíos(Benito, Joel, Nefta, Rut, Lea, Arme) y sus familias por sus palabras que me dieron, me ayudaron a seguir adelante, y a mis primos porque aunque hemos tomado caminos diferentes, estamos donde debemos estar.

A mi tía Mayra y mi tío Claudio por su apoyo y por permitirme quedar un tiempo en su casa; sé que no fui el mejor inquilino, pero estaré eternamente agradecido porque lo que hicieron me ayudo a seguir encariñándome con la carrera.

A mi tío Coco y a su familia, porque me recibieron una semana en su casa; fueron mi familia por esa semana y me hicieron sentir como en casa. Aunque no me quede a vivir con ustedes, les agradezco su apoyo en mi decisión y su amor que a la fecha sigo recibiendo de ustedes.

A todos mis profesores que me dieron clases, en especial a la Dra. Rosa García, al Dr. Marco González, a mi tutor Dr. Héctor Ramírez porque me hicieron apasionarme por el conocimiento, por encontrar más aplicaciones de las matemáticas, también porque me permitieron faltar a las clases por el futbol y porque no renunciaron a mi a pesar de dormirme en sus clases. Un especial agradecimiento al Mtro. Héctor Abelleyra por guiarme en mi camino de la preparatoria, pues junto con el Prof. Fernando Contreras me motivaron a estudiar Ciencias de la Computación.

Por último a todos mis amigos de la universidad gracias por acompañarme en este tramo de la vida. A mis amigos del módulo(Ivonne, Brian, Erick, Chino, Cuama, Cuauhtle, Jefaso, Luis, Emilio, Josué, Santiago, Steve) por los juegos de futbol, los juegos de mesa, las comidas, las carcajadas, los trabajos en equipo, por todo; A mis amigos de las canchas, los gordibuenos(Domi, Kiko, Heber, Crack, Ramón, Migue, Lalo, War, Capi, Román, Miguel, y todos los demás del cubo), Ely mi mejor amiga, Kiko por ser mi compañero de camión y mi mejor amigo de la universidad, e Ilse gracias por tu amistad. De todos me llevo grandes recuerdos que no pienso ni quiero olvidar, y bien lo saben. Gracias porque hicieron de este tramo de vida algo increíble.

En general gracias a todos, lo que hicieron en mi es algo que no se imaginan, no hay palabras, ni hojas para expresar todo lo que siento en estos momentos. Que Dios los bendiga siempre y no les diré que estarán orgullosos de mí, porque deberían estarlo y no por lo que he logrado. Deben estar orgullosos porque hasta el día de hoy soy feliz con las decisiones que he tomado, quizá me he arrepentido en algún momento, pero me encanta lo que he hecho. Bien dicen que venía a la universidad a jugar, a dormir y por último a estudiar. Lo que no saben es que jugar me hacía ser un mejor estudiante, incluso faltando por jugar salí mucho mejor que cuando me lesione y solo me dediqué a estudiar. Jugar me daba la fuerza para dar lo mejor de mí. Y si me dormía en clase, no lo hacía por falta de respeto, si no porque en la noche estudiaba y mi cuerpo necesitaba fuerzas para jugar y para estudiar.

No estén decepcionados por lo que no hice, lo que pude hacer o lo que hago, no es que no he dado lo mejor de mí, porque si lo doy, incluso en el futbol, que es lo que más me gusta, no he ganado mas que un torneo. Al contrario siéntanse alegres porque donde estoy, estoy feliz y quiero seguir siendo feliz. Renunciaré a cualquier logro terrenal por dar amor, porque solo dar amor nos hará estar felices.

Gracias a toda las personas que han estado estos 24 años conmigo, y también a aquellos que estuvieron por lapsos, porque esta tesis no es de la carrera, dejo aquí una parte de toda mi vida.

RESUMEN

Una huella de audio (Audio fingerprint) se puede considerar como un pequeño extracto de cierto objeto de audio[2]; esta huella es similar a la huella dactilar en el ser humano, es única. Algunos ejemplos de la aplicación de la huella de audio son identificar archivos de audio sin necesidad de la marca de agua, recuperación de información o criptografía.

Para la obtención de la huella digital de un audio, es necesario que éste pase por varios procesos como el pre-procesamiento, el procesamiento, la extracción de características y el modelado de huella.

En la fase de pre-procesamiento se usan filtros para quitar ruido(frecuencias no deseadas), o también para reconstruir la señal, si ésta tuvo pérdida de información debido a la digitalización, entre otras. Para la fase de procesamiento, se transforma la señal de audio para que ésta dependa de la frecuencia y se construya un espectrograma. En el proceso de extracción de características, se analiza el espectrograma para encontrar particularidades que pueden prevalecer a pesar del ruido de la señal. Finalmente en la fase de modelado de huella, se usan las características de la fase de extracción y se construye una huella digital de audio. Para finalizar, la huella se compara dentro de una base de datos para determinar los audios que son similares.

En este documento se explica el funcionamiento de cada proceso. Se habla de la aplicación de los filtros pasa baja, pasa alta, pasa banda y elimina banda; se muestra cómo se puede transformar la señal mediante la transformada en tiempo corto de Fourier, al igual que se muestra cómo construir el espectrograma y cómo de éste obtener los picos espectrales. Por último se explican dos algoritmos de modelado de huella, las imágenes binarias y los puntos de referencia.

En esta tesis se propone el desarrollo de una herramienta para pre-procesar archivos de audio, procesarlos, analizarlos y obtener su respectiva huella de audio para poder experimentar dentro de una base de datos y obtener resultados.

ÍNDICE GENERAL

CAPÍTULO 1

INTRODUCCIÓN

1.1 INTRODUCCIÓN

El procesamiento del audio digital ha proporcionado grandes avances, en diferentes ámbitos tales como la música, un ejemplo claro son las tiendas digitales; la interacción de humanos y máquinas mediante el habla, ahora se puede controlar objetos con nuestra voz; la localización de objetos, mediante pulsos eléctricos, con una computadora se puede escuchar a que distancia está el objeto; la sismología, mediante sonidos de la Tierra se puede determinar la estructura de la misma, etc.[8].

Otro avance muy grande es que el audio digital ayuda a comprimir los datos de un archivo para que se pueda transmitir de manera más rápida; también mediante la compresión el audio puede ocupar menos espacio en la memoria sin que éste pierda información; por último el uso del audio digital ha permitido su sencilla manipulación para analizar datos, obtener patrones, modificar la señal por medio de filtros y obtener resultados a los problemas de los diferentes ámbitos ya mencionados.

El procesamiento de audio inicia desde que se convierte la onda analógica en onda digital, lo cual provoca que algunos datos (dependiendo de las características de las muestras) se pierdan. Algunas veces es información que el oído humano no puede detectar, pero para la computadora siempre será información valiosa.

El uso de filtros ayuda a manipular el audio digital a nuestra conveniencia. Se puede usarlos antes y después del análisis de la onda digital
. Cuando el audio es digitalizado, es necesario pre-procesar la información para limpiar y/o modificar la señal. También el uso de los filtros ayuda a extraer características de una onda y que éstas se puedan clasificar para encontrar patrones.

En este documento se consideran aspectos para encontrar la huella digital de un audio. En [2] menciona una forma general de encontrar la huella digital, en la que, aplicaciones tales como Shazam, se basan para encontrar y comparar la huella dentro de una base de datos. Como todo en la vida, los algoritmos para encontrar la huella no son perfectos, por lo que en este documento igual se pretende explicar algunas variantes de los algoritmos para que el lector, dependiendo su criterio, elija los algoritmos y variantes que mejor le convengan.

En la actualidad para conocer el comportamiento de una huella digital de audio es necesario conocer algún lenguaje de programación que permita diseñar e implementar un sistema que extraiga características de patrones distintivos de la información. En esta

tesis se desarrolla un sistema que permite a un usuario interactuar con un archivo de audio de manera que se puedan extraer características para modelar una huella digital sin necesidad de conocer algún lenguaje de programación.

1.1.1 Objetivos

Objetivo General

Desarrollar una herramienta que permita procesar señales de audio a partir de las cuales es posible extraer huellas de audio con utilidad en la extracción de patrones distintivos en la información.

Objetivos específicos

- Revisar y analizar el funcionamiento de los filtros de pre-procesamiento de audio digital.
- Obtener y analizar cada una de las fases de la extracción de la huella digital de audio.
- Estudiar el estado de arte de la huella digital de audio.
- Crear un sistema que permita procesar señales de audio.
- Llevar a cabo pruebas funcionales del sistema.

1.1.2 Alcances y Limitaciones

Alcance

A partir de esta tesis se crea una herramienta que extrae huellas de audio digital mediante la extracción de características. Permite modificar los parámetros en el pre-procesamiento de audio para modelar una huella según los criterios del usuario. La herramienta también faculta la realización de experimentos para visualizar el comportamiento de la huella modelada respecto a otras.

Limitaciones

El trabajo desarrollado en esta tesis respecta al pre-procesamiento de huella de audio y la extracción de características mediante imágenes binarias y puntos de referencia.

1.2 MARCO TEÓRICO

En esta sección se presentan conceptos relacionados a la huella digital de audio. Se mencionará desde la definición de un audio, el muestreo, la necesidad de los filtros, el análisis de las muestras, la extracción de características y la creación de la huella.

En la actualidad, es común que el usuario almacene archivos de música en su dispositivo móvil, memoria USB, etc., y que ésta pueda ser reproducida en cualquier altavoz. Antes de llegar a esta época, la reproducción de la música pasó por los CD´s, los casete y los disco de vinilo. Estos últimos no usaban el proceso digital para su reproducción y/o almacenamiento, si no que todo su proceso era analógico. Por eso es correcto cuando alguien mencione que el sonido de un disco de vinilo tiene más calidad que una canción mp3.

Como se mencionó, el audio digital tiene una pequeña pérdida de información cuando ésta se digitaliza, pero ¿Por qué se trabaja el audio digital en vez del audio analógico?

En la física se considera que el sonido consiste en una vibración mecánica de un medio elástico(gaseoso, líquido, sólido) y la propagación de esta vibración a través de las ondas. Por lo que se infiere que el audio analógico es una onda. Matemáticamente una onda tiene infinidad de valores, pero no se puede almacenar o trabajar con todos en una computadora, pues ésta es discreta, y tiene un limite. Por esa razón se crea el audio digital, aunque pueda existir una pérdida de información, la gran ventaja que éste tiene es la manipulación de los datos, es muy extensa y rápida.

Watkinson menciona que “un audio digital tiene las mismas características que un audio analógico, ya que ellos son totalmente transparentes y reproducen la onda original sin ningún error”[19]. El audio analógico pasa por un proceso(digitalización), donde transforma la señal analógica en una señal digital. La digitalización se basa en el muestreo y la cuantización, lo cual producirá una discretización.

El muestreo (Sampling) es un proceso que consiste en obtener un valor mediante el voltaje en intervalos regulares; el valor de cada muestra es redondeado al valor más cercano, ya que como antes se mencionó el audio digital es discreto. La muestra es el valor de la señal v(n) tomado en un tiempo t=n. La muestra a su vez necesita de otros procesos tales como la cuantización. La cuantización es el proceso de seleccionar los números para representar los niveles de voltaje a cada muestra. Si se toma un rango de

cuantización [0,100], 0 representaría el valor más negativo de la señal y a su vez, 100 correspondería al nivel más positivo de la señal; este proceso puede causar un pequeño error de redondeo y causar distorsión

¿Qué problemas trae consigo el muestreo(sampling)?

Se muestra una gráfica simulando una señal analógica(Ver figura 1.1); usando la definición de muestra, ¿Cuántas muestras serán necesarias para que éstas sean usadas para regenerar la señal?.

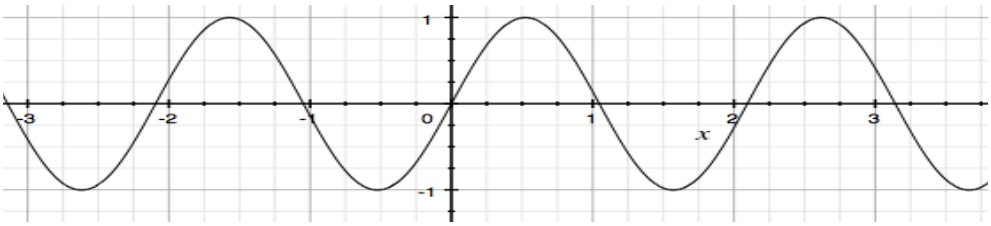


Figura 1.1 Simulación de una onda analógica

Se dispone que cada número entero del eje de las x para poder tomar una muestra(Ver figura 1.2).

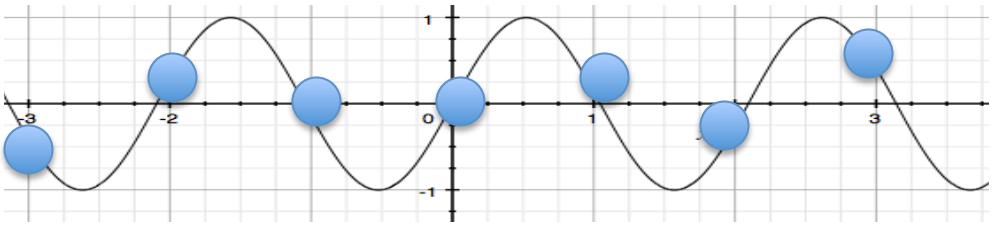


Figura 1.2 Muestreando la señal en cada número entero del eje de las x

Las coordenadas de los puntos de izquierda a derecha son:

- P1(-3,-0.412)
- P2(-2,0.279)
- P3(-1,-0.141)
- P4(0,0)
- P5(1,0.141)
- P6(2,-0.279)
- P7(3,0.412)

Si se unen los puntos para reconstruir la señal original, se tendría algo como la figura 1.3

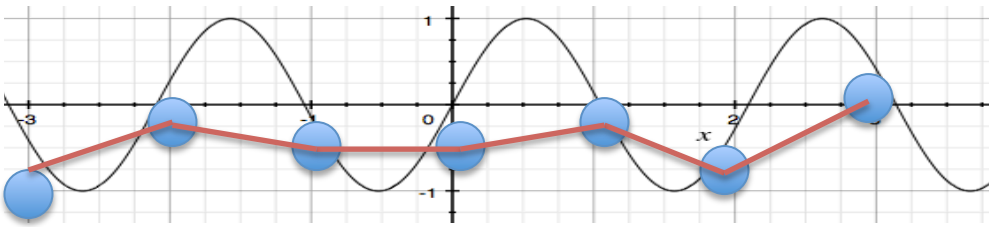


Figura 1.3 La línea roja representa una posible reconstrucción de la señal a partir de los puntos dados.

Como se puede notar, usando esos puntos, no es posible reconstruir la señal original. Smith define el aliasing como ”el fenómeno de cambio de frecuencia de sinusoides durante el muestreo”. Entonces cómo se puede saber, ¿Cuántas muestras se debe obtener de la señal?. De esta pregunta se deriva el teorema fundamental del muestreo, conocido como teorema de muestreo de Shannon o el teorema de muestreo de Nyquist el cual dice que “La frecuencia de muestreo debe ser al menos dos veces más alta que la señal de entrada”.

Otro problema que se pueda originar del sampling es el jitter. Apogee(Empresa que provee dispositivos profesionales de audio a nivel mundial) menciona que el jitter se puede definir como “la desviación no deseada de una señal periódica de el momento ideal”. Para entender esto, es necesario conocer que las muestras usan un word clock. Un word clock es aquella señal de reloj que indica el momento en el que se tomará una muestra.

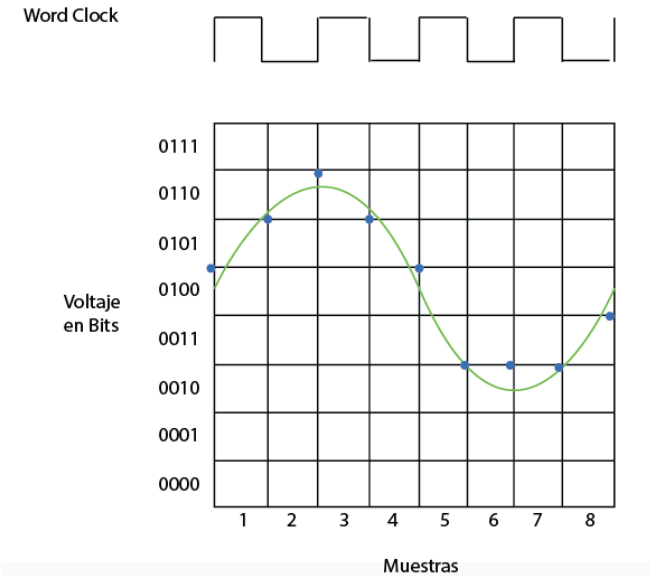


Figura 1.4 Representación de cómo trabaja el Word clock en el muestreo.

Se puede visualizar en la figura 1.4 el word clock(parte superior) es constante y en cada flanco de subida ó de bajada se toma una muestra de la señal. El problema viene cuando por alguna razón el reloj no es constante o la muestra tiene un pequeño retraso.

Se observa en la figura 1.5 que el word clock no es constante, por lo que la muestra no es la correcta y causa una distorsión en la señal.

Una vez digitalizado un audio, si éste tiene ligeras distorsiones puede alterar el análisis de una señal, o producir un resultado no deseado. ¿Cómo solucionar este problema?. Una solución es volver a digitalizar el sonido, quizá si no se tenía un equipo sofisticado, esta vez se haga con un equipo que muestre una mejor resolución y se cuide que el jitter no aparezca. Pero sí se usa un equipo sofisticado y aún existe distorsión, lo que se puede aplicar es un pre-procesamiento (uso de filtros) al audio digital, para manipularlo a nuestra conveniencia.

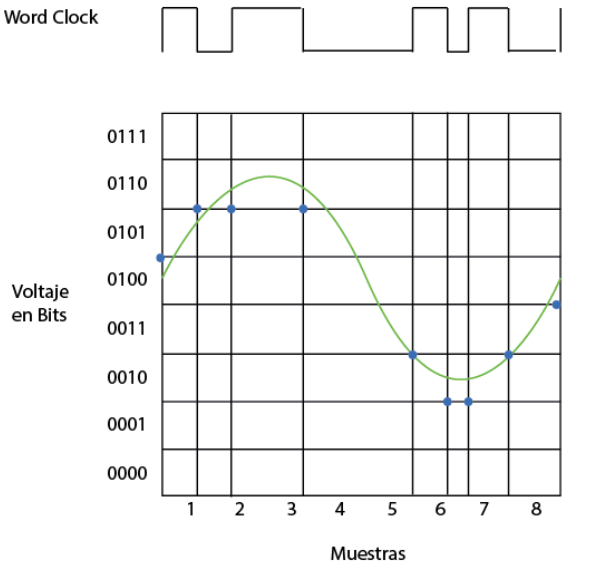


Figura 1.5 Representación del Jitter

1.2.1 Estéreo y mono

Una señal monofónica, es aquella que es grabada con un solo micrófono, mientras que una señal estereofónica es aquella que es grabada con un par de micrófonos iguales. Al escuchar en estéreo, por el canal izquierdo se escucha lo que se grabó con un micrófono, y por el canal derecho se escucha lo que se grabó con el otro micrófono; mientras que al escuchar en mono, en ambos canales se escucha lo mismo.

Se deduce que la señal de un audio con dos canales, es de mejor calidad que el mono, pero para analizarla es conveniente convertir aquellos archivos de estéreo en mono, ya que se podría tener dos huellas digitales de un audio en estéreo, una huella se obtendría del canal de la izquierda y el otro del de la derecha.

Para convertir de estéreo a mono, se puede conservar el canal izquierdo, conservar el canal derecho ó aplicar la media aritmética a las amplitudes de las señales en un cierto tiempo, siendo N la nueva señal, I el canal izquierdo y D el canal derecho, la fórmula sería la siguiente:

$$N(t) = \frac{I(t) + D(t)}{2} \tag{1.1}$$

1.2.2 Decibeles

El bel representa la intensidad del sonido mediante la relación de la señal de entrada y la de salida, su nombre se debe al físico Alexander Graham Bell. El bel significa que se cambia por un factor de diez. Ejemplo si un sistema tiene 4 bels de amplificación, ésta produce una señal de salida con 10000(10*10*10*10) veces la potencia de entrada. Un decibelio se representa con la siguiente fórmula en relación a la señal de entrada y la señal de salida.

$$db = 10 \log\left(\frac{Senal\ Salida}{Senal\ Entrada}\right) \tag{1.2}$$

Por lo general se trabaja con la amplitud de la señal, no con la potencia; así que se usa la relación que existe entre la amplitud(A) y potencia(P).

$$A = \sqrt{P} \tag{1.3}$$

1.2.3 Filtros

El audio digital presenta muchas ventajas para poder manipular la información. Los filtros son una forma de la manipulación del audio. Los filtros digitales son usados para dos propósitos generales:

1. “Separación de señales que han sido combinadas.
2. Restauración de señales que han sido distorsionadas de alguna manera.” [8].

Un filtro es una operación matemática que de una secuencia de números(señal de entrada), los modifica produciendo otra secuencia de números(Señal de salida).

Con el uso de filtros se puede atacar los problemas causados por el muestreo, como se menciona anteriormente, los filtros restauran las señalas que han sido distorsionadas. También se aplica el uso de algunos filtros en el pre-procesamiento para separar señales combinadas. Un ejemplo sería si se tiene un audio de una conversación en un centro comercial; solo interesa la conversación, no el ruido generado por el ambiente, por lo que se puede aplicar algunos filtros para quitar el ruido. De este tipo de ejemplos, sobresalen algunos filtros ideales, tales como:

- Filtro Pasa Baja: Este filtro solo deja pasar la parte baja de las frecuencias. Es decir en la figura 1.6, el filtro solo deja pasar las frecuencias que son menores a 20,000 Hz.

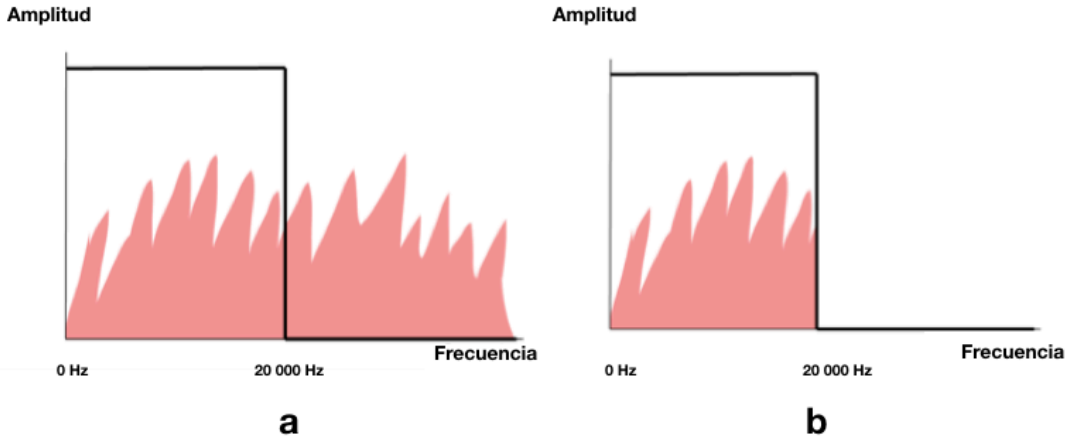


Figura 1.6 a) Representación del filtro pasa baja y la señal, b) Resultado de la aplicación del filtro pasa baja

- Filtro Pasa Alta: Este filtro es lo contrario al anterior, deja pasar solo las frecuencias altas. Es decir en la figura 1.7, el filtro solo deja pasar las frecuencias que son mayores a 20,000hz
- Filtro Pasa Banda: Deja pasar varias frecuencias dentro de un rango comprendido (figura 1.8).
- Filtro Elimina Banda: Al contrario del filtro anterior, éste deja pasar frecuencias que no estén dentro del rango(figura 1.9).

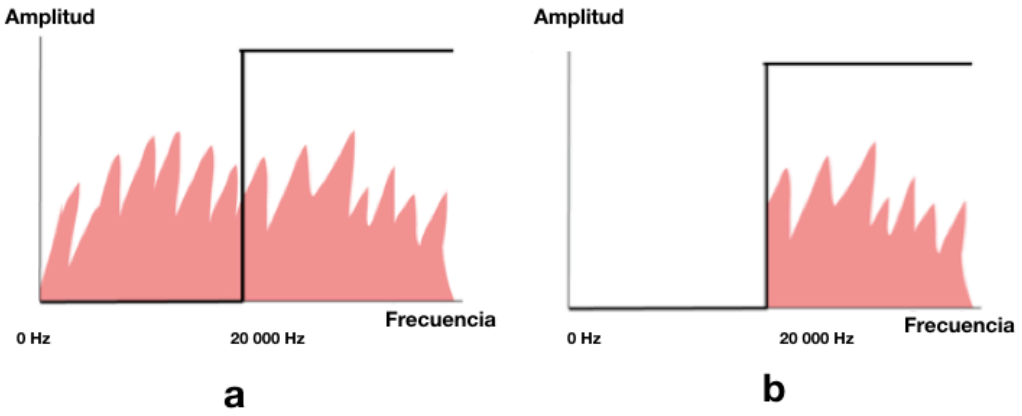


Figura 1.7 a) Representación del filtro pasa alta y la señal, b) Resultado de la aplicación del filtro pasa alta

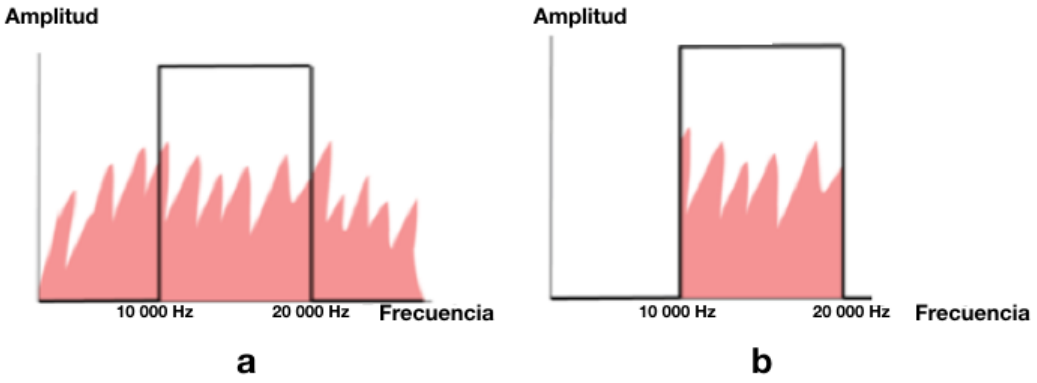


Figura 1.8 a) Representación del filtro pasa banda y la señal, b) Resultado de la aplicación del filtro pasa banda

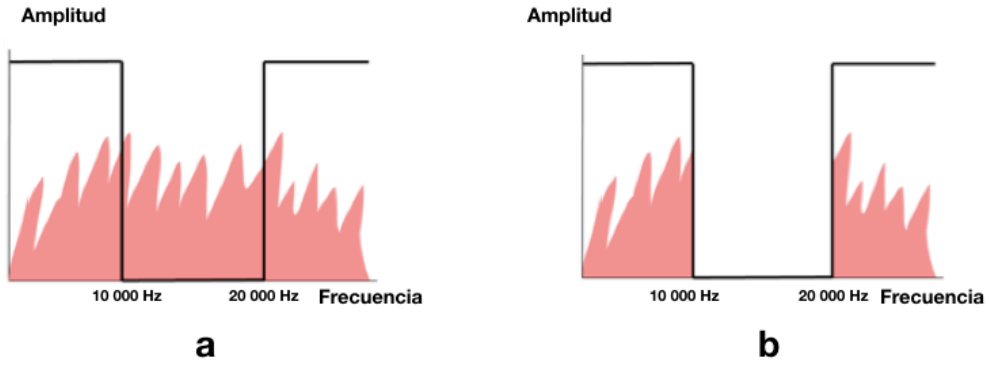


Figura 1.9 a) Representación del filtro elimina banda y la señal, b) Resultado de la aplicación del filtro elimina banda

Los filtros ideales son irrealizables, debido a que éstos implican tener una respuesta de impulso antes del tiempo en el cual se aplica la función impulso [18]. Debido a este problema se han creado filtros (no ideales) que se asimilan a los anteriores, tales como Window Sinc, Butterworth, Chebyshev, entre otros. Estos filtros aunque han sido analizados respecto al filtro pasa baja, se pueden aplicar simulando los demás(pasa alta, pasa banda y elimina banda apéndice C).

1.2.3.1 Características de Filtros

Cualquier filtro no ideal cumple con las siguientes características(Figura 1.10).

- Banda pasante: Conjunto de frecuencias para las cuales el filtro deja pasar desde la señal de entrada hasta la señal de salida.
- Banda rechazo: Conjunto de frecuencias que el filtro no deja pasar
- Frecuencia de corte o borde de la banda: Es aquella en la que la ganancia cae generalmente 3dB(atenuación) por debajo de la ganancia alcanzada(F1,F2). Se encuentra entre la banda pasante y la banda de transición.
- Frecuencia central(F0): Es la media geométrica de las frecuencias extremas.
- Ancho de Banda o banda de transición(Bw):Es la diferencia entre las frecuencias extremas, cuanto mayor anchura de banda es menor la calidad del filtro.
- Calidad(Q): Especifica la eficiencia del filtro(f0/Bw).

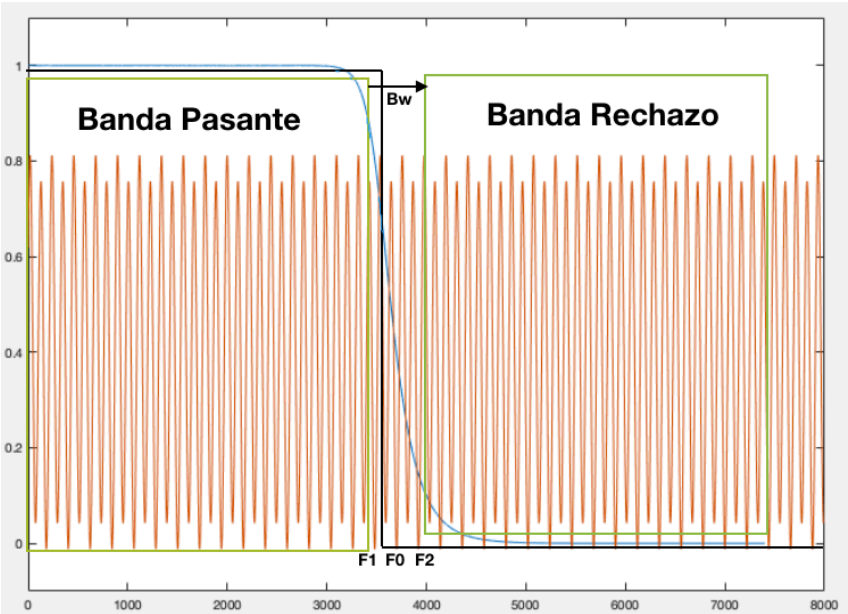


Figura 1.10 Características de los filtros. La línea de color negro representa el filtro ideal(pasa baja), y la línea azul presenta el filtro real o no ideal.

1.2.3.2 Filtros con respuesta al impulso

Este tipo de filtro se caracteriza por operar en el tiempo-discreto. Se pueden clasificar en varios grupos, en este caso se hablará de dos tipos:

- FIR (Respuesta finita al impulso)
- IIR (Respuesta Infinita al impulso)

1.2.3.2.1 Respuesta Finita al Impulso(FIR)

Su salida se basa en la suma de las entradas actuales y anteriores de la señal.

$$y[t] = b_0 x[t] + b_1 x[t+1] + \cdots + b_{N-1} x[t-N+1] = \sum_{k=0}^{N-1} b_k x[t-k] \qquad (1.4)$$

Donde:

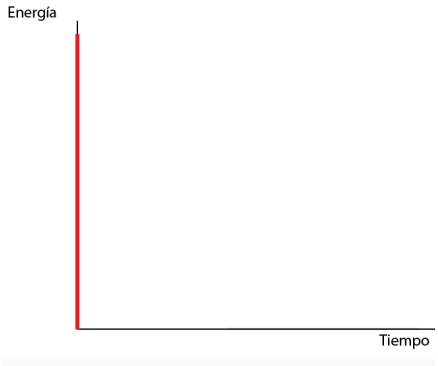
$X[t]$ es la señal de entrada.

$y[t]$ es la salida.

b_k son los coeficientes del filtro

N es el número de coeficientes en el filtro.

Esta ecuación se puede escribir de otra forma, usando el concepto de respuesta impulsional y convolución. La respuesta impulsional es la salida de un sistema cuando se le aplica en la entrada una señal impulso. La señal impulso se define como la figura 1.11 y la función 1.5:



$$\delta[t] = \begin{cases} 1 & t = 0 \\ 0 & t \neq 0 \end{cases} \qquad (1.5)$$

Figura 1.11 Gráfica de señal impulso, cuando t=0 la energía será 1.

Si esta señal $\delta[t]$ se introduce a un sistema LTI (Apéndice A) (figura 1.12), se obtiene una salida(respuesta impulsional) $g[t]$ la cual caracterizará al sistema. Lo que esto quiere decir que con la respuesta impulsional se puede conocer la respuesta a otra entrada.

Figura 1.12 Diagrama de la respuesta impulsional.

El sistema puede ser estático y/o dinámico. El sistema estático solo da la respuesta en el tiempo en que se pide, es decir, se tiene un sistema con $y[x] = 3x$, si la señal de entrada vale en $u[0] = 1$, la salida $y[0] = 3$, si $u[1] = 2$, la salida $y[1] = 6$, en cambio el sistema dinámico necesita de los valores actuales y anteriores de la entrada, por lo que la entrada debe ser obligatoriamente una función.

Como se ha observado la entrada, el sistema y salida de funciones, se puede cometer el error de expresar la salida como la multiplicación de la función de la entrada y del sistema.

$$y[t] = f[t] * u[t] \tag{1.6}$$

La función de salida se representa como la convolución de dos funciones. La convolución es un operador que convierte dos funciones en una tercera, y se representa por la siguiente forma(para funciones continuas):

$$y(t) = \int_{-\infty}^{\infty} a(k)b(t-k)\,dk \tag{1.7}$$

donde a y b son las funciones que se convolucionarán y y es la función resultante de esa convolución.

La convolución de dos funciones discretas se representa por la siguiente forma:

$$y[t] = \sum_{-\infty}^{\infty} a[k]b[t-k] \tag{1.8}$$

donde a y b son las funciones que se convolucionarán y y es la función resultante de esa convolución.

Como se observa la función convolución es la multiplicación de las funciones y se suman con todas las multiplicaciones anteriores(sistema dinámico). Como se vio anteriormente la función $g[t]$ (respuesta impulsional) ayuda a conocer la respuesta a cualquier entrada por lo que se puede sustituir $b[k]$ por $g[k]$ (Apéndice B), y también se puede notar que cuando la señal impulso está en t=0, la señal vale 1, por lo que para valores t<0 la función del sistema valdrá 0, pero para valores mayores a 0, la respuesta al sistema será diferente de 0, pues a partir de $\delta[0]$ ya existirá un valor y como en un sistema dinámico se requiere de valores anteriores, entonces los valores de $y[t]$ podrán ser diferentes de 0, por lo que la sumatoria de la convolución puede iniciar en k=0.

La salida $y[t]$ puede escribirse como la convolución de la señal de entrada $x[t]$ con la respuesta impulsional $g[t]$:

$$y[t] = \sum_{k=0}^{N-1} x[k] \cdot g[t-k] \tag{1.9}$$

Hablando en dominio de la frecuencia se tendría que:

$$Y[\phi] = X[\phi] * G[\phi] \tag{1.10}$$

donde Y, X y G son las transformadas de las señales de y, x y g respectivamente.

Un ejemplo de este tipo de filtro es el sinc (Window Sinc); está basado en los filtros ideales (pasa baja). Tomando la inversa de la Transformada de Fourier de este filtro ideal, muestra el filtro kernel (respuesta impulsional)(Figura 1.13). Esta curva es la representación de la función sinc = sin (x)/x.

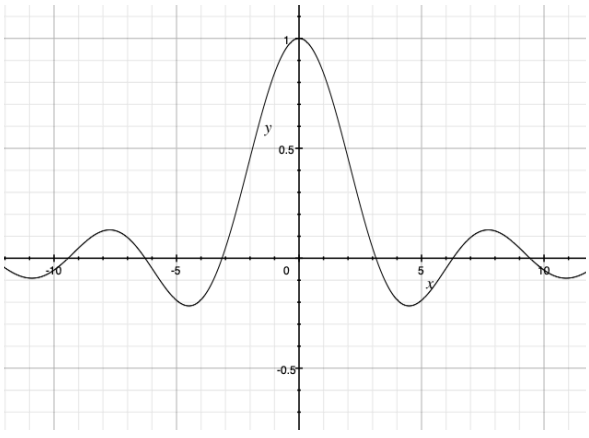


Figura 1.13 Respuesta impulsional al aplicar la Inversa de la transformada de Fourier al filtro ideal pasa bajo.

Cuando se aplica este filtro con cualquier señal de entrada, se forma un perfecto pasa baja, el problema es que la función sinc continúa hasta el infinito positivo y negativo sin caer a una amplitud continua 0. Para las matemáticas la longitud infinita no sería un problema, pero para las computadoras si.

La solución al problema anterior es hacer dos modificaciones. Primero se truncan M+1 muestras elegidas simétricamente alrededor del lóbulo central, donde M es par. Todas las muestras fuera de M+1 son 0 o son ignoradas. Segundo la función se recorre a la derecha, de manera que estuviera de 0 a M[8].

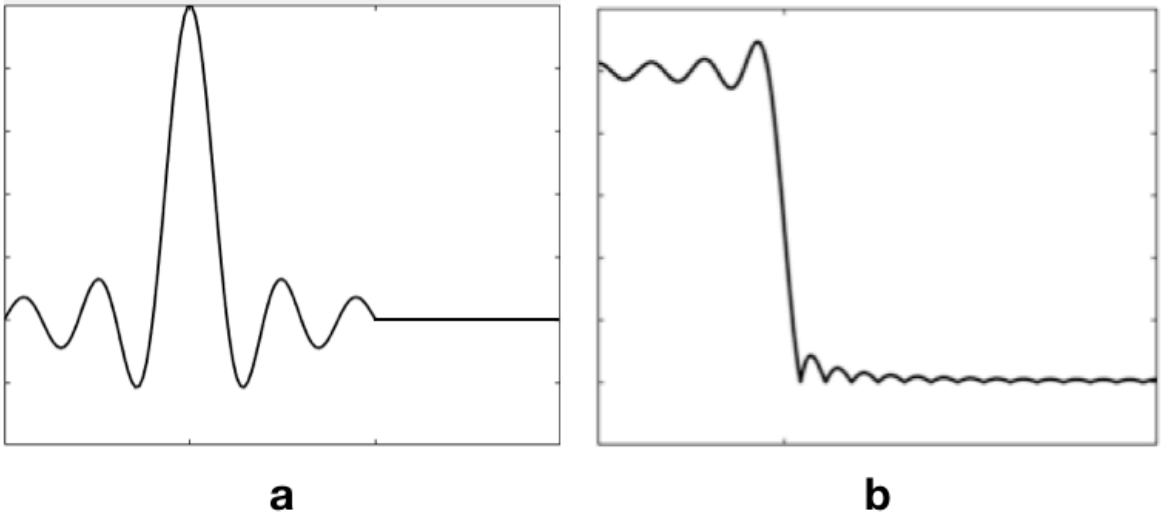


Figura 1.14 a) Filtro kernel truncado b) Transformada del filtro Window Sinc truncado.

Como se puede notar en la figura 1.14b en la banda de paso hay una ondulación excesiva y en la banda de paso hay una atenuación deficiente, por lo que se recurre a usar una ventana (Blackman, Hamming, en la sección 2.1.4.5 se explicará con mas detalle), para reducir la brusquedad de los extremos truncados y así mejorar la respuesta en frecuencia.

Como se observa en la figura 1.15, la banda de paso ahora es plana, y la banda de rechazo es buena que no se puede notar a simple vista. Para aplicar este filtro se necesitan algunos parámetros:

- N, que es equivalente a M el número de puntos a truncar, se recomienda que sea par.
- Wn si es un escalar, será la frecuencia de corte del filtro pasa baja y pasa alta, si es un vector de 2 escalares, w1<w2 se diseñará un filtro pasa banda o elimina banda.

El último parámetro será especificar la ventana(Blackman, Hamming) en la cual el tamaño de ésta es equivalente a N+1.

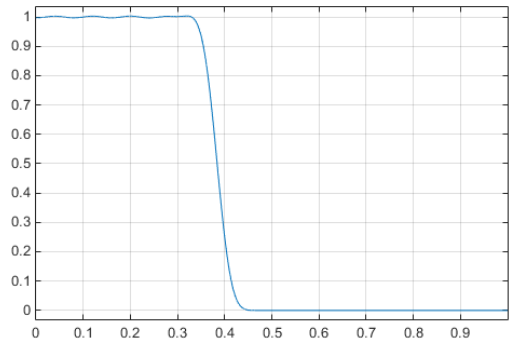


Figura 1.15 Respuesta en frecuencia del filtro Window Sinc.

1.2.3.2.2 Respuesta Infinita al impulso(IIR)

Este tipo de filtro se caracteriza por tener una retroalimentación de la señal de la salida(recursión). Si su entrada es un impulso, la salida tendrá una infinidad de valores no nulos, es decir nunca volverá a su estado de reposo. Su salida se basa en entradas anteriores, actuales, y en salidas anteriores que se almacenan en memoria.

$$y[t] = b_0x[t] + b_1x[t-1] + \cdots b_Nx[t-M] - a_1y[t-1] - a_2y[t-2] - \cdots - \quad a_Ny[t-M] = \sum_{k=0}^{N-1} b_kx[t-k] - \sum_{k=1}^{N-1} a_ky[t-k] \tag{1.11}$$

Donde:

$x[t]$ es la señal de entrada.

$y[t]$ es la salida.

a_k, b_k son los coeficientes del filtro

N es el número de coeficientes en el filtro.

Algunos ejemplos de este tipo de filtro IIR son Butterworth y Chebyshev.

La figura 1.16 muestra la respuesta en frecuencia del filtro Chebyshev pasa bajo. Como se puede notar una buena ondulación produce una mala caída, pero una mala ondulación, produce una buena caída. Cuando la ondulación se establece en 0%, el filtro se denomina plano o Butterworth[8]. Butterworth se considera por ser plano en la banda pasante.

En estos filtros se debe tener en cuenta las frecuencias de corte(F1,F2), la ondulación en la banda pasante y la ondulación en la banda rechazada(rp,rs). La ondulación también es conocida como la atenuación, que representa la caída que el filtro

permite en la banda pasante y en la banda rechazada(Figura 17, ver apéndice C para ver representación de en los filtros pasa alta, pasa banda, elimina banda).

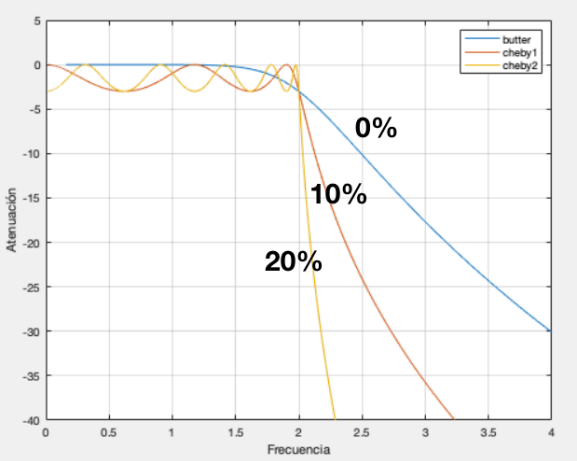


Figura 1.16 Representación frecuencial de los filtros Chebyshev y Butterworth.

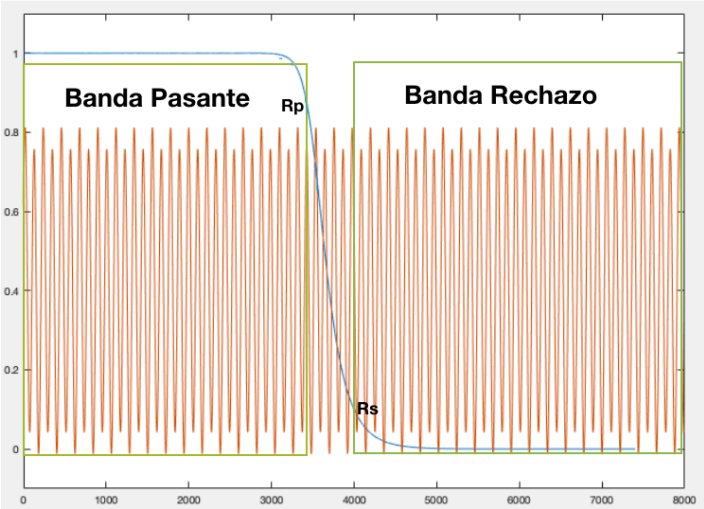


Figura 1.17 Representación gráfica de las atenuaciones (Rs, Rp)

Teniendo en cuenta los parámetros de la figura 1.17 y de la sección 1.1.3.1 (F1, F2, Rp y Rs), se puede obtener el orden(n). Para el filtro Butterworth se usa la siguiente fórmula:

$$n = \frac{\log\left(\frac{10^{\frac{Rs}{10}} - 1}{10^{\frac{Rp}{10}} - 1}\right)}{2\log\left(\frac{F2}{F1}\right)} \tag{1.12}$$

Para el filtro Chebyshev se utiliza la siguiente fórmula para obtener el orden(n):

$$n = \frac{\operatorname{arccosh}\left(\sqrt{\frac{10^{0.1Rp}-1}{\varepsilon}}\right)}{\operatorname{arccosh}\left(\frac{F2}{F1}\right)} \tag{1.13}$$

donde:

$$\varepsilon = \sqrt{10^{0.1Rs}-1} \tag{1.14}$$

El orden(n) es importante porque es el que determina que tan suave es la caída entre la banda de paso y la banda de atenuación, mientras el orden sea más grande, la caída será menos suave, pero se debe tomar en cuenta que puede afectar las atenuaciones y las bandas de paso y atenuación(Apéndice D).

Algo importante es que estas atenuaciones se miden en decibeles y por lo general se trabaja en amplitud, por lo que hay que tener en cuenta la ecuación 1.3, en la que representa la relación entre el poder(dB) y la amplitud(A).

1.2.4 Análisis de una señal de audio

1.2.4.1 Transformada de Fourier

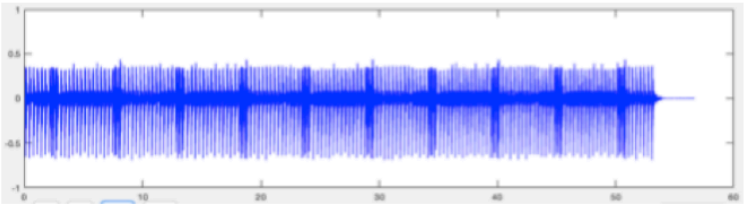
Como se describió en la sección 1.2.3, los filtros separan señales que están combinadas ó arreglan señales distorsionadas, pero para aplicarlos es necesario analizar la señal de audio para determinar las frecuencias que añaden el ruido en la señal; conociendo esto se puede determinar los parámetros tales como la frecuencia de corte, la atenuación entre otros; después se podrá aplicar los filtros a la señal x(t).

Una señal de audio depende del tiempo x(t). Para analizar una señal, ésta se debe transformar al dominio de la frecuencia, para descubrir aspectos que serían difíciles de observar en la representación temporal (figura 1.18).

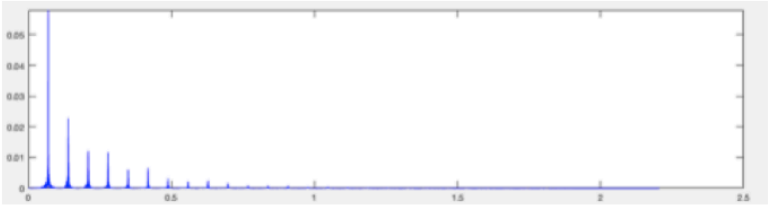
Jean Baptiste Joseph Fourier, propuso que cualquier señal podría ser representada por una suma de senos y cosenos. Si se descompone cualquier onda, se pueden analizar las diferentes frecuencias que están en la señal original. La transformada de Fourier está dada por la siguiente formula:

$$X(f) = F\big(x(t)\big) = \frac{1}{2\pi}\int_{-\infty}^{\infty} x(t)e^{-i\omega t}dt \tag{1.15}$$

donde $x(t)$ es la señal de entrada y $w= 2\pi f$



a



b

Figura 1.18. a) Se muestra una señal periódica,
b) Se muestra los componentes en frecuencia de la señal.

Usando la transformada de Fourier se lleva la señal del dominio del tiempo al dominio de la frecuencia(Figura 1.19).

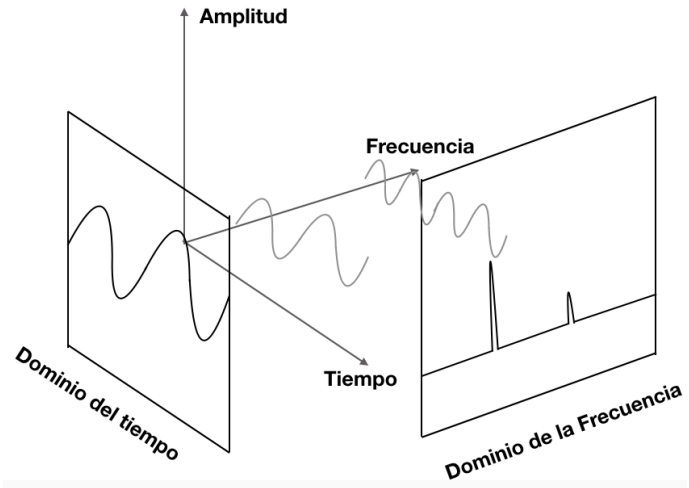


Figura 1.19 Descomposición de Fourier.

1.2.4.2 Transformada Discreta de Fourier

La señal analógica se digitaliza para ser analizada por una computadora, por lo que la Transformada de Fourier no ayudaría a pasar la señal al dominio de la frecuencia, dado que ahora la señal es finita y Fourier necesita una señal infinita.

Para transformar una señal digital, se requiere la Transformada Discreta de Fourier, que hace lo mismo que la Transformada normal, solo que para una señal discreta. La DFT está representada por la siguiente fórmula.

$$X[w_k] = \frac{1}{N}\sum_{r=0}^{N-1} x_r e^{-i(w_k t)} \tag{1.16}$$

donde x_r es la señal de entrada, k=0,1,2…. N-1, N es el número de muestras, y $w_k=2\pi k/N$.

1.2.4.3 Espectrograma

Cuando se analiza una señal en el dominio de la frecuencia, la función X(f) representa el espectro de una señal. El espectro es variable con respecto al tiempo; para observar estas variaciones es necesario que el resultado de varios barridos se almacene y se comparen las amplitudes, de aquí el espectrograma.

El espectrograma es una imagen que muestra la evolución temporal del espectro de una señal (figura 1.20). El espectrograma se compone del tiempo(eje y), la frecuencia(eje x) y la amplitud(Color, en este caso va desde azul fuerte, que es baja amplitud, hasta amarillo brilloso, que significa alta amplitud); se puede deducir que la imagen muestra el tiempo en el que las frecuencias tienen mayor amplitud.

A partir de este espectrograma, se pueden extraer características para poder modelar una huella de audio digital

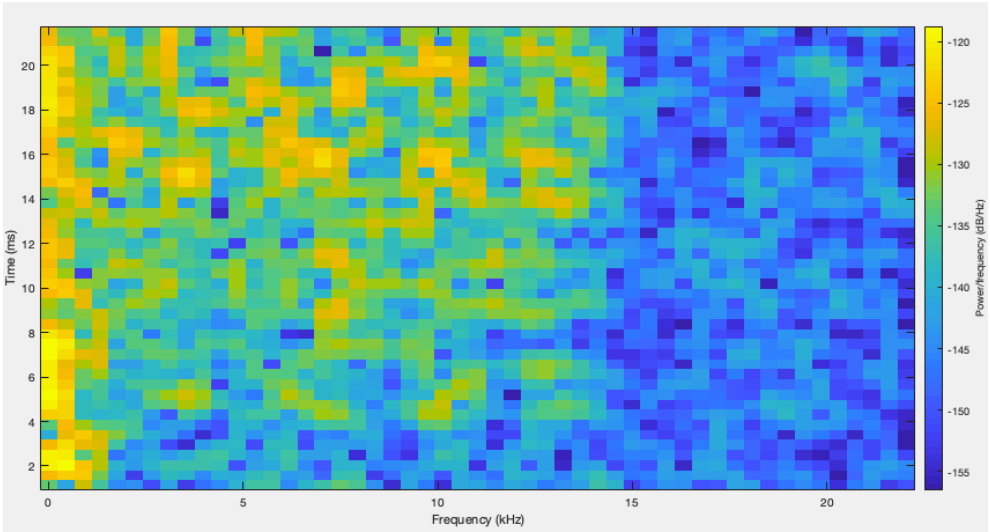


Figura 1.20 Espectrograma de una onda de audio.

Para poder formar el espectro es necesario usar trozos de señal, es decir, si se obtiene el espectro de la onda por segundos, entonces se tendría el espectro del primer segundo, luego se obtiene el espectro del siguiente segundo y así sucesivamente, hasta formar una imagen de cómo evoluciona el espectro. Para cortar la señal en trozos y obtener su espectro respectivamente, se usará la transformada discreta de Fourier en tiempo corto.

1.2.4.4 Transformada Discreta de Fourier en Tiempo Corto

La STDFT consiste en dividir la parte de la señal, que se asume que es estacionaria, en diferentes trozos y los transforma. Con estos trozos transformados se puede construir el espectrograma de la onda de audio[22].

En esta parte se usarán las funciones ventanas, que son 0 excepto en un pequeño periodo; estas ventanas permitirán dividir la señal en los tramos que se deseen.
La STDFT se compone de la siguiente fórmula:

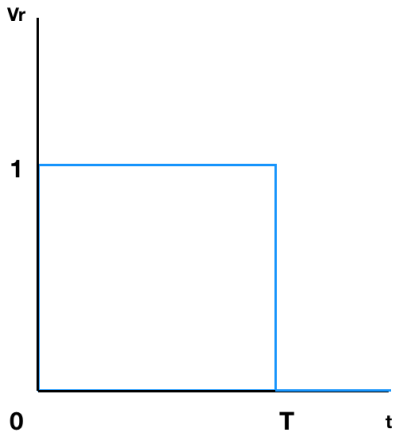
$$X[m, w_k] = \sum_n x[n]v[n-m]e^{-iw_k m} \tag{1.17}$$

donde x es la señal a transformarse, k=0,1,2…. N-1, m=0,1,2…N-1, N es el número de muestras, w_k=2πk/N con m=0,1,2…N-1 y v es la función ventana de longitud L.

1.2.4.5 Ventanas

Las ventanas permiten visualizar el tramo de la señal que se desea, análogamente como una ventana en una pared en la vida real, permite visualizar una determinada imagen de un cierto paisaje. Si se quisiera ver a la derecha de la imagen del paisaje que se tiene, se debería construir una ventana a la derecha de laque ya se tenía.

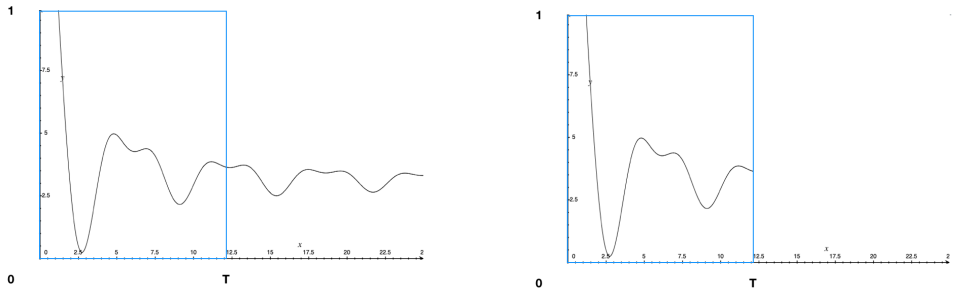
La ventana mas simple es la rectangular, que se define mediante las fórmula 1.18 donde t es el tiempo y $[0, T]$ es el intervalo que ésta durará(Figura 1.21).



$$v_r[t] = \begin{cases} 1 & si\ t \in [0,T] \\ 0 & para\ t\ otro\ valor \end{cases} \quad (1.18)$$

Figura 1.21 Representación gráfica de la Ventana rectangular.

Cuando se multiplica la función ventana por la señal, esto muestra el trozo de la señal deseada(figura 1.22).



a b

Figura 1.22 a) Multiplicación de la señal por la ventana b) Resultado de la multiplicación de la figura a.

En la figura 1.23 se muestra el dominio de la frecuencia de la función ventana. Como toda función ventana, ésta tiene un lóbulo principal y varios lóbulos laterales. El lóbulo principal se centra en cada componente de frecuencia de la señal del dominio del tiempo y los lóbulos laterales se aproximan a cero[21].

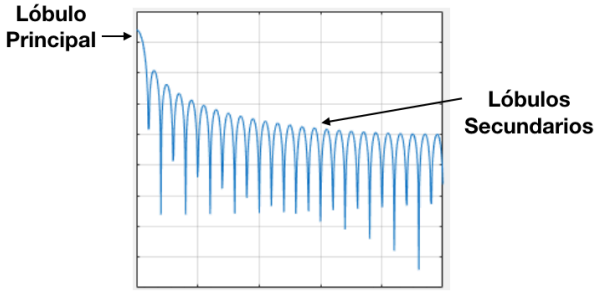


Figura 1.23 Se muestra el espectro de la función ventana rectangular, señalando el lóbulo principal y los lóbulos secundarios.

El efecto del ventaneo no causa problemas en el dominio del tiempo, pero en el dominio de la frecuencia, se deben tener en cuenta dos factores, la reducción en la resolución y las fugas espectrales. El problema de la reducción en la resolución se deriva principalmente por la anchura del lóbulo principal y el grado de fugas se deriva por la amplitud relativa del lóbulo principal a los lóbulos laterales. La anchura del lóbulo principal y la amplitud relativa de los lóbulos secundarios dependen de la longitud de la ventana[20].

Para poder elegir alguna ventana se puede tomar en cuenta que la rectangular tiene el lóbulo principal más estrecho, pero tiene los mayores lóbulos laterales en comparación a ventanas como Hamming o Hann; estas últimas tienen un lóbulo principal que es dos veces más ancho que la ventana rectangular generalmente, pero tienen amplitudes menores en los lóbulos secundarios(figura 1.24).

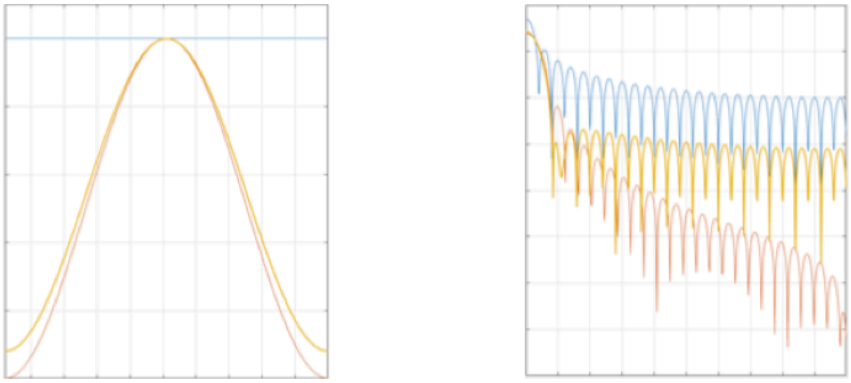


Figura 1.24 a) Ventanas en el dominio del tiempo b)Ventanas en el dominio de la frecuencia. Ventana rectangular(color azul), ventana Hann(color amarillo) y ventana Hamming(color rojo).

1.2.4.6 Traslape

Cuando se ha aplicado la transformada corta de Fourier, la señal está formada por trozos como en la figura 1.25 (a), pero se recomienda hacer un traslape entre las ventanas como en la figura 1.25 (b).

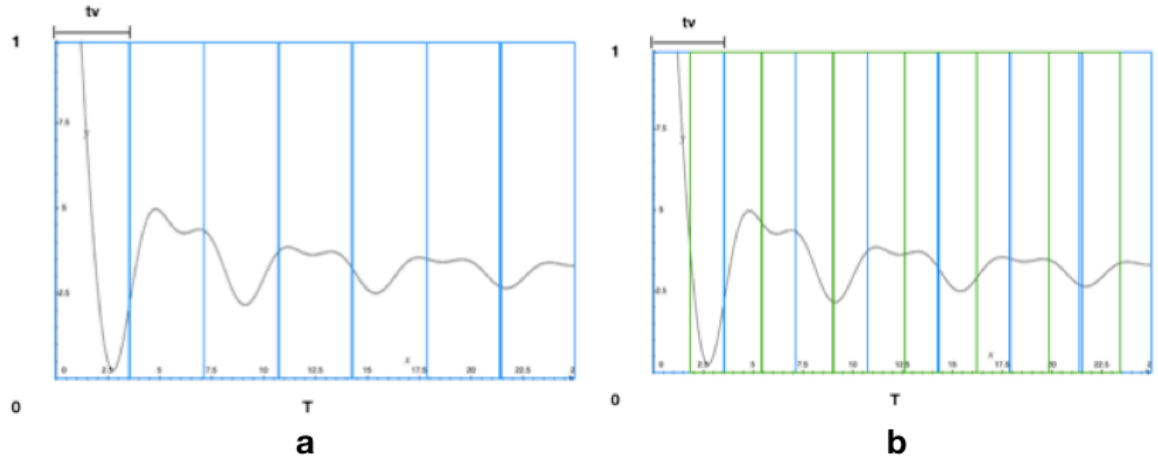


Figura 1.25 a) Se muestra las ventanas sin traslaparse b) Las ventanas están traslapadas en un 50%

El traslape funciona como un zoom en la que estira la escala del tiempo, lo que permite una mejor visibilidad en los cambios de la frecuencia con respecto al tiempo. Mientras más porcentaje de muestras traslapadas, mejor visualización tendrá el espectrograma(Figura 3.14).

1.2.4.7 Modelado de Huella de Audio Digital

En esta fase se obtienen características del audio digital para que a partir de éstas se construya la huella de audio digital. Para esta tesis se proponen dos algoritmos de modelado, las imágenes binarias y los puntos de referencia. Estos algoritmos se basan en el espectrograma, obtenido en la fase anterior, y tratan de conservar las características(puntos) que sean resistentes al ruido.

Cabe la pena mencionar que la huella digital de un audio es la concatenación de varias huellas que se forman por el audio dividido en *x* intervalos; estos intervalos pueden estar o no traslapados.

1.2.4.7.1 Imágenes Binarias

Para modelar la huella, se usarán imágenes binarias a partir del espectrograma. Se visualiza el espectrograma como una imagen y se encuentra la media global de la intensidad; si la intensidad del pixel rebasa esa media entonces se pinta el pixel de color blanco, si es menor, se pintará de color negro(Figura 1.26). Para no quitar información relevante del espectrograma en[26] se propone que la media se multiplique por un umbral(0.1, 0.5, 0.7), para pintar más pixeles de color blanco.

Para este algoritmo, se tomará la huella como un vector de tamaño n. En la sección 3.1.3.1 se explica más acerca del tamaño y la concatenación de vectores para formar la huella.

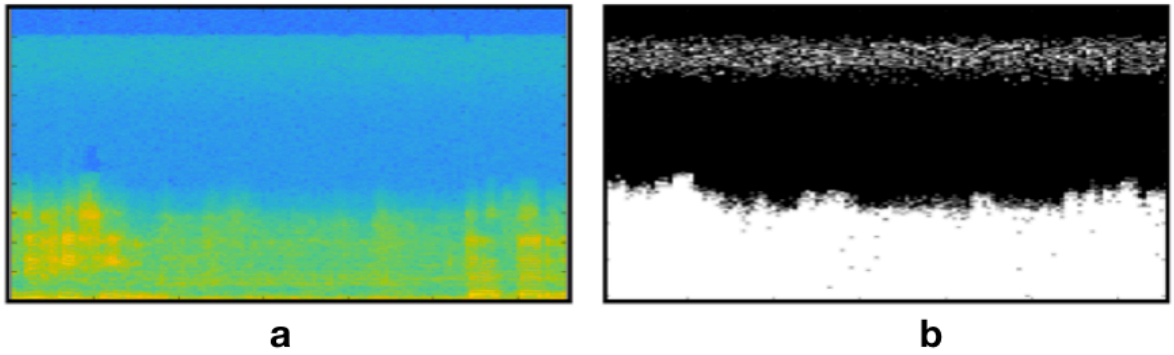


Figura 1.26 a) Representación del espectrograma b)Representación de imagen binaria basándose del espectrograma de la figura a

Por cada imagen binaria se formará una huella de tamaño n. En [26] recomienda un tamaño n=48. Se dividirá la imagen binaria en n/2 partes horizontales y después n/2 partes verticales. Se suman los pixeles en blanco de cada parte horizontal y se guarda el valor en las casillas 1 hasta n/2, del vector respectivamente. Se realiza lo mismo en las partes verticales de la imagen binaria y se colocan los resultados en las casillas n/2 +1 hasta n respectivamente así como lo muestra la figura 1.27.

Se puede observar que cada espectrograma obtenido del archivo de audio es una huella. En [26] obtiene una huella de cada 96 ms, es decir mas de 10 huellas por segundo; en [27] se propone que un audio de dure 3 minutos debería tener 10,000 huellas, es decir mas de 50 huellas por segundo; en [28] usa una huella de 25 ms, es decir 40 huellas por segundo. Se concluye que no hay un tamaño predefinido para la huella.

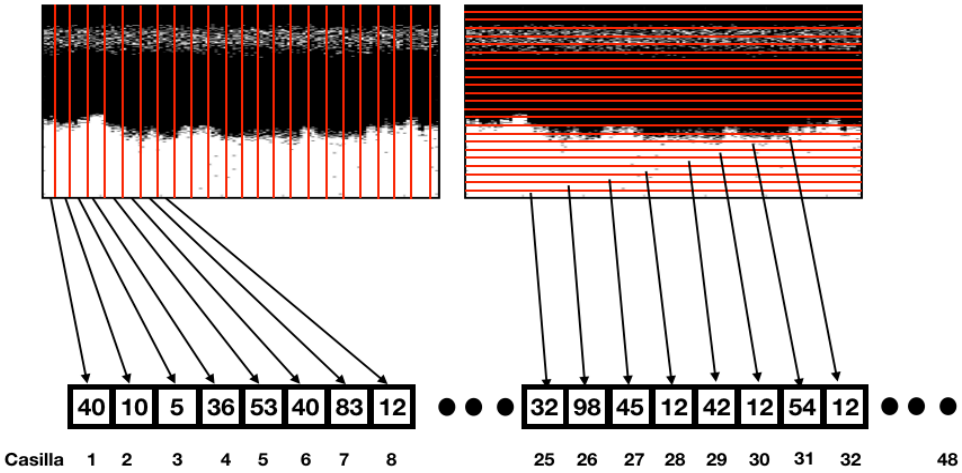
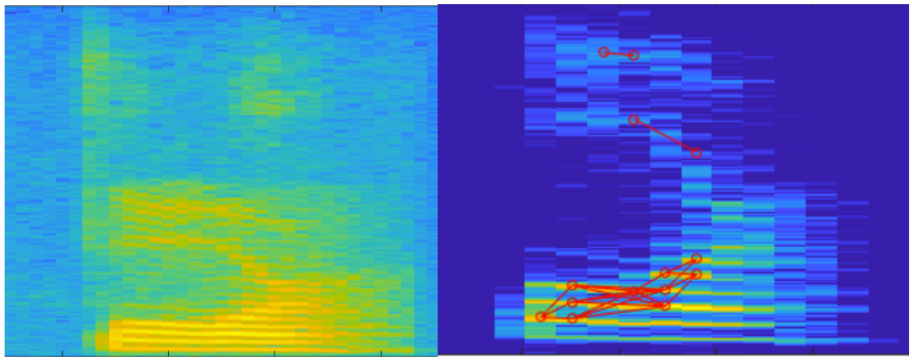


Figura 1.27 Representación del vector de tamaño n. Este vector es considerado una huella digital de audio.

1.2.4.7.2 Puntos de Referencia

Otra forma de obtener la huella digital de audio es cuando se toman los puntos de referencia, que se obtienen a partir de los picos de intensidad del espectrograma(Figura 1.28). En esta sección se explicará cómo obtener la huella a partir de estos picos. En [11] menciona que estos picos en el espectrograma tienen más probabilidad de sobrevivir a las distorsiones o a los filtros que se le puedan aplicar al audio.

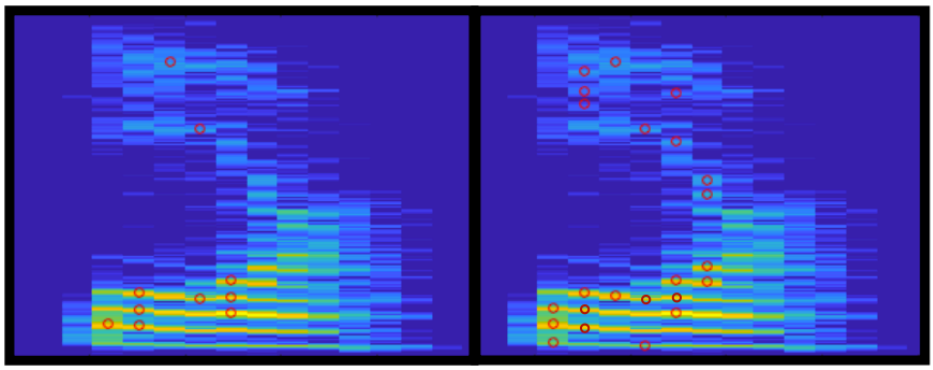


a **b**

Figura 1.28 a) Espectrograma b) Espectrograma con los puntos de referencia.

Existen varios picos de intensidad en el espectrograma, pero no todos pueden ser puntos de referencia; para que un pico sea un punto de referencia, sus vecinos deben ser puntos con intensidades inferiores. Para eliminar picos, se usará el ancho de la vecindad para ver si el pico tiene vecinos cercanos. Dentro de ese ancho de vecindad, el punto de

referencia debe tener vecinos con intensidades inferiores, por lo que si el ancho es más grande, menos puntos de referencia existirán(Figura 1.29).



a b

Figura 1.29 a) Espectrograma con 10 puntos de referencia con un ancho de vecindad=30 b) Espectrograma con 19 puntos de referencia con un ancho de vecindad=23.

Cuando se obtiene el mapa de constelaciones(posibles puntos de referencia), el patrón de puntos debe ser el mismo para el audio de consulta y el audio coincidente. Si se coloca el mapa de constelaciones de una canción de la base de datos, y el mapa de constelaciones de una muestra corta de audio coincidente, de unos pocos segundos de duración, en una pieza de plástico transparente, y luego se desliza la última sobre la primera, en algún momento significativo el número de puntos coincidirá cuando se ubique el desplazamiento de tiempo adecuado y los dos mapas de constelación estén alineados(Figura 1.30).

Comparar los mapas de constelaciones es una forma poderosa de hacer coincidir dos audios, en presencia de ruido y/o eliminación de características. Este procedimiento reduce el problema de búsqueda a un tipo de "astronavegación", en el que un pequeño parche de puntos de constelación de tiempo y frecuencia se debe ubicar rápidamente dentro de un gran universo de puntos encontrados en mil millones de segundos en la base de datos[11]. En [31] Yang propone resolver el problema con el uso de esquemas Hashing.

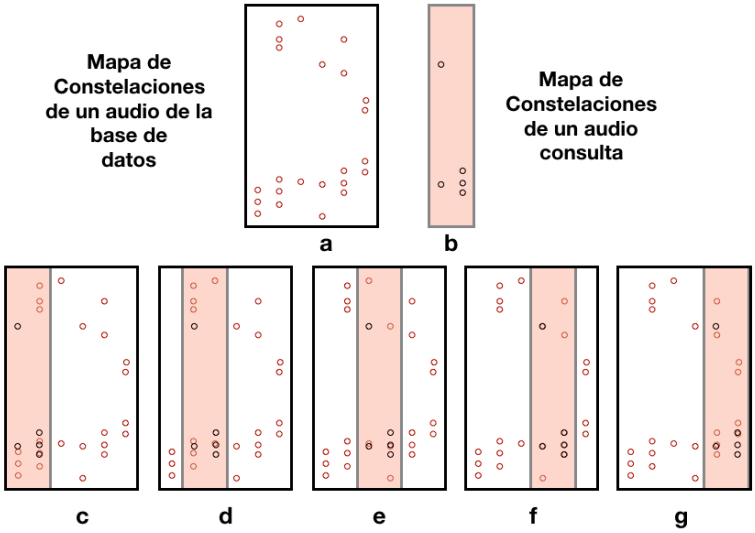


Figura 1.30 c, d, e, f y g muestra cómo se comparan los mapas de constelaciones. a y b mapas de constelaciones. f)Muestra la mejor coincidencia.

Comparar punto por punto es tardado, por lo que se debe comparar varios puntos al mismo tiempo. Este grupo de puntos se denomina zona objetivo[11]. Se divide los puntos referencia en zonas. Cada zona tiene un punto de anclaje(que puede estar o no dentro de la zona) y está delimitado por una cierta frecuencia y un cierto tiempo; ningún punto que esté fuera de este alcance puede pertenecer a la zona objetivo. Con el punto de anclaje y con cada punto dentro de la zona se formará un arreglo de 3 casillas; en la primer casilla estará la frecuencia del punto de anclaje, en la segunda casilla estará la frecuencia del punto referencia del grupo, y en la tercer casilla estará la diferencia del tiempo entre el punto de anclaje y el punto de referencia.

Estas direcciones estarán vinculadas a otro arreglo de dos casillas, en la primer casilla estará el tiempo del punto de anclaje y en la segunda casilla estará el id del audio.

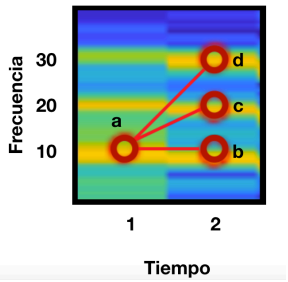


Figura 1.31 Se muestra cómo b, c y d están dentro de una zona objetivo, el ancla de esa zona es el punto a.

En la tabla 1.1 se muestra un ejemplo de cómo se forman los arreglos a partir de los puntos de referencia y el punto de anclaje de la figura 1.31.

Punto	Frecuencia del ancla.	Frecuencia del punto	Diferencia del tiempo t_ancla-t_punto	Tiempo del ancla	Id_song
b	10	10	1	1	1
c	10	20	1	1	1
d	10	30	1	1	1

Tabla 1.1 Direcciones de los hashes unidos al arreglo donde se encuentra el id de la canción con respecto a la figura 1.31.

Para encontrar la huella similar en la base de datos, se hace los mismo que cuando se comparan los mapas de constelaciones, es decir que en cierto tiempo los hashes de consulta y los de un audio de la base de datos van a coincidir. A diferencia que en este cotejamiento se comparan dos puntos de referencia. Se debe tener en cuenta que si no hay puntos similares, no se encontrará ninguna huella en la base de datos.

CAPÍTULO 2

TRABAJO RELACIONADO

2.1 TRABAJO RELACIONADO

En esta sección se explican algunos de los trabajos que se relacionan al uso de la huella digital, para que el lector se familiarice con lo que se ha hecho hasta el día de hoy.

El objetivo principal de la huella digital de audio es establecer una igualdad de dos objetos multimedia, no comparando los objetos en su totalidad, si no comparando pequeñas fracciones de ellos. Las ventajas del uso de la huella son:

- Reduce memoria en el almacenamiento, ya que las huellas son pequeñas en comparación con los objetos multimedia.
- El ruido o las frecuencias irrelevantes son eliminadas de las huellas.

Generalmente un sistema de huella digital consiste de dos componentes, un método de extracción de huellas y un método de búsqueda eficiente para encontrar la huella similar en una base de datos.[9]

2.1.1 Huella digital vs marca de agua

La extracción de información a partir de un audio se ha llevado a cabo a lo largo de varias décadas; anteriormente era muy común usar las marcas de agua(una técnica para ocultar información dentro del archivo sin modificar la calidad [4]) para poder preservar los derechos de autor de un audio y que éste no sea usado de manera ilícita.

La marca de agua genera excelentes resultados cuando se trata de comparar el archivo completo y sin distorsiones. La complejidad de la marca de agua permanece constante a comparación con la huella, que su complejidad aumenta a medida que la base de datos crece[7], pero en [2] muestra que la huella de audio es menos vulnerable a los ataques ya que si se trata de modificar esta huella, se puede alterar la calidad de sonido, mientras que si se modifica la marca de agua, ésta podría dejar intacta la calidad del audio, y éste podría ser usado de manera no autorizada y ninguna computadora lo detectaría. Por lo que en cuestión de fiabilidad, la huella digital es mejor.

2.1.2 Fases para la obtención de una huella digital

En [2] se presenta una forma general de cómo obtener la huella digital de un audio, tal como se puede observar en la figura 2.1.

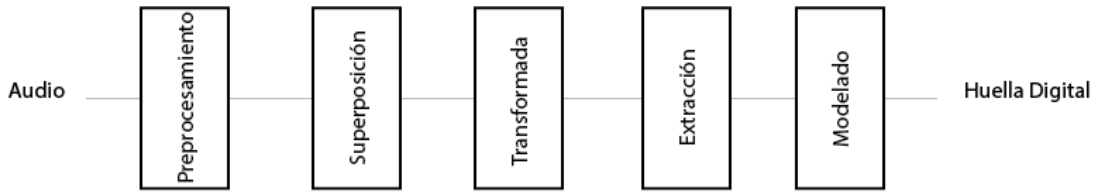


Figura 2.1 Forma general de modelar una huella.

Para encontrar la huella digital en un audio es necesario aplicar un pre-procesamiento, en donde se usan filtros para quitar ruido(frecuencias no deseadas), o también para reconstruir la señal, si ésta tuvo pérdida de información debido a la digitalización.

El pre-procesamiento se realiza de distintas maneras dependiendo la necesidad. En [6] se sugiere aplicar un filtro pasa alta para quitar el ruido(en este caso el ruido son embarcaciones por debajo de los 1000Hz). En [5] se usa el pre-procesamiento para convertir la entrada a una señal mono y reducir la frecuencia de muestreo eliminando información que no es relevante para la percepción humana y así poder enfocarse en las características importantes de la señal. En [2] se sugiere que el audio necesita simular un canal(band filtering), para poder aplicar una normalización o una pre-énfasis.

En la superposición la señal se divide en cuadros de un tamaño comparable a la velocidad de variación de los eventos acústicos subyacentes, ya que en [2] sugiere que la señal puede considerarse como estacionaria, es decir que son constantes en sus parámetros estadísticos sobre tiempo en un intervalo de unos pocos milisegundos.

Como la señal depende del tiempo, ésta se transforma para que dependa de la frecuencia. La Transformada Discreta de Fourier(DFT), la Transformada Rápida de Fourier(FFT), la Transformada de Fourier en Tiempo Corto(STFT), la Transformada Wavelet, entre otras, realizan esta conversión. Como la señal es discreta no se puede utilizar una transformada de funciones continuas, por lo que se tiene que usar transformadas discretas. Si se considera usar la STFT, al dividir la señal es necesario minimizar las discontinuidades del principio y del final de los intervalos, por lo que es necesario aplicar una función ventana a cada lapso.

Transformada la señal, se procede a construir un espectrograma. Un espectrograma es una representación gráfica del dominio de la frecuencia, en el que partiendo de éste se pueden extraer características y modelar una huella

digital(fingerprint). Existen técnicas que se pueden usar para la extracción de características del audio. En [3] usa PCA(Análisis de Componentes principales), ya que reduce la dimensionalidad de un conjunto de datos. Para extraer las características del llanto de los bebes usa varios algoritmos tales como el LPC(Codificación Predictiva Lineal) que es una de las técnicas más usadas en la codificación de voz; la MFCC(Coeficientes Cepstrales en las Frecuencias de Mel) que son los coeficientes para la representación del habla basados en la percepción auditiva humana; también usa un cocleograma que es el que representa los patrones de excitación de la membrana basilar(en el oído interno). En [11] se obtiene la huella a partir de los puntos de referencia que se basan a partir de los picos de intensidad. En [5] se obtienen la huella a partir de los puntos de referencia, que se obtienen a base de los picos espectrales. En [26] la huella digital es un vector formado por suma de pixeles de color blanco de una imagen binaria obtenida de un espectrograma.

En el bloque de modelado de huellas se recibe generalmente una secuencia de vectores de características[2]. Para estos modelos se pueden usar las energías del audio, el ritmo de la canción(BPM), secuencias de características, entre otros. Un modelo para crear la huella es el modelo oculto de Márkov, en el que determina parámetros desconocidos u ocultos de una cadena de datos a partir de parámetros conocidos.

Cuando se deduce la huella de audio, se necesita comparar las huellas con una base de datos, como lo hace Shazam[5](una aplicación móvil permite la identificación de música, usa la huella de una señal de audio para determinar que canción se está reproduciendo), se usan redes neuronales para clasificar el audio tal como la hace [3] con las enfermedades mediante el llanto de los niños, ó como lo hace [6] mediante el clasificador KNN para determinar el tipo de embarcación. Shazam[1] usa la técnica del vecino más cercano para encontrar los mejores resultados en la base de datos, después realiza una puntuación mediante el algoritmo genético Grano Fino para determinar el audio que más se asemeja.

2.1.3 Aplicaciones de la huella digital de audio

La huella digital de audio tiene aplicaciones en la música y en la biometría de voz. En [10] se menciona que la huella digital tiene 4 aplicaciones en la música:

- Monitoreo de transmisión: Generar automáticamente listas de reproducciones de estaciones de radio, televisión o transmisión web.

- Audio Conectado: Los usuarios pueden identificar la canción presionando un botón.
- Tecnología de filtrado P2P: Las canciones, con derechos de autor, se pueden bloquear de la red para que estas no se puedan compartir o descargar de manera ilícita.
- Organización de la biblioteca de música automática: Los archivos con metadatos inconsistentes, faltantes o erróneos se pueden corregir automáticamente con los metadatos correctos.

En [33] se menciona que la huella digital de audio se usa para reconocer la voz de una persona para conocer si es quien dice ser.

Shazam[5] reconoce la canción un audio que el usuario graba con su teléfono, una vez reconocida devuelve al móvil datos tales como el nombre, el artista, el álbum, y foto de álbum. El usuario graba un audio corto y Shazam determina su huella digital. Para determinar la huella, Shazam aplica un pre-procesamiento al audio, el cual consiste en convertir el archivo de audio en un formato común para que éste pueda ser analizado; de vez en cuando este paso involucra convertir el audio a mono y reducir la frecuencia de muestreo para quitar el ruido o las frecuencias no deseadas, para enfocarse en las características importantes de la señal; para la siguiente fase se calcula cuantas muestras se deben considerar para aplicarles una transformada en el dominio de la frecuencia. Después se seleccionan características que pueden ser usadas para caracterizar el audio. De las características que se extrajeron, se deduce la huella del audio. Esta huella se convierte a valores Hash[11] y ésta se compara en una base de datos de millones de huellas de audio de canciones. Convertir la huella a valores hash, acelera la búsqueda para identificar la canción. Identificada la canción, el sistema devuelve los datos de la canción y estos aparecen en el móvil.

Existen varias aplicaciones móviles para el reconocimiento de una canción. Otra aplicación es Sound Hound; es similar a Shazam en el objetivo de reconocer una canción, la diferencia es que Sound Hound permite que el usuario además de grabar la canción, éste la pueda cantar, tararear o alguna combinación de las anteriores. La aplicación debe analizar y distinguir si la entrada de sonido es música monofónica, música polifónica, palabras habladas, sonido cantado, un zumbido, cualquier otro sonido, o una combinación de los anteriores. Cuando se determina que tipo de sonido es, se obtienen las características del archivo de audio(huella digital) y se compara con varias bases de datos para encontrar la canción correcta[12].

En [13] también se crea un sistema de reconocimiento musical basado en la huella digital de audio. Éste expone que Shazam es un aplicación móvil que basa su algoritmo en el clásico algoritmo Phillips para extraer la huella digital y que este algoritmo es robusto, pero la eficiencia es insatisfactoria; por lo que propone hacer una transformada Wavelet que es una referencia cuantitativa aplicada a la recuperación musical [14]; al usar esta transformada, el tamaño de las huellas de audio se reduciría. También propone usar el espectro de energía coclear ya que las huellas tendrían mejor precisión y robustez[15].

En [16] presenta dos aplicaciones de la huella de audio. La primera es detectar un audio duplicado, incluso si estos difieren en calidad ó en duración. La detección de audios duplicados es útil para limpiar automáticamente grandes colecciones de audio. La segunda aplicación consiste en generar una miniatura de audio; esta miniatura es una sección representativa, de 15 segundos, de la canción. Estas miniaturas ayudan a mejorar la exploración de audio de listas de canciones pequeñas o grandes. Este trabajo usa un Motor de Reconocimiento de Audio(RARE) que convierte un segmento de audio en 64 números de punto flotante(huella digital), e identifica los clips que utilizan una distancia euclidiana ponderada. RARE ha demostrado ser muy resistente a las distorsiones del audio original.

En [17] la huella de audio se utiliza para identificar rápidamente eventos de sonido recurrentes en grabaciones de varios días. Esta aplicación se basa en una técnica de huellas que usa los picos de energía en el tiempo y la frecuencia, elimina problemas de encuadre y quita los niveles de ruido de fondo. Esta técnica es muy efectiva para identificar sonidos estructurados, pero no puede encontrar repeticiones de sonidos más orgánicos. Los sonidos estructurados son sonidos que siguen un patrón, tales como el timbre del teléfono; los sonidos orgánicos son aquellos sonidos que no crea una computadora; por lo general no siguen un patrón tales como el habla.

Dentro del open source existen herramientas que ayudan a la extracción de características de un audio, entre ellas están Echoprint, Chromaprint y Dejavu. The Echo Nest liberó el código de fingerprint e identificación de canciones al cual se le llamó Echoprint, pero fue adquirida en 2014 por Spotity. Aunque en la actualidad la comunidad de usuarios está activa, BBEVA Labs concluye que el sistema necesita al menos 20 segundos para obtener buenos resultados, logrando un 89% de acierto; a parte de que solo obtiene buenos resultados si el audio es directamente obtenido de su fuente original[23]. Chromaprint forma parte del proyecto AcousticID el cual tiene 8.3 millones de audios para comparar. Chromaprint compara audios mediante el uso de espectogramas y

el procesamiento de imágenes[24]. Dejavú es un código que se encuentra en GITHUB. Su implementación está en Python; lo mismo que las aplicaciones anteriores se usa para reconocer la canción que está siendo reproducida, su algoritmo no está hecho para el reconocimiento de voz[25].

CAPÍTULO 3

EXPERIMENTACIÓN Y RESULTADOS

3.1 IMPLEMENTACIÓN

En este capítulo se presentan detalles de la implementación y resultados obtenidos al procesar audio y extraer características de la huella de audio.

En las siguientes secciones se realizan experimentos con la extracción de características de las imágenes binarias y de los puntos de referencia. Los experimentos, resultados y algunas imágenes de esta tesis se obtienen mediante el uso de la herramienta creada llamada PAEX-HUDI(Pre-procesamiento de Audio para la extracción de la Huella Digital).

PAEX-HUDI permite pre-procesar señales de audio, analizarlas, extraer huella y realizar experimentos con diferentes audios comparando su huella digital. En la figura 3.1 se observa la pantalla principal de la herramienta, donde aparece en la parte superior el menú donde el usuario puede cargar un audio(menú archivo), pre-procesar, analizar y realizar experimentos.

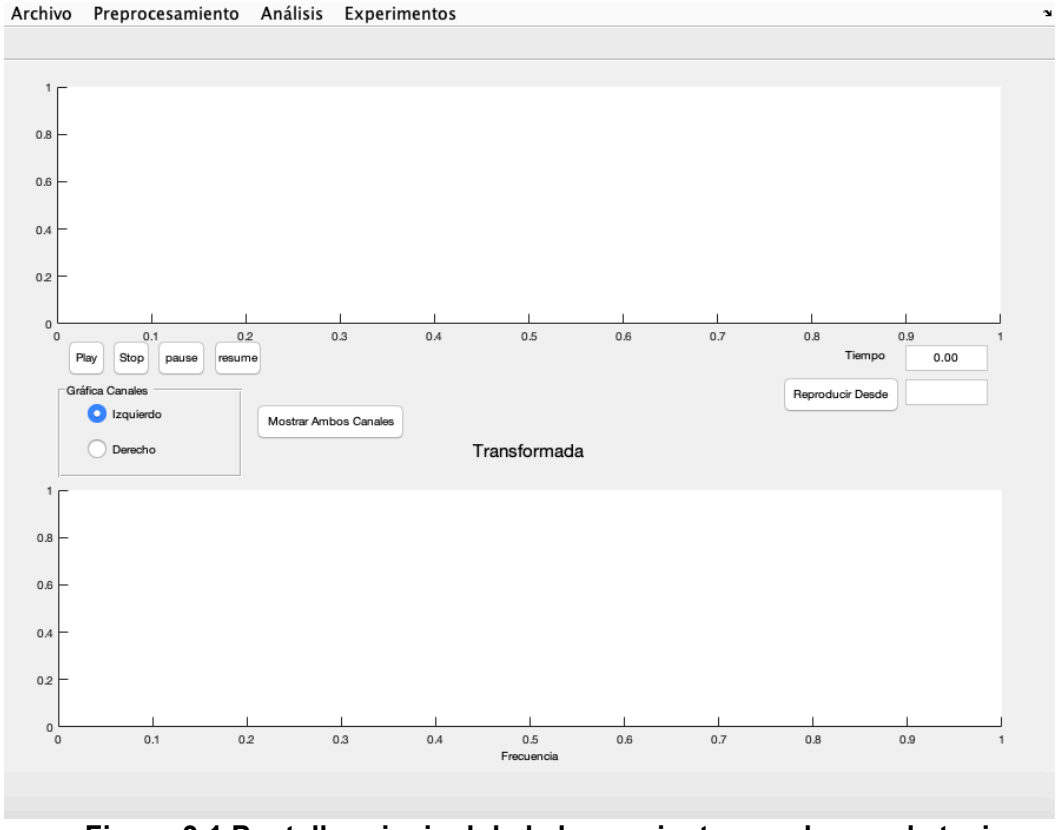


Figura 3.1 Pantalla principal de la herramienta creada para la tesis

PAEX-HUDI se programó en matlab; este lenguaje permite el uso de funciones predeterminadas para obtener la transformada de Fourier y el espectrograma; se usaron

funciones de procesamiento de imágenes y de audio tales como convertir matrices a imágenes y la aplicación de los filtros Butter, Chebyshev y Window Sinc. Para la comparación de huellas en el método de imágenes binarias, se creó una base de datos con el driver JDBC para MySQL con la herramienta database toolbox.

PAEX-HUDI le permite al usuario visualizar la señal y la transformada de un audio en un cierto tiempo para que éste determine si se debe aplicar un pre-procesamiento al audio. En el pre-procesamiento se puede aplicar la conversión estéreo a mono y se pueden aplicar los filtros vistos en el capítulo 1 (figura 3.2). El usuario también puede analizar el audio como espectrograma y a partir de éste modelar la huella digital mediante los métodos vistos en el capítulo 1. Al analizar el audio, la herramienta le permite al usuario visualizar la huella en un archivo (imágenes binarias) o en una imagen(puntos de referencia) (figura 3.3), para determinar si se debe modificar parámetros como el umbral, el ancho de vecindad, zona objetivo o los máximos por frame dependiendo el método. Por último PAEX-HUDI le permite realizar experimentos para los dos métodos de modelado de huella. El usuario debe seleccionar los audios con los que se experimentará y el programa determinará las huellas que son semejantes(figura D.7 y D.8)Visualice el apéndice D para ver una descripción mas detallada del sistema.

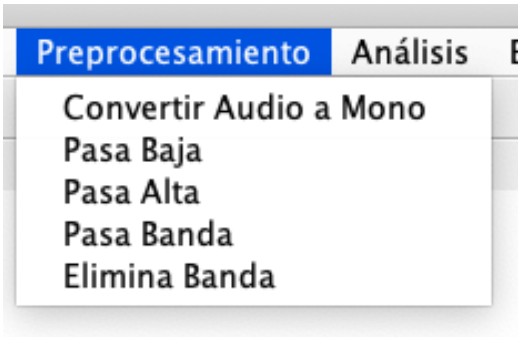


Figura 3.2 Menú Pre-procesamiento

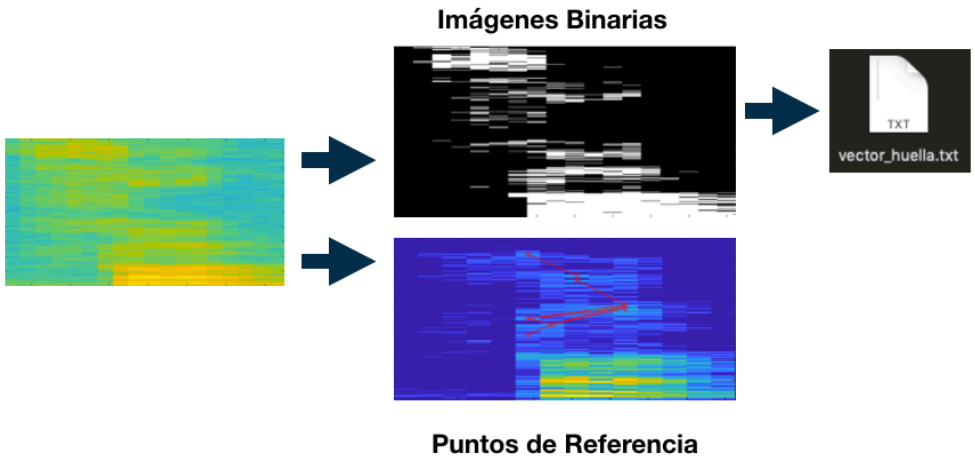


Figura 3.3 Visualización de huella en las imágenes binarias y en los puntos de referencia.

3.1.1 Pre-procesamiento

En la fase de pre-procesamiento, los filtros permiten el paso de las frecuencias deseadas y elimina las frecuencias no deseadas; como se puede observar en la sección 1.2.3 existen 4 filtros que se pueden aplicar; para poder aplicarlos es necesario hacer uso de los filtros no ideales; para esta tesis se propone el uso de Window Sinc, Butterworth y Chebyshev.

- Window Sinc: Para poder aplicar este filtro se requiere los M puntos a truncar, la frecuencia de corte Wn(depende el filtro, si es pasa baja o pasa alta solo se requiere una frecuencia de corte, si es pasa banda o elimina banda se requieren dos frecuencias de corte) y la ventana para evitar las ondulaciones en la banda eliminada.
- Chebyshev y Butterworth: Como se vio anteriormente Butterworth nace de Chebyshev, por lo que usan los mismos requerimientos para poder aplicarlos. Se requiere el orden del filtro N y la frecuencia de corte Wn(igual que en Window Sinc, éstos necesitan una o dos frecuencias dependiendo del filtro no ideal). Mientras mayor sea el orden, el filtro produce una mejor caída como se observa en el apéndice D, pero si se necesita crear el filtro a base de las atenuaciones(Rs, Rp), la ó las frecuencias de corte y las frecuencias en donde el filtro se empieza a aplicar Ws, se usan las fórmulas 12, 13 y/ó 14, para obtener el orden y que el filtro respete las especificaciones necesarias.

En la figura 3.4 se puede observar el espectro del audio de la canción “Returning Empty Handed- Underotah” filtrado con un pasa banda de 300 a 5000 Hz. Se puede observar en la figura 3.4a el filtro Window Sinc con 10 puntos truncados y una ventana Blackman. En la figura 3.4b se observa el filtro Chebyshev con un orden N=4. En la figura 3.4c se observa el filtro Butterworth con atenuación Rp=0.005 y Rs=0.007 y frecuencias Ws1=200 y Ws2=5500. Esto da como resultado un orden N=2.

Cabe recalcar que esta fase no es necesaria, pero esto depende del audio. Si se supone que los audios que se quieren analizar se encuentran grabados en una cabina de audio especializada, donde no existe el paso del ruido, no es necesario aplicar filtros; si se aplica un filtro innecesariamente, puede afectar las frecuencias y el análisis del audio.

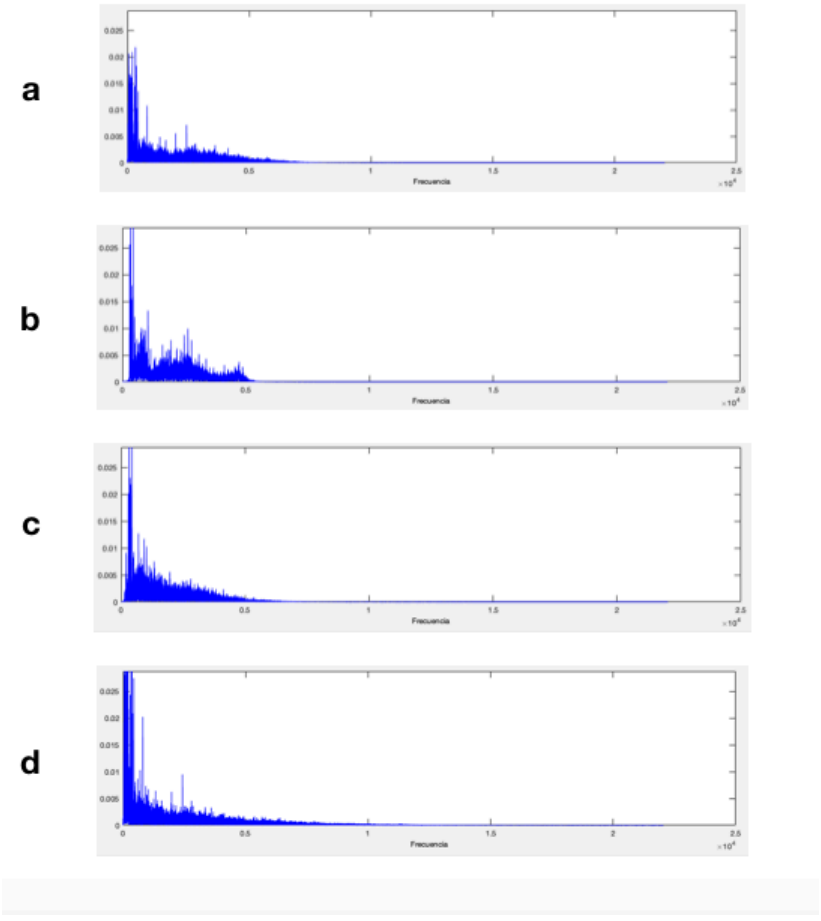


Figura 3.4 a)Espectro de un audio filtrado con pasabanda Window Sinc , b)Chebyshev, c)Butterworth. d)Espectro sin filtro del audio.

3.1.2 Espectrograma

Cuando se crea un espectrograma, es necesario tener en cuenta el tipo y tamaño de la ventana, y la cantidad de muestras traslapadas.

- La ventana se utiliza para poder aplicar la transformada de Fourier en tiempo corto, como se explica en la sección 2.1.4.5 de una muestra de audio se obtendrán varios espectros y la unión de esos espectros dará como resultado el espectrograma.
- Las muestras traslapadas darán una mejor resolución en las intensidades en el eje de la frecuencia como se explica en la sección 2.2.4.6.

Si la función 1.17 se visualiza en forma matemática, ésta es una función con dos variables independientes, pero si se visualiza en forma computacional, ésta es una función con dos parámetros, en otros términos son las entradas de una matriz $X[m,w]$, donde m señala el tiempo que ha transcurrido; m tiene un valor inicial igual a 1 y un valor máximo equivalente al tamaño de la ventana. La variable *w* es la encargada de señalar la frecuencia, la cual tiene un valor inicial igual a 1 y un valor final a 22000. La matriz $X[m,w]$ se puede visualizar como una tabla, donde *m* es el número de filas y *w* el número de columnas, y a su vez se puede visualizar como una imagen donde *m* es el ancho de la imagen y *w* es la altura; cada pixel de la imagen equivale a cada celda de la matriz X, sólo que en la matriz cada celda representa en número la intensidad del espectro y en la imagen cada pixel representa con un color la intensidad de la frecuencia, un color amarillo brilloso para frecuencias altas y azul fuerte para frecuencias bajas. El espectrograma no es más que una forma visual de representar los datos de una matriz(figura 3.5).

Cuando se obtiene cada valor de las casillas de la matriz X, se usa la fórmula 1.17 donde m=1 hasta T, donde T es el tamaño de la ventana y w=1 hasta 22000, por lo que se tendrán 22000T casillas por cada espectrograma. No se debe olvidar que m indica el tiempo que la señal ha transcurrido, pero ésta no se mide en segundos o milisegundos, ésta se mide en muestras. En el capitulo 1 se menciona cuanto debe valer la frecuencia de muestreo (*fs*). ¿Pero qué significa *fs*?. La frecuencia de muestreo significa que por cada segundo se tendrá *fs* muestras, por lo que si *fs*=8000, se tendrán 8000 muestras por cada segundo ó bien cada muestra vale 0.000125 segundos.

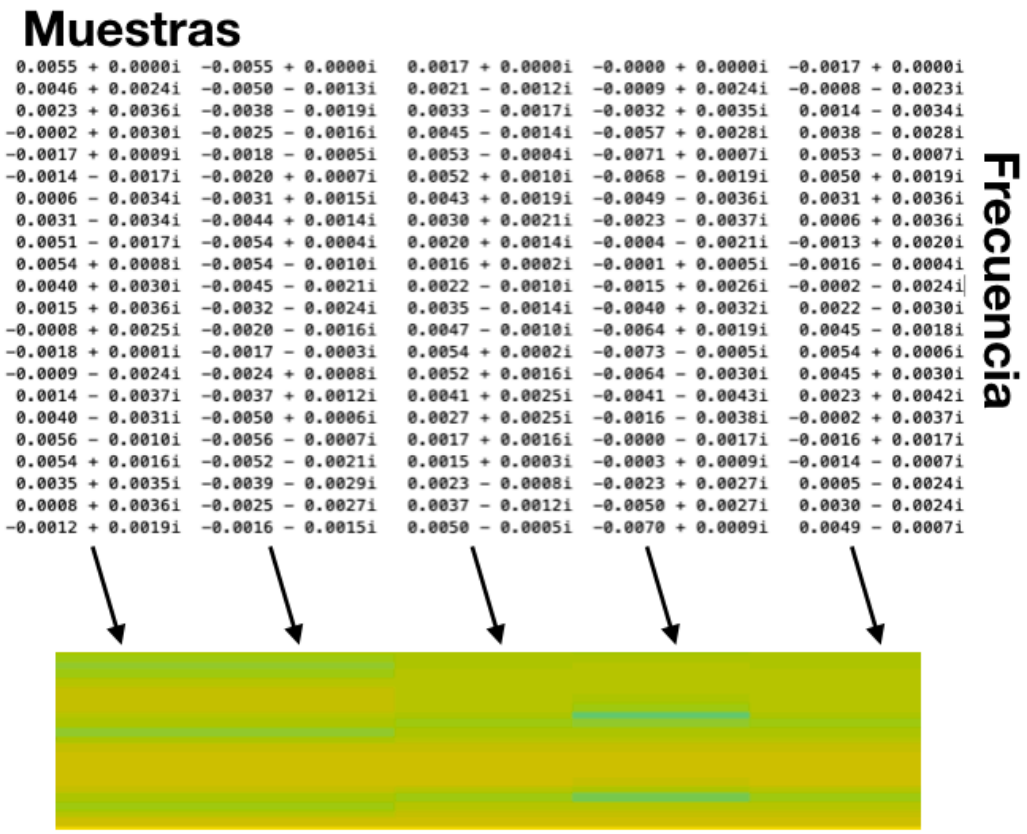


Figura 3.5 Se observan los datos de una matriz visualizados en un espectrograma.

3.1.3 Extracción de características y modelado de la huella digital

Cuando la señal de audio se ha transformado y se ha creado su espectrograma, ésta se puede analizar para obtener las características, o bien los puntos del espectrograma que permitirán modelar la huella digital de audio. En esta sección se juntan las dos fases ya que el modelado de la huella depende del algoritmo de extracción de características.

3.1.3.1 Modelado a partir de imágenes binarias

Como se mencionó en el capítulo 1 se dice que una huella digital de audio puede ser un conjunto de varias huellas, pero esta definición puede ser confusa para el lector, por lo que antes de adentrarnos al modelado a partir de imágenes binarias, se aclarará este punto.

En el capitulo 2, se menciona que una huella es un extracto de un audio, la cuál ayuda a no comparar un audio o un objeto multimedia con otros en su totalidad, si no se compararán pequeñas fracciones de éstos para determinar que tan semejantes son.

Se considera en este ejemplo la huella digital de un audio como un vector de 48 casillas. Se supone un audio de 10 segundos con un *fs* = 8000, si se obtiene la huella digital de todo el audio, el espectrograma sería una matriz de 22000*8000*10=1760000000=1.76*10^9 datos, al convertir el espectrograma a una imagen binaria, la imagen tendría 1.76*10^9 pixeles, y de ésta se obtendría la huella, por lo que se tendría 1.76*10^9 datos analizados y almacenados en nuestra huella de 48 casillas. Lo que se pretende mostrar es que un conjunto de datos demasiado grande se reduce a 48 casillas; esto podría generar que audios que no se parecen tengan la misma huella digital.

Se puede evitar el problema anterior dividiendo el audio de 10 segundos en 10 segmentos, como lo muestra la figura 3.6. En la figura 3.6a se puede notar cómo de un audio de 10 segundos se obtiene un vector, y en la figura 3.6b se puede observar cómo del mismo audio se obtienen 10 vectores, por lo que para esta figura la huella sería un vector con 480 casillas(tomando en cuenta que en el capítulo 1 se menciona que por cada segmento se obtiene un espectrograma, y del espectrograma se obtiene una imagen binaria y de cada imagen binaria se extrae un vector de 48 casillas; por lo que si el audio se dividió en 10 segmentos, se tendrá 10 vectores de 48 casillas, de ahí se dice que la huella digital de audio de la figura 3.6b tiene 480 casillas). Esta división reduce la posibilidad de que audios diferentes tengan la misma huella digital.

Cuando se inicia la extracción de características del espectrograma, éste se debe convertir a una imagen binaria, por lo que se necesita obtener el espectro *E* de cada valor de la matriz $X[m,w]$ para eso se utiliza la fórmula 1.17. Una vez obtenido el espectro de la matriz, se necesita normalizar como lo hace [29] para quitar propiedades, en particular el brillo(fórmula 3.1) y el contraste y para poder trabajar con los valores dentro del rango [0,1]. Para normalizarla matriz se usa la fórmula 3.2.

$$E[m,w] = modulo^2(X[m,w]) \tag{3.1}$$

$$Xn[m,w] = \frac{E[m,w] - minimo(E)}{maximo(E) - minimo(E)} \tag{3.2}$$

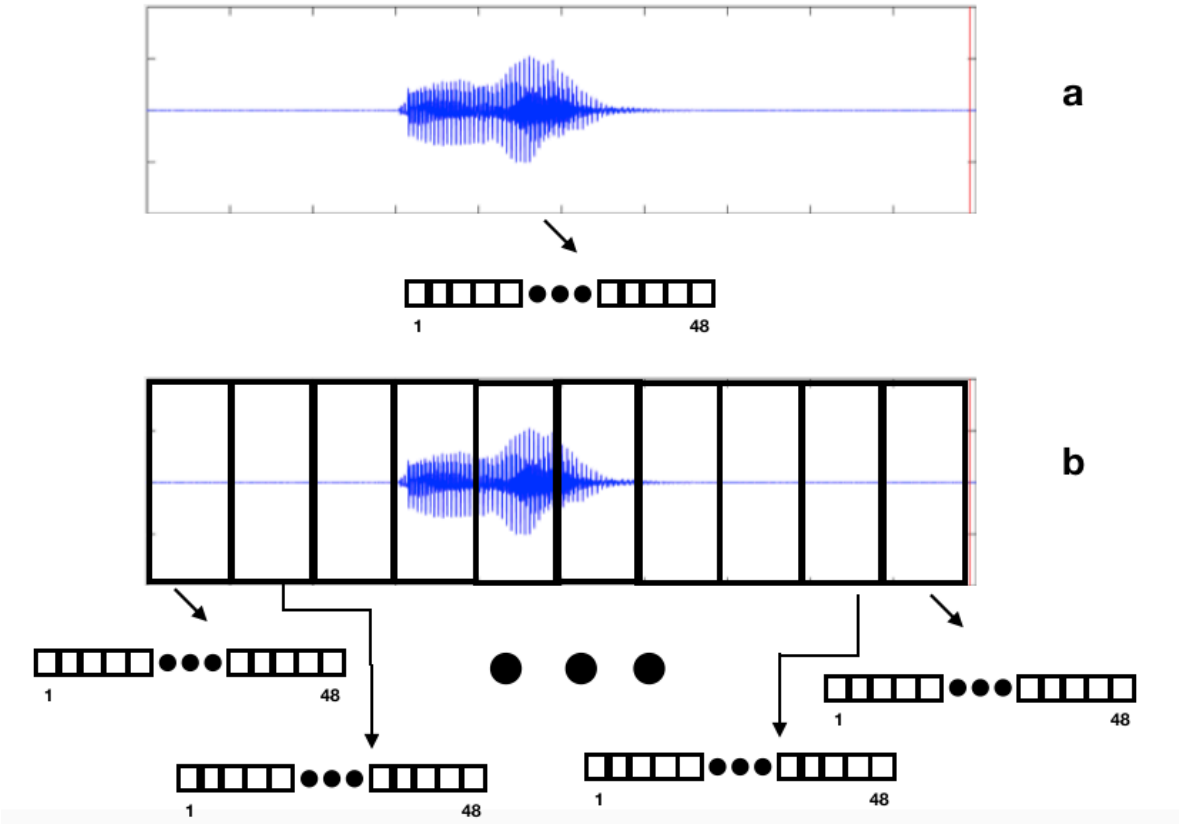


Figura 3.6 a)Un audio con una huella b)El mismo audio de a con 10 huellas de un audio.

En este momento la matriz está normalizada por lo que se obtiene la media(*med*) con la fórmula 3.3, para que en la imagen binaria (I[*m*,*w*]) se pinten los pixeles de color blanco cuando las casillas $Xn[m,w]$ rebasen el valor de la media; si este valor no es rebasado, el pixel se deberá pintar de color negro. En este paso se usa una variable conocida como umbral. Esta variable se multiplica por la media y dependiendo el valor permite pintar más o menos pixeles de color blanco.

$$med = \frac{\sum_{i=1}^{m}\sum_{j=1}^{w} Xn[m,w]}{m * w} \tag{3.3}$$

donde *m* es el tamaño de la ventana, *w* son las frecuencias y *Xn* matriz normalizada.

Una vez obtenida la imagen binaria, se debe crear la huella, es decir se debe crear el vector de 48 casillas. Se divide la imagen en 24 filas y en 24 columnas. Cada suma de pixeles blancos de cada fila o columna se guarda en una casilla del vector. Se tiene 48 sumas(24 sumas de las filas y 24 sumas de las columnas), por lo que cada suma corresponde a cada casilla del vector.

Cuando se divida la imagen binaria la última fila y la columna, no tendrá el mismo ancho que las demás filas y columnas respectivamente. Para que esto suceda *m* y *w*

deben ser múltiplos de 24 en la creación del espectrograma, lo cual no será del todo posible ya que se tienen w=22000 frecuencias y este número no es múltiplo de 24.

Creado el vector, se almacena en la base de datos. La herramienta que se creó reconoce si se tienen uno o varios vectores y por cada vector se crea un id diferente compuesto del nombre del audio + guión bajo + número del vector. Ejemplo si se tiene el vector 23 del audio 3.wav, el nombre quedaría 3_23.

Se crea una base de datos para almacenar todas las huellas o vectores de los audios que se van a analizar. La base de datos tiene una tabla que tiene 49 columnas; la primer columna(id) es el identificador del vector y es la llave de la tabla; esta columna es de tipo carácter con un tamaño de 7. A partir de la segunda columna y hasta la 49 se almacenan las casillas del vector, por lo que estas columnas son de tipo entero. En la figura 3.7 se puede observar cómo se almacenan las huellas en la base de datos. Se ve que si sólo es un vector, el id es sólo un número, pero si la huella tiene varios arreglos, entonces el id es la unión del número de la huella, seguido de un guión bajo, seguido del número del vector.

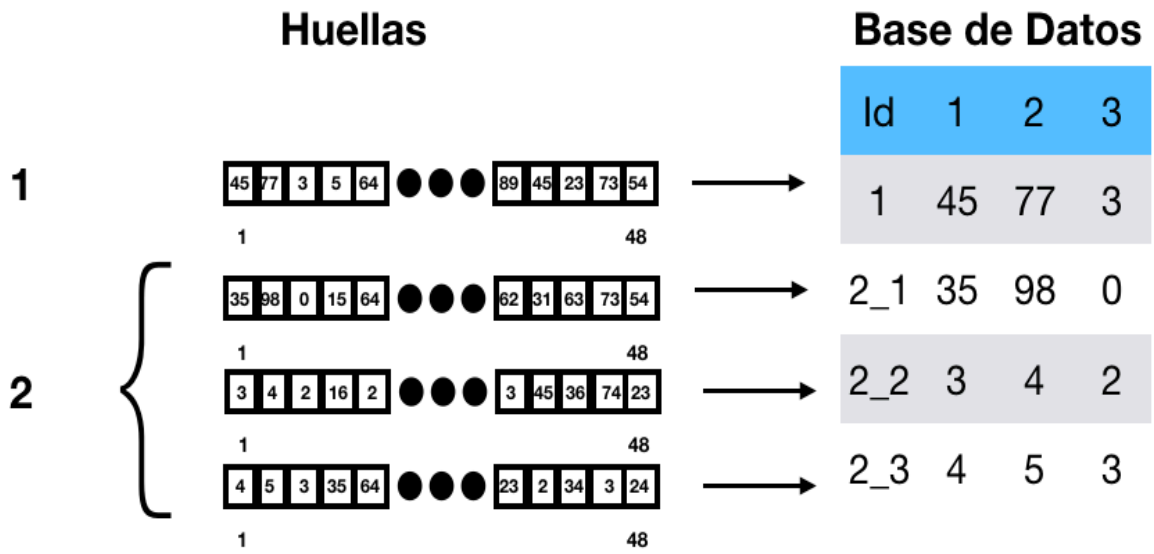


Figura 3.7 Se muestra cómo se almacenan las huellas en la base de datos

Antes de experimentar, se necesitan dos tipos de audios; uno es el audio que se va a comparar y el segundo tipo es o son los audios con los que el primer tipo se va a comparar. Cuando se obtienen las huellas de los archivos,se almacenan en la base de datos. Se debe tomar en cuenta que el mismo pre-procesamiento aplicado al audio que se va a comparar, se debe aplicar al audio o a los audios con los que se va a comparar. Cuando se experimenta se extraen las huellas de la base de datos, se concatenan en el

caso de que los audios tengan varias huellas para formar una sola huella, y se comparan casilla con casilla la huella de cada audio de tipo dos, con la huella de tipo uno.

La comparación de casilla con casilla de las huellas de los audios se realiza de la siguiente forma se toma v1 como la huella de tipo 1, v2 como la huella de tipo 2, t1 el tamaño de v1, t2 el tamaño de t2 y j=1, c=0:

1. Si las huellas son del mismo tamaño, se compara cada par de casilla de v1 y v2 desde la casilla i=1 hasta la casilla i=t1

$$v1(i+c) == v2(i) \tag{3.4}$$

   Si cumple la comparación entonces el contador incrementa casillas_iguales(j)=casillas_iguales(j)+1.
2. La huella v1 se necesita mover por todo v2 para ver si la huella v1 se asemeja de una mejor manera, por lo que se repite el paso 1 incrementando c=c+1 y j=j+1.
3. Cuando se ha movido v1 por todo v2, se toma el mayor valor del vector casillas_iguales() y ese valor indica cuantas casillas tienen de semejanza v1 y v2

En el caso de que el tamaño de v1 > v2, i=1 hasta t2. Y para el caso de que el tamaño v2<v1, i=1 hasta t1. Para ambos casos se repite el mismo algoritmo.

El algoritmo anterior permite comparar la exactitud de las casillas. Si las casillas que se están comparando no son iguales, entonces el contador casillas_iguales(j) no se incrementa. Si no se requiere comparar la exactitud de las casillas y lo único que se quiere es ver si las casillas a comparar son parecidas, se usa una variable llamada desfase *d*. Es decir, si la casilla v1(i1)=1030 y v2(i1)=1031, se puede notar que no son iguales; el algoritmo anterior no incrementaría el contador, pero se puede observar que la diferencia no es mucha por lo que se puede concluir que la comparación no es exacta pero es similar. Esta variable de desfase se le sumará y restará al valor de v1(i+c) para que si v2(i) está dentro del rango, la variable de casillas_iguales(j) se incremente, así como se puede visualizar en el siguiente condicional.

$$Si\ (v2(i) < v1(j+c) + d)\ y\ (v2(i) > v1(j+c) - d)$$
$$casillas\ iguales(j) = casillas\ iguales(j) + 1 \tag{3.5}$$
$$Fin\_si$$

Cada máximo valor de casillas iguales se almacena en otro vector *mch(k)* (máximo comparación huellas) donde *k* representa los audios de tipo 2 que se han comparado. De este vector se toman los resultados que sonn iguales a t1, eso quiere decir que todas las casillas del audio k son iguales a t1. Las huellas que cumplan con lo anterior, son las huellas que se asemejan al audio que se está comparando.

Como se está comparando la exactitud de las huellas si t1=96 y mch(3)=95, el algoritmo de comparación rechazaría la comparación y concluiría que el audio 3 no se parece, así que aquí se utilizará otro umbral se nombrará como mci(máximo casillas iguales). Si mch(k)>mci entonces la huella del audio k es semejante a la huella del audio a comparar. El umbral mci debe ser menor o igual a t1.

3.1.3.2 Modelado a partir de puntos de referencia

En esta sección se explica el modelado de huella a partir de los puntos de referencia. Cabe mencionar que este modelado de huella no divide la huella de todo el audio, en huellas pequeñas como el modelado anterior.

A partir del espectrograma se necesita encontrar los puntos la de referencia que sean características de la señal de audio. Como se explicó en la sección 1.2.4.7.2 existen varios picos de intensidad en un espectrograma, pero no todos pueden ser puntos de referencia. Un pico espectral (t_0, f_0), donde t representa la coordenada del tiempo y f la coordenada de la frecuencia, es un punto de referencia si cumple con la ecuación 3.6:

$$P(t_0, f_0) > P(t, f) \tag{3.6}$$

donde $(t, f) \in [(t_0 - a, t_0 + a) \text{ X } (f_0 - a, f_0 + a)]$, a es la anchura de la vecindad y P determina la intensidad del punto[32]. Por lo que entre más grande sea a, menos puntos de referencia se encontrarán. En la figura 3.8 se muestra un pico de intensidad y su vecindad con el que se va a comparar.

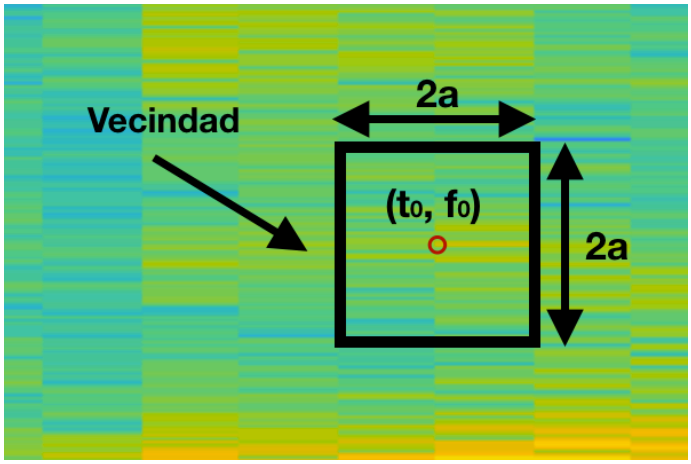


Figura 3.8 Se muestra el pico de intensidad (t_0,f_0) y la vecindad con la que se compara.

Estos puntos de referencia pueden persistir a la aplicación de filtros, por lo que en la figura 3.9 se muestran los puntos de referencia de una señal de audio de una persona; en la figura 3.9a el audio es virgen y en la figura 3.9b se le aplica un filtro pasa banda de

300 a 5000 Hz. En la figura 3.9*a* aparecen 22 hashes, mientras que en la figura 3.9b aparecen 24 hashes. Se puede observar en la parte superior de ambas imágenes la diferencia de hashes.

Los puntos de referencia obtenidos forma un mapa de constelaciones como se observo en la figura 1.30. También se explicó anteriormente que se necesita comparar un pequeño parche de puntos dentro de una base que tiene millones de puntos, por lo que en este documento se propone comparar varios puntos en lugar de comparar punto por punto. Para esta comparación se usan las tablas hash. Las tablas hash asocian valores con claves para una eficiente búsqueda de datos; en este caso, se asociarán los puntos a las filas de una tabla como se muestra en la tabla 1.1.

La unión de dos puntos será un hash. El primer punto deberá ser el anclaje y el segundo deberá ser un punto dentro de la zona objetivo como se observa anteriormente en la figura 1.31. La zona objetivo se limita por un rectángulo, donde la base es zt y la altura es zf. Las variables zt y zf indican los límites del tiempo y la frecuencia para saber si un punto (t, f) se encuentra dentro de nuestra zona objetivo. Para esta tesis el punto de anclaje será el primero de la zona objetivo, pero no formará hashes con puntos de referencia que tengan el mismo tiempo. Se supone un pico de anclaje (t_a, f_a) y un pico (t, f) formará un hash si:

$$t_a < t \leq t_a + zt \quad \text{y} \quad f_a \leq f \leq f_a + zf \tag{3.7}$$

Como se observa en la figura 3.10 los puntos (a,b) y (a,c) forman hashes mientras que (a,d) no, porque d no está dentro de la zona objetivo donde a es el punto de anclaje.

Cuando se han obtenido los hashes del audio digital, éstos se compararan con los hashes de audios que están en una base de datos. Los audios que tengan hashes similares con el audio a comparar, son los que tienen huella similar. En la figura 3.11 se muestra el audio a comparar con 32 hashes y el audio que más se asemeja en 4 hashes. Se observa el cambio de color de los hashes que son similares.

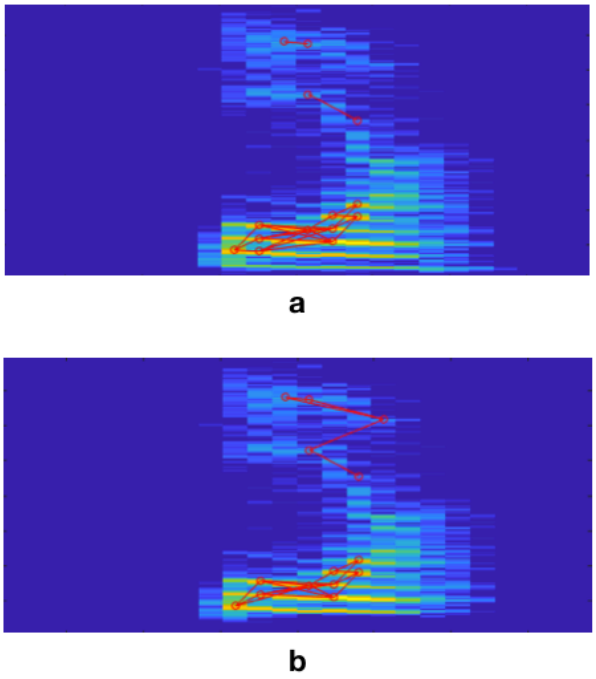


Figura 3.9 a)Hashes de audio virgen b)Hashes de un audio filtrado.

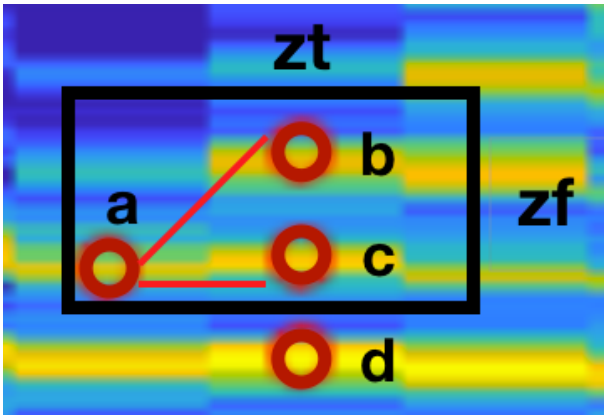


Figura 3.10 Representación de los puntos de referencia que forman hashes.

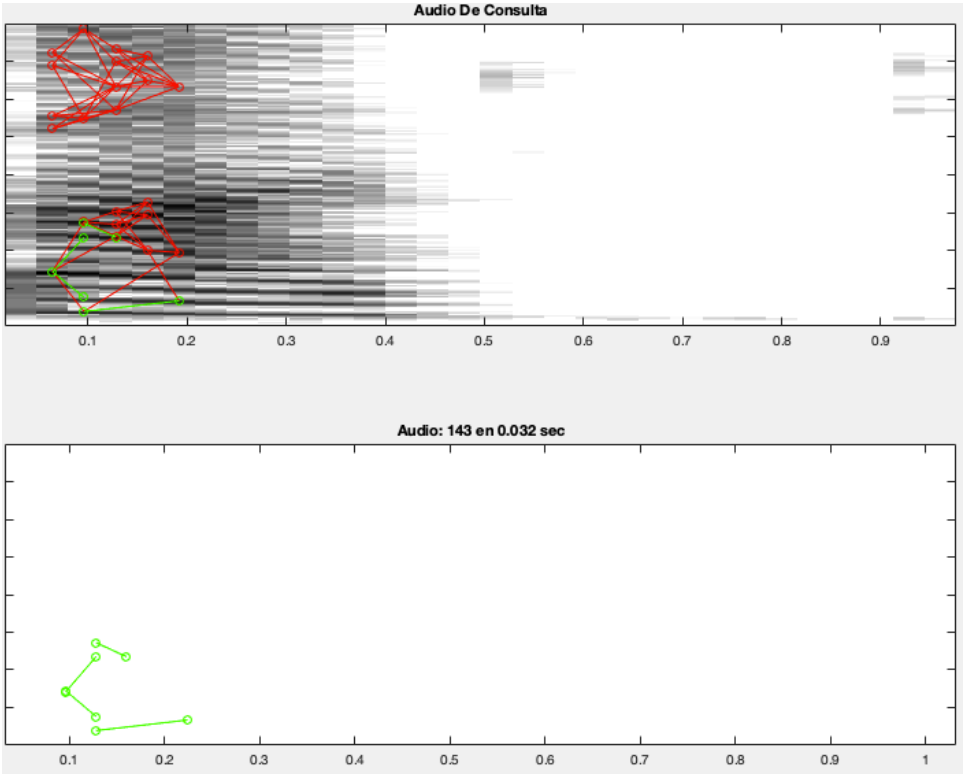


Figura 3.11 Representación de dos audios que tienen 4 hashes semejantes.

3.2 EXPERIMENTACIÓN Y RESULTADOS

En esta sección se muestran los resultados de algunos experimentos comparando las huellas digitales que existen en una base de datos, en el caso de las imágenes binarias, y en matrices, en el caso de puntos de referencia.

3.2.1 Descripción de los experimentos

En el capítulo 2 se muestran algunas fases del modelado de huella. Cada fase tiene parámetros que ayudan a quitar el ruido y dejar las frecuencias importantes. En este apartado se analizan algunas fases usando diferentes parámetros.

La primer fase es el pre-procesamiento en el que se quita las frecuencias no deseadas mediante el uso de filtros como el pasa alta, baja, banda y elimina banda. En [26] sugiere aplicar un filtro pasa banda desde los 300 Hz hasta los 5000 Hz. Este filtro se aplica con el fin de dejar las frecuencias que utiliza la voz humana. En la figura 3.12, se puede observar cómo cambia la señal, el espectro y el espectrograma de una señal sin filtro y de una señal con filtro. En las imágenes a, c y e se muestran la señal, el espectro y el espectrograma respectivamente de una señal no filtrada, mientras que en b, d y f se muestran los mismo componentes de una señal filtrada con un pasa banda de 300 Hz a 5000 Hz. Se puede notar cómo en las segundas imágenes, las gráficas tienen menos información, pues se ha quitado las frecuencias que no son necesarias, en este caso se analizará las señal de la voz humana.

En la fase de superposición se debe determinar el tamaño en el que se dividió el audio por medio de ventanas. En los primeros experimentos al ser audios de duración de un segundo, no se dividieron éstos por ser pequeños, por lo que nuestra huella digital de audio abarcaba un segundo. Para la segunda ronda de experimentos, se dividió el audio en 36 ms. En la figura 3.13 se muestra cómo se divide el audio en ventanas de 36 ms. El usuario puede dividir el audio a su conveniencia; entre mas pequeñas sean las ventanas, mas grande será la huella digital del audio. Hay que recalcar que esta fase solo está disponible para las imágenes binarias.

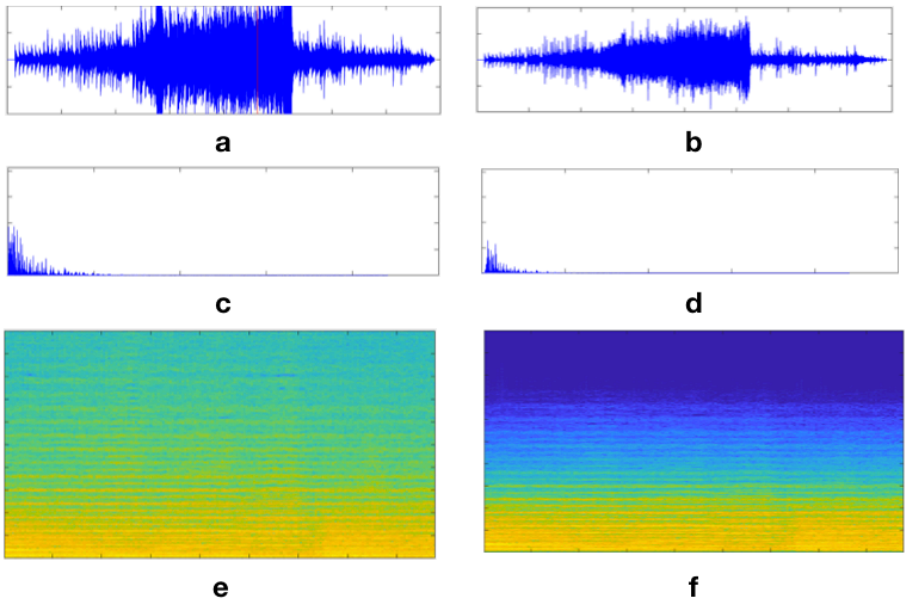


Figura 3.12 Comportamiento de la señal, espectro y espectrograma de una señal y de la misma señal sin filtro.

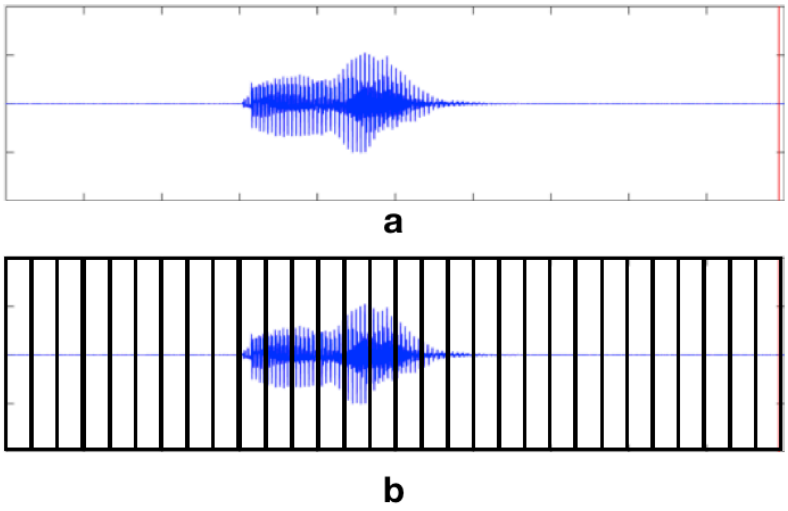


Figura 3.13 a) Se muestra una señal que no se divide b) Se muestra una señal que se ventanea cada 36 ms.

Para la fase de la transformada se usará la transformada discreta de Fourier donde se formará el espectrograma de cada división de la fase anterior. Se puede elegir una ventana rectangular, Hamming o Hann.. Otros parámetros para contemplar en la creación del espectrograma son el tamaño del espectrograma y las muestras solapadas.

Se recomienda que las muestras solapadas sean la mitad del tamaño del espectrograma, para que éste tenga una mejor visualización. En la figura 3.14a se muestra una espectrograma sin muestras solapadas y en la figura 3.14b se muestra un espectrograma con la mitad de muestras solapadas.

En las fases de extracción y modelado se puede crear la huella mediante imágenes binarias y puntos de referencia. Si se usan las imágenes binarias, en la extracción de características se debe elegir un umbral para poder formar las imágenes binarias(En la figura 3.15 se muestra imágenes con diferentes umbrales); en la fase de modelado, esta imagen se guarda en un vector de 48 casillas. Si se crea la huella con los puntos de referencia, la extracción de características dependen de parámetros tales como la anchura de la vecindad, máximos por frame, zona de frecuencia y zona de tiempo. Para la fase de modelado se eligen los puntos de referencia encontrados y se forman hashes.

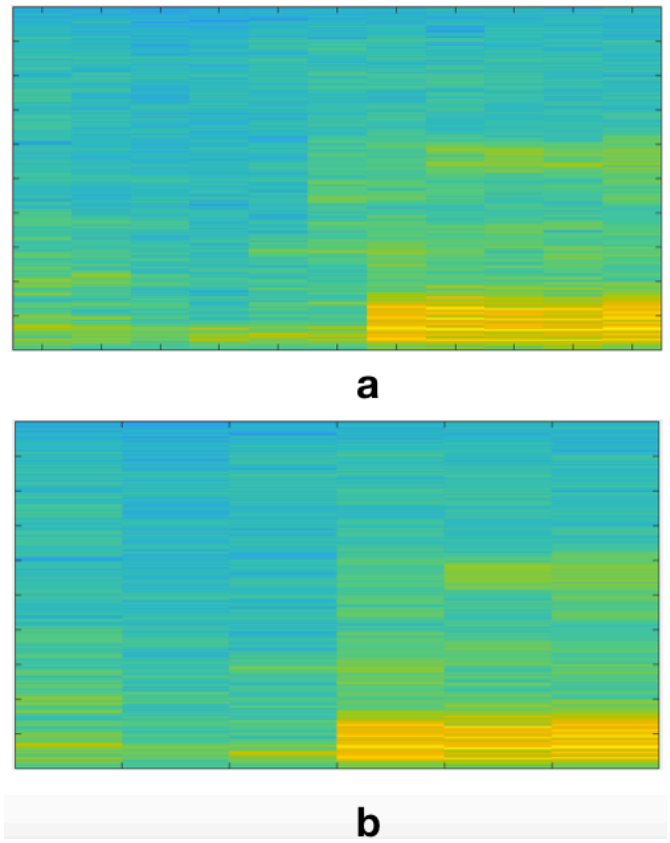


Figura 3.14 a) Espectrograma sin muestras solapadas b) Espectrograma con la mitad de muestras solapadas. Se puede notar un zoom en el eje del tiempo.

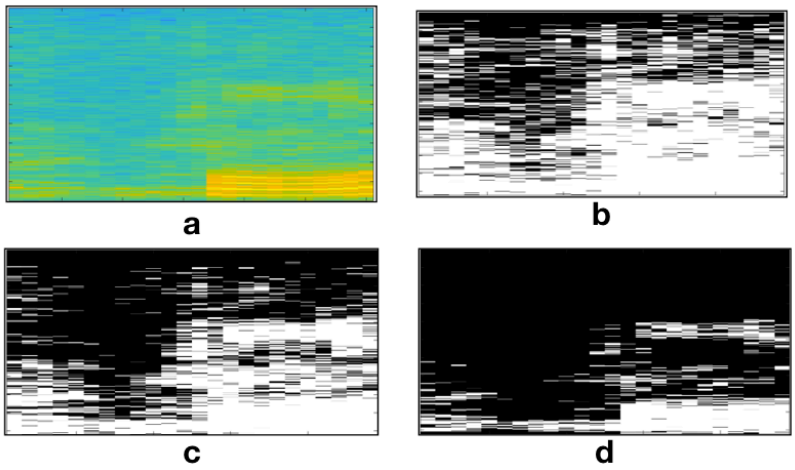


Figura 3.15 a) Espectrograma de un audio, las imágenes b, c y d son imágenes binarias con 0.9, 1 y 1.2 de umbral respectivamente.

3.2.2 Resultados

En esta sección se muestran y explican los resultados obtenidos de una base de datos de una página web(Wolfram Data Repository), donde se descargan audios de distintas personas diciendo la palabra five o four.

Se realizaron 22 experimentos usando 200 archivos de audios en los que la gente menciona la palabra five(número 5 en inglés). Los audios son de distintas personas, ya que para estos experimentos lo que se quiere es que no importando los parámetros, PAEX-HUDI no debe encontrar una huella similar. Al ser archivos de duración máxima de un segundo se optó por crear un solo vector(tamaño = 48) para la huella de cada archivo. Se configura el espectrograma con los parámetros tamaño de espectrograma=tamaño_audio/8, ventana Hamming, y la mitad de muestras traslapadas. En la figura 3.16 se observan algunos comportamientos de la huella digital que muestra PAEX-HUDI, se puede notar que los espectrogramas y las imágenes binarias tienen un parecido, pero la herramienta no detecta que sean iguales.

En los experimentos del 1 al 21 PAEX-HUDI no reconoce dos huellas como iguales ya que no existen distintos audios de la misma persona. En el experimento 22 PAEX-HUDI detecta huellas idénticas, pero esto se debe a que las intensidades de los espectrogramas de 6 audios no rebasan la media multiplicada por el umbral, por lo que toda la imagen binaria es pintada de color negro.

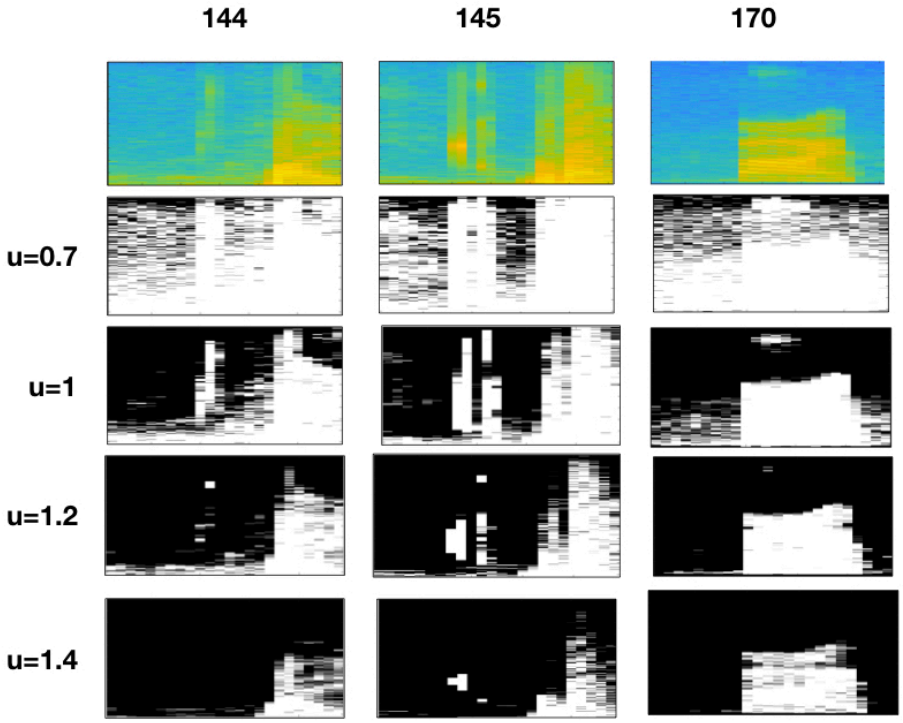


Figura 3.16 Se muestra el comportamiento de 3 audios con diferentes umbrales(u).

El umbral usado en [26] es de 0.1, 0.5 y 0.7. En los experimentos del 6 al 22 se usa un umbral mayor a 1, pero se debe tener cuidado. Antes de obtener la media global de la intensidad del espectrograma, se normaliza los valores de la intensidad para ubicarlos dentro del rango [0,1] como se realiza en [29]. En la multiplicación del umbral(0.1,0.5,0.7) con la media global, el resultado se ubica dentro del mismo rango[0,1], pero con un umbral mayor a 1 no se puede asegurar eso. Cuando se multiplica la media global por 1.5, se presentan valores mayores a 1, por lo que ninguna intensidad de los 6 espectrogramas está por encima de este valor, lo que causa que las imágenes se pinten de color negro. Se concluye que se puede usar un umbral mayor a 1, pero no es recomendable.

Se realizaron 10 experimentos donde se toman 170 audios de los experimentos anteriores, y se agregan 30 audios en los que la voz de algunas personas se repite de 2 a 4 veces. Para estos experimentos se modela una huella mas pequeña(tamaño=32 ms). Se comparan tramos mas pequeños del audio. En promedio cada huella tiene 1532 casillas. En la tabla F.2 se muestra el resultado de los experimentos. Se usa la misma configuración de la obtención del espectrograma de los experimentos anteriores; se usa el

parámetro umbral y desfase. En los experimentos del 7 al 10 se agrega un filtro Pasa Banda, para filtrar solo las frecuencias de habla que están entre 300 y 5000 Hz[26].

En estas pruebas se puede observar que no se encuentran huellas con todas las casillas iguales. En los experimentos 4, 6 y 8 existen huellas que son semejantes(tienen más de 1300 casillas semejantes). Dentro de esas huellas semejantes se pueden encontrar audios de la misma persona (en este caso el audio 141, 142 y 143 son de la misma persona y los audios 147 y 148 que son igual de una misma persona).

En el experimento 8, el par de huellas que alcanzaron el máximo número de casillas semejantes fueron el audio 142 y 143. En la figura 3.17 se observa como se asemejan los espectrogramas y las imágenes binarias de los audios 141, 142 y 143; en la figura 3.18 se visualiza lo mismo que en la figura anterior, pero de los audios 147 y 148. Las huellas de estos audios no son exactas pero alcanzan un considerable número de casillas semejantes, por lo que se puede determinar que la huella es la misma.

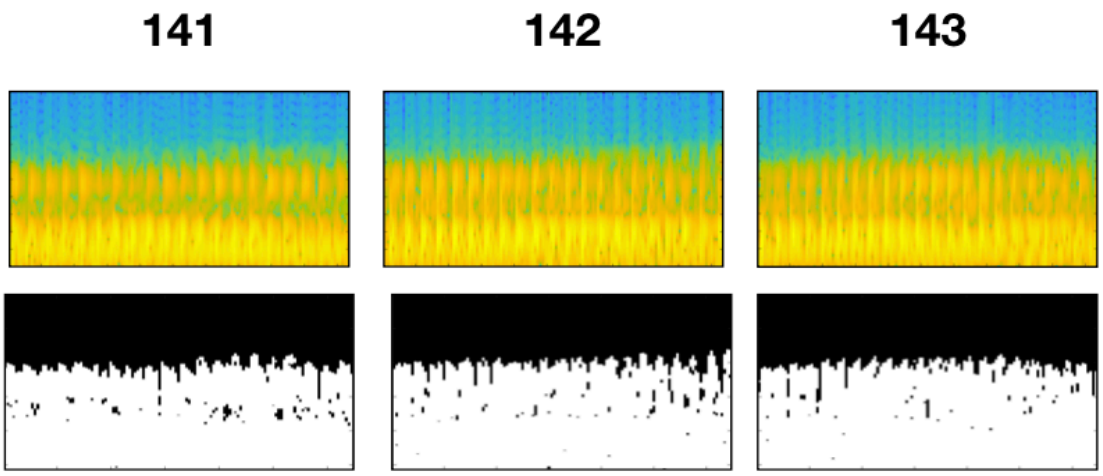


Figura 3.17 Espectrogramas e imágenes binarias de los audios 141, 142 y 143

Con los mismos datos de estos últimos experimentos se realiza el experimento de los puntos de referencia. La anchura de vecindad es 30 y 40 sólo para el experimento 2; se deben encontrar 5 picos máximos(son los picos de intensidad que se encuentran por frame), zona objetivo tiempo=63 y 73 para el 4to experimento y zona objetivo frecuencia=31 y 41 para el 4to experimento; cuando se modifican estos parámetros se modifica el número de hashes por audio y por consecuente el número de hashes por segundo. En la figura 3.19 aparecen los hashes semejantes del audio 144 y 145 y en la figura 3.20 aparecen los hashes semejantes entre el audio 147 y 148. De color verde aparecen los hashes semejantes.

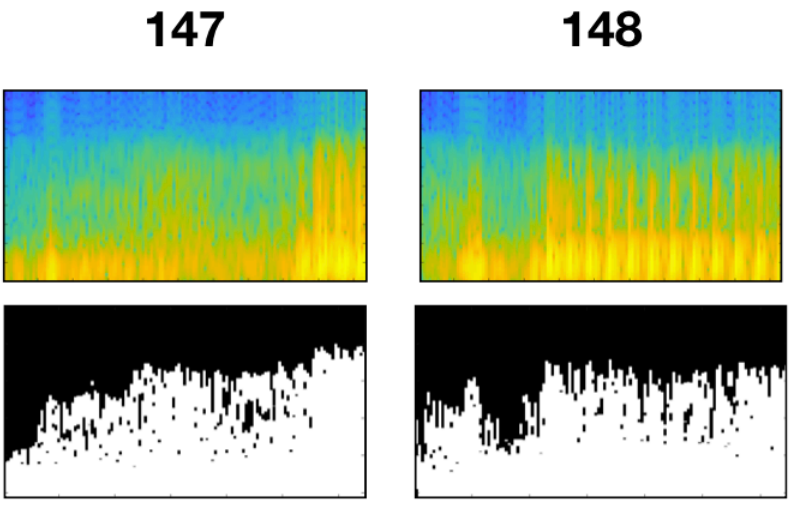


Figura 3.18 Espectrogramas e imágenes binarias de los audios 147 y 148

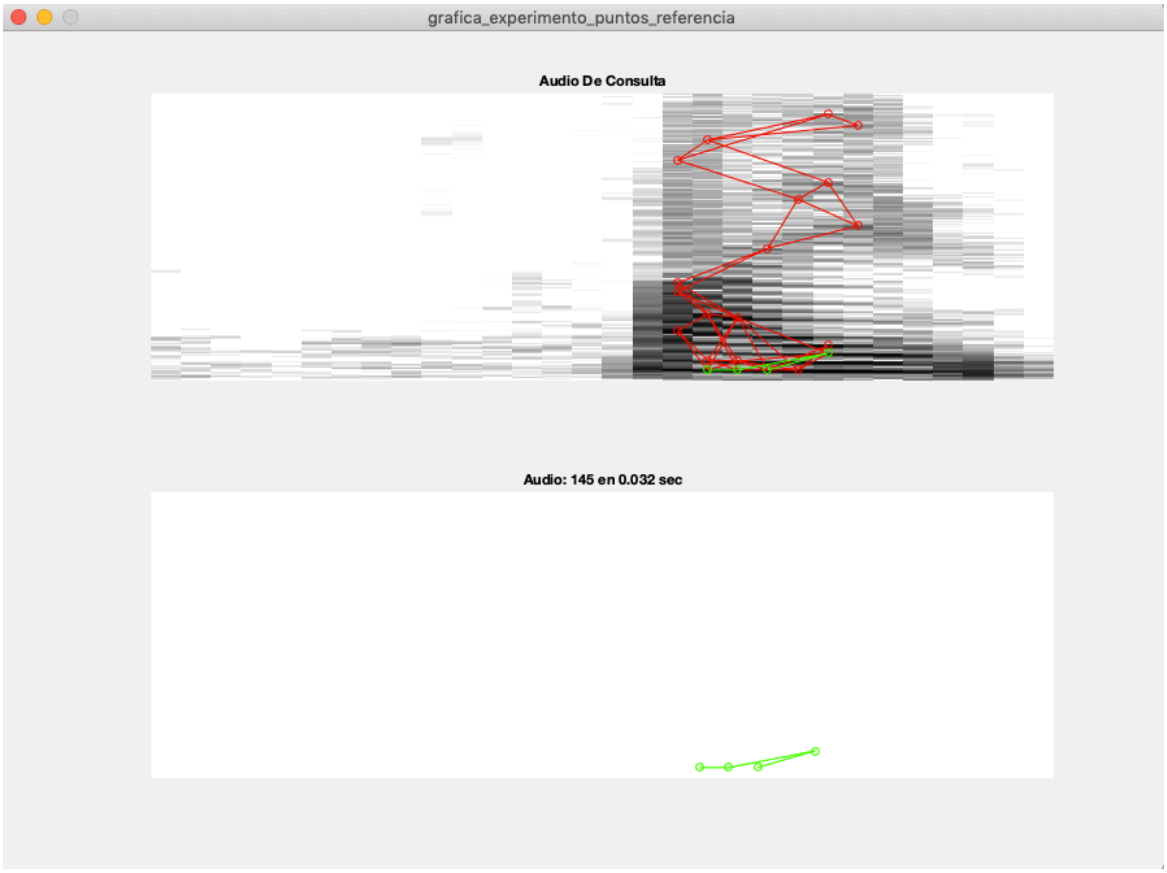


Figura 3.19 Comparación de hashes entre audio 144 y 145

Los últimos siete experimentos(cinco con imágenes binarias y dos con puntos de referencia) se realizan con 100 audios donde se graba a las personas mencionando la palabra “four”(cuatro en inglés), cada persona graba 5 veces su voz, por lo que se tiene

20 diferentes personas en el banco de audio. En estos experimentos se utilizan parámetros que se han usado en los experimentos anteriores para visualizar si generan los mismos resultados.

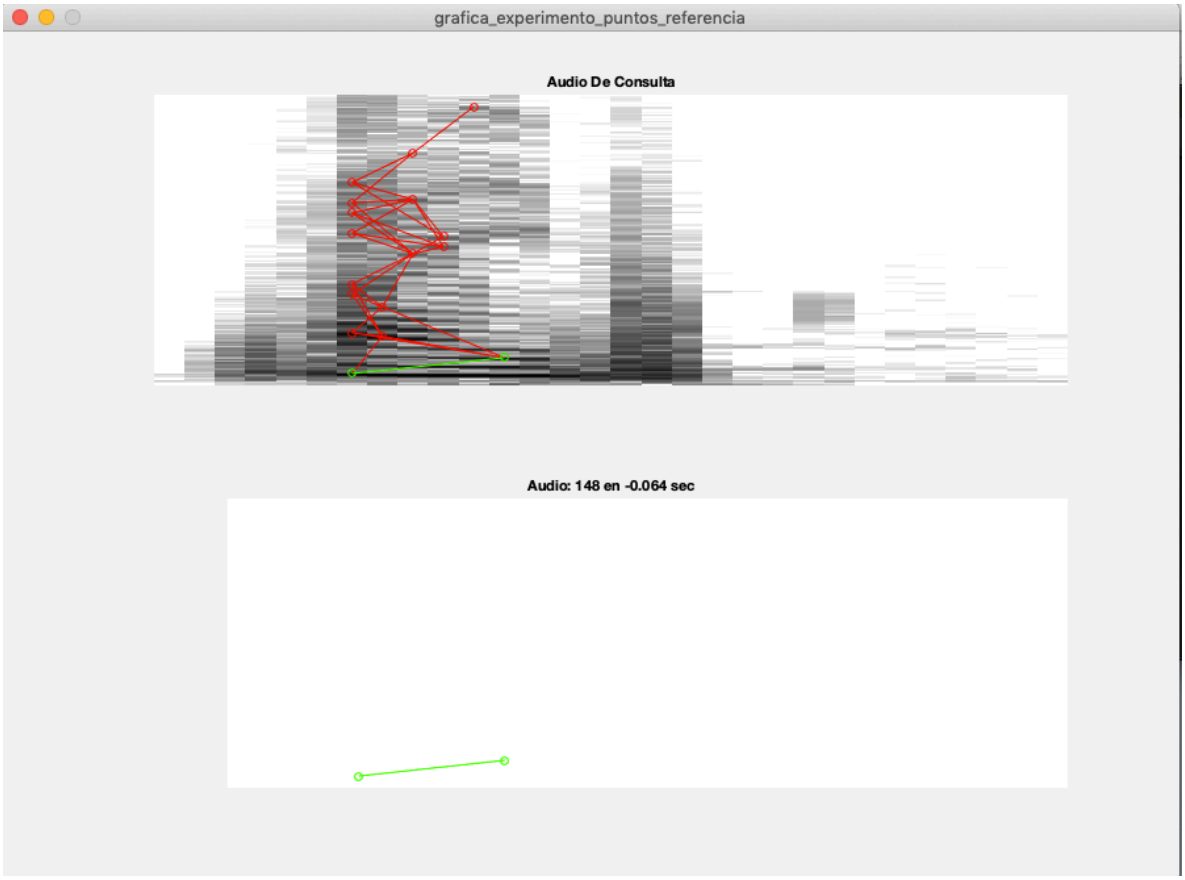


Figura 3.20 Comparación de hashes entre audio 147 y 148

Para las imágenes binarias se usan se aplica el filtro pasa banda de 300 a 5000 Hz, un desfase=30 y un umbral=1. Los resultados aparecen en la tabla 3.1, donde se muestra el máximo de casillas iguales(mci); este parámetro es un límite que las huellas comparadas deben rebasar para concluir que las huellas son semejantes, mientras este valor sea más pequeño, mas huellas coincidentes se encontrarán. Por cada experimento, cada audio se compara con los 99 restantes, la herramienta devuelve las huellas semejantes, pero no todas las semejanzas son buenas. En la tabla se muestran las huellas semejantes que son buenas.

Experimento	Máximo casillas iguales(mci)	Huellas semejantes
1	1500	2
2	1490	4
3	1480	12
4	1470	21
5	1460	28

Tabla 3.1 Resultados de experimentos con el algoritmo de imágenes binarias con audios de personas diciendo "Four".

Para el experimento 1 y 2 de la tabla 3.2, se usa la configuración del experimento 1 y el experimento 4 de los puntos de referencia. Se usan estas configuraciones porque fue donde se encontraron mas huellas semejantes buenas.

Experimento	Anchura de vecindad	Máximos por frame	Zona Objetivo Frecuencia	Zona Objetivo Tiempo	Total Hashes	Hashes por segundo	Huellas Semejantes
1	30	5	31	63	1018	10.226	39
2	30	8	41	73	1490	14.9667	46

Tabla 3.2 Resultados de experimentos con el algoritmo puntos de referencia con audios de personas diciendo "Four".

Como se puede observar en la tabla 3.2 desde el experimento 1 hubo un mejor rendimiento que los experimentos de imágenes binarias de la tabla 3.1; basta con aumentar los máximos por frame y la zona objetivo, haciendo que el número de hashes incremente y existan más coincidencias de huella digital

CAPÍTULO 4

CONCLUSIONES

4.1 CONCLUSIONES

En este capítulo se hablarán de las conclusiones sobre la herramienta que se ha creado, y los algoritmos utilizados para la creación de la huella digital de audio.

El uso de PAEX-HUDI le permite al usuario procesar audios digitales en donde puede extraer características y modelar una huella de audio, sin necesidad de programar. El usuario puede analizar los algoritmos, de las imágenes binarias y de los puntos de referencia, y modificarlos para visualizar su comportamiento y usar el que mejor le convenga. A su vez se pueden hacer experimentos para comparar los audios de una carpeta.

El algoritmo de las imágenes binarias no es muy preciso, cuando se convierten las intensidades en pixeles de color blanco o negro, dependiendo del umbral, se introduce mucho ruido; si el umbral es mayor a 1, el ruido desaparece, pero como las intensidades se normalizan entre 0 y 1, se corre el riesgo de tener una imagen llena de color negro. A pesar de eso, es una manera de obtener la huella digital de audio; y usando más algoritmos de procesamiento de imágenes se puede hacer más robusto este modelado.

El algoritmo de puntos de referencia es un poco más preciso que el de las imágenes, pues detecta los puntos más altos de un espectrograma, aunque de vez en cuando detecta puntos de ruido, pero eso se considera cuando se comparan los hashes, ya que no se necesita una exactitud en la comparación de los hashes, lo que se busca es un audio que tenga los más hashes semejantes posibles.

A pesar de estas conclusiones la herramienta permite hacer uso de cualquiera de los dos algoritmos con las modificaciones que mejor convengan. Dependiendo del tipo de audio que se quiera analizar(voz, instrumento musical, ruido, etc.), puede que los algoritmos se comporten de manera diferente.

APÉNDICES

Apéndice A (Sistema Lineal)

Un sistema es cualquier proceso a través del cual las señales se convierten en otras. Un sistema lineal LTI(Linear Time Invariant), es un sistema que es lineal e invariante en el tiempo. Muchos procesos físicos pueden considerarse como sistemas LTI, entre ellos los filtros para modificar las señales de audio. Para saber si un sistema es linear, debe cumplir las siguientes propiedades:

1.-Homogeneidad: Dada una señal x(t), si se multiplica por un factor *a,* la señal de salida y(t), debe ser multiplicada por el mismo factor *a* (figura A.1).

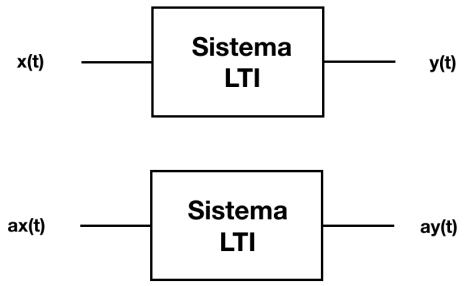


Figura A.1 Propiedad de Homogeneidad de un sistema LTI

2.- Aditividad: Si $y_1(t)$ y $y_2(t)$ son salidas de un sistema LTI de dos señales de entrada $x_1(t)$ y $x_2(t)$ respectivamente, entonces $y_1(t) + y_2(t)$ es la salida del mismo sistema LTI de la entrada $x_1(t) + x_2(t)$(Figura A.2).

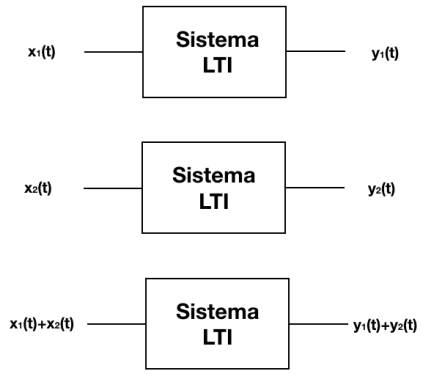


Figura A.2 Propiedad de aditividad de un sistema LTI.

Si el sistema cumple con las propiedades anteriores entonces cumple con el principio de linealidad o superposición en el que dice que si la señal de la entrada es una

combinación lineal de varias señales, la salida es la misma combinación lineal de las respuestas del sistema a cada una de las señales(Figura A.3):

Figura A.3 Principio de linealidad o superposición.

La invariabilidad dice que un sistema es invariante con el tiempo. Esto significa que los parámetros del sistema no van cambiando a través del tiempo y que por lo tanto, una misma entrada muestra el mismo resultado en cualquier momento(Figura A.4).

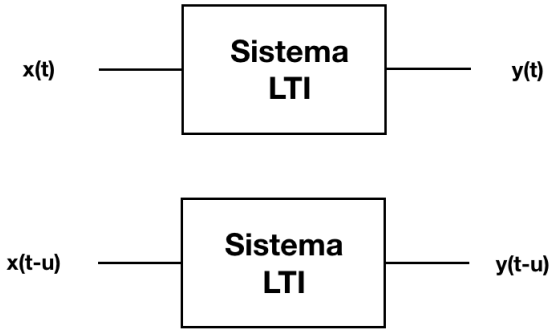


Figura A.4 Invariabilidad en el tiempo de un sistema.

Si el sistema cumple con la linealidad y la invariabilidad entonces el sistema es LTI.

Apéndice B (Función impulso)

Usando la función impulso(1.5) se puede escribir cualquier señal x[t] como una combinación de impulsos desplazados como lo muestra la figura B.1 y la ecuación 1.9.

$$x[t] = x[-2]\delta[t+2] + x[-1]\delta[t+1] + x[0]\delta[t] + x[1]\delta[t-1] + x[2]\delta[t-2] \quad \text{(B.1)}$$

Si se quisiera obtener el valor de la función cuando t=2, se sustituye en la ecuación B.1:

$$x[2] = x[-2]\delta[2+2] + x[-1]\delta[2+1] + x[0]\delta[2] + x[1]\delta[2-1] + x[2]\delta[2-2] \quad \text{(B.2)}$$

Se reduce:

$$x[2] = x[-2]\delta[4] + x[-1]\delta[3] + x[0]\delta[2] + x[1]\delta[1] + x[2]\delta[0] \quad \text{(B.3)}$$

de la función impulso se sabe que $\delta[0] = 1$ y que para cualquier parámetro diferente de 0 la función valdrá 0, por lo que:

$$x[2] = (3)(0) + (1)(0) + (2)(0) + (0)(0) + (-2)(1) \quad \text{(B.4)}$$

Se reduce y se obtiene que x=-2.

Como se puede escribir cualquier señal como una suma de impulsos, se obtiene la fórmula:

$$x[t] = \sum_{k=-\infty}^{k=\infty} x[k]\delta[t-k] \quad \text{(B.5)}$$

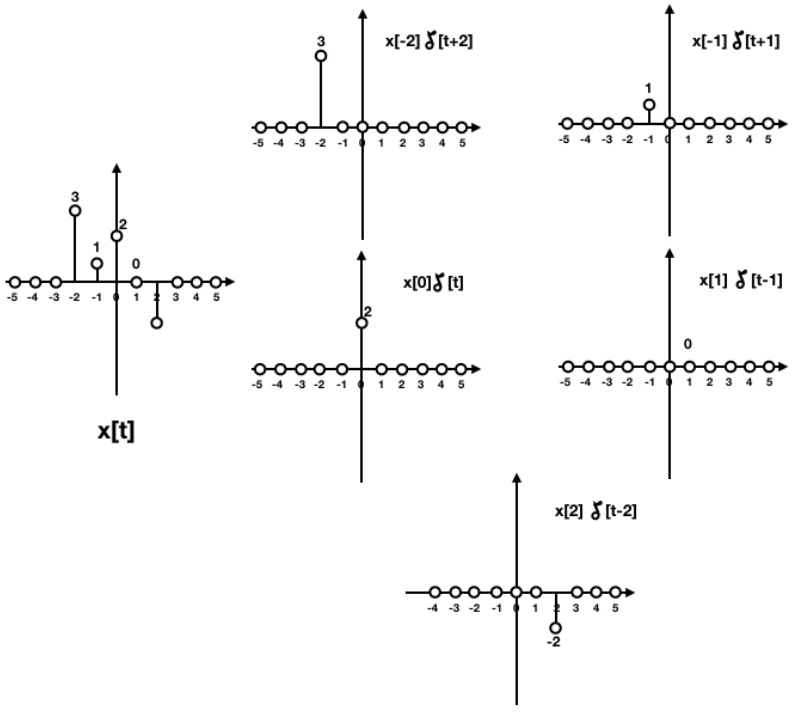


Figura B.1 Representación de la señal x[t] como combinación de impulsos desplazados.

Ahora se define g[t] como la salida del sistema LTI cuando la entrada es la función impulso.

Como el sistema es invariante en el tiempo entonces:

Como el sistema LTI cumple el principio de superposición entonces se tiene que:

$$x[t] = \sum_{k=-\infty}^{k=\infty} x[k]\delta[t-k] \quad \text{Sistema LTI} \quad y[t] = \sum_{k=-\infty}^{k=\infty} x[k]g[t-k]$$

A g[t] generalmente se le conoce como respuesta al impulso de un sistema LTI discreto, y algunos autores la denotan con la letra h[t]

Apéndice C (Filtros no ideales)

Se muestran las atenuaciones(Rs,Rp) y frecuencias de corte(F1, F2) de los filtros pasa alta, pasa banda y elimina banda. Con estos 4 parámetros se obtiene mediante las formulas 1.12, 1.13 y 1.14 el orden del filtro dependiendo si es para Butterworth ó Chebyshev respectivamente. En las figuras C.1, C.2 y C.3 se puede mostrar cómo se comportan los filtros de aproximación contra los filtros ideales.

- Rp Atenuación u ondulación en la banda pasante.
- Rs Atenuación u ondulación en la banda eliminada.
- F1, F2 Frecuencia de corte.
- Filtro con color azul el filtro aproximado(Butterworth).
- Filtro ideal de color negro.
- Señal color naranja.

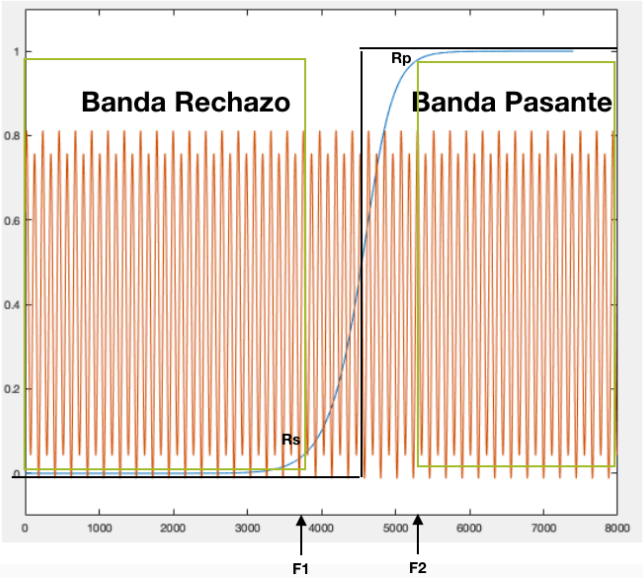


Figura C.1 Representación del filtro aproximado para el filtro ideal Pasa Alta con sus parámetros

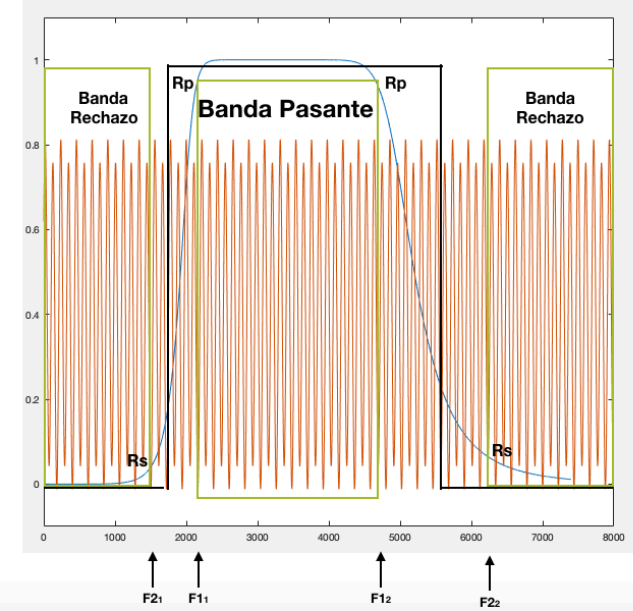


Figura C.2 Representación del filtro aproximado para el filtro ideal Pasa Banda con sus parámetros. Como se puede notar este filtro es la aplicación del filtro Pasa Alta y del filtro Pasa Baja.

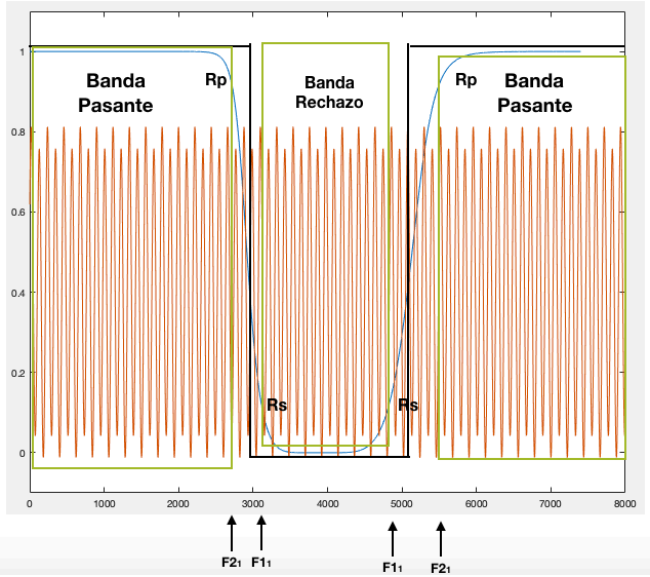


Figura C.3 Representación del filtro aproximado para el filtro ideal Elimina Banda con sus parámetros. Como se puede notar este filtro es la aplicación del filtro Pasa Baja y del filtro Pasa Alta.

Apéndice D (Comparación del orden del filtro)

Se muestra la comparación del orden del filtro Butterworth aplicado como pasa baja(Figura D.1).

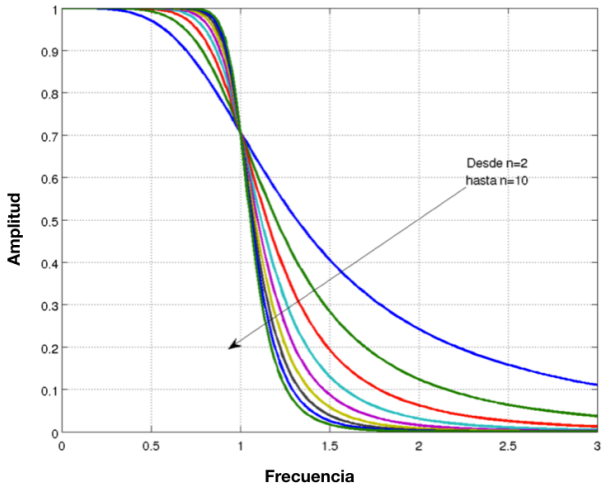


Figura D.1 Comparación de orden con el filtro Butterworth.

Como se puede observar mientras el orden crece, la caída se hace menos suave, pues entre más se le exija al filtro, mayor será su orden[A].

La figura D.2 muestra la comparación de los diferentes orden del filtro Chebyshev. Como se puede observar, se comporta similar a Butterworth, entre mejor caída, el orden será mayor, la cuestión es que entre mayor sea el filtro, la ondulación será cada vez peor en la banda de paso[8].

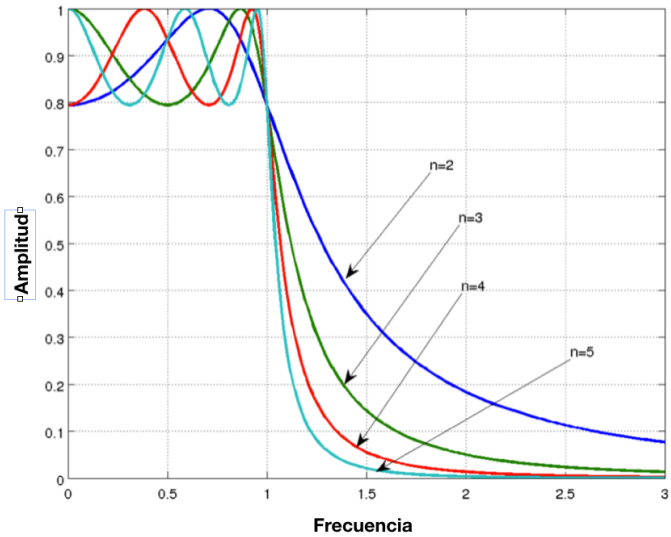


Figura D.2 Comparación de orden con el filtro Chebyshev.

Apéndice E (Descripción de la herramienta desarrollada)

Cuando el usuario abre el programa, lo primero que debe hacer es elegir el audio que se va procesar. En la interfaz gráfica(figura E.1) el usuario puede reproducirlo, desde el inicio o en un tiempo en específico, pararlo, visualizar el canal de reproducción que desee(o incluso los dos), y visualizar la frecuencia del audio en el tiempo que se está reproduciendo.

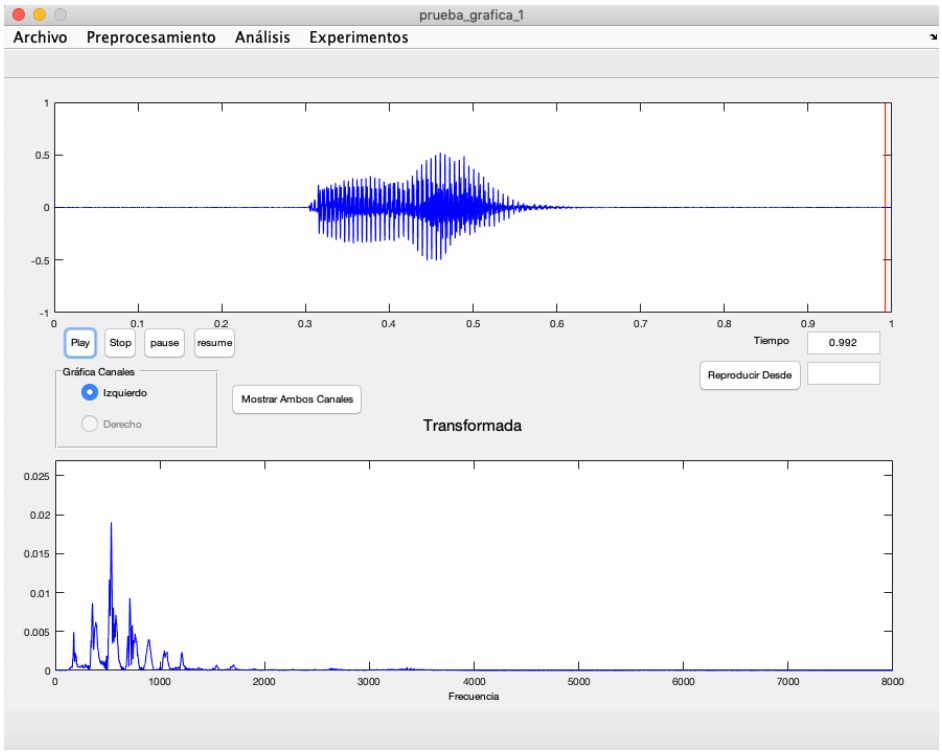


Figura E.1 Interfaz Gráfica del programa, en la imagen aparece como un audio se está reproduciendo.

E.1 Menú Pre-procesamiento

Para analizar el audio, se necesita convertir el audio a mono, si éste estuviera en estéreo. Se le permite al usuario que elija la conversión que mejor le convenga:

1. Promedio
2. Conservar el Canal Izquierdo
3. Conservar el Canal Derecho

Para la conversión por el promedio se usa la fórmula 1.1. Si el audio no está en Mono, el programa no permitirá el análisis ó procesamiento del archivo.

El usuario puede pre-procesar el audio mediante los filtros pasa alta, baja, banda ó elimina banda. No se debe olvidar que el objetivo de estos filtros es separar señales o restaurar señales distorsionadas. Un ejemplo sería que si se quiere analizar el habla en un audio, se aplica un pasa banda, ya que las frecuencias del habla se encuentran entre los 250 y 6000 Hz, por lo que no es necesario analizar las frecuencias menores a 250Hz y mayores a 6000Hz. Así que dependiendo el sonido a analizar, se puede aplicar los diferentes filtros.

Para la aplicación de los filtros, el usuario puede elegir, entre los filtros no ideales, tales como Window Sinc, Butterworth y Chebyshev. Para el filtro Window Sinc en pasa baja y alta se pide como parámetros el orden (N) y la frecuencia de corte. Para Window Sinc en pasa banda y elimina banda se pide el orden(N) y las frecuencias de corte. Para el resto de los filtros existe una configuración básica y una avanzada. En la configuración básica se pide el orden del filtro y las frecuencias de corte y se aplicará el filtro Buterrworth. En la avanzada se pide como parámetros las atenuaciones(Rp, Rs) y las frecuencias de corte(F1,F2); en esta configuración el programa obtiene el orden mediante las fórmula 1.12, 1.13 y 1.14 dependiendo del filtro. El programa, una vez obtenido el orden, aplicará el filtro a la señal de audio.

E.2 Menú Análisis

En este menú el usuario puede visualizar cómo se comporta el audio en el modelado de huella digital. A partir del espectrograma el usuario puede decidir si trabaja con el modelado de huella a partir de imágenes binarias o a partir de puntos de referencia(picos).

Se crea el espectrograma(Figura E.2) de forma básica ó avanzada. Si el usuario usa la configuración básica, se creará un espectrograma de 3 segundos desde el inicio del audio; se usará una ventana Hamming y se dividirá el audio en 8 partes donde el 50% de las muestras de la ventana serán solapadas. En la configuración avanzada, el usuario elige el tamaño y el tipo de ventana que mejor le convenga. También elige el inicio y el final del audio con un máximo de 15 segundos; incluso puede elegir el porcentaje de muestras solapadas.

Si el usuario decide trabajar con las imágenes binarias el programa le mostrará una ventana donde aparece la media de intensidad del espectrograma, así el usuario podrá decidir el umbral por el que multiplicará la media para pintar de blanco y de negro la

imagen. Se vuelve a recalcar que se debe tener cuidado con el umbral cuando éste sea mayor a 1(figura E.3).

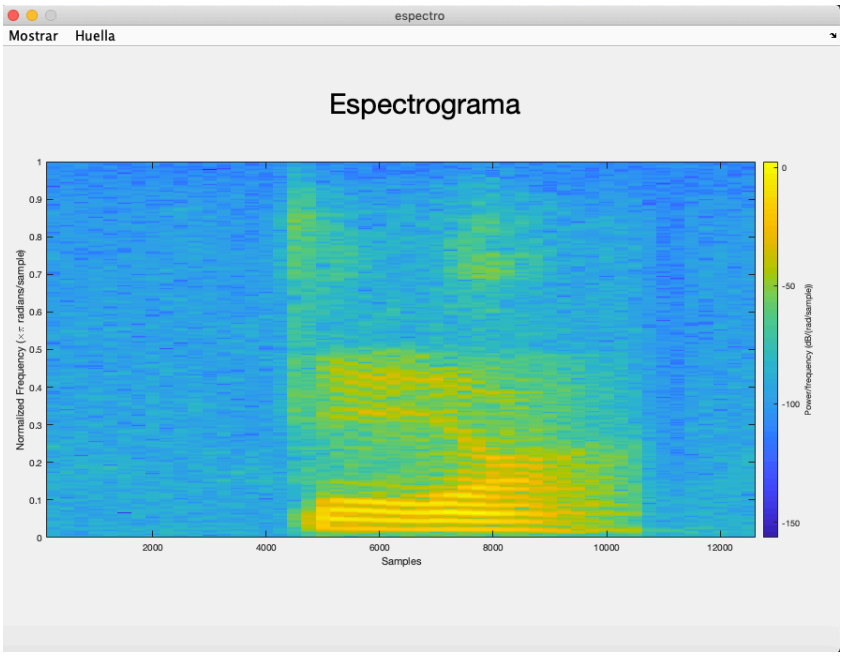


Figura E.2 Espectrograma.

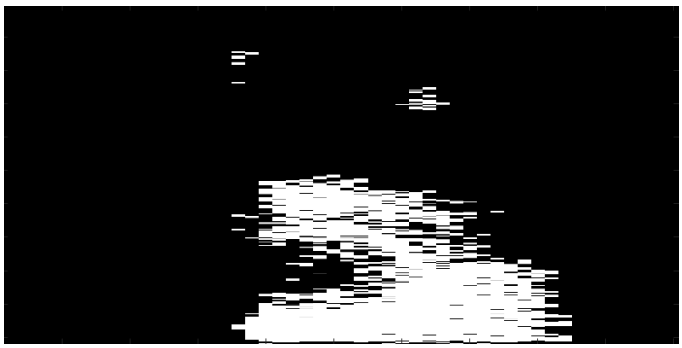


Figura E.3 Imagen binaria del espectrograma de la figura E.2, en la que el umbral es multiplicado por 1.4.

Una vez que el usuario ha convertido el espectrograma a imagen binaria, puede obtener en un archivo (vector_huella.txt) la huella digital del audio.

Si el usuario decide trabajar con los picos, se debe seleccionar en el menú de huella, “obtener huella” y aparecerá un cuadro de diálogo en el que se debe elegir la

opción picos. Después aparecerá una ventana en donde el usuario debe escribir el ancho de la vecindad, picos máximos por frame, zona objetivo tiempo y zona objetivo frecuencia. Como se vio anteriormente estos parámetros incrementan o disminuyen la cantidad de hashes(Figura E.4).

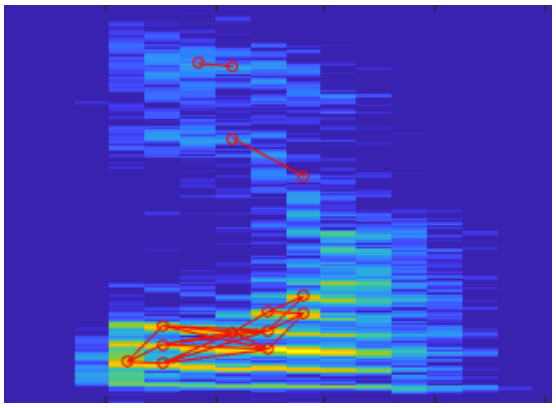


Figura E.4 Resultado de modelar la huella con puntos de referencia(Picos).

E.3 Menú Experimentación

Una vez que el usuario haya analizado y observado las dos formas de modelar la huella digital, en este menú puede realizar experimentos ya sea por imágenes binarias o por puntos de referencia, lo único que tiene que hacer es ubicar en el programa donde se encuentran los audios y dar algunos parámetros para que se pueda realizar el experimento. Se recomienda que los audios para analizar estén en una sola carpeta y que estos se nombren con la numeración decimal empezando desde el 1.

Antes de iniciar el experimento de imágenes binarias, se deberá encender el servidor Mysql(Figura E.5). Cuando el usuario ha encendido el servidor, éste debe elegir el experimento de imágenes binarias donde aparecerá una ventana en donde se debe ubicar la ruta de la carpeta de los audios, luego tiene que elegir el tamaño de cada muestra(El usuario puede dividir el audio en los ms que el prefiera), también debe elegir un filtro, configurar el espectrograma y por último debe elegir el umbral para multiplicar con la media de la intensidad del espectrograma(Figura E.6). Cuando el programa haya guardado la huella o huellas de cada audio en la base de datos, el programa debe comparar cada huella con las restantes. Una vez finalizado la comparación el usuario debe elegir un limite para que las comparaciones(huellas similares) que rebasen ese limite se muestren en un cuadro de diálogo; estas huellas similares representaran los audios que son similares.

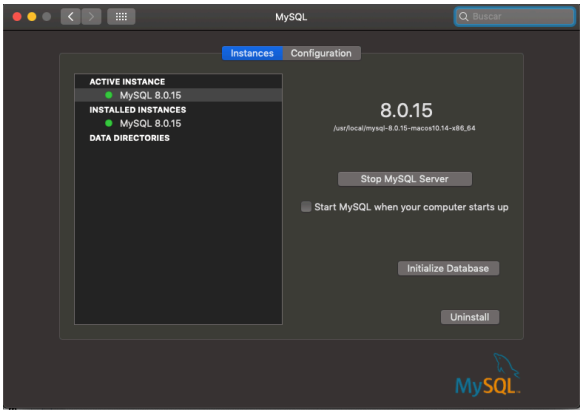


Figura E.5 Ventana para encender el servidor Mysql.

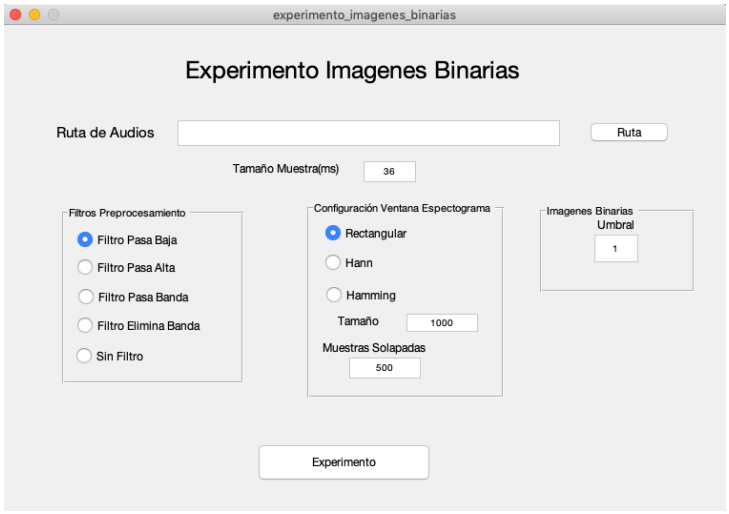


Figura E.6 Ventana de configuración para el experimento imágenes binarias.

Para el experimento puntos de referencia, no es necesario hacer el uso de una base de datos(prender un servidor), así que cuando el usuario elija este experimento aparecerá una ventana en donde deberá elegir la ruta de la carpeta de los audios para el experimento, elegir el audio que se va a comparar, y escribir parámetros para los puntos de referencia tales como el ancho de la vecindad, los picos máximos por frame y la zona objetivo del tiempo y la frecuencia(figura E.7). En este experimento el audio a comparar, se cotejará con toda los audios de la carpeta y el programa mostrará en cuadros de diálogo, los audios semejantes y con cuantos puntos. También al final mostrará una gráfica de los puntos de referencia del audio que se comparó y del audio que mas se asemejó como en la figura E.8.

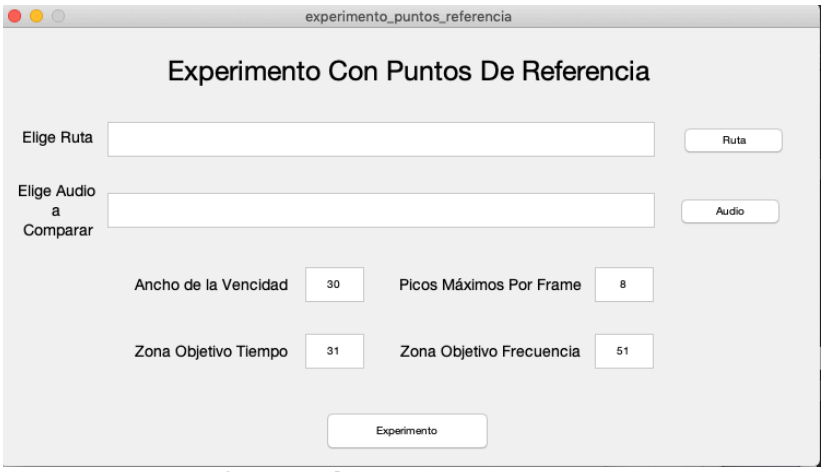


Figura E.7 Ventana de configuración para el experimento de puntos de referencia.

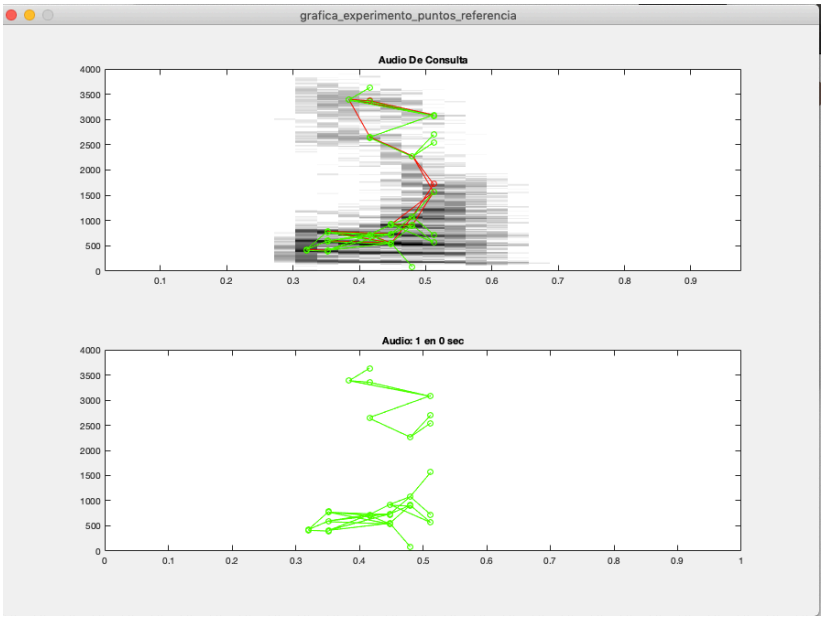


Figura E.8 Gráfica de los puntos de referencia del audio de la consulta y el audio que mejor similitud tiene.

BIBLIOGRAFÍA

1. Agüera B., Gfeller B., Guo R., Kilgour K., Kumar S., Lyon J., Odell J., Ritter M., Roblek D., Shafiri M., Velimirovic M. (2017). Now Playing: Continuous low-power music recognition. *arXiv preprint arXiv:1711.10958*.
2. Cano, P., Batlle, E., Kalker, T., & Haitsma, J. (2005). A review of audio fingerprinting. *Journal of VLSI signal processing systems for signal, image and video technology*, *41*(3), 271-284.
3. Amaro, E. (2008). *Caracterización Automática del Llanto de Bebés para su Estudio con Modelos de Clasificación* (Doctoral dissertation, Master's Thesis, INAOE, Tonantzintla, Puebla, México).
4. Boney, L., Tewfik, A. H., & Hamdy, K. N. (1996, June). Digital watermarks for audio signals. In *Multimedia Computing and Systems, 1996., Proceedings of the Third IEEE International Conference on* (pp. 473-480). IEEE.
5. Seyoum, S., Alfonso, L., van Andel, S. J., Koole, W., Groenewegen, A., & van de Giesen, N. (2017). A Shazam-like household water leakage detection method. *Procedia Engineering*, *186*, 452-459.
6. Zurek, E. E., Gamarra, A., Margarita, R., Escorcia, G., José, R., Gutierrez, C., ... & García, X. (2016). ANÁLISIS ESPECTRAL PARA EL RECONOCIMIENTO DE HUELLAS ACÚSTICAS. *Journal of Research of the University of Quindio*, *28*(1).
7. Gomes, L. D. C., Cano, P., Gomez, E., Bonnet, M., & Batlle, E. (2003). Audio watermarking and fingerprinting: For which applications?. *Journal of New Music Research*, *32*(1), 65-81.
8. Smith, S. (2013). *Digital signal processing: a practical guide for engineers and scientists*. Elsevier.
9. Haitsma, J., & Kalker, T. (2002, October). A highly robust audio fingerprinting system. In *Ismir* (Vol. 2002, pp. 107-115).
10. Haitsma, J. A. (2002). Audio Fingerprinting: A New Technology to Identify Music. *Unclassified Report 2002*, *824*.
11. Wang, A. (2006). The Shazam music recognition service. *Communications of the ACM*, *49*(8), 44-48.
12. Master, A. S., Stonehocker, T. P., Levitt, B. J., Huang, J., & Mohajer, K. (2016). *U.S. Patent No. 9,280,598*. Washington, DC: U.S. Patent and Trademark Office.
13. Fan, Y., & Feng, S. (2016, December). A Music Identification System Based on Audio Fingerprint. In *2016 4th Intl. Conf. on Applied Computing and Information*

Technology (ACIT), 3rd Intl. Conf. on Computational Science/Intelligence and Applied Informatics (CSII), and 1st Intl. Conf. on Big Data, Cloud Computing, Data Science & Engineering (BCD) (pp. 363-367). IEEE.

14. Kamaladas, M. D., & Dialin, M. M. (2013, February). Fingerprint extraction of audio signal using wavelet transform. In *2013 International Conference on Signal Processing, Image Processing & Pattern Recognition* (pp. 308-312). IEEE.
15. Jianhua Meng,Ning Chen.A Algorithm Based on Gammachirp Cochlear Energy Spectrum Feature Extraction Of Audio Fingerprint. Journal Of East China University Of Science And Technology 2015 10:41-5.
16. Burges, C. J., Plastina, D., Platt, J. C., Renshaw, E., & Malvar, H. S. (2005, March). Using audio fingerprinting for duplicate detection and thumbnail generation. In *Acoustics, Speech, and Signal Processing, 2005. Proceedings.(ICASSP'05). IEEE International Conference on* (Vol. 3, pp. iii-9). IEEE.
17. Ogle, J. P., & Ellis, D. P. (2007, April). Fingerprinting to identify repeated sound events in long-duration personal audio recordings. In *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on* (Vol. 1, pp. I-233). IEEE.
18. Ibarra, R., & López, M. S. (1999). *Principios de teoría de las comunicaciones*, Limusa.
19. Watkinson, J. (2013). *Introduction to digital audio*. Focal Press.
20. Oppenheim, A. L., Schafer, R. W. and Buck, J. R., *Tratamiento de señales en tiempo discreto*, Prentice Hall, Segunda edición, 1999.
21. Entendiendo Transformada de Fourier y Ventaneo (2017). - National Instruments. EU. Recuperado de: www.ni.com de Notas Técnicas.
22. Moya, I. (2009). *Análisis tiempo-frecuencia de la señal de vibración de un cambiador de tomas en carga* (Proyecto Fin de Carrera, Leganés, España).
23. Otero E. (2015). Huellas digitales acústicas y software libre. España. Recuperado de: https://labs.beeva.com/
24. Lalinsky, L. (2018). Chromaprint. Recuperado de: https://acoustid.org/
25. (2013). Dejavú. Recuperado de: https://github.com/
26. Ouali, C., Dumouchel, P., & Gupta, V. (2016). A spectrogram-based audio fingerprinting system for content-based copy detection. *Multimedia Tools and Applications*, *75*(15), 9145-9165

27. Miller, M. L., Rodriguez, M. A., & Cox, I. J. (2005). Audio fingerprinting: nearest neighbor search in high dimensional binary spaces. *Journal of VLSI signal processing systems for signal, image and video technology*, *41*(3), 285-291.
28. Jang, D., Yoo, C. D., Lee, S., Kim, S., & Kalker, T. (2009). Pairwise boosted audio fingerprint. *IEEE transactions on information forensics and security*, *4*(4), 995-1004.
29. Roca E, (2017). *Audio fingerprinting para la identificación automática de contenidos audiovisuales,* Tesis Final De Grado, Catalunya, España.
30. Gupta, V. N., Boulianne, G., & Cardinal, P. (2012). CRIM's content-based audio copy detection system for TRECVID 2009. *Multimedia Tools and Applications*, *60*(2), 371-387.
31. Cheng Yang, "MACS: Music Audio Characteristic Sequence Indexing For Similarity Retrieval", in IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, 2001
32. Müller, M. (2015). *Fundamentals of music processing: Audio, analysis, algorithms, applications*. Springer.
33. Borja, C. T., & Bueno, Á. G. (2006). Sistemas Biométricos. *Recopilado de: https://www.dsi.uclm.es/personal/MiguelFGraciani/mikicurri/Docencia/Bioinformatica/web_BIO/Documentacion/Trabajos/Biometria/Trabajo% 20Biometria. pdf*.