UNIVERSIDAD AUTÓNOMA DE MADRID

ESCUELA POLITÉCNICA SUPERIOR

PROYECTO FIN DE CARRERA

MEJORA DE LA ROBUSTEZ FRENTE AL RUIDO EN UN SISTEMA DE BÚSQUEDA RÁPIDA DE AUDIO EN AUDIO

Andrés Martín López

Ingeniería de Telecomunicación

Mayo 2015

MEJORA DE LA ROBUSTEZ FRENTE AL RUIDO EN UN SISTEMA DE BÚSQUEDA RÁPIDA DE AUDIO EN AUDIO

AUTOR: Andrés Martín López
TUTOR: Doroteo Torre Toledano

Área de Tratamiento de Voz y Señales (ATVS)
Dpto. de Tecnología Electrónica y de las Comunicaciones
Escuela Politécnica Superior
Universidad Autónoma de Madrid
Mayo de 2015

Agradecimientos

Tras muchos años de esfuerzo y dedicación, este periodo de mi vida parece llegar a su fin. Se podría decir que estuvo repleto de infinidad de momentos y estados de ánimo. Momentos difíciles, en los que la frustración parecía ser una losa demasiado pesada para continuar, y momentos de satisfacción personal y autorrealización. De lo que no cabe duda es que todo ello hizo de mí una persona más fuerte y capaz, y por todo ello, el esfuerzo realizado mereció la pena. Todo esto no hubiera sido posible sin la presencia de tantísimas personas importante en mi vida, que han sabido tener paciencia y brindarme su incondicional apoyo en todo momento, a las cuales estaré eternamente agradecido.

No podría empezar sin mencionar a mi tutor Doroteo Torre Toledano, que fue quien me brindó la oportunidad de trabajar y aprender con el durante el desarrollo de este proyecto. Quisiera agradecer todos los meses de dedicación y paciencia que tuvo conmigo, así como su amabilidad y trato cercano, ha sido un placer trabajar a tu lado, ¡muchísimas gracias!

También me gustaría mencionar a mis padres, Santos y Begoña, por su constante apoyo y dedicación tanto a mí como a mis hermanos Pablo y Miguel. Ellos nos educaron en valores de esfuerzo, responsabilidad, respeto, generosidad entre muchos otros, sin los cuales nunca hubiera podido llegar a ser quien soy, ni haber conseguido nada de lo que he conseguido.

Como podría olvidarme de todos mis compañeros de fatigas que hacían que el día a día en la escuela fuera un poco más llevadero. Por esos días de risas, por esos planes que nunca nos faltaban los fines de semana, por esos viajes juntos, ha sido un verdadero placer haber coincidido con todos vosotros. Me alegra especialmente que todos hayamos salido adelante y tengamos un gran futuro por delante, y espero que sigamos siendo leales amigos como hasta ahora.

Por último no puedo despedirme sin hablar de mi Erasmus, el cual podría decir sin ninguna duda fue el mejor año de mi vida. La segunda familia que formamos ese año seguirá estando ahí espero por muchos años.

De todo corazón, gracias.

Andrés Martín López

Resumen

El objetivo de este proyecto es la mejora de un sistema de búsqueda de audio en audio en relación a la robustez frente al ruido. Se parte de un sistema ya existente desarrollado por el grupo ATVS, el cual es bastante vulnerable frente a ambientes ruidosos. Se implementarán una serie de mejoras basadas en técnicas de compensación de canal así como en algoritmos desarrollados a partir del análisis del comportamiento del sistema.

La memoria comienza con una introducción a los diferentes sistemas de recuperación de información de audio, y al efecto del ruido en este tipo de sistemas. Se incluye referencias al estado del arte de las diferentes técnicas de extracción de coeficientes característicos así como las técnicas de compensación y normalización que se utilizan para paliar el efecto del ruido sobre dichos coeficientes.

El sistema se subdivide en dos módulos, el primero encargado de la extracción de coeficientes característicos a partir de muestras de audio, y el segundo que trata de realizar un alineamiento temporal entre los vectores característicos de la muestra original y la muestra contaminada por ruido. La primera parte del desarrollo del proyecto está encaminada a conseguir una extracción más robusta de vectores de coeficientes, y la segunda parte consiste en realizar un estudio de los errores para tratar de detectarlos y así hacer el algoritmo más preciso.

Todos los pasos llevados a cabo en el proyecto vendrán seguidos de una evaluación de la precisión de forma sistemática, tanto para señales sin ruido, con ruido controlado, y grabaciones reales.

Palabras clave

Sistema de búsqueda de audio en audio, robustez frente al ruido, fingerprinting, coeficientes MFCC, CMVN, ruido, distancia espectral, trayectorias temporales.

Abstract

The objective of this project is to improve an audio in audio searching system regarding the robustness against noise. The starting point is an existing system developed by the ATVS group, which is quite vulnerable to noisy environments. Several improvements based on channel compensation techniques will be implemented as well as new algorithms developed from the analysis of the system behavior.

The report begins with an introduction to the different systems of audio information retrieval and the effect of noise in them. It includes references to the state of the art of various feature extraction methods and the compensation and normalization techniques used to mitigate the effect of noise on the characteristic coefficients.

The system is divided into two modules, the first one is responsible for the extraction of characteristic coefficients from audio samples, and the second one performs a temporal alignment between the feature vectors of the original sample and the sample contaminated by noise. The first part of the project development aims to achieve a more robust feature extraction against noise. The last part consists of making a study of the errors in order to identify them and improve the accuracy of the algorithm.

The accuracy of the system will be systematically checked in all the steps carried out in this project both for signals without noise, signals contaminated by monitored noise, and real recordings.

Keywords

Audio in audio searching system, robustness against noise, fingerprinting, MFCC coefficients, CMVN, noise, spectral distance, time paths.

Índice de contenidos

Índice de figuras

Índice de tablas

Capítulo 1:

Introducción

1.1 Motivación

El uso de aplicaciones de recuperación de información musical es cada vez más común en la actualidad. Se trata de sistemas que identifican archivos de audio (normalmente música) a partir de una grabación corta de la misma. Esto permite al usuario, por ejemplo, conseguir una canción que ha escuchado en la radio pero no sabe cuál es. Un ejemplo de todo esto es Shazam [1], que recupera en tiempo real información de millones de archivos musicales. Existe también una iniciativa de evaluaciones internacionales en recuperación de información musical llamada MIREX (Music Information Retrieval Evaluation eXchange) [2], que es una referencia esencial en este ámbito.

Por otro lado, existen aplicaciones de sincronización de audios, es decir, a partir de una muestra grabada, no solo se identifica la pista a la que pertenece, sino que también se indica el momento en el que transcurre sobre la misma. En relación con este tipo de tecnología existe *AudibleMagic*, una compañía estadounidense que ha realizado estudios sobre el tema y tiene varios productos en el mercado. Este proyecto está enfocado a este último tipo de aplicaciones.

Debido al auge de este tipo de tecnología, existe la necesidad de que el reconocimiento sea fiable de modo que no se cometan errores de detección. El principal problema que afecta a este tipo de sistemas es el ruido y las distorsiones en grabación del audio y por tanto es necesario dotar a estos sistemas de mecanismos de defensa para lograr un porcentaje de éxito de detección elevado.

Para ello, se investiga en la línea de la aplicación y creación de algoritmos matemáticos que permitan reconocer detecciones erróneas y posteriormente corregirlas. Es muy importante tener en cuenta el coste computacional de dichos algoritmos, puesto que se trata de sistemas en tiempo real, y esto puede ser un factor condicionante. En este sentido, se estudiarán varias vías de mejora:

- Técnicas de compensación de efectos de ruido y canal sobre características MFCC:
    - Cepstral Mean Normalization (CMN)
    - Cepstral Mean and Variance Normalization (CMVN)

- Análisis de evolución temporal de detecciones.

Además, debido a la gran diversidad de aplicaciones que puede tener este tipo de sistemas, es necesario hacer un análisis en distintos tipos de condiciones, así como con

diferentes tipos de terminales, para crear una solución que abarque un mayor número de situaciones.

1.2 Objetivo

El objetivo del proyecto es mejorar el rendimiento del sistema de búsqueda rápida de audio en audio, en el sentido de lograr un porcentaje de éxito de detección mayor en todos los ambientes. Para ello se evaluara el rendimiento del programa aplicándole una serie de muestras de audio previamente grabadas y se examinara cómo afecta el ruido incluido en la grabación de las muestras de audio.

Posteriormente, después de un análisis exhaustivo de los resultados obtenidos, se procederá a la creación de un algoritmo que modifique el programa primario, para que el porcentaje de acierto en detección, así como el error medio de detección sean mejores. El método de procedimiento se basará en hacer pruebas sistemáticas para comprobar que el algoritmo funciona mejor.

El resultado final esperado será un sistema con una tasa de acierto de detección más elevada así como una precisión del orden de milisegundos, para muestras de audio grabadas en ambientes relativamente ruidosos.

1.3 Metodología de trabajo

En el desarrollo de este proyecto podemos diferenciar cuatro fases distintas, que son:

1. ***Documentación Bibliográfica y obtención de muestras:***
    - Estudio de las tecnologías de reconocimiento de audio basadas en MFCC (Mel Frencuency Cepstral Coefficient)
    - Estudio de aplicaciones de reconocimiento musical como Shazam.
    - Elección de audio de referencia, por ejemplo audio de películas que formarán una base de datos suficientemente amplia para nuestros experimentos.
    - Grabación de muestras de dichos audios en distintos ambientes y con distintos terminales.

2. ***Experimentación y pruebas:***

- Aplicación de las muestras grabadas sobre el programa primario para obtener el rendimiento de sistema en inicio.
- Análisis de parámetros característicos obtenidos de los resultados de los test (análisis de ráfagas).
- Pruebas sobre dichos parámetros para orientar el futuro algoritmo de robustez frente al ruido.

3. ***Realización de la mejora:***

- Creación de un algoritmo basado en los experimentos, que permita incrementar el rendimiento del sistema.
- Modificar el código del programa para integrar el algoritmo.
- Comprobación de la eficiencia del nuevo programa, tanto en resultados como en coste computacional.
- Repetir varias veces el punto tercero hasta cumplir los objetivos deseados.

4. ***Escritura de la memoria:***

- Recopilación de todos los experimentos realizados, así como las mejoras propuestas en una memoria en la cual se expondrán las conclusiones finales del proyecto.

1.4 Estructura de la memoria

Capítulo 1: Introducción

Capítulo en el que se detallan las motivaciones que han llevado al desarrollo del proyecto, así como los objetivos que se pretenden alcanzar.

Capítulo 2: Estado del arte

Este capítulo contiene un detallado estado del arte sobre las técnicas de robustez frente al ruido más comunes así como de técnicas de procesamiento de audio en general. La primera parte contiene información acerca de los diferentes tipos de sistemas de recuperación de información a partir del audio. Posteriormente se hace referencia a las técnicas comunes de procesado de audio para todos estos sistemas

descritos anteriormente encaminándonos hacia el tipo de sistema objeto de este proyecto. A continuación se entra en detalle en los sistemas de búsqueda de audio en audio y finalmente se mencionan las técnicas más comunes que se emplean para hacer de estos, sistemas más robustos frente al ruido.

Capítulo 3: Diseño y desarrollo

En este capítulo se propondrán todas las mejoras que se incorporarán al programa objeto del proyecto. Se comienza con una descripción detallada del programa de partida, de todos sus componentes y de cómo es su funcionamiento. Posteriormente se detallan paso a paso las mejoras que se han implementado y la forma en que se acoplan al programa principal. Se trata de un capítulo breve en el cual no se entra en detalle sobre la mejora obtenida con su incorporación.

Capítulo 4: Pruebas y resultados

Este será el capítulo donde se pongan a prueba todas las mejoras introducidas en el apartado anterior. El punto de partida será una evaluación del potencial del programa primario para contrastar las mejoras que se van incorporando. Posteriormente se van introduciendo las mejoras de modo que cada una venga a complementar a la anterior. Se realizan exhaustivas pruebas para comprobar que se logra la mejora deseada antes de continuar. Finalmente se prueba todo el sistema en conjunto y se comprueba si se han alcanzado los objetivos marcados desde un principio.

Capítulo 5: Conclusiones y trabajo futuro

En este último capítulo se expondrán brevemente las conclusiones obtenidas de todo el trabajo realizado, en términos de rendimiento del programa y de objetivos alcanzados. Finalmente se hará una pequeña mención a aquellos aspectos que pueden ser objeto de estudio para un trabajo futuro, con el objetivo de mejorar aún más las prestaciones del programa.

Capítulo 2:

Estado del Arte

2.1. Recuperación de información de audio

El sonido puede ser una fuente de información extraordinariamente rica, lo que hace posible la extracción de gran cantidad de información diferente del mismo. Existen numerosas tecnologías dedicadas a este propósito, cada una de las cuales se centra en la obtención de un tipo determinado de información. Se denominan tecnologías ASC (Audio Signal Classification) y todas ellas se basan en el procesado de audio mediante diferentes técnicas con el objetivo final de obtener una serie de características del mismo, y usar esas características para extraer la información deseada, normalmente empleando técnicas de reconocimiento de patrones.

2.1.1 Clasificación de audio

El ser humano clasifica señales de audio inconscientemente en su vida normal, identifica el sonido de un teléfono, un grito o una puerta que se cierra. Esto puede parecer muy sencillo, pero los procesos mentales involucrados siguen sin ser completamente entendidos y los sistemas automáticos siguen sin superar el grado de exactitud de los humanos. Además puede llegar a ser más complicado, como por ejemplo identificar una dolencia de un paciente en función del sonido que haga al respirar o la avería de un coche dependiendo del ruido que haga el motor. Es por ello, que existe interés en poder clasificar estos sonidos de forma precisa mediante tecnologías de clasificación de audio, puesto que un sistema de tipo artificial puede tener mucho más potencial y puede servir de gran ayuda en situaciones concretas [3].

Como es obvio, el audio se puede clasificar de muchas formas distintas en función de su contenido, y es por ello que existirá un tipo de aplicación de extracción de información de audio para cada función específica. Uno de las funcionalidades más comunes en las que los estos sistemas están involucrados es en el reconocimiento de música o voz [15 y 16]. También se puede realizar una clasificación por género musical, o identificar un acento [14] o un idioma en una grabación de voz, o como veremos más adelante, algo más específico como identificar al locutor o las palabras que este dice.

2.1.1.1 Aplicaciones de los sistemas ASC

Los sistemas ASC son utilizados como base de muchas aplicaciones de procesado de audio. El reconocimiento de voz es una de las más comunes, donde las señales son clasificadas en fonemas y posteriormente ensambladas en palabras. Sin embargo, el audio, y en concreto la voz humana, contiene mucha más información que solo

palabras, también contiene información sobre el acento, el idioma o incluso un estado emocional.

En otros casos, no estamos interesados en el reconocimiento de todo lo que se dice, sino sólo en la identificación de ciertas palabras clave (Keyword Spotting) que nos permitan, por ejemplo, buscar conversaciones interesantes o determinar el tema de un dialogo para así procesarlo de forma más precisa.

Normalmente, los ASC destinados al análisis de señales musicales, esperan recibir una señal musical, y aquellos destinados a análisis de voz, esperan lo propio. Es por ello que existen los sistemas segmentadores y clasificadores de audio que particionan el audio en clases como voz, música, ruido, silencio, voz+música, etc., y son encargados de encaminar el audio a unos sistemas u otros en función del tipo de audio encontrado.

Una de las aplicaciones más obvias a las que los sistemas ASC dan soporte, es a la creación y manejo de bases de datos multimedia. Mediante la clasificación de audios, son capaces de agilizar la creación de estas bases de datos ordenando por ejemplo música en función de su género.

2.1.1.2 Análisis del entorno acústico

Los sistemas ASC son la forma más básica de un grupo más general de sistemas denominado ASA (Auditory Scene Analysis), el cual centra su trabajo en el análisis de todo el entorno acústico, en lugar de un solo tipo de sonidos. El entorno acústico está formado por la mezcla de muchos sonidos audibles en cualquier momento. La primera parte del procesado de señal de los sistemas ASA, es la segmentación, donde el entorno acústico es descompuesto en grupos de señales provenientes de diversas fuentes.

A partir de ahí es donde los sistemas ASC identifican cuál de las fuentes sonoras contienen música, cuál de ellas contiene voz, etc.

2.1.1.3 Tipos de audios que clasifican los ASC.

- ***Ruido***

Desde el punto de vista teórico existen varios tipos de ruido. Nos referimos al ruido blanco, el ruido rosa, y otros tipos de ruidos denominados coloreados. Sin embargo desde el punto de vista perceptual el concepto de ruido cambia, y nos referiremos normalmente a señales que tengan gran cantidad de su energía en altas frecuencias, y esta energía no está armónicamente distribuida. En cualquier

caso, hay que tener en cuenta que el concepto de ruido depende de la aplicación, puesto que un ruido es una señal no deseada para la aplicación en concreto. Así, por ejemplo el golpe de una puerta puede ser ruido en una aplicación de reconocimiento de voz, pero puede ser la señal de interés en una aplicación de detección de eventos acústicos, por ejemplo centrada en la seguridad.

- ***Sonidos naturales***

Se trata de sonidos no generados o no influenciados por la acción del hombre. El sonido del viento, el sonido que hace cualquier animal, el sonido del agua al caer, etc.

- ***Sonidos artificiales***

Son aquellos generados o influenciados por la acción del hombre, excluyendo la música y la voz humana. Ejemplos de este tipo de sonidos son aquellos provenientes de las máquinas, coches, etc. La fuente que genera estos sonidos puede ser caso de estudio de sistemas ASC.

- ***Voz***

La voz es el sonido generado por el aparato fonador humano. Se puede clasificar la voz por el idioma utilizado, por el hablante, o por el estado emocional del hablante, entre otras. También se puede clasificar por palabras usadas o por fonemas emitidos. Finalmente también se pude clasificar en voz normal o voz patológica en caso de que el hablante padezca alguna dolencia que se refleje en el habla.

- ***Música***

Son sonidos artificiales creados a partir de instrumentos. Existen muchas formas de clasificar la música, según el número de instrumentos, el tipo de instrumentos, el género, el compositor etc.

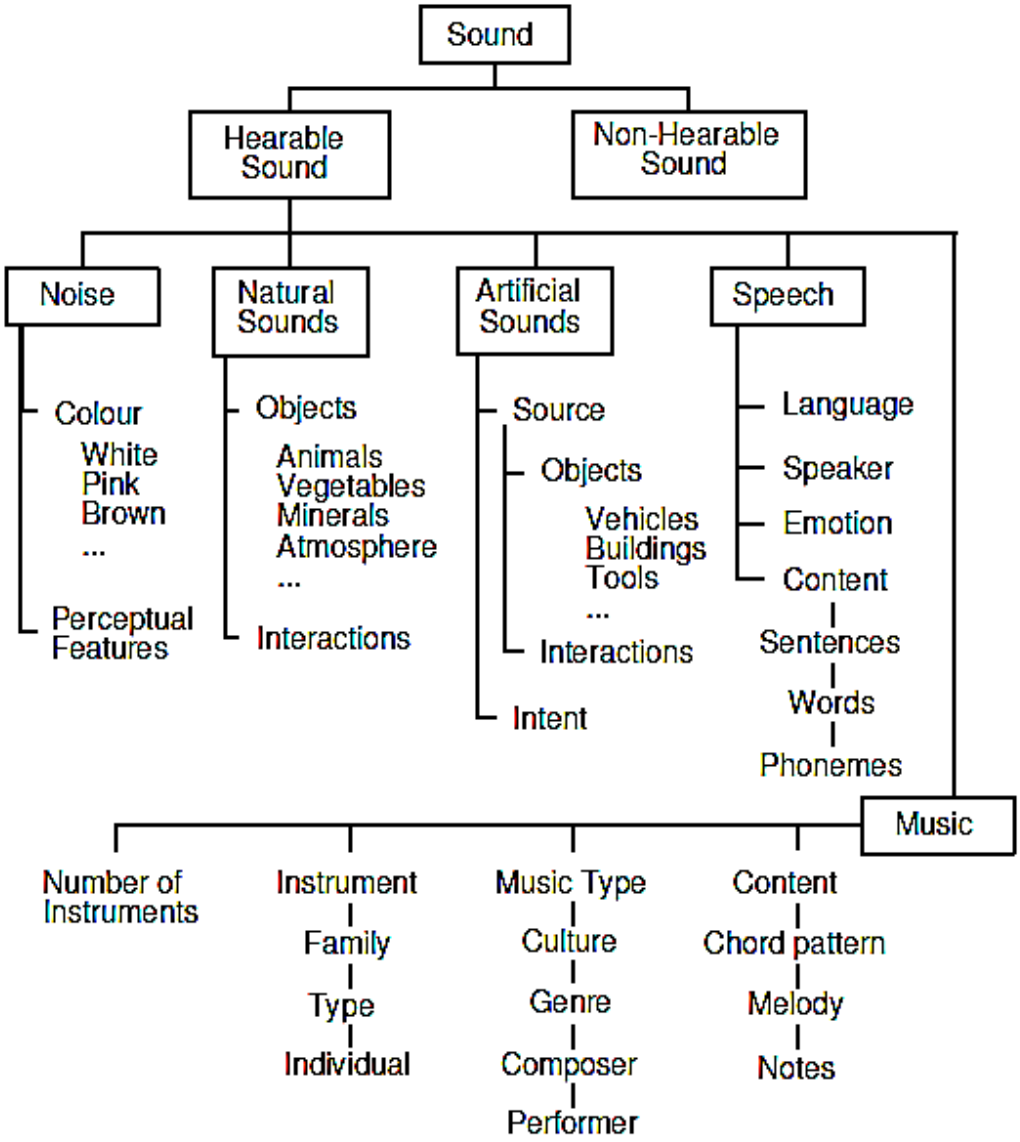


Figura 2.1: Ejemplo de una posible clasificación de los distintos sonidos.[3]

2.1.2 Recuperación de información musical

Los sistemas de recuperación de información musical son un tipo de sistemas ASC que identifican características musicales en el audio. Existen sistemas que detectan la similitud entre dos canciones, sistemas que detectan el tempo de una obra, la melodía etc. Existe una gran cantidad de líneas de investigación a este respecto, para lograr cada vez caracterizar de forma más precisa la música. A este respecto existe una gran comunidad internacional que se centra en la investigación en este campo denominada MIREX (Music Information Retrival Evaluation eXchange) [2].

2.1.2.1 MIREX

MIREX es una comunidad coordinada y dirigida por la IMIRSEL (International Music Information Retrieval Systems Evaluation Laboratory) cuya sede se encuentra en la universidad de Illinois (UIUC). IMIRSEL fue fundada con el ánimo de crear la infraestructura necesaria para la evaluación científica de diferentes técnicas utilizadas por investigadores interesados en el mundo de la recuperación de información musical.

MIREX organiza congresos anuales que reúnen a investigadores de todo el mundo, en los cuales se exponen diferentes soluciones para los problemas a tratar en ese año. La estructura del programa es la siguiente, a principios de año se publica en su página web las tareas que se quieren desarrollar durante este año y que serán tratadas en el congreso que se celebra a finales de año, cada vez en un país diferente, siendo en el año 2015 en Málaga del 26 al 30 de octubre. Los participantes deben elegir uno de los temas propuestos y desarrollarlo individualmente reportando un informe final con un formato determinado que será posteriormente evaluado.

En las conferencias se realizan actividades en las cuales se fomentan discusiones sobre las últimas y más novedosas ideas que se están desarrollando ahora mismo, siempre relacionadas con el mundo de la caracterización musical. Para este propósito se organizan sesiones en las que no se requiere de resultados ni evaluaciones, simplemente se espera una lluvia de ideas interesantes de todos los participantes.

2.1.3 Sistemas de búsqueda de audio en audio

Los sistemas de búsqueda de audio en audio tienen como objetivo la búsqueda de coincidencias entre dos audios. Se pueden considerar sistemas ASC en el sentido de que tratan de clasificar audios mediante la similitud entre ellos. La idea es conseguir una coincidencia entre un audio contaminado por ruido y distorsión y ese mismo audio sin perturbaciones. Para ello se suele tener previamente procesado el audio original para poder realizar la búsqueda de la muestra contaminada de forma eficiente.

Existen diferentes tipos de sistemas de búsqueda de audio en audio:

- **Sistemas de búsqueda unitaria**

  Aquellos que realizan una búsqueda de un audio que contenga a una muestra grabada y la respuesta del sistema es una coincidencia probable entre ellos.

Dentro de este grupo se encuentran los sistemas de reconocimiento de música como puede ser Shazam o Soundhound.

- **Sistemas de sincronismo**

  Son sistemas similares a los anteriormente descritos, pero que no solo indican una coincidencia probable, sino un instante de tiempo de ocurrencia de una muestra sobre el audio original. Audible Magic y su producto Media Syncronization destaca como representante de este tipo de sistemas, asimismo, el sistema objeto de estudio de este proyecto también pertenece a este grupo.

2.1.3.1 Audible Magic

Audible Magic es una compañía que desarrolla software dedicado al reconocimiento automático de contenido multimedia (ACR, Automatic Content Recognition) [5]. Fue creada en 1999 con el objetivo de aportar una nueva tecnología en el mundo de la identificación de audio, desde la identificación de contenidos acústicos, hasta la sincronización. A día de hoy se trata de una de las empresas punteras en el sector, desarrollando aplicaciones independientes con fines comerciales. Cuenta con una enorme base de datos con más de diez millones de canciones, así como acuerdos con los estudios más importantes como NBC, Fox, Warner Bros, Sony Pictures, etc.

Entre las aplicaciones desarrolladas por Audible Magic queremos hacer mención a una de ellas denominada Media Syncronization. Esta aplicación permite sincronizar eventos mediante la captura de audio con un teléfono móvil. Se basa en la sincronización de una señal de audio desconocida con un conjunto de pistas de referencia que contiene la base de datos. Permite la activación y seguimiento de actividades en el tiempo, como puede ser la aparición de información extra sobre un programa de televisión en tú teléfono o Tablet mientras estás viendo el programa. Permite también la participación interactiva del usuario, así como la aparición de publicidad en tiempo real.

El sistema es rápido y preciso, con una resolución de 25ms, lo que equivale a un fotograma. También es fácil de integrar en cualquier dispositivo, ya que está disponible para los principales sistemas operativos. Los precios varían en función del número de usuarios para cada aplicación pero parte desde los 1000 $/mes para aplicaciones con bases de datos local y de 2000$/mes para el resto.

2.2. Técnicas comunes en procesamiento de audio

2.2.1 Procesamiento de señales analógicas

Lo que tienen en común todas las tecnologías descritas anteriormente, es que todas trabajan con parámetros característicos extraídos del audio digital. El procesamiento digital de señales es el proceso por el cual se transforma una señal analógica en datos digitales (cadenas de bits) para poder ser tratados mediante software. El proceso comienza con una conversión analógica/digital:

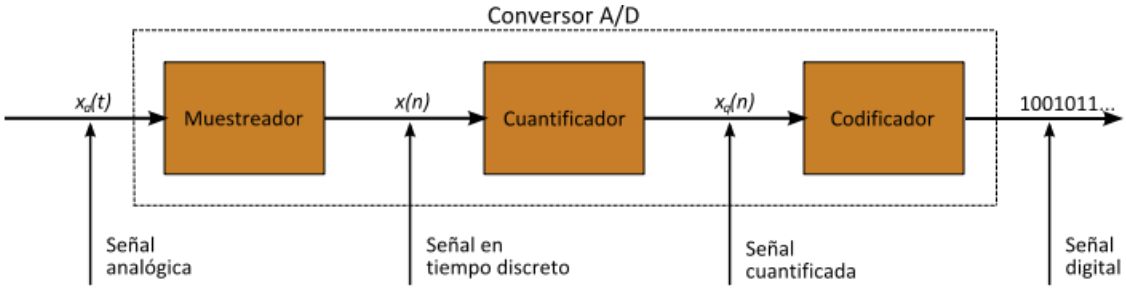


Figura 2.2: Conversor analógico/digital.

2.2.1.1 Muestreo

La señal analógica se muestrea siguiendo el criterio de Nyquist, el cual dice que la frecuencia de muestreo debe ser al menos el doble que la frecuencia máxima de la señal analógica. El oído humano es capaz de percibir sonidos de hasta 20KHz, por lo que una frecuencia de muestreo habitual para señales de audio suele ser 44.1KHz.

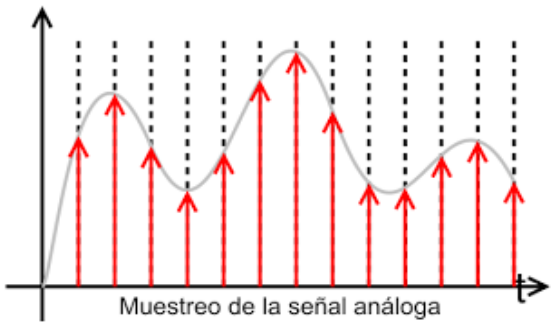


Figura 2.3: Muestreo de señal analógica. [7]

El valor de la n-ésima muestra tomada de la señal analógica tiene un valor de amplitud igual al valor de la señal analógica en ese instante (T_s es el periodo de muestreo).

$$x(n) = x_a(nT_s), \qquad -\infty < n > \infty$$

2.2.1.2 Cuantificación

El objetivo de la cuantificación es transformar la señal discreta de entrada de amplitud continua, en una señal discreta cuya amplitud esté dentro de un conjunto finito de valores posibles. Se obtiene mediante el establecimiento de niveles de cuantificación y encasillando cada muestra aleatoria dentro de uno de estos niveles. El número de niveles de cuantificación establecerá la resolución del cuantificador, e influirá en la cantidad de ruido de cuantificación que se introduce a la señal. En audio es común trabajar con cuantificadores de 16 bits o incluso de 24 bits.

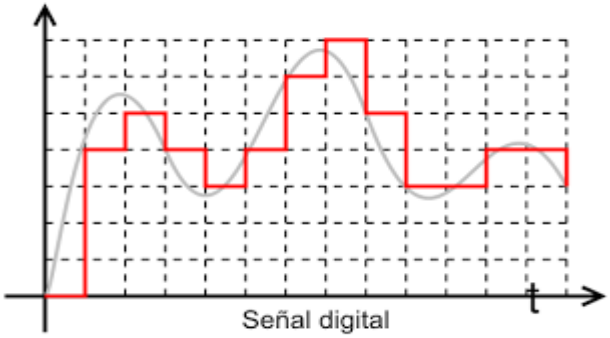


Figura 2.4: Cuantificación de señal discreta. [7]

2.2.2 Parametrización del audio

Se define parametrización de audio como el proceso por el cual se obtienen características relevantes de la señal de audio digitalizada. La idea es transformar la señal digital en una serie de coeficientes característicos que suelen estar basados en la forma en la que el oído humano responde ante los estímulos sonoros. El proceso tiene varios pasos como veremos a continuación.

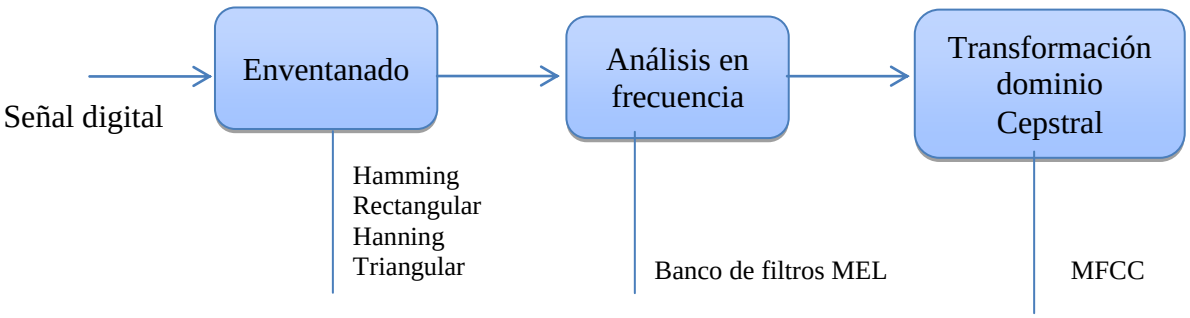


Figura 2.5: Parametrización del audio. [7]

2.2.2.1 Enventanado

La señal de audio es un proceso no estacionario, lo que dificulta el análisis de la misma mediante una transformación en frecuencia. Debido a este problema, se debe tratar descomponer la señal en segmentos cuasi-estacionarios sobre los que se pueda aplicar dicha transformación. Este proceso se denomina enventanado, y consiste en la obtención de tramas de la señal del orden de decenas de milisegundos. Cada una de estas tramas es multiplicada por una función limitada en el tiempo de modo que fuera de ese intervalo su valor sea nulo. Se trata por tanto de agrupar la señal digital *x(n)* en bloques de *N* elementos, y multiplicarlos en el tiempo por una ventana *w(n).*

El tamaño de ventana elegido (N) determina la resolución en frecuencia de la representación. Con ventanas cortas se obtiene una buena resolución temporal (mayor capacidad de discriminar entre eventos próximos temporalmente) sin embargo se consigue peor resolución frecuencial y viceversa. Las ventanas utilizadas habitualmente se sitúan entre los 20 y 30 *ms* con las que se consigue un compromiso razonable entre resolución en tiempo y en frecuencia para la señal de voz. Las ventanas más comunes son:

Ventana	Fórmula
Rectangular	$\mathrm{w(n)} = 1 \qquad 0 < n < N$
Triangular	$\mathrm{w(n)} = \frac{\mathrm{N}}{2} - \lvert n - \frac{N-1}{2} \rvert \qquad 0 < n < N$
Hanning	$\mathrm{w(n)} = \frac{1}{2} - \frac{1}{2}\cos\left(\frac{2\pi n}{N}\right) \qquad 0 < n < N$
Hamming	$\mathrm{w(n)} = \frac{27}{50} - \frac{23}{50}\cos\left(\frac{2\pi n}{N}\right) \qquad 0 < n < N$

Tabla 2.1: Principales enventanados para procesamiento de audio.

Las características más reseñables de las ventanas son la anchura del lóbulo central y la amplitud de los lóbulos laterales. Para conseguir una buena resolución en frecuencia se requiere un lóbulo central lo más estrecho posible. Por otro lado, los lóbulos laterales son los causantes de distorsión armónica por lo que cuanto menor sea su amplitud, tendremos una señal con menor distorsión.

De todas las ventanas descritas, la ventana rectangular es la que tiene un lóbulo central más estrecho, sin embargo sus lóbulos laterales decaen lentamente por lo que generan un tipo de distorsión conocida como "*efecto de ripple*". La ventana Hamming cumple un buen compromiso en ambos requisitos por lo que la hace ser una de las más usadas en procesamiento de audio. Existen muchos otros tipos de ventana como pueden ser Bartlett, Blackman o Kaiser.

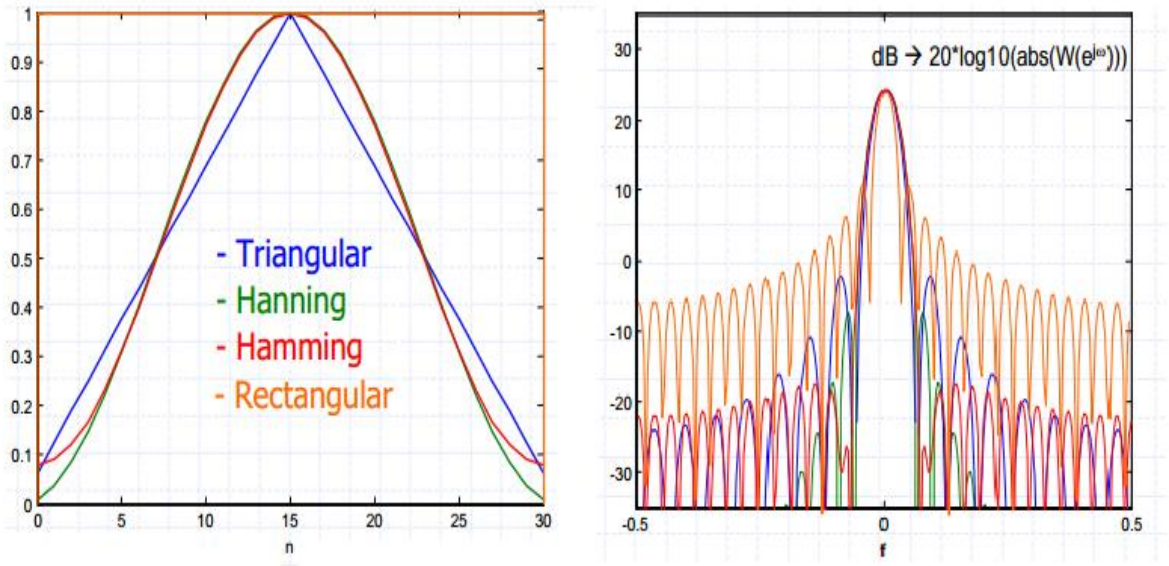


Figura 2.6: Principales ventanas en tiempo y en frecuencia (log). [8]

2.2.2.2 Análisis en frecuencia

El siguiente paso para la parametrización de la señal de audio será la obtención de su espectro en frecuencia. Para ello se lleva a cabo un análisis de Fourier sobre la señal enventanada:

$$S_x(t,k) = \sum_{n=0}^{N-1} x[n]w[n]\mathrm{e}^{-j\frac{2\pi n}{N}k} \quad , \quad 0 \le k < N$$

Donde $\mathrm{S_x(t,k)}$ es la transformada discreta de Fourier de la trama t-ésima en la que se descompuso la señal para el enventanado. N es la longitud de la trama y en este caso el número de puntos de la FFT.

Es común el uso de la transformada rápida de Fourier (FFT) para calcular la transformada discreta de Fourier (DFT), debido a que el coste computacional es mucho menor, por lo que para sistemas de reconocimiento en tiempo real es mucho más

ventajosa. El coste computacional de la FFT es de $O(n\,log\,n)$, mientas que el de la DFT es $O(\mathrm{n}^2)$. Del análisis en frecuencia se obtiene el espectrograma.

2.2.2.3 El espectrograma

El espectrograma es la representación gráfica de la amplitud del espectro de la señal a lo largo del tiempo. Es el resultado del análisis frecuencial de tramas enventanadas de una señal. Representa la energía de la señal en cada banda de frecuencias en cada instante de tiempo, por lo que representa una definición muy precisa de cómo es la señal y sus características.

Para su creación se calcula la FFT de una de las tramas enventanadas en las que se divide la señal, y los valores de la amplitud de la transformada de Fourier se representan sobre una gráfica. Seguidamente se desplaza la ventana sobre la trama siguiente y se vuelven a calcular estos valores y se representan seguidos sobre la misma gráfica. Esta operación se repite a lo largo de toda la señal, de modo que al final tengamos una representación de las variaciones de energía en cada banda de frecuencia a lo largo del tiempo.

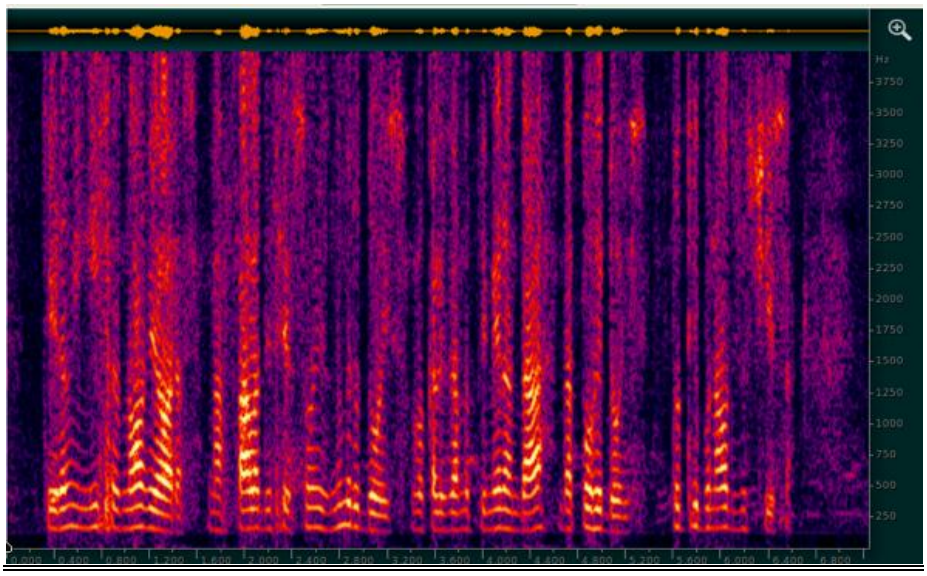


Figura 2.7: Espectrograma de una señal de audio.

2.2.2.4 Escalas perceptuales

Como vimos en el apartado anterior, el espectrograma es una representación lineal en frecuencia, es decir, representa todas las bandas de frecuencia de forma lineal, asignándole a cada una su cantidad de energía. Sin embargo, el sistema auditivo

humano, y en particular la percepción de los sonidos, se ha venido estudiando desde hace muchos años por una disciplina conocida como psicoacústica. Una de las conclusiones más claras de esta disciplina es que la percepción humana del sonido no sigue un comportamiento lineal respecto a las diferentes bandas de frecuencia. Es por eso que, para tratar de modelar este peculiar comportamiento, se han desarrollado una serie de escalas perceptuales que se ajustan más a lo que realmente percibe como sonido el ser humano. Dentro de este grupo de escalas perceptuales, podemos destacar las siguientes:

2.2.2.4.1 La escala BARK

La escala Bark es una escala psico-acústica propuesta por Eberhard Zwicker en 1961. Posteriormente, Heinrich Barkhausen propuso una nueva medida subjetiva sobre las bajas frecuencias, y la escala adoptó su nombre. La escala está compuesta por 24 bandas críticas, que corresponden con el rango de 1 a 24 Barks. Los valores de la escala en Hertzios son los siguientes:

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

Tabla 2.2: Bandas críticas de la escala BARK. [9]

Se establece unos anchos de banda distintos para cada rango de frecuencias. Esto se debe a que el oído humano no se comporta de forma lineal ante los estímulos, teniendo una resolución mayor a baja frecuencia que a alta. Esto quiere decir que para un humano es más fácil diferenciar dos sonidos graves que dos sonidos agudos, debido a la propia fisiología del oído. La conversión de frecuencia en Hertzios a Barks es:

$$Ancho\ de\ Banda\ (Hz) = \frac{52548}{z^2 - 52.56z + 690.39} \quad con\ z\ en\ Barks$$

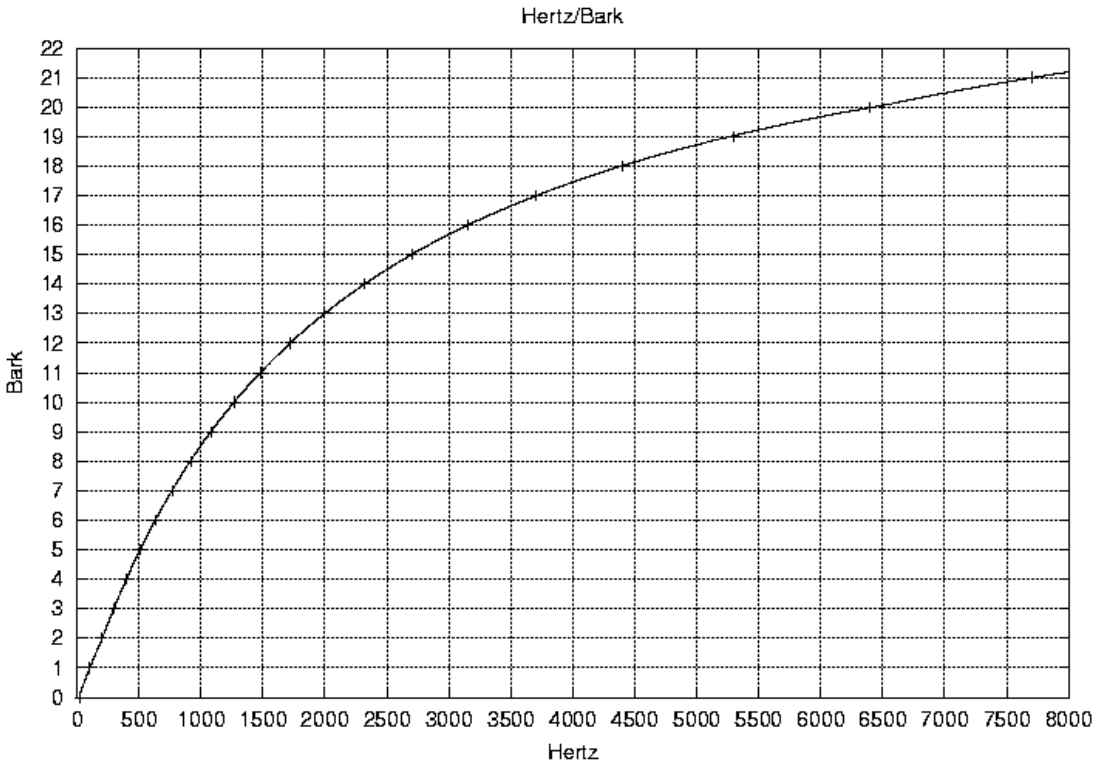


Figura 2.8 Escala Bark.[9]

2.2.2.4.2 La escala MEL

La denominada escala MEL por Stevens, Volkman y Newman en 1937 (proveniente del término *melody*), se trata de una escala perceptual de tonos musicales. Los valores MEL representan frecuencias de tonos equidistantes entre ellos desde el punto de vista perceptual. Esta valoración de equidistancia entre tonos, es llevada a cabo por oyentes experimentados y modelado como una función logarítmica.

El punto de referencia de la escala MEL son los 1000 MELs, que equivalen a 1000 Hz. A partir de unos 500 Hz, los intervalos que representan incrementos de pitch iguales, son cada vez más grandes, hasta los 10 KHz. Como resultado, cuatro octavas

de la escala Hertz son comprimidas en dos de la escala MEL a partir de los 500 Hz. (una octava es el intervalo entre un tono musical y otro cuya frecuencia sea el doble o la mitad).

La relación entre la escala MEL y la escala Hertz establecida por Stevens y Volkman en 1937 es por tanto la siguiente:

$$m = 2595 \log_{10}(1 + \frac{f}{700})$$

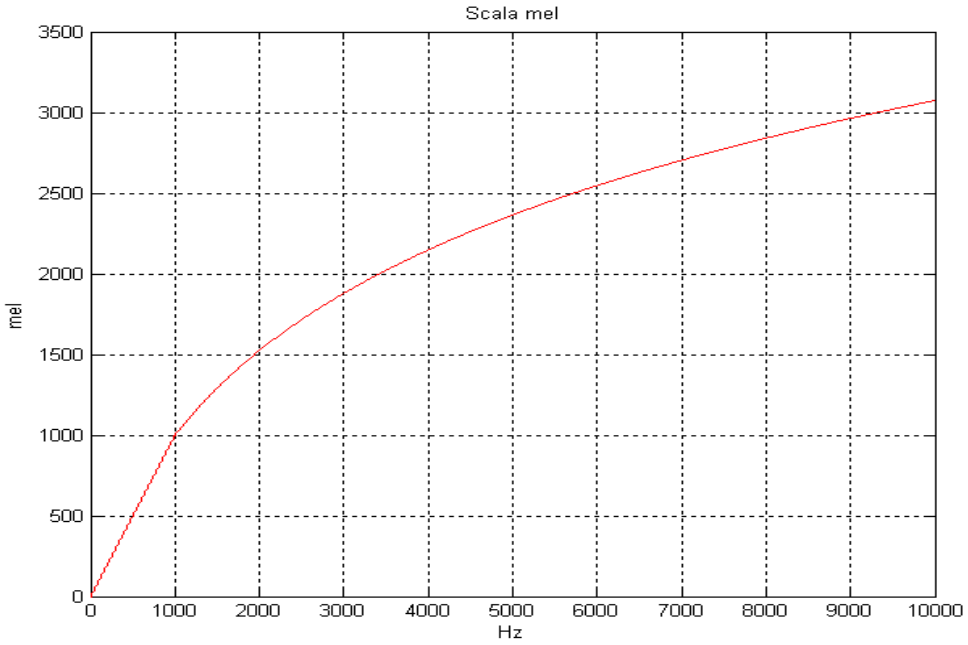


Figura 2.9: Escala MEL vs escala Hertz.

La escala MEL se caracteriza por ser lineal a bajas frecuencias (hasta 1KHz) y logarítmica para altas frecuencias (a partir de 1KHz). Es la escala en la que se basa la obtención de coeficientes espectrales MFCCs (Mel-Frequency Cepstral Coefficients), que son ampliamente utilizados en el mundo del procesamiento de audio como veremos más adelante.

2.2.3 Algoritmos de extracción de coeficientes espectrales

Los sistemas de análisis de audio en tiempo real se caracterizan por brindar una respuesta rápida tras el análisis de un audio de entrada. Para ello llevan a cabo un procesado de la señal digitalizada. Normalmente este procesado es muy costoso computacionalmente, por lo que para que sea efectivo tiene que trabajar con cadenas de datos relativamente cortas para poder ser eficiente. Esta es la premisa por la cual se trata de descomponer el complejo espectro de la señal en una serie de parámetros que lo caractericen fielmente, pero reduzcan la carga de datos a unas centenas de bits con las que sí se pueda trabajar fácilmente.

Existen muchos tipos de coeficientes característicos, pero lo más usados en tecnologías de análisis de audio debido a su precisión y robustez son los llamados coeficientes MFCCs (Mel Frecuency Cepstral Coefficients).

2.2.3.1 Coeficientes MFCC

Los MFCC, son los parámetros que se obtienen del análisis de la señal de audio mediante el banco de filtros MEL. Se usan en gran cantidad de aplicaciones de reconocimiento de voz así como en el campo de recuperación de información musical, como puede ser identificación de géneros musicales, o como en este caso medidas de similitud entre dos pistas de audio.

2.2.3.1.1 Banco de filtros MEL

La naturaleza del oído humano hace que la percepción de sonidos no sea lineal con respecto al espectro de frecuencias. Por tanto parece lógico pensar que la forma de operar con el espectro para obtener las características más reconocibles para su análisis debe ser también no lineal. En este punto es donde se introduce la idea del banco de filtros que modelados de forma adecuada, permitirán obtener esa deseada resolución frecuencial no lineal.

El banco de filtros MEL está basado en filtros triangulares equidistantes a lo largo de la escala MEL. Los filtros se solapan entre ellos de tal forma que la frecuencia máxima del filtro *n* es la misma que la frecuencia mínima del filtro *n+2, y* coincide con el máximo de respuesta en amplitud del filtro n+1.

$$f_{n,max} = f_{n+2,min} = f_{n+1}^{A\,max}$$

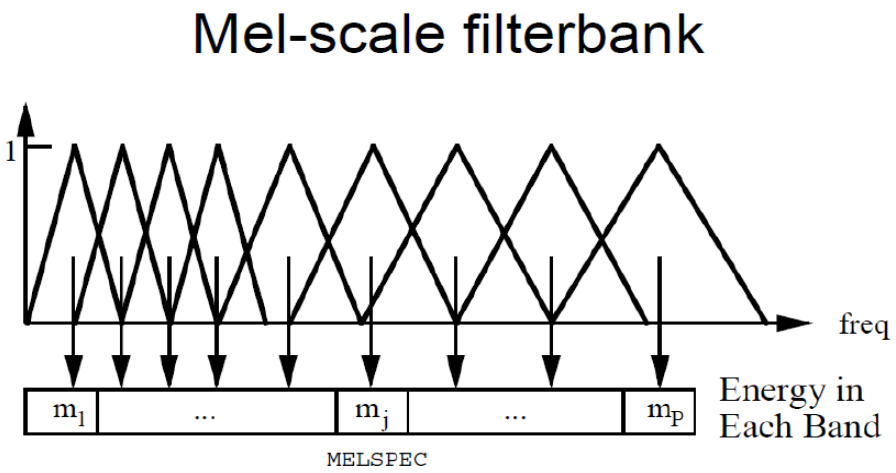


Figura 2.10: El banco de filtros MEL. [11]

Como se observa, la concentración de filtros es mayor para frecuencias más bajas, que se corresponde con la pendiente mayor para bajas frecuencias de la escala MEL e indica que existe más información útil a la hora de caracterizar una señal para dichas frecuencias. Cada uno de los filtros viene definido por la siguiente ecuación:

$$H_m(k) = \begin{cases} \dfrac{kf_s/N - f_b(m-1)}{f_b(m) - f_b(m-1)}, & f_b(m-1) \le kf_s/N \le f_b(m) \\ \dfrac{f_b(m-1) - kf_s/N}{f_b(m+1) - f_b(m)}, & f_b(m) \le kf_s/N \le f_b(m+1) \\ 0, & resto \end{cases}$$

El rango de frecuencias útiles del banco de filtros MEL, suele ir desde el cero hasta la frecuencia de Nyquist, sin embargo puede ser útil limitar en banda el banco de filtros con el propósito de rechazar bandas no deseadas, en las que la señal no tenga energía útil para su caracterización. Para ello se puede fijar una frecuencia mínima inferior y una frecuencia máxima superior.

2.2.3.1.2 Obtención de coeficientes MFCC

Los coeficientes MFCCs, son una representación de la envolvente espectral de la señal de audio, así pues de ellos se extraen importantes características de la señal. Concretamente, el coeficiente C_0 indica la energía de la señal y el coeficiente C_1 indica el balance global de energía entre bajas y altas frecuencias.

Los coeficientes MFCCs se calculan a partir de la cantidad de energía de la señal en cada banda delimitada por cada uno de los filtros del banco de filtros (m_i). La forma de obtener estos coeficientes es la siguiente:

1) Se calcula la FFT de un frame de una señal de audio conservando solo el módulo.
2) Se aplica el banco de filtros MEL tomando como entrada dicho módulo.
3) Se calcula el logaritmo de la salida de cada filtro MEL.
4) Se aplica la transformada discreta del coseno DCT sobre el resultado, conservando sólo los N primeros coeficientes (habitualmente entre 9 y 20).
5) Los MFCC serán los valores resultantes.

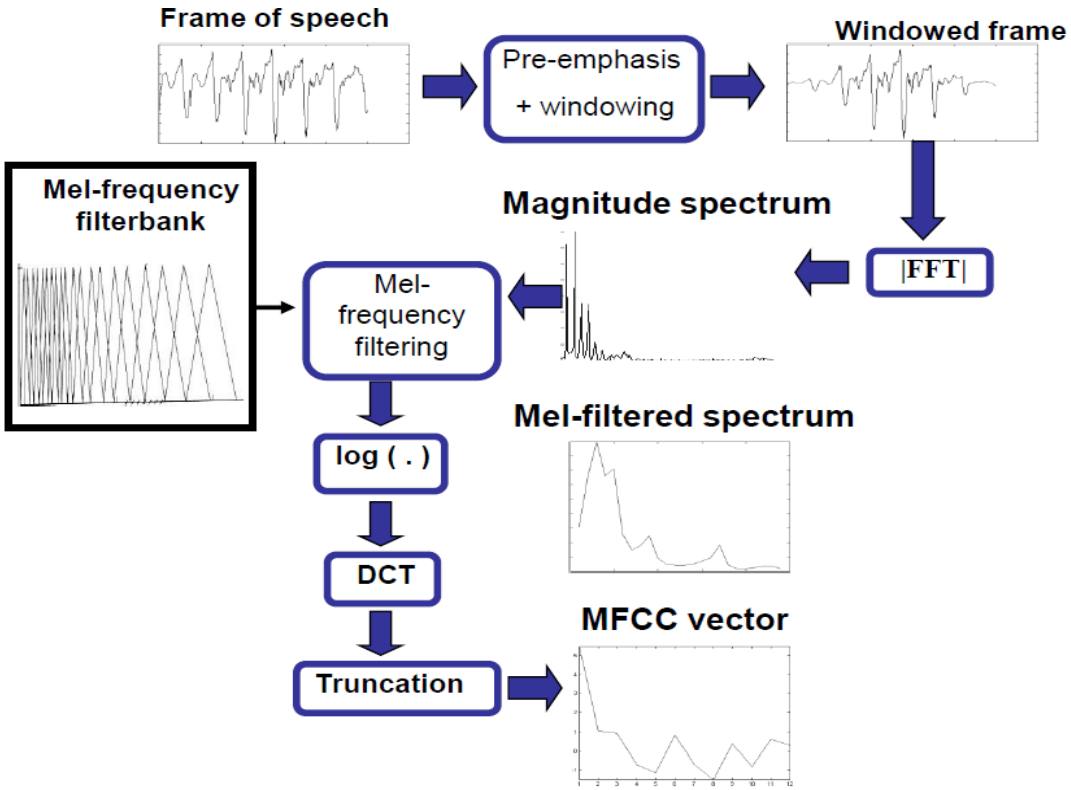


Figura 2.11: Proceso obtención de MFCCs. [12]

El proceso matemático de obtención de coeficientes MFCCs es el siguiente:

1) Se realiza el análisis en frecuencia de la seña de entrada $x[n]$, mediante el enventanado $w[n]$ y la transformada de Fourier dependiente del tiempo (Short-Time Fourier Transform, STFT):

$$S_x(t,k) = \sum_{n=0}^{N-1} x[n]w[t-n]\mathrm{e}^{-j\frac{2\pi n}{N}k} \ , \qquad 0 \le k < N$$

2) Después se aplica el banco de filtros MEL, y se calculan los logaritmos de las energías a la salida de cada uno de los filtros:

$$X_t^l(m) = 10log_{10}\left[\sum_{k=0}^{N-1} |S_x(t,k)|^2 H_m(k)\right], \qquad 0 \le m < N_b - 1$$

Donde $\mathrm{X_t^l(m)}$ representa la log-energía en la banda m, para la trama t.

3) Finalmente, se aplica la transformada discreta del coseno (DCT) a las log-energías en las distintas bandas.

$$X_t^c(i) = \sum_{k=0}^{N_b-1} X_t^l(m)\cos\left(i(m-0.5)\frac{\pi}{N_b}\right), \qquad 0 \le i < N_c - 1$$

Donde $X_t^c(i)$ son los coeficientes MFCC y N_c es el número de coeficientes considerados útiles. Este número suele ser menor que el número de filtros utilizados, puesto que se considera que para frecuencias elevadas, estos parámetros contienen muy poca información, y por tanto para no aumentar el coste computacional del proceso, se descartan.

Una característica importante de los MFCCs es que debido al empleo de la transformada discreta del coseno (DCT) se concentra la mayor parte de la energía de la señal en los primeros coeficientes de la DCT, de modo que preservando sólo los primeros coeficientes de la DCT se preserva una buena parte de la energía contenida en la señal. El mismo principio se emplea para comprimir imágenes, por ejemplo en las imágenes JPEG, aunque empleando una DCT bidimensional en lugar de unidimensional. Otra característica importante de los MFCCs es que debido a la utilización de la DCT, los distintos coeficientes MFCC obtenidos muestran un nivel muy bajo de correlación, lo que permite emplear modelos más sencillos en etapas posteriores.

El número de coeficientes útiles con los que trabaja habitualmente en reconocimiento de voz es de 13, y este es también el número de coeficientes que emplearemos en este trabajo.

2.3 Técnicas empleadas por los sistemas de búsqueda de audio en audio

El objetivo de los sistemas de búsqueda de audio en audio es obtener una correspondencia entre dos audios mediante el análisis frecuencial de estos. Existen diferentes técnicas para conseguirlo, dependiendo de la forma en la que se obtienen los *fingerprints* o huellas espectrales de los audios. Por un lado están los sistemas tipo SHAZAM, y por otro lado los sistemas basados en comparación de coeficientes espectrales, como el Medya Syncronization, desarrollado por Audible Magic.

2.3.1 Shazam

Shazam es una aplicación para telefonía móvil cuya funcionalidad es la identificación de música. Utiliza el micrófono incorporado que tienen los teléfonos móviles para poder grabar una pequeña muestra sonido de la música que este sonando a su alrededor. A partir de dicha grabación, crea una huella digital acústica y se compara en una gran base de datos para encontrar coincidencias. Si la búsqueda tuvo éxito, el usuario recibe información de la canción que suena, y enlaces directos a otros servicios de compra de música por internet.

El funcionamiento de este sistema está basado en el análisis de espectrogramas. La clave es la obtención de los picos más significativos del mismo y la codificación como una secuencia de datos para poder realizar la búsqueda. La dificultad reside en mejorar la robustez frente a problemas que serán inherentes a la propia grabación, como será el ruido de fondo, así como la propia distorsión del micro, o algoritmos de compresión del teléfono adaptados para la grabación de voz y no de música.

2.3.1.1 Fingerprints partir del espectrograma

La forma de calcular los *fingerprints* desarrollada por Shazam se basa en el análisis del espectrograma de la muestra de audio. Se realiza un análisis a todo el espectrograma identificando los picos de energía, que serán seleccionados solo sí tienen una cantidad de energía significativamente mayor que la de sus vecinos más próximos. La elección de los picos espectrales se fundamenta en que estos serán menos vulnerables frente al ruido o distorsiones. La densidad de puntos se tomará de tal forma que se realice una cobertura uniforme de todo el espectrograma. De esta forma, el complicado análisis de un espectrograma, quedará reducido a una constelación de puntos, como se observa a continuación:

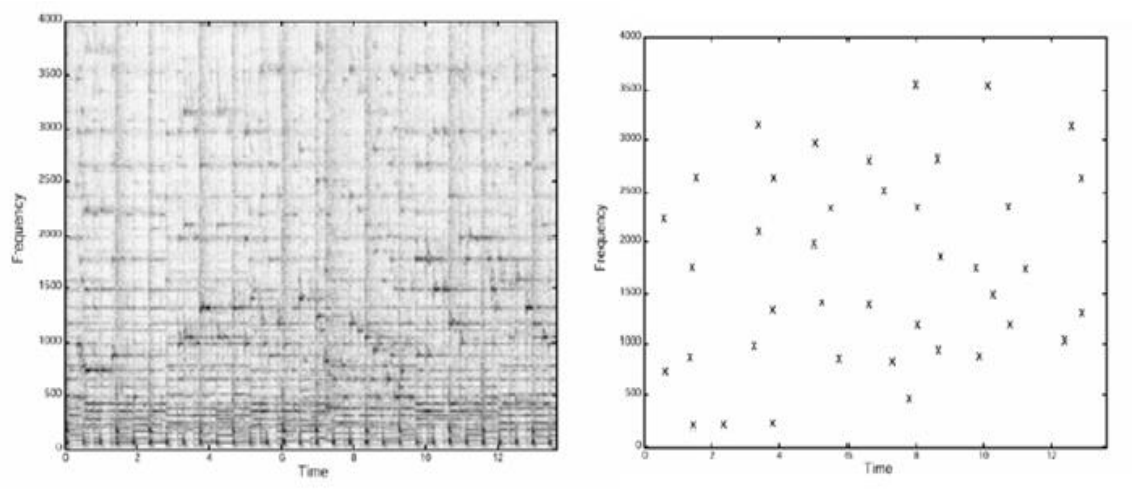


Figura 2.12: Constelación de picos de un espectrograma.[13]

Gracias a la simplificación del espectrograma, el problema se reduce a buscar una constelación de puntos dentro de un universo de puntos que tiene la base de datos donde están registradas todas las constelaciones de las pistas originales. Se define una correspondencia de un pico como la existencia de éste tanto en el audio original como en la muestra grabada con un margen suficientemente pequeño de distancia entre ellos. De esta forma, será posible obtener una coincidencia entre ambas constelaciones, si se obtiene una serie de correspondencias alineadas en el tiempo.

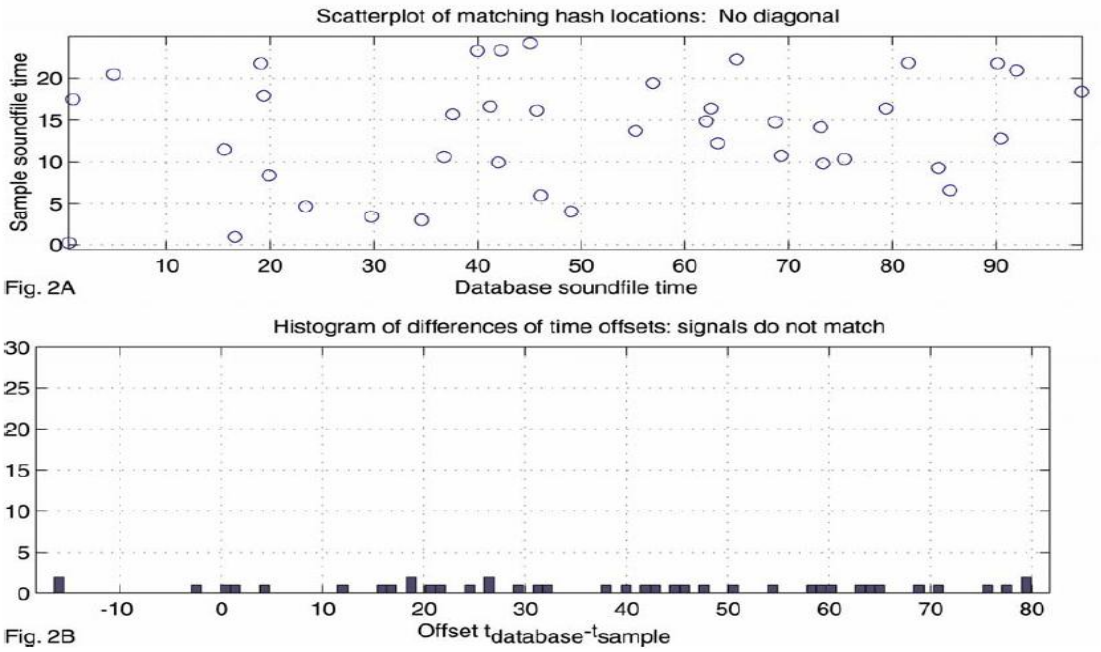


Figura 2.13: SHAZAM: Diagrama de detecciones no alineadas.

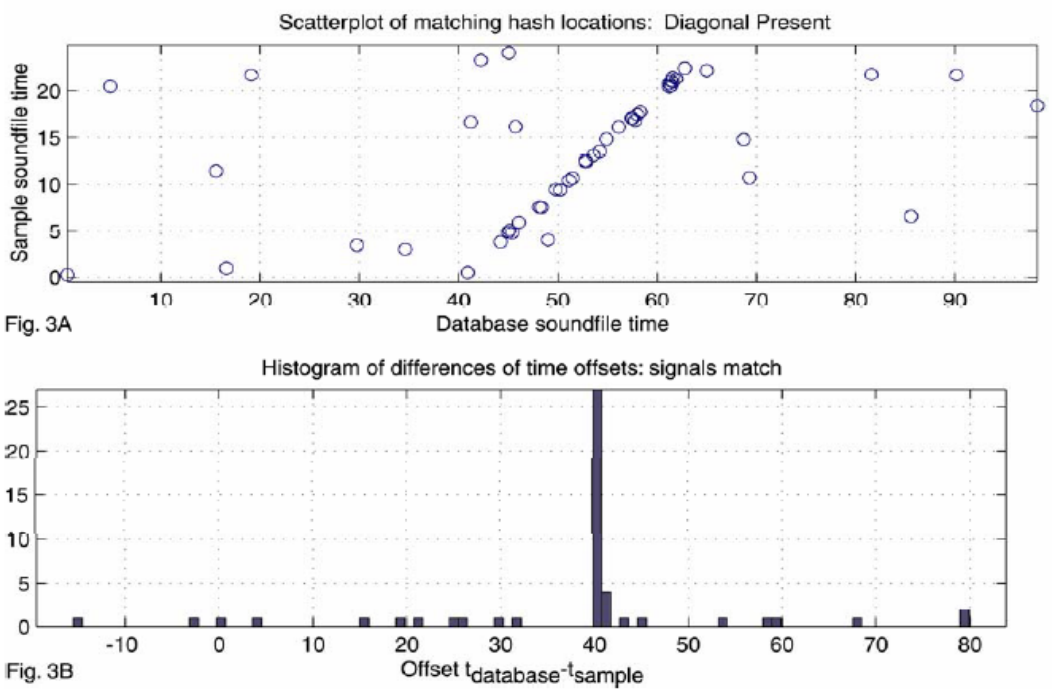


Figura 2.14: SHAZAM: Diagrama de detecciones alineadas.

Si se produce una serie de correspondencias alineadas en el tiempo, el histograma que representa la latencia de dichas correspondencias mostrará un valor máximo claramente definido, que corresponde con la diferencia de tiempo entre ambas pistas, y será síntoma de coincidencia encontrada.

2.3.2 Búsqueda lineal de coeficientes

El software de este proyecto es un programa basado en la búsqueda mediante comparación directa de coeficientes espectrales. Los sistemas de búsqueda mediante comparación directa de coeficientes basan su funcionamiento en el análisis del espectro, al igual que el sistema Shazam. Sin embargo en este caso, en lugar de buscar picos frecuenciales, se extraen una serie de coeficientes espectrales por cada trama en la que se subdivide la muestra y desplazando las tramas, se consigue transformar la señal de audio en una secuencia de vectores de coeficientes espectrales. El tipo de coeficientes más utilizados para este tipo de sistemas son los coeficientes MFCC, que como vimos anteriormente, están basados en la forma en que el sistema auditivo humano percibe el sonido (escala MEL).

En este caso, el problema reside en alinear dos vectores de coeficientes basándose en la distancia espectral que los separa. Esta distancia se calcula como la distancia entre los vectores definidos por los valores de los coeficientes de ambas muestras. Normalmente se intenta alinear una muestra de unos pocos segundos con audios de

varios minutos, horas o incluso días, por lo que el vector de coeficientes de la muestra se desplaza a lo largo de todo el vector del audio original calculando la distancia total en cada superposición. De este modo, cuando se obtenga el valor de distancia más pequeño, se podrá afirmar que en ese momento es cuando más se parecen ambas muestras y por tanto tendremos también el valor de latencia entre ellas. El tipo de búsqueda más básico es el de búsqueda lineal, aunque puede ser acelerado mediante diversas técnicas de *clustering* y vecinos cercanos. En este proyecto todavía no estamos interesados en acelerar el sistema de búsqueda, sino sólo en mejorar su robustez frente al ruido.

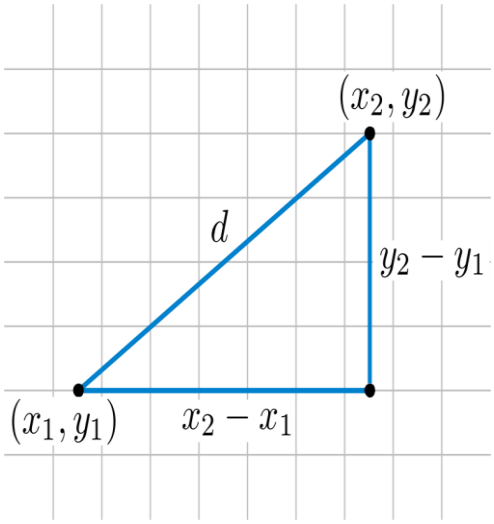
2.3.2.1 Concepto de distancia espectral

El término distancia espectral hace referencia a la magnitud de la desigualdad entre dos vectores de coeficientes espectrales. Para el cálculo de esta distancia, se tienen en cuenta la diferencia entre todos los coeficientes de ambos vectores uno a uno. Existen muchas formas de calcular esta distancia. En concreto tantas como distancias posibles definidas sobre el espacio vectorial de dimensión *N*, siendo *N* el número de coeficientes de los parámetros espectrales que utilicemos. En este proyecto vamos a emplear únicamente dos métricas: distancia City Block y distancia Euclídea.

2.3.2.1.1 Distancia Euclídea

La distancia Euclídea entre dos puntos es la distancia ordinaria. Para un espacio bidimensional de coordenadas cartesianas, la distancia Euclídea se define como la longitud de la recta que separa ambos puntos:

$$\mathrm{D_E} = \sqrt{(\mathrm{x_2} - \mathrm{x_1})^2 + (\mathrm{y_2} - \mathrm{y_1})^2}$$



Para un espacio n-dimensional, como es el caso del espacio de coeficientes MFCC:

$$D_E = \sqrt{\sum_{i=1}^{n} (p_i - q_i)^2}$$

Donde p_i y q_i representan la coordenada *i* de los vectores p y q respectivamente, que contienen los coeficientes MFCC.

2.3.2.1.2 Distancia City Block

La distancia City Block o L_1, es la suma de las diferencias absolutas de las coordenadas, o en este caso de los valores de los coeficientes. También llamada distancia Manhattan, debido a la similitud de la forma de calcular esta distancia con la forma de moverse por una ciudad "cuadriculada". Por tanto para un espacio bi-dimensional tenemos:

$$D_1 = |x_2 - x_1| + |y_2 - y_1|$$

Y para el espacio n-dimensional, como por ejemplo el espacio de coeficientes MFCC:

$$D_1 = \sum_{i=1}^{n} |p_i - q_i|$$

Donde $p_i = (p_1, p_2, \dots, p_{n)}$ y $q_i = (q_1, q_2, \dots, q_{n)}$ son los vectores que representan el k-esimo coeficiente MFCC.

2.3.2.1.3 Cálculo de distancias acumuladas entre secuencias de frames

Hasta ahora hemos visto cómo se calcula tanto la distancia Euclídea como la distancia City Block para cada par de frames. Si se trata de comparar secuencias de frames (como en realidad funciona el programa), se calculan de la siguiente manera:

- **Euclídea**

$$D_{E(key,RTkey)} = \sum_{k=1}^{M} \sqrt{\sum_{i=1}^{n} (C_{k,i}^{key} - C_{k,i}^{RTkey})^2}$$

- **City Block**

$$D_{1(key,RTkey)} = \sum_{k=1}^{M} \sum_{i=1}^{n} |C_{k,i}^{key} - C_{k,i}^{RTkey}|$$

Donde M es el número de coeficientes MFCC, en nuestro caso 13, y n es la longitud del vector de coeficientes, que para el programa se ha establecido en 8 segundos de muestra, que equivalen a 500 frames.

2.4 Técnicas de robustez frente al ruido

2.4.1 El ruido

Dentro del mundo de las comunicaciones, existen dos factores que corrompen la señal, se trata del ruido y la distorsión. La distorsión la degradación de la señal debido a factores como la no linealidad del canal o a limitaciones de los sistemas que actúan sobre la misma. El ruido es un elemento independiente que también degrada la calidad de la señal, lo que puede acarrear consecuencias negativas en el procesado de la misma. Se han desarrollado diversas técnicas que tienen como objetivo eliminar la mayor cantidad de ruido posible para optimizar el rendimiento de los procesadores de señal.

Se puede definir el ruido como "toda señal no deseada, que interfiere en la comunicación, procesamiento o medida de otra señal portadora de información" [1]. Esta definición es completamente general, y antes de continuar, será preciso realizar una clasificación más detallada de los distintos tipos de ruido así como de su procedencia. Por tanto, el ruido se puede clasificar en:

- **Ruido aditivo acústico**

  Como ruido aditivo acústico, se considera todo tipo de ruido que proviene de cualquier tipo de fuente que coexiste en el mismo entorno acústico y que, por tanto, se suma a la señal deseada.

- **Señales interferentes (ruido aditivo en el canal)**

  Se trata de cualquier otra señal que aparezca en el mismo canal, que no es de interés, y que se mezcla con la señal deseada sumándose también a la misma.

- **Distorsiones de la señal en el canal (ruido convolutivo)**

  Al pasar la señal por un canal no ideal siempre se suelen producir distorsiones de la misma, que en muchos casos se pueden modelar como la convolución de la señal con la respuesta al impulso del canal. Un ejemplo típico en audio es el efecto de la respuesta en frecuencia de los micrófonos. No se trata, por tanto, de una señal que se añade a la original, aunque en muchas ocasiones es útil referirse a estas distorsiones como ruido convolutivo.

- **Reverberación**

  Es un tipo de distorsión provocada por la propagación multitrayecto que se da sobre todo en espacios cerrados. La reverberación produce la adición de versiones retardadas y atenuadas del sonido principal.

- **Eco**

  Se trata de otro tipo de distorsión generada por el acople entre altavoces y micrófonos (eco eléctrico) o aquella generada por las ondas sonoras que rebotan en el entorno acústico y regresan en forma de distorsión (eco acústico). El eco acústico se produce por el mismo fenómeno que la reverberación. La diferencia entre ambos es que el eco se percibe como un segundo sonido en lugar de como una modificación del sonido principal (ello debido a un retardo elevado y a una atenuación no demasiado pronunciada).

Todos estos tipos de ruido y distorsión tienen un campo de estudio propio, que en los últimos años ha avanzado en gran medida gracias al desarrollo de diversas técnicas de procesado de señal. Estas técnicas están siempre orientadas a reducir los efectos negativos que todos estos tipos de ruido y distorsión ejercen sobre las señales. A partir de este punto, se analizará el ruido aditivo acústico, que es el que afectará en mayor medida al desarrollo de este proyecto, y será el que se deberá tratar de compensar para lograr un sistema de búsqueda de audio más robusto frente a este tipo de ruido.

2.4.1.1 Reducción de ruido directamente en la señal de audio.

Existen técnicas de supresión de ruido que permiten reconstruir una señal de audio en la que el ruido se ha atenuado respecto a la señal. Estas técnicas suelen suponer que el ruido es una señal indeseada que se añade a la señal deseada. Están por tanto orientadas fundamentalmente a la supresión de ruido aditivo. Estas técnicas suelen operar mediante filtrado de señal para optimizar parámetros objetivos como la SNR (Signal to Noise Ratio) y el MSE (Mean Squared Error). En concreto el objetivo es aumentar la SNR (es decir conseguir una mayor diferencia entre la potencia de la señal y la del ruido) minimizando el MSE (que mide la distorsión de la señal respecto a la señal original sin ruido). Estos métodos llevan a cabo un procesado espectral de la señal obtenida mediante micrófonos. El número de micrófonos utilizados para realizar la grabación es un factor muy importante. Cuantos más micrófonos haya, mejores estimaciones de ruido se pueden hacer, y con ellas desarrollar filtros adaptados que procesen la señal final. Por ejemplo, si dentro del mismo entorno acústico se tienen

micrófonos separados, unos dedicados a grabar la señal deseada y otros centrados en grabar el ruido ambiente, la obtención del canal de señal con el ruido atenuado se simplifica. Sin embargo en la mayoría de los casos solo se tiene un micrófono, por lo que serán necesarias técnicas más complejas para el filtrado. Las técnicas de estimación y eliminación de ruido (directamente sobre la señal de audio) más importantes desarrolladas hasta la fecha se pueden englobar en los siguientes grupos:

- **Filtrado lineal adaptativo**

  Esta técnica consiste en pasar la señal ruidosa por un filtro lineal que se adapta al ruido que se quiere eliminar, atenuando la componente del mismo e intentando distorsionar la señal lo menos posible. Dentro de esta categoría destacan los filtros de Wiener.

- **Substracción espectral**

  Reducen el ruido mediante la sustracción del espectro estimado del ruido.

- **Basados en modelos**

  Los métodos de reducción basados en modelos están orientados mayormente a la eliminación de ruido en señales de voz. Estos modelos de voz contienen información de la señal de voz, es decir, componentes que pertenecen a la misma. Por tanto, su funcionamiento está basado en resaltar esos componentes de la señal frente a otros que no se consideran de voz. Entre ellos destaca el filtrado de Kalman.

Todos estos modelos actúan sobre la señal de audio, y han sido desarrollados para compensar el ruido principalmente sobre señales de voz, y hacer estas más reconocibles por otro interlocutor. El resultado del proceso es una señal de audio completa en la que la componente de ruido ha sido atenuada.

En principio sería posible aplicar una técnica de reducción de ruido basada en estos modelos al audio antes de extraer los coeficientes con los que trabaja nuestro sistema (MFCC). En nuestro caso, sin embargo, no se ha considerado esta posibilidad sino que únicamente se han estudiado técnicas de compensación del ruido que operan en el dominio de las características MFCC. En nuestro caso esto es válido ya que el sistema de búsqueda de audio en audio no genera una señal de audio, sino que sólo tiene que analizar dicho audio para determinar finalmente la sincronización entre dos muestras de audio. Por ello se considera innecesario recurrir a toda la complejidad requerida para regenerar una señal de audio y se tratará de reducir directamente el efecto del ruido sobre los parámetros con los que opera el sistema: los MFCC.

A continuación analizaremos los efectos del ruido sobre estos coeficientes. Este análisis nos dará pistas sobre cómo tratar de compensar dichos efectos. Para este análisis consideraremos dos tipos de ruidos: los convolutivos en los que la señal se convoluciona con la respuesta al impulso de un sistema lineal e invariante (ejemplo típico de este ruido es la distorsión debida al canal, por ejemplo el telefónico o la respuesta de un micrófono de un móvil), y los aditivos en los que el ruido se suma a la señal (ejemplo típico de este ruido es el ruido de fondo de una escena acústica).

2.4.1.2. Ruido convolutivo sobre los MFCC y compensación.

El ruido convolutivo es aquella degradación de la señal debida a la no-idealidad del canal por donde se transmite la señal. Nos referimos a respuestas en frecuencia no lineales de elementos del sistema como pueden ser los micrófonos que captan el audio. Esta distorsión no lineal afecta por tanto al espectro de la señal, y por consiguiente a los coeficientes espectrales que obtengamos de ella. Será objetivo principal de este proyecto tratar de compensar el efecto de este ruido, el cual, a diferencia del ruido aditivo es más fácil de modelar y eliminar, como veremos a continuación.

Se define por tanto la señal obtenida como combinación de la señal original *s(t)* a través del canal *h(t)* y el ruido aditivo *n(t):*

$$\mathrm{y(t)} = \mathrm{s(t)} * \mathrm{h(t)} + n(t)$$

Que en el dominio de la frecuencia se define como:

$$\mathrm{Y(t,\omega)} = \mathrm{H(w)S(t,\omega)} + N(t,\omega)$$

Si suponemos que la grabación se realiza en un entorno en silencio y tranquilo, podemos suponer que la componente de ruido aditivo *n(t)* es despreciable respecto a la señal *s(t).* Aplicando esta hipótesis sobre las fórmulas anteriores, podemos definir la señal recibida como la convolución en el dominio temporal de la señal limpia con la respuesta al impulso del canal *h(t).* Esta convolución se transforma en una multiplicación en el dominio espectral:

$$\mathrm{Y(t,\omega)} = \mathrm{H(w)S(t,\omega)}$$

Multiplicación que, a su vez, se transforma en una suma en el dominio log-espectral:

$$\log(|Y(t,\omega)|^2) = \log(|H(\omega)|^2) + \log(|S(t,\omega)|^2)$$

Si se aplica la transformada discreta del coseno (DCT) a la anterior ecuación, se obtiene la relación entre los parámetros cepstrales de la señal limpia, los de la voz contaminada y los del canal:

$$C_y(t,\tau) = C_x(t,\tau) + C_h(\tau)$$

O lo que es lo mismo:

$$MFCC_y = MFCC_x + MFCC_h\,, \quad donde\ MFCC_h = cte$$

Lo que significa que el efecto del ruido convolutivo sobre los coeficientes cepstrales es simplemente un desplazamiento en los parámetros de la señal limpia. Este desplazamiento es constante para cada componente cepstral, si asumimos que las características del canal no cambian con el tiempo, como parece razonable.

Una forma sencilla de eliminar el ruido convolutivo de los coeficientes MFCC consiste simplemente en calcular la media de cada coeficiente en un fichero y eliminar a continuación dicha media de cada coeficiente. Dado que la influencia del canal se refleja únicamente en un término constante, al eliminar la media se eliminará la influencia del canal en los MFCC. Esta forma sencilla de eliminar (en realidad reducir por las discrepancias entre los cálculos reales y la teoría) el ruido convolutivo se denomina Cepstral Mean Substraction (CMS) o Cepstral Mean Normalization (CMN) y ha venido empleándose con regularidad en el reconocimiento de voz, entre otros ámbitos.

2.4.1.3 Ruido aditivo sobre coeficientes MFCC y compensación.

2.4.1.3.1 Distorsión a la salida del banco de filtros

Si se define x_i y n_i como las muestras de señal de audio y del ruido aditivo respectivamente, $y_i = x_i + n_i$ representa la señal de audio contaminada por el ruido aditivo. La energía de cada frame se calcula por tanto como:

$$E_y = \sum_{i=1}^{I} {y_i}^2 = \sum_{i=1}^{I} ({x_i}^2 + {n_i}^2 + 2x_i n_i)$$

Asumiendo que el ruido y la señal tienen media cero y son estadísticamente independientes:

$$\sum_{i=1}^{I} x_i n_i \approx 0 \quad \rightarrow \quad E_y = E_x + E_n$$

Lo cual se cumple para cualquier parámetro de energía de la señal. Definiendo $X_b(t)\ y\ N_b(t)$como la energía de la señal y del ruido a la salida del filtro b, se tiene:

$$Y_b(t) = X_b(t) + N_b(t)$$

Y en escala logarítmica $(x_b(t) = \log(X_b(t)))$:

$$y_b(t) = \log\left[e^{x_b(t)} + e^{n_b(t)}\right]$$

Esta expresión define como afecta el ruido aditivo a la salida de los bancos de filtros en el proceso de extracción de coeficientes MFCC.
Estos efectos se pueden observar en la figura siguiente, y se caracterizan por:

- El ruido aditivo produce una distorsión no lineal en escala logarítmica (a la salida del banco de filtros).
- En las regiones en las que el ruido aditivo tiene una energía mayor, la señal queda totalmente enmascarada por el ruido
- Como la obtención de coeficientes MFCC se lleva a cabo mediante una transformación lineal de la energía (normalmente mediante DCT (Discrete Cosine Transform)), los dos efectos anteriores, también estarán presentes en dichos coeficientes.

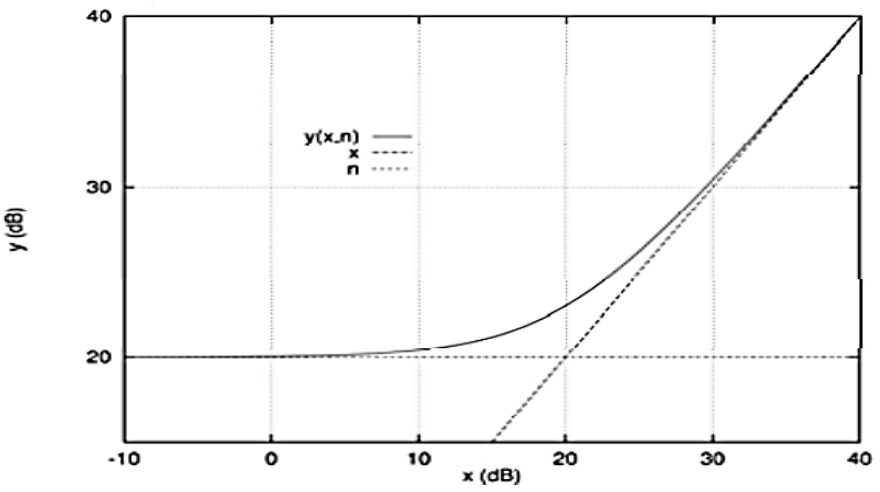


Figura 2.15: Distorsión de energía con un nivel de ruido constante (20dB).

2.4.1.3.2 Distorsión de la función densidad de probabilidad

Veremos ahora un ejemplo gráfico de cómo afecta el ruido aditivo a una fdp gaussiana, para obtener una estimación de cómo afectaría a la fdp de la señal de audio. Para ello se representa una función gaussiana de media 15dB y desviación estándar 2dB con distintos niveles de ruido, desde 0 a 15 dB:

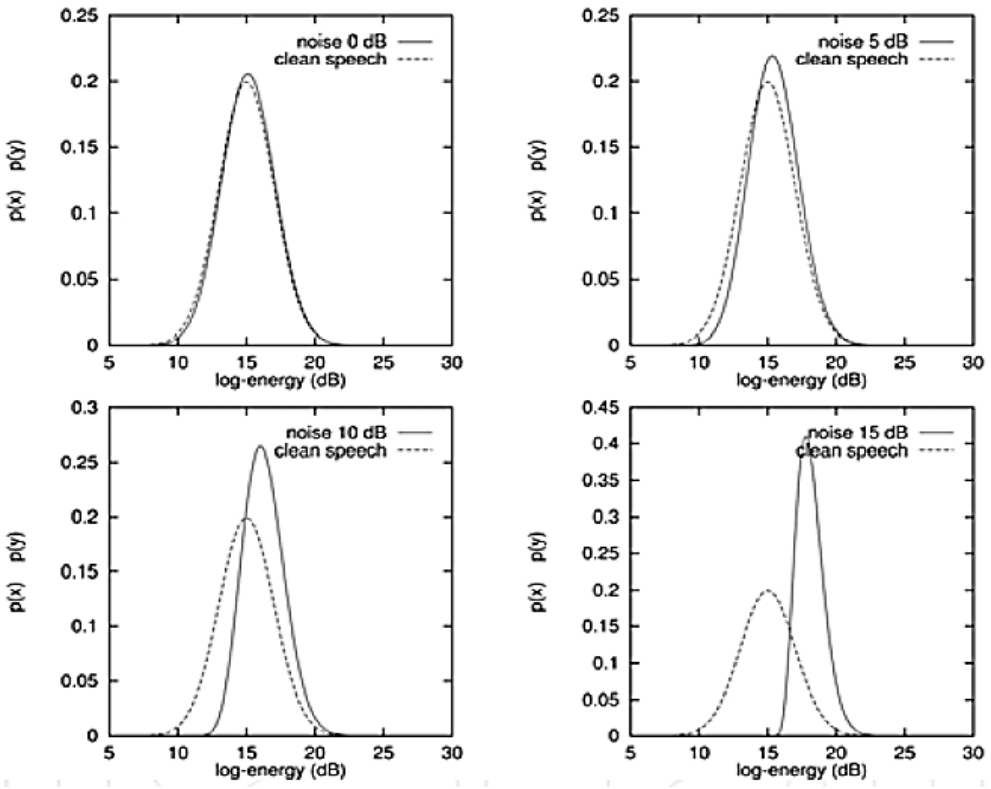


Figura 2.16: Distorsión de fdp Gaussiana con ruido aditivo.

Se observan los siguientes efectos del ruido sobre funciones densidad de probabilidad:

- Desplazamiento en la media
- La desviación estándar se reduce
- Debido a la no-linealidad del ruido, la fdp también queda distorsionada, no siendo una distribución Gaussiana.

La función densidad de probabilidad de la señal de audio tenderá a tener media cero, pero al verse afectada por el ruido aditivo, esta media se desplazará, así como su varianza que se verá reducida. Es por eso que la técnica conocida como Cepstral Mean

Normalization (CMN) se puede modificar para que normalice no sólo la media sino también la varianza, para de ese modo ser capaz de compensar en parte los efectos del ruido, no sólo convolutivo sino también aditivo. Esta técnica se denomina normalización de media y varianza sobre coeficientes espectrales (Cepstral Mean and Variance Normalization, CMVN).

2.4.1.4 CMVN

La técnica CMVN (Cepstral Mean and Variance Normalization) tiene como objetivo abordar el problema de la distorsión convolutiva y el rudio aditivo. Como se puso de manifiesto en apartados anteriores, el efecto del ruido convolutivo consiste en un desplazamiento constante de los parámetros de la señal limpia en el dominio cepstral. La idea será por tanto eliminar ese desplazamiento introducido por la no linealidad del canal de transmisión, que puede ser considerado como una componente continua. Para eliminar dicha componente, se puede substraer la media del vector de coeficientes cepstrum o MFCC. Esta técnica es conocida como CMN, que es la predecesora de la técnica CMVN. La técnica CMVN además tiene en cuenta las varianzas, y su propósito es eliminar el desplazamiento introducido por el ruido convolutivo a la vez que se consigue un valor unitario de varianza del vector de coeficientes MFCCs para el segmento de interés, lo que consigue compensar también parte del efecto del ruido aditivo.

Existen dos variantes del método, en función de la forma de calcular los estadísticos para la normalización (media y varianza). Cada una de las cuales es más conveniente para un tipo de sistemas u otros, dependiendo del tiempo de respuesta necesario.

2.4.1.4.1 Normalización segmentada

La normalización segmentada se caracteriza por ser un proceso no causal, es decir, la salida del sistema dependerá de valores futuros del vector de coeficientes. Se calcula la media y la varianza a través de una ventana deslizante centrada sobre el frame a normalizar.

Una vez obtenidos estos estadísticos, se realiza la normalización mediante la resta de la media y la división por la varianza, quedándonos por tanto media cero y varianza unitaria a lo largo de todo el vector de coeficientes MFCCs.

$$Media: \qquad \mu_t(k) = \frac{1}{N} \sum_{n=t-\frac{N}{2}}^{n=t+\frac{N}{2}-1} x_n(k)$$

$$Varianza: \qquad \sigma_t^2(k) = \frac{1}{N} \sum_{n=t-\frac{N}{2}}^{n=t+\frac{N}{2}-1} (x_n(k) - \mu_t(k))^2$$

Donde N es el tamaño de la ventana deslizante y k representa el coeficiente k-ésimo. Finalmente, el vector de coeficientes MFCC quedará:

$$\boldsymbol{Normalización}: \qquad \boldsymbol{x_t^{norm}(k) = \frac{x_t(k) - \mu_t(k)}{\sigma_t(k)}}$$

De este modo el coeficiente que queda normalizado es el del centro de la ventana deslizante. Las ventajas de este tipo de normalización son que los parámetros estadísticos son independientes del entorno acústico que se dé, y también proporciona una rápida adaptación a cambios en las condiciones ruidosas.

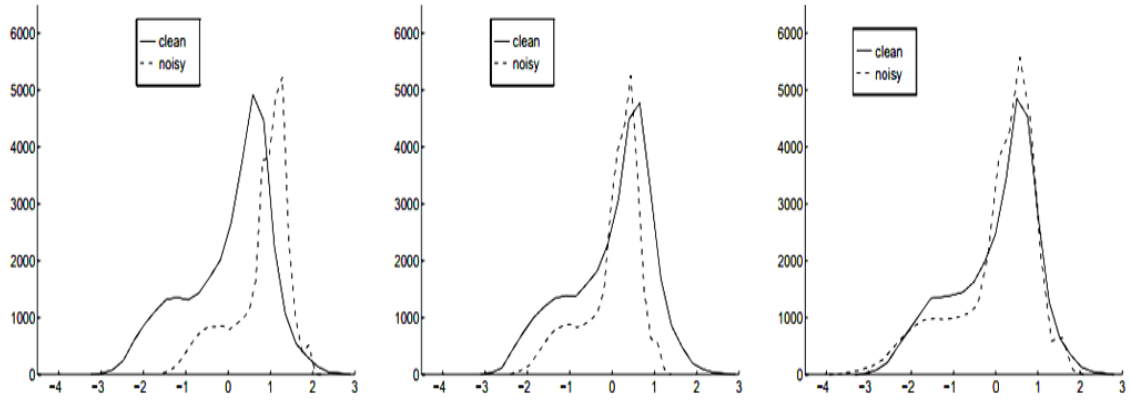


Figura 2.17: Efectos de CMVN sobre coeficientes espectrales. *[17]*

El principal inconveniente de este tipo de normalización es el retardo que introduce al procesamiento debido al tamaño de la ventana que se utiliza. A efectos prácticos, una ventana de aproximadamente 1 segundo de longitud es suficiente para garantizar una normalización robusta de coeficientes. Tanto el uso de memoria como el tiempo de procesado dependen directamente del valor del tamaño de la ventana. Es por ello para aplicaciones de respuesta semi-inmediata como reconocimiento de locutor, este factor de tiempo es fundamental, y debe ser minimizado al máximo desde el punto de vista de la implementación.

2.4.1.4.2 Normalización recursiva

Para solucionar este problema, este tipo de aplicaciones que requieren una respuesta rápida utilizan una técnica diferente. Esta técnica consiste en calcular los coeficientes normalizados de forma recursiva, de modo que el coeficiente actual no depende del valor de los coeficientes futuros, sino pasados, y por tanto el retardo de procesamiento virtualmente no existe. Los coeficientes media y desviación estándar se calculan recursivamente del siguiente modo:

$Media$: $\mu_k(i) = \alpha\mu_k(i-1) + (1-\alpha)x_k(i)$

$Desviación\ estandar$: $s_k(i) = \alpha s_k(i-1) + (1-\alpha)x_k^2(i)$

$Varianza$: $\sigma_k^2(i) = s_k(i) - \mu_k^2(i)$

$\boldsymbol{Normalizado}$: $\boldsymbol{x_k^{norm}(i) = \frac{x_k(i) - \mu_k(i)}{\sigma_k(i)}}$

Donde α es el denominado factor de olvido, que se usa para reducir paulatinamente el efecto de los frames anteriores. Distintos experimentos han demostrado que un valor de *α = 0.995* proporciona una estimaciones estables [3]. Para que la normalización recursiva sea efectiva, debe haberse hecho una normalización previa para calcular los valores de los coeficientes $\mu_k(i-1)\ y\ s_k(i-1)$ de modo que se puedan calcular los siguientes coeficientes a partir de ellos.

Combinando ambas normalizaciones se puede lograr un tamaño de ventana significativamente menor sin reducir la calidad del reconocimiento posterior. Para ello, se debe realizar la normalización previa de los N primeros frames, calculando la media y la varianza como sigue:

$Media$: $\mu_t(i) = \frac{1}{N}\sum_{t=1}^{N} x_t(i)$

$Varianza$: $\sigma_t^2(i) = \frac{1}{N}\sum_{t=1}^{N}[x_t^2(i)] - [\mu_t(i)]^2$

$\boldsymbol{Normalizado}$: $\boldsymbol{x_{t=N}^{norm}(i) = \frac{x_{t-N}(i) - \mu_t(i)}{\sigma_t(i)}}$

De este modo se establece la normalización para el primer frame con un retardo de N frames, y todos los siguientes frames son calculados por medio de la normalización recursiva. Esto quiere decir que todos los frames son calculados con un retardo de N frames, pero con un coste en memoria tremendamente inferior. Se estima entre un 50 % y un 80%la reducción del tamaño de ventana para realizar la normalización comparada con la normalización segmentada, con los mismos valores de rendimiento en reconocimiento.

Como se comentó anteriormente, la normalización recursiva es especialmente útil para sistemas de reconocimiento rápido como los VAD (Voice Activity Detector), puesto que permite una normalización mucho más rápida para ofrecer una respuesta en tiempo real. Sin embargo, el sistema de búsqueda de audio en audio, aun siendo un sistema en tiempo real, no requiere una respuesta tan inmediata. Concretamente, el sistema con el que se trabaja en este proyecto requiere de al menos 8 segundos de muestra para ser procesada. Por tanto la incidencia del tiempo extra de procesamiento debido al uso de normalización segmentada es insignificante. Sin embargo se tiene en cuenta la normalización previa del modo recursivo para adaptarlo al principio y al final del vector en el modo segmentado. Es decir, los primeros y últimos frames del vector de coeficientes se normaliza respecto a los N/2 primeros y últimos y el resto se hace con el método de ventana deslizante de N frames.

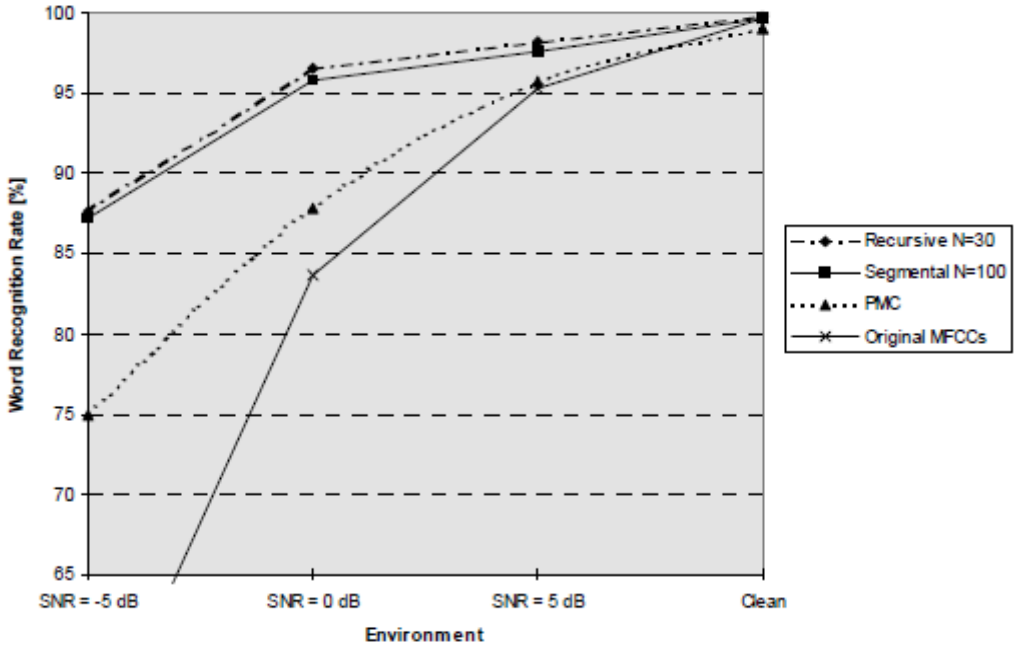


Figura 2.18: Rendimiento de varios métodos de robustez frente al ruido. *[18]*

La gráfica muestra el rendimiento de un sistema de reconocimiento de voz en función de la técnica de normalización y robustez frente al ruido utilizada. Una gráfica

similar a esta será introducida tras el análisis del funcionamiento del sistema de búsqueda de audio en audio con el uso de la normalización de media y varianza. En esta se confronta el rendimiento entre CMVN recursiva y CMVN segmentada así como el rendimiento sin normalización para diferentes niveles de SNR (Signal to Noise Ratio). Se observa como la normalización recursiva tiene un rendimiento muy similar a la segmentada, pero utilizando un tamaño de ventana mucho menor. Es decir, para lograr los mismos resultados en términos de rendimiento, utilizando normalización segmentada tenemos que utilizar una ventana mayor (100 frente a 30), que conlleva un aumento del tiempo de procesamiento, así como un coste en memoria superior. La ganancia respecto a la no normalización es más que evidente, y se acentúa más para niveles de ruido superiores (SNR menor).

2.4.1.4.3 Efectos CMVN sobre coeficientes MFCCs.

A continuación se muestra un ejemplo de cómo un vector de coeficientes MFCC es afectado por el ruido, así como de los efectos que tiene la normalización sobre él:

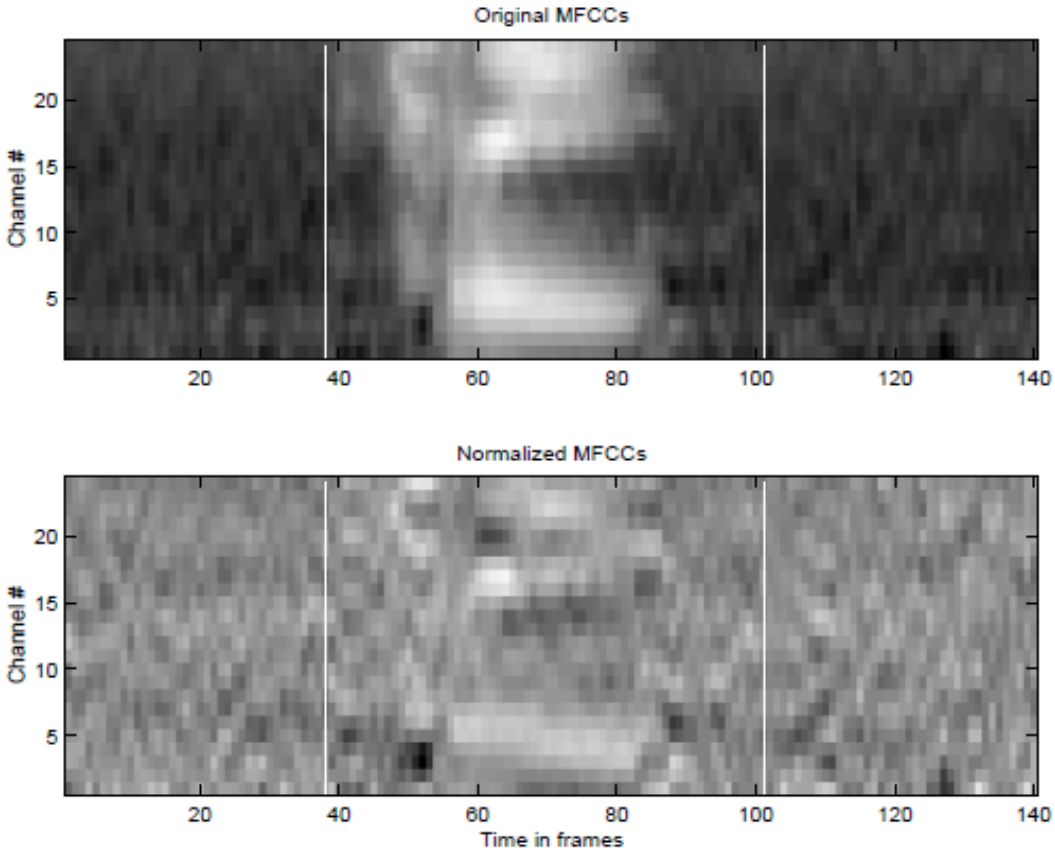


Figura 2.19: Coeficientes MFCCs originales y con normalización en un entorno sin ruido. *[18]*

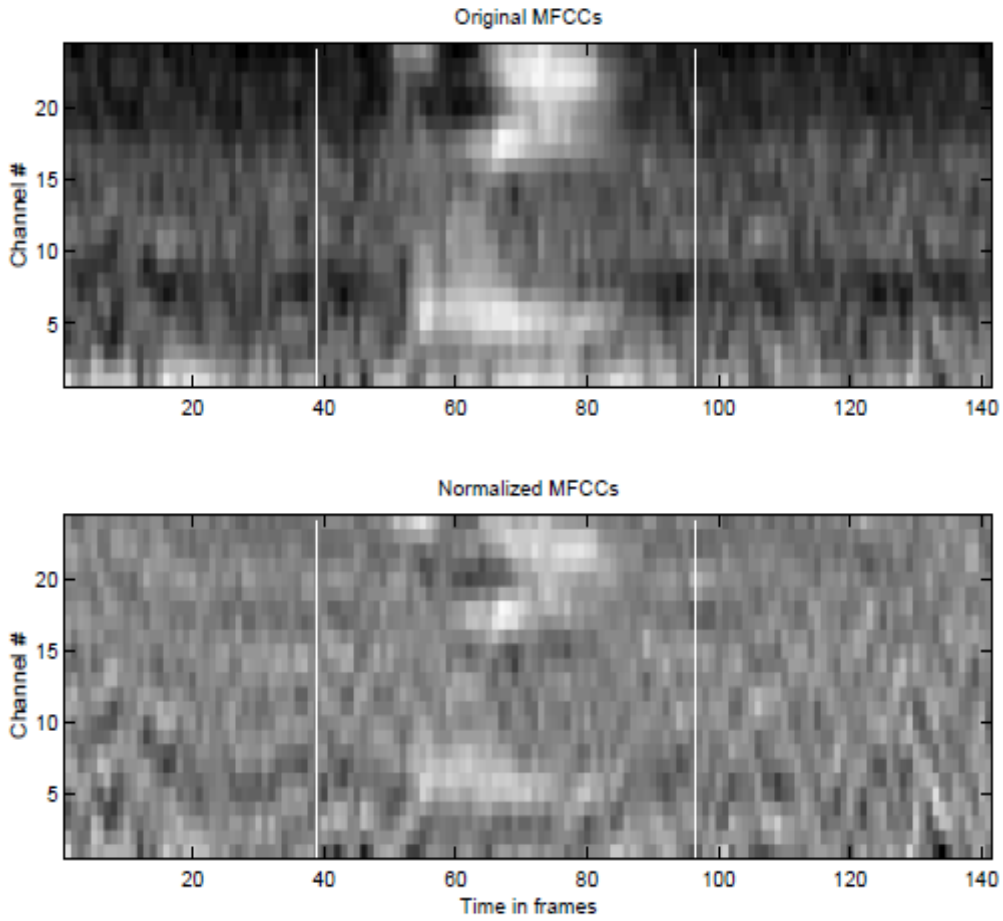


Figura 2.20: Coeficientes MFCCs originales y con normalización en un entorno ruidoso. *[18]*

Como se puede observar comparando el primer gráfico de ambas figuras, la existencia de ruido enmascara la señal original, observándose una representación de coeficientes MFCCs muy diferente entre ambas. Sin embargo, si nos fijamos en las figuras inferiores, que representan las anteriores pero normalizadas, es fácil darse cuenta de que las diferencias entre ambas son mucho menos significativas. Esto significa que el ruido afecta mucho menos a los coeficientes normalizados.

Centrando la atención en la primera figura, es fácil darse cuenta que la normalización de un vector de coeficientes limpios hace que los espectrogramas parezcan mucho más ruidosos, y por tanto parece más difícil de obtener características diferenciadoras en ellos. Sin embargo muchas pruebas de reconocimiento indican que este hecho no afecta al rendimiento de los sistemas que utilizan esta técnica.

2.4.1.5 CSN

CSN (Cepstral Shape Normalization) es una técnica de normalización cuyo objetivo es el aumento de la robustez frente al ruido en sistemas de reconocimiento de audio. El método está basado en el cambio que sufre de la forma de las distribuciones de vectores de coeficientes característicos de una señal en presencia de ruido. El método CSN normaliza la forma de dichas distribuciones mediante el uso de exponenciales.

Este método ha sido utilizado con gran éxito en aplicaciones de reconocimiento de voz en condiciones ruidosas y con baja SNR. Se aplica a partir del método CMVN y proporciona una mejora extra en el rendimiento. El principio fundamental en que se basa el sistema CSN es que las funciones distribución de los coeficientes característicos para cada dimensión, pueden ser aproximadas a una función distribución Gaussiana normalizada. Se trata por tanto de encontrar un parámetro que de forma a la distribución para poder aplicarlo con objeto de minimizar el efecto del ruido.

De manera análoga a como la técnica CMVN vino a complementar a la técnica CMN, incluyendo una normalización de la varianza, CSN viene a complementar a CMVN incluyendo una normalización en la forma de las distribuciones de vectores de coeficientes. Se trata por tanto de una técnica más avanzada y moderna.

Capítulo 3:

Diseño y desarrollo

3.1 El programa de partida

3.1.1 Esquema general

El objetivo del programa es la sincronización de 2 pistas de audio, una de ellas audio fuente original, y la otra una muestra grabada con un micrófono de un dispositivo móvil. Se desea obtener una precisión bastante elevada, del orden de decenas de milisegundos.

El tratamiento de ambas pistas es muy similar, sin embargo, las huellas del audio original se procesan offline, mientras que para la muestra grabada se realiza online. Dicho procesamiento consiste en adecuar el audio a un formato manejable, extraer promedios de parámetros característicos y por último se compara una con la otra y se mide el grado de similitud entre ellas. La respuesta del programa es un valor de tiempo del audio original, que indica el momento, medido desde el inicio del audio original, en el cual ambas muestras se parecen más.

El esquema básico del programa se muestra a continuación:

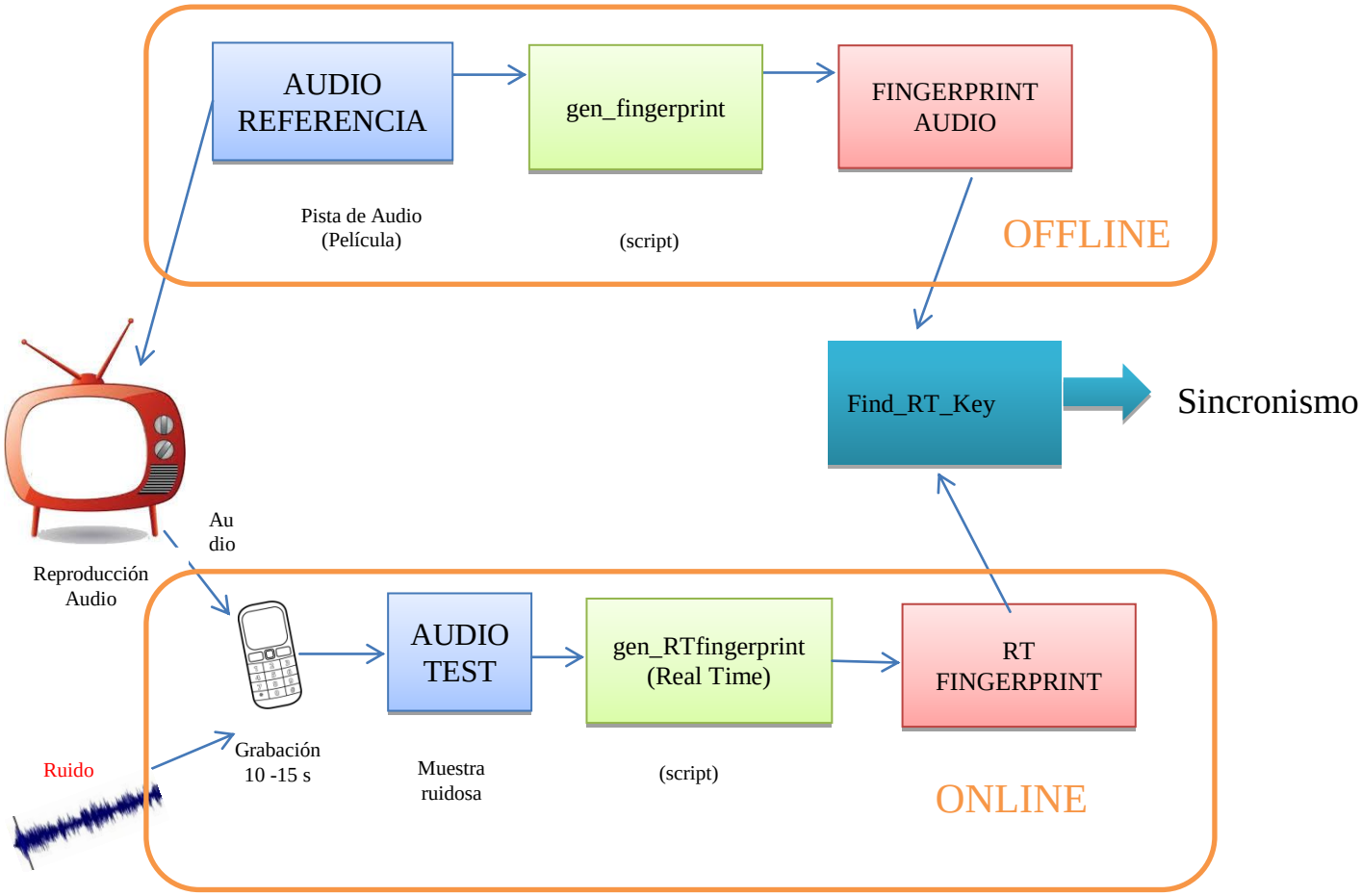


Figura 3.1: Esquema básico del sistema.

3.1.2 Componentes del programa

El programa está compuesto por una serie de scripts y algoritmos que se detallan a continuación:

1. ***config.h:*** Fichero con los datos de configuración (número de MFCC, separación entre frames, número de frames para calcular las medias etc.).

2. ***gen_mfcc:*** Genera un fichero con coeficientes MFCCs a partir de un fichero de audio. Este fichero de audio tiene que ser en formato WAV 16KHz mono. A la salida entrega un fichero de coeficientes MFCC (en este caso 13 por frame, con frames cada 16ms).

3. ***gen_fingerprint.sh:*** Script que adecúa la entrada de audio (en cualquier formato a formato WAV 16KHz mono, mediante SOX. Posteriormente llama a *gen_mfcc* con el audio adaptado y finalmente llama al ejecutable *gen_key_from_mfcc,* al que le pasa como parámetro el fichero de MFCCs generado en el paso anterior.

4. ***gen_RTfingerprint.sh:*** Script que adecúa la entrada de audio (para la muestra grabada), y recorta la muestra desde un instante que se le pasa como argumento hasta un tiempo igual al definido en *config.h* (en este caso 8 segundos). Posteriormente llama a *gen_mfcc* con el audio adaptado y finalmente llama al ejecutable *gen_RTkey_from_mfcc* al que le pasa como parámetro el fichero de MFCCs generado en el paso anterior.

5. ***gen_key_from_mfcc:*** Ejecutable en C que crea los ficheros *.key* para el audio original. Promedia los coeficientes MFCCs que recibe a la entrada en intervalos definidos en *config.h* (en este caso ventana de 0.8s y un promedio cada 0.4s). Devuelve el fichero *.key* correspondiente.

6. ***gen_RTkey_from_mfcc:*** Ejecutable en C que crea los ficheros *.key* para la muestra grabada. Promedia los coeficientes MFCCs que recibe a la entrada en intervalos definidos en *config.h* (en este caso ventanas de 0.8s y un promedio cada 0.016s). Devuelve el fichero *.key* correspondiente.

7. **find_RTkey:** Ejecutable en C que calcula el desfase entre las dos pistas de audio basándose en las distancias mínimas entre vectores de coeficientes MFCCs. Puede ser configurado de modo que se use como cómputo de medida, la distancia Eucídea o la distancia City Block. Devuelve un instante de tiempo y una distancia de detección.

3.1.3 Generadores de Fingerprints

Las huellas o *fingerprints* son ficheros de datos obtenidos a partir de los vectores de coeficientes MFCCs. Se realizan promedios de estos vectores según una ventana de tiempo que varía en función del tipo de *fingerprint* que se quiere obtener. A continuación se muestra un esquema de cómo funcionan los generadores de *fingerprints* del sistema, tanto para el auido original (*gen_fingerprint*), como para la muestra grabada (*gen_Rtfingerprint*):

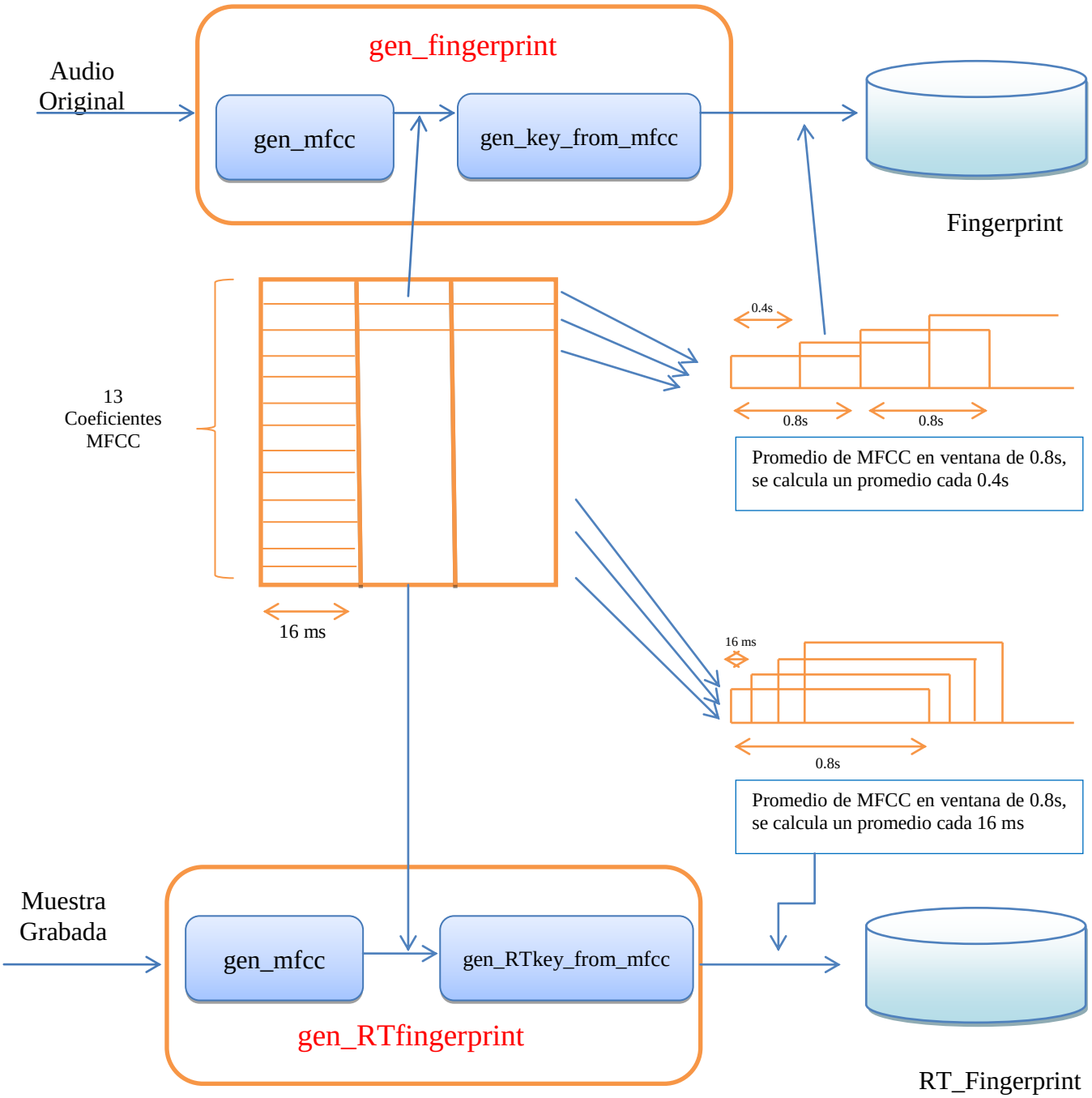


Figura 3.2: Generadores de fingerprints.

El tamaño de las pistas originales es mucho mayor que el de las muestras grabadas, por lo tanto, los promedios de coeficientes se calculan en intervalos de tiempo mayores, con el fin de no aumentar en exceso el tamaño de los *fingerprints*. En este caso se utiliza un intervalo de 0.4 segundos. Por otro lado el intervalo utilizado para la muestra grabada será el que mida la precisión del sistema. Como las muestras son de tamaño muy inferior se fija en un valor de 0.016 segundos, que aporta una precisión bastante razonable.

3.1.3.1 Tasa de compresión

Una de las premisas de los sistemas de búsqueda de audio en audio es que sean capaces de dar una respuesta rápida, puesto que se trata de sistemas en tiempo real por lo general. Para que esto sea posible, el tiempo de procesamiento debe ser reducido, cosa que no sería posible conseguir sin una tasa de compresión muy elevada a la hora de generar las huellas o *fingerprints*.

A continuación se mostrará un ejemplo de la magnitud de esta compresión para las muestras con las que se ha trabajado en este proyecto. El audio original que se ha analizado se trata de una pista de una longitud de dos horas aprox. (7200s) en formato mp3 a 192 Kbit/s. Las muestras contaminadas tienen una longitud de 8 segundos:

- Tamaño de la pista de audio original

$$Size = 7200s * 192\frac{Kbit}{s} * \frac{1\ Kbyte}{8.192\ Kbits} = \mathbf{168750\ Kbytes}$$

- Tamaño del fingeprint del audio original

$$Fingerprint\ Size = 7200s * \frac{promedio}{0.4\ s} * \frac{13\ Mfcc}{promedio} * \frac{1\ Float}{Mfcc} * \frac{4\ bytes}{Float}$$
$$= \mathbf{936\ Kbytes}$$

- Tamaño del fingeprint de la muestra grabada

$$RTFingerprint\ Size = 8s * \frac{promedio}{0.016\ s} * \frac{13\ Mfcc}{promedio} * \frac{1\ Float}{Mfcc} * \frac{4\ bytes}{Float}$$
$$= \mathbf{26\ Kbytes}$$

- **Tasa de compresión (para el fingerprint de audio original)**

$$TC = \frac{168750\ Kbytes}{936\ Kbytes} \approx \mathbf{180}$$

3.1.4 Algoritmo de búsqueda

El algoritmo de búsqueda que emplea el programa primario es un algoritmo lineal, es decir, se calculan todas las distancias entre cada pareja de vectores. Como los promedios de la muestra grabada son de 0.016s y los de la muestra original 0.4s, cada promedio de la muestra original será comparado con 25 promedios de la grabación, lo que hará que la precisión aumente hasta los 16ms.

Es necesario aclarar aquí que en esta fase del desarrollo se buscaba una evaluación de la precisión del algoritmo, no de su velocidad. En una implementación realista un algoritmo de búsqueda lineal no sería en ningún caso una solución satisfactoria por sus malas características en cuanto a escalabilidad.

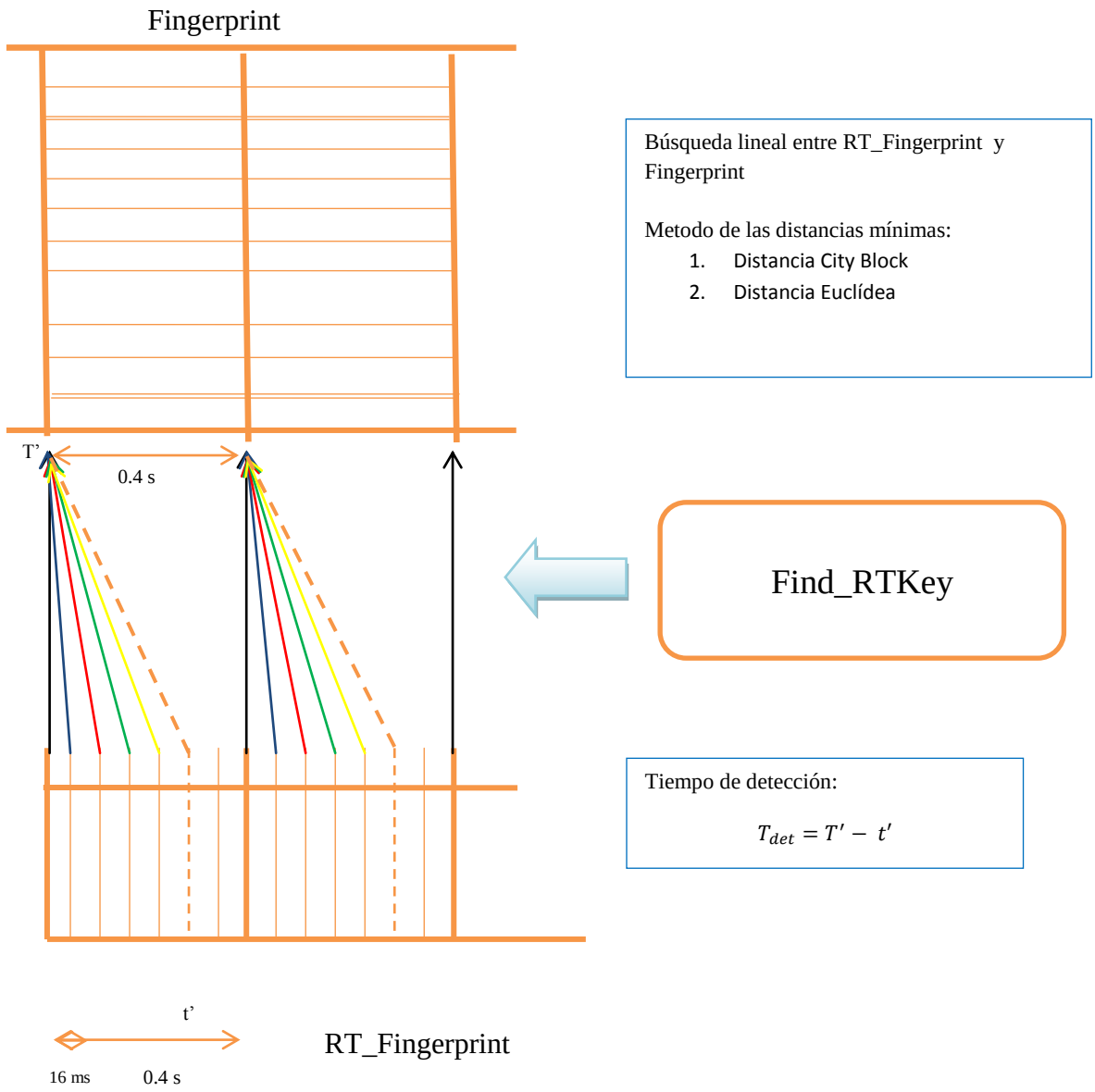


Figura 3.3: Algoritmo de búsqueda.

3.2 Datos de prueba

El desarrollo de este proyecto se ha basado en el análisis de un audio de referencia perteneciente a la pista de audio de una película, cuya duración aproximada es de dos horas. Esta contiene una diversidad de momentos acústicos, alterna partes musicales que se repiten, partes de diálogos, silencios etc., que hace que su análisis sea bastante complejo y por tanto significativo respecto al rendimiento que puede tener con cualquier tipo de pista de audio.

Se han tomado una serie de muestras aleatorias mediante un proceso de grabación con varios teléfonos móviles en diferentes situaciones del audio original reproducido en una sala de cine real, aunque sin público presente. Los dos dispositivos utilizados son un iPhone y un HTC, y las diferentes situaciones se describen a continuación, en función de la localización del aparato en el momento de la grabación:

- **Regazo**

  El móvil apoyado sobre el regazo, sin bloquear el micrófono con ningún elemento físico. Se trataría de un uso habitual para este tipo de aplicaciones. El nivel de ruido para este caso, en las muestras que hemos empleado en este proyecto, es mínimo.

- **Bolsillo**

  En este caso la grabación se toma desde el móvil introducido dentro del bolsillo, y por tanto con ropa o incluso el cuerpo bloqueando el micrófono parcialmente. El nivel de audio recogido es menor y se captura adicionalmente el ruido propio al introducir el teléfono en el bolsillo. Esta muestra se encuentra en segundo lugar en cuanto a nivel de ruido se refiere en nuestros experimentos.

- **Bolso**

  La última y peor de las grabaciones es aquella en la que el teléfono es introducido dentro de un bolso para grabar. En este caso el nivel de audio que se recoge es mínimo. Además, el ruido generado por el hecho de meter el móvil en el bolso, cerrar la cremallera, etc. es muy elevado, llegando en ocasiones a enmascarar totalmente el audio objeto de grabación.

Se cuenta con una muestra de cada una de estas situaciones para cada uno de los dos teléfonos empleados.

3.3 Diseño de pruebas

Tanto las pruebas como las mejoras realizadas en este proyecto siguen una evolución ascendente. Se parte de un programa inicial sin ningún tipo de algoritmo de robustez frente al ruido, al que se van incorporando mejoras sobre las anteriores, y así sucesivamente. Tras cada mejora, se realizan las pruebas oportunas para comprobar que dicha mejora es realmente efectiva antes de continuar el desarrollo. Este proceso se puede dividir en tres fases principales: primero un análisis inicial de las prestaciones del sistema, una segunda en la que se integran mecanismos de normalización ya conocidos, y una tercera de innovación, en la cual se desarrolla un algoritmo propio, estudiando el comportamiento del sistema.

En este apartado describiremos con más detalle estos procesos, dejando para el Capítulo 4 la presentación de los resultados obtenidos con cada uno de ellos.

3.3.1 Evaluación inicial

Se trata del primer paso realizado en este proyecto. Consiste en poner a prueba el sistema de partida con las muestras grabadas. Estas muestras tienen una longitud aproximada de un minuto, pero para obtener unos resultados que sean más objetivos, se descompondrán en muchas muestras recortándolas mediante una ventana deslizante con un desplazamiento de 0.016s. De este modo, de una grabación de 1 minuto, obtendremos aproximadamente 3200 muestras de 8 segundos. Teniendo en cuenta que tenemos 6 grabaciones distintas, el número total de búsquedas que se realizan se acerca a las 20.000, lo cual conforma un banco de pruebas suficientemente grande en cuanto a número. Este banco de pruebas será utilizado en todos los caso en los que se realice cualquier mejora.

3.3.2 Primera mejora: CMVN sobre coeficientes MFCC

Como se comentó anteriormente, la técnica de *Cepstral Mean and Variance Normalization*, consiste en una transformación lineal de coeficientes para conseguir unos estadísticos con media cero y varianza unitaria. Esto minimiza el efecto del ruido estacionario y del ruido gaussiano, que tiende a desplazar las medias y reducir las varianzas.

Llevado al desarrollo software, esto se traduce en la incorporación de un módulo especial que se sitúa entre el generador de coeficientes MFCCs y el generador del *fingerprint*. Tomará por entrada secuencias de coeficientes MFCCs y a la salida se tendrán estas mismas secuencias pero normalizadas respecto a su media y varianza:

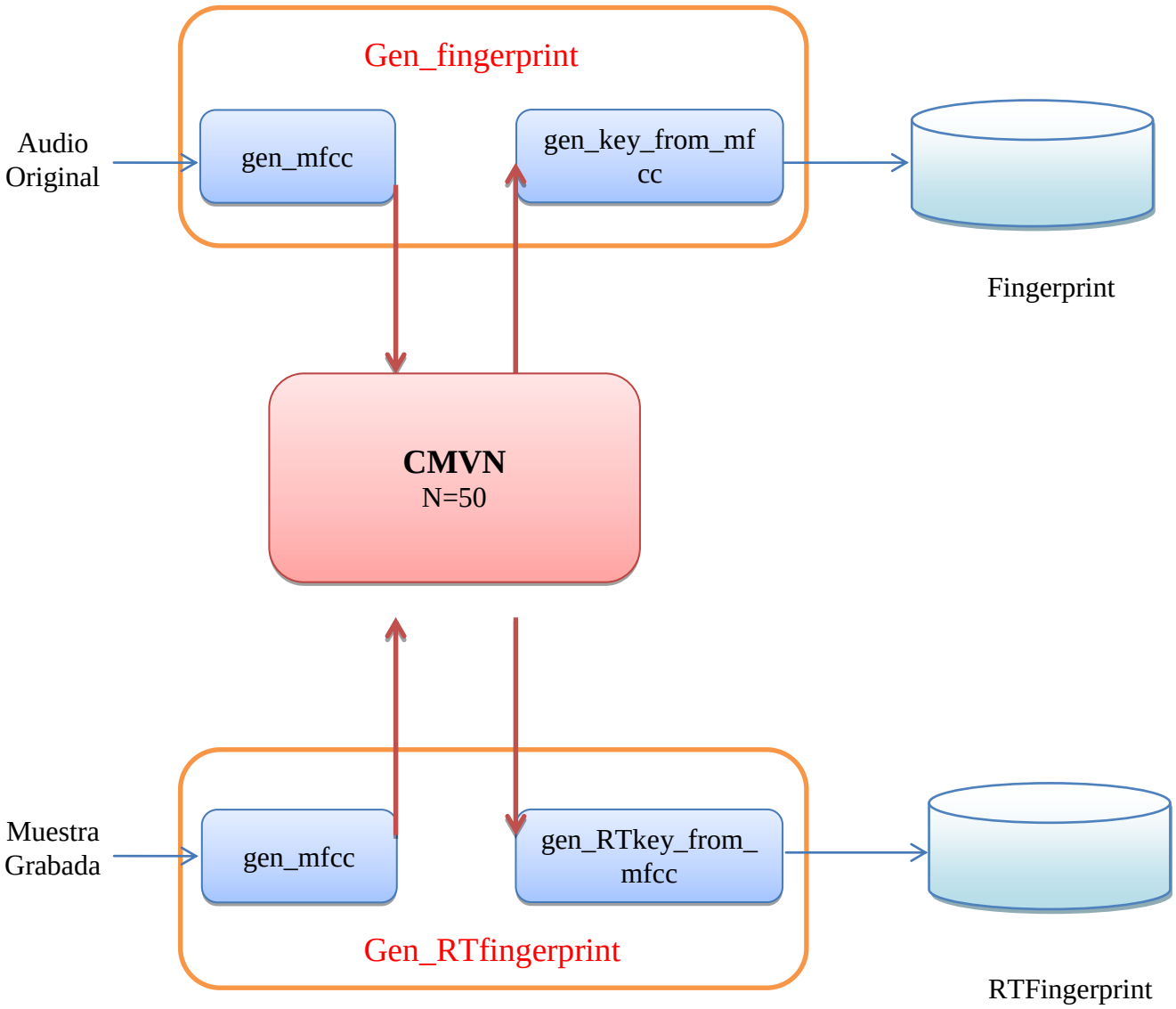


Figura 3.4: CMNV sobre coeficientes MFCC.

La normalización de media y varianza se lleva a cabo sobre una ventana deslizante de 0.8 segundos, que equivale a 50 frames. (muy próximo a la recomendación de [2] , que es 1 segundo) En este punto no se han calculado todavía los promedios por lo que es válido tanto para el *.key* como para el *.RTkey*.

3.3.3 Segunda mejora: CMVN sobre ficheros *.key*

Aparte de la normalización sobre coeficientes MFCC, en este proyecto, se propone la normalización de los ficheros *.key*, que contienen los promedios de los coeficientes MFCC normalizados. Es decir, se lleva a cabo una normalización doble para compensar en mayor medida el efecto del ruido. Como se verá más adelante, esta doble normalización ofrece mejores resultados que la normalización simple.

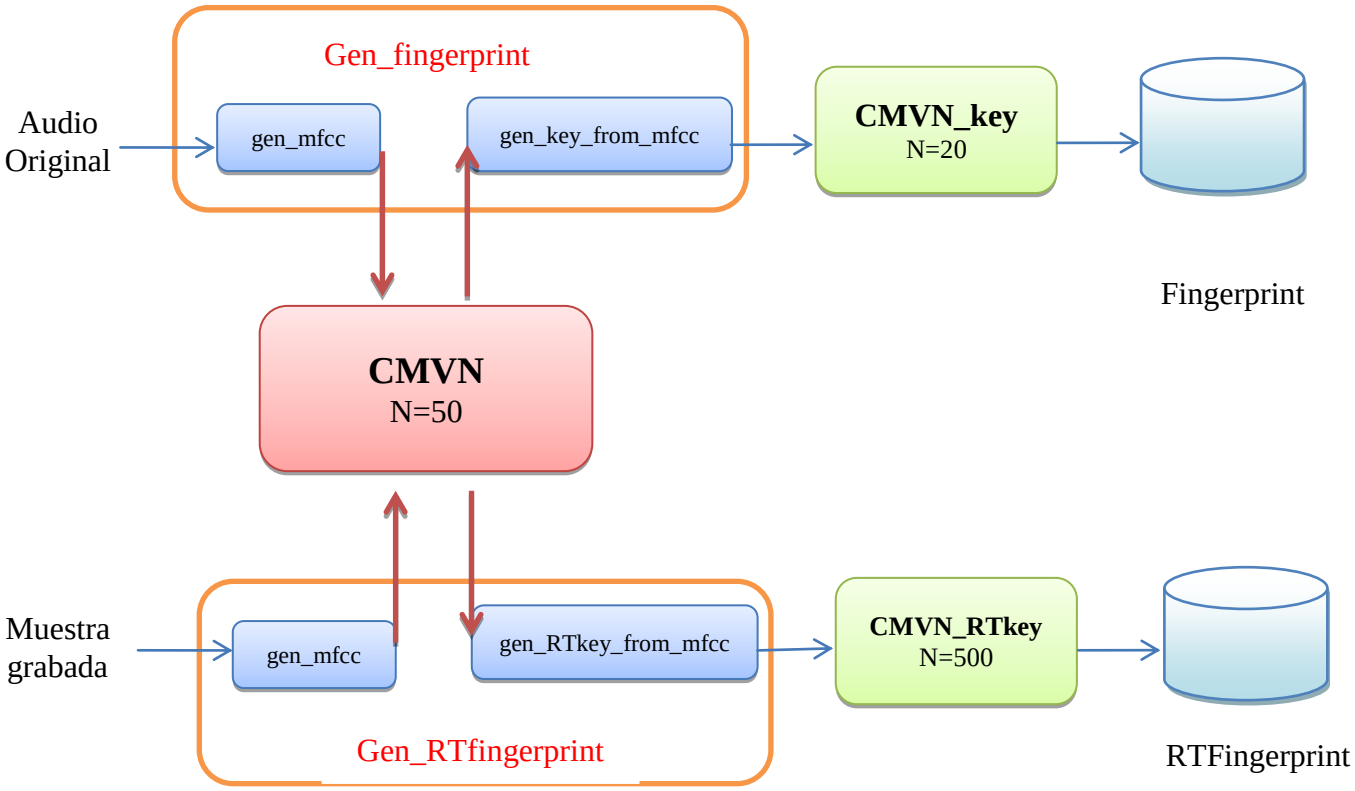


Figura 3.5: CMNV doble.

En este caso la ventana elegida es mayor, y equivale al tamaño completo de la muestra de audio grabada, es decir 8 segundos. Como el *gen_key_from_mfcc*, crea promedios cada 0.4 segundos, la ventana para el audio original es de 20 frames (8/0.4). Por otro lado el *gen_RTkey_from_mfcc* crea promedios cada 0.016 segundos, y por tanto, su ventana equivale a 500 frames (8/0.016).

3.3.4 Análisis de trayectorias temporales

Como se explicó en el funcionamiento del programa, la toma de decisiones respecto a las detecciones, se basa en la distancia mínima entre los MFCC's agrupados en el *RTfingerprint*, con los MFCCs agrupados en el *fingerprint* de la señal original. Estas comparaciones se realizan con toda la huella del audio original, por lo que se obtiene muchas distancias.

Hasta ahora, nos limitábamos a usar la mínima de ellas, que nos proporcionaba un valor de detección, pero a partir de ahora, tendremos en cuenta las *M-Best* distancias, es decir las M distancias más pequeñas detectadas, que nos proporcionarán las M detecciones más probables para cada muestra de 8 segundos que se analiza. El objetivo de esto, es averiguar si en una detección errónea, la opción correcta se encuentra dentro de las M más probables y si es así, tratar de crear un algoritmo basado en el análisis de una trayectoria de detecciones que obtenga como resultado, que esa detección es la que más se adecua, a pesar de no ser la que menor distancia mínima tiene a priori.

A partir de las *M-Best* mejores detecciones, el algoritmo de análisis de trayectorias hará lo siguiente:

A. Se define una ventana deslizante de N frames, en función de la longitud de la trayectoria que se quiere estudiar.

B. Se calculan para los *M-best,* todas las trayectorias posibles dentro de la ventana, esto es, aquellas que tienen una evolución temporal lineal durante toda la ventana.

C. Se calcula la distancia mínima de toda la trayectoria, en función de la distancia mínima de todas las detecciones de dicha trayectoria.

D. El valor de retorno será la primera detección de la trayectoria cuya distancia mínima total sea la menor.

De esta forma se pueden obtener interesantes resultados, puesto que podemos terminar ofreciendo la detección correcta en un frame que habíamos detectado de forma incorrecta, así como ser capaces de encontrar que una detección era en realidad errónea, puesto que si no se consigue ninguna trayectoria posible, lo más probable es que esa detección sea en un punto muy ruidoso y por tanto con alto riesgo de producir un error.

Veremos ahora un ejemplo del funcionamiento de este algoritmo en el modo en que ha sido empleado en este proyecto. Se analiza una ventana de 10 detecciones y se

tienen en cuenta las 10 detecciones más probables. Posteriormente las detecciones que no siguen una evolución temporal lineal son eliminadas. Es decir aquéllas para las que en el conjunto de 10-Best ninguna detección cumple:

$$Detección_{(i,t+0.016)} = Detección_{(i,t)} + 0.016\ s$$

Detecciones					
4649,984	79,568	79,616	79,68	79,6	79,616
79,616	79,584	79,568	79,632	79,616	79,664
79,568	79,648	79,6	79,648	79,648	79,6
79,6	79,552	79,536	79,6	79,696	79,712
79,52	79,632	79,52	79,552	79,568	79,632
79,552	79,536	79,584	79,536	79,664	79,568
79,536	~~4650~~	79,648	79,584	79,584	79,68
79,504	79,616	79,632	79,568	79,68	79,696
79,632	79,52	79,552	79,616	79,552	79,584
79,584	79,6	79,664	79,664	79,632	79,648

Tabla 3.1: Ejemplo de algoritmo de análisis de trayectorias (1).

Se alinean las trayectorias validas, y las descartadas se eliminan fijándose a 0:

trayectorias					
4649,984	4650	0	0	0	0
79,616	79,632	79,648	79,664	79,68	79,696
79,568	79,584	79,6	79,616	79,632	79,648
79,6	79,616	79,632	79,648	79,664	79,68
79,52	79,536	79,552	79,568	79,584	79,6
79,552	79,568	79,584	79,6	79,616	79,632
79,536	79,552	79,568	79,584	79,6	79,616
79,504	79,52	79,536	79,552	79,568	79,584
79,632	79,648	79,664	79,68	79,696	79,712
79,584	79, 6	79,616	79,632	79,648	79,664

Tabla 3.2: Ejemplo de algoritmo de análisis de trayectorias (2).

Se calcula la distancia total de cada trayectoria, obteniendo por tanto un candidato a resultado final, siendo este aquel cuya trayectoria sume la menor distancia total: Se observa que la trayectoria número 3 es la que menor distancia total tiene, por tanto,

nuestro candidato, en este caso, será la primera detección de la trayectoria 3, que corresponde con ***79,568s.***

Num	Trayectoria (última)	Distancia total
1	0	1E+15
2	79,792	2545,75
3	79,744003	2342,08
4	79,776001	2425,83
5	79,695999	2486,6
6	79,727997	2422,17
7	79,711998	2422,39
8	79,68	2573,58
9	0	1E+15
10	79,760002	2355,36

Tabla3.3: Ejemplo de algoritmo de análisis de trayectorias (3).

Capítulo 4:

Pruebas y resultados

4.1 Análisis del programa básico

Como primer paso para el desarrollo del proyecto, se realizará un análisis exhaustivo del rendimiento del programa primario, para poder extraer las primeras conclusiones, así como ver en qué medida y cuáles son los márgenes de mejora. Para ello, se realizan pruebas sistemáticas sobre un audio original, en este caso la pista de audio de una película de la cual se tienen las muestras anteriormente mencionadas. La forma de proceder por tanto será la siguiente:

- Obtención del archivo con los coeficientes MFCC del audio original, ejecutando el script "*gen_mfcc*", y posteriormente, a partir de este archivo de MFCC, se obtiene el archivo .key a través del script "gen_key_from_mfcc", ambos son ejecutados desde "*gen_fingerprint*".

- Obtención del archivo de coeficientes MFCC y posterior *.key* de la muestra ruidosa. Para ello se usa el script "gen_RTKey_from_mfcc", que genera la "Real Time" *.key* de la muestra a análisis.

- Se ejecuta el programa de búsqueda "*find_RT_key*" al cual se le pasa como argumentos los ficheros *.key* obtenidos anteriormente. La salida de este programa será un dato de tiempo, el cual corresponde con el momento del audio original en el que ambas muestras son más similares.

- Se repite esta operación para todo el banco de pruebas, que como comentamos anteriormente está formado por recortes de las muestras grabadas cada 16 ms, dando lugar a un total de cerca de 20.000 muestras comparables.

Todas estas pruebas serán repetidas para los dos tipos de distancia espectral que admite el programa de búsqueda "*find_RT_key*":

- Distancia First Order (City Block)
- Distancia Euclídea

Realizamos las pruebas para cada una de las distancias con el objetivo de comprobar cuál de ellas ofrece mejores resultados, para decantarnos por una de ellas en un futuro. Como prueba inicial, se evaluará sobre el audio original, recortes del propio audio original, como forma de comprobar que el funcionamiento del sistema es el correcto a priori.

Todas las pruebas que vamos a realizar nos darán como resultado una serie de valores de tiempo de sincronismo. Sin embargo, para poder evaluar la precisión del

sistema, debemos saber a priori, cuales son los verdaderos valores de desfase entre ambas muestras para poder compararlos con los obtenidos de realizar la búsqueda. Para ello hay que realizar un sincronismo muy preciso entre las muestras grabadas y la señal original mediante la función correlación cruzada, que es una función que mide la similitud de dos señales a lo largo del tiempo aplicando distintos desplazamientos a una de ellas. Donde se encuentre el máximo de dicha función será el instante en que ambas muestras son más similares y por tanto indicará el desplazamiento de una respecto de la otra.

El cálculo de la función de correlación cruzada es muy costoso computacionalmente, por lo que sería inviable utilizar esta técnica para un sistema rápido de búsqueda de audio en audio. Es más, para el audio con el que se trabaja en este proyecto, de dos horas de duración, ya sería muy costoso analizarlo entero, por lo que se recorta la zona de la cual se han tomado las muestras.

Para este análisis, se tomó un fragmento de 3 minutos del audio original y se hizo la correlación cruzada con cada una de las muestras mediante MATLAB, a través de la siguiente función:

```
[Corr vect] = xcorr(original, sample,'none');
```

Donde `corr` es el vector Y con los datos de la correlación cruzada entre en audio original (original) y la muestra (sample), y `vect` es el vector X. La siguiente gráfica representa la función correlación cruzada entre la muestra original "*spanish.wav*" y la muestra "*bolsillo.wav*":

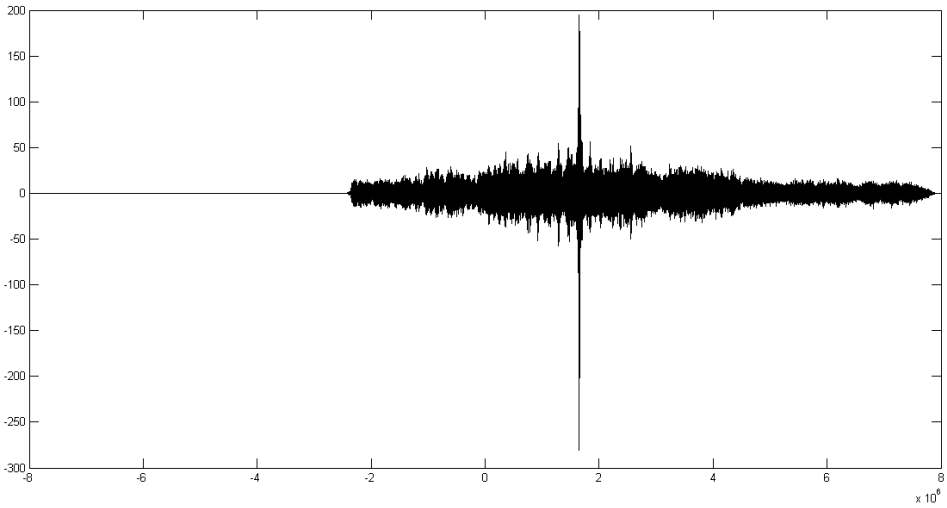


Figura 4.1: Función correlación cruzada entre audio original y muestra.

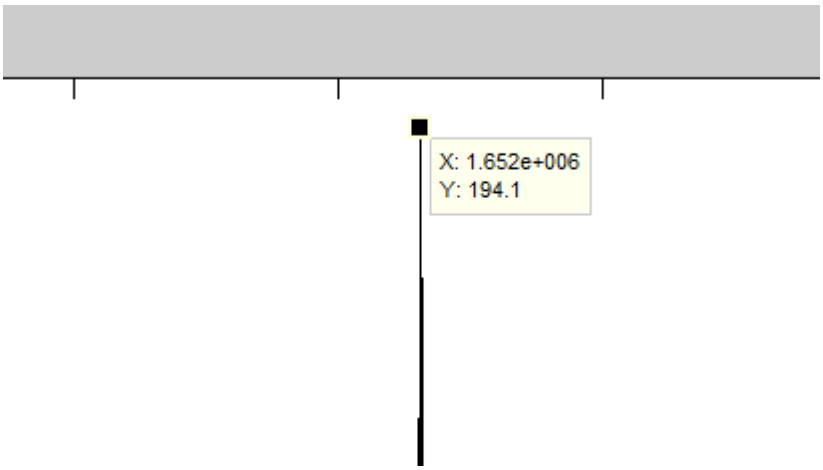


Figura 4.2: Detalle de la función correlación cruzada.

Como se observa, la función de correlación cruzada, tiene un máximo muy claro, que representa el punto donde ambas señales son más parecidas entre sí. Dicho máximo indica el desplazamiento del vector que contiene los valores de "*bolsillo.wav*" sobre el vector de "*spanish.wav*", y por tanto indica el desfase entre ambas. De este modo, por ejemplo para este caso, el máximo se encuentra en el punto 1652256. Por tanto el desfase existente entre ambas es de 1652256 muestras. Como se trata de un audio wav, su frecuencia de muestreo es de 44100 Hz, por lo que:

Desplazamiento =1652256 /44100 = ***37.46612s***

Este proceso se ha repetido para localizar el desplazamiento de cada muestra respecto al inicio de la pista de audio original con muy elevada precisión y servirá como base para las medidas de rendimiento posteriores.

4.1.1 Resultados de los test sobre el programa primario

Calculando la latencia exacta de cada una de las grabaciones, podemos determinar tanto el porcentaje de acierto de las detecciones que ofrece *find_RT_key*, como el error medio en segundos de dichas detecciones. Se establece un margen de 0.3 segundos para considerar una detección correcta (es decir un error de 0.29s se considera un acierto y un error de 0.31s se considera un error). Los valores de error medio en segundos, son aquellos calculados para detecciones correctas, es decir, aquellas que no difieran más de 0.3 segundos tanto por exceso como por defecto

Los resultados obtenidos se muestran en las siguientes tablas:

	%acierto	error (ms)
Audio original	100	16

Tabla 4.1: Rendimiento audio original.

Al evaluar el audio original sobre sí mismo, el rendimiento es óptimo, y solo se produce un error medio de 16 ms que se corresponde con un error medio de un frame.

Para el resto de muestras, los resultados obtenidos son los siguientes:

IPHONE				
Dist:	First Order		Euclidea	
	%acierto	error (ms)	%acierto	error (ms)
Regazo	83.29	57.5	77.93	68.1
Bolsillo	71.13	33.8	76.49	47.0
Bolso	16.85	72.4	21.51	70.7

Tabla 4.2: Rendimiento de las muestras del Iphone.

HTC				
Dist:	First Order		Euclidea	
	%acierto	error (ms)	%acierto	error (ms)
Regazo	80.81	63.4	82.43	45.1
Bolsillo	44.95	55.0	50.97	43.3
Bolso	12.64	66.7	11.92	70.8

Tabla 4.3: Rendimiento de las muestras del HTC One.

El objetivo de esta evaluación es evaluar por un lado los errores de sincronismo que probablemente fuesen detectados como tal por un humano (>0.3s) y por otro la precisión en el sincronismo en los casos en que la detección posiblemente fuese considerada correcta por un humano (<0.3s). En cualquier caso el umbral de 0.3s no se ha ajustado y se ha tomado de forma heurística, dependiendo en cualquier caso de la aplicación en que se quiera emplear el sistema.

Conclusiones respecto al rendimiento

Estos primeros resultados, desvelan la necesidad de realizar mejoras importantes, puesto que el rendimiento es bastante pobre. En las muestras poco ruidosas (Regazo), se sobrepasa ligeramente el 80% de acierto en detecciones. Para las muestras semi-ruidosas (Bolsillo), hay una clara diferencia entre la muestra grabada por un terminal y por otro, que puede ser debida a la calidad del micrófono y de los mecanismos de mejora de audio de ambos, sin embargo, los resultados tampoco son muy satisfactorios. Finalmente, para las muestras ruidosas, el resultado es bastante desastroso en ambos casos, puesto que en el mejor de ellos llegamos al 21% de acierto.

El error medio es un claro indicador de la exactitud de las detecciones y se puede observar que aun estando dentro de una zona considerada de acierto, hay bastante variación entre los frames detectados (errores de cerca de 60 ms corresponden a un salto de 4 frames respecto al instante correcto):

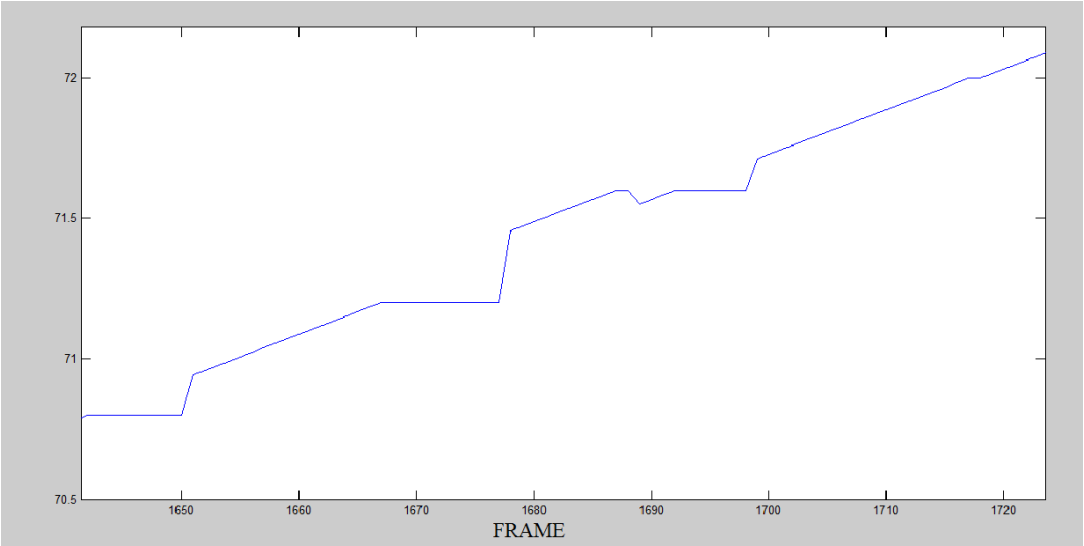


Figura 4.3: Evolución temporal de detecciones.

En la gráfica se observa como las detecciones no siguen una línea recta respecto al tiempo, sino que hay detecciones que se repiten durante varios frames. Este hecho hace que se acumulen diferencias de tiempo respecto al instante correcto de sincronismo, lo que da lugar al error medio que mostraban las tablas.

Conclusiones respecto a la distancia espectral

Las medias obtenidas para cada una de las distancias son las siguientes:

	%acierto	error (s)
FIRST	0,516	0,0581
EUCLIDEA	0,535	0,0575

Tabla 4.4: Resultados medios del programa básico.

La distancia Euclídea ofrece mejores resultados, tanto en porcentaje de acierto como en error medio, aunque bien es cierto que no son demasiado significativos por ahora, por tanto, en las futuras mejoras, se evaluara de nuevo el rendimiento que ofrecen ambas con el objetivo de elegir la más adecuada.

4.2. Primera mejora: CMVN

4.2.1 CMVN sobre coeficientes MFCC

La primera mejora que se introduce al programa primario es la normalización previa de los vectores de coeficientes, que como se explicó en capítulos anteriores, proporciona una mejora significativa, en cuanto a minimización del efecto de ruido y distorsión se refiere. Para evaluar esta mejora, se realiza de nuevo el análisis completo de todo el banco de muestras y se evalúan las dos distancias espectrales. Los resultados obtenidos son los siguientes:

IPHONE				
	EUCLIDEA		FIRST ORDER	
	%acierto	Error(ms)	%acierto	Error(ms)
regazo	100	11.1	100	12.5
bolsillo	87,32	9.4	86,69	8.9
bolso	60,98	29.2	72,63	33.3

Tabla 4.5: Rendimiento para muestras del Iphone con CMNV.

HTC				
	EUCLIDEA		FIRST ORDER	
	%acierto	error (ms)	%acierto	error (ms)
regazo	98.88	16	99,91	19.5
bolsillo	83,07	39.8	82,66	12.6
bolso	62,39	15.4	66,73	15.3

Tabla 4.6: Rendimiento para muestras del HTC One con CMNV.

	%acierto	error (ms)
FIRST ORDER	84.77	17
EUCLIDEA	82,10	20.1

Tabla 4.7: Rendimiento medio con CMNV para las diferentes distancias.

En este caso, los resultados obtenidos indican un mejor resultado tanto en porcentaje de acierto como en error medio para la distancia First Order. Si bien es cierto la diferencia sigue sin ser significativa, por lo que en pruebas sucesivas se seguirán evaluando ambas.

4.2.2 CMVN sobre coeficientes MFCC y sobre Fingerprint

El algoritmo de normalización de media y varianza para vectores de coeficientes espectrales (CMVN), está concebido en principio para normalizar secuencias de coeficientes MFCC. Sin embargo, a modo de prueba, en este proyecto, se ha propuesto su uso también para normalizar los vectores huella o *fingerprints*. Estos vectores están compuestos por promedios de coeficientes MFCC previamente normalizados, por lo que en principio no existe ningún inconveniente para aplicar de nuevo CMVN. Al emplear esta técnica, estamos llevando a cabo una doble normalización de coeficientes espectrales y las prestaciones así obtenidas son las siguientes:

IPHONE				
Dist:	First Order		Euclidea	
	%acierto	error(ms)	%acierto	error(ms)
Regazo	100	11.2	100	10.9
Bolsillo	98,53	14.1	99,09	10.1
Bolso	86,63	13.1	84,32	29.8

Tabla 4.8: Rendimiento para muestras del IPhone con CMNV doble.

HTC				
Dist:	First Order		Euclidea	
	%acierto	error(ms)	%acierto	error(ms)
Regazo	100	25.5	100	25.2
Bolsillo	88,97	10.1	90,53	13.2
Bolso	76,01	15.6	74,04	21.8

Tabla 4.9: Rendimiento para muestras del HOT One con CMNV doble.

	%acierto	error (ms)
FIRST	91.69	14.9
EUCLIDEA	91.33	18.5

Tabla 4.10: Rendimiento medio con CMNV doble para las diferentes distancias.

Ambas distancias obtienen unos valores muy similares en cuanto a porcentaje de acierto y error medio. La distancia First Order sigue teniendo una ligerísima ventaja en ambos aspectos, sin embargo, se seguirá experimentando con ambas distancias para llegar a una decisión final. Asimismo, la mejora global implementando CMNV es evidente, y queda reflejada en las siguientes gráfica:

- **Precisión en las detecciones (% de acierto)**

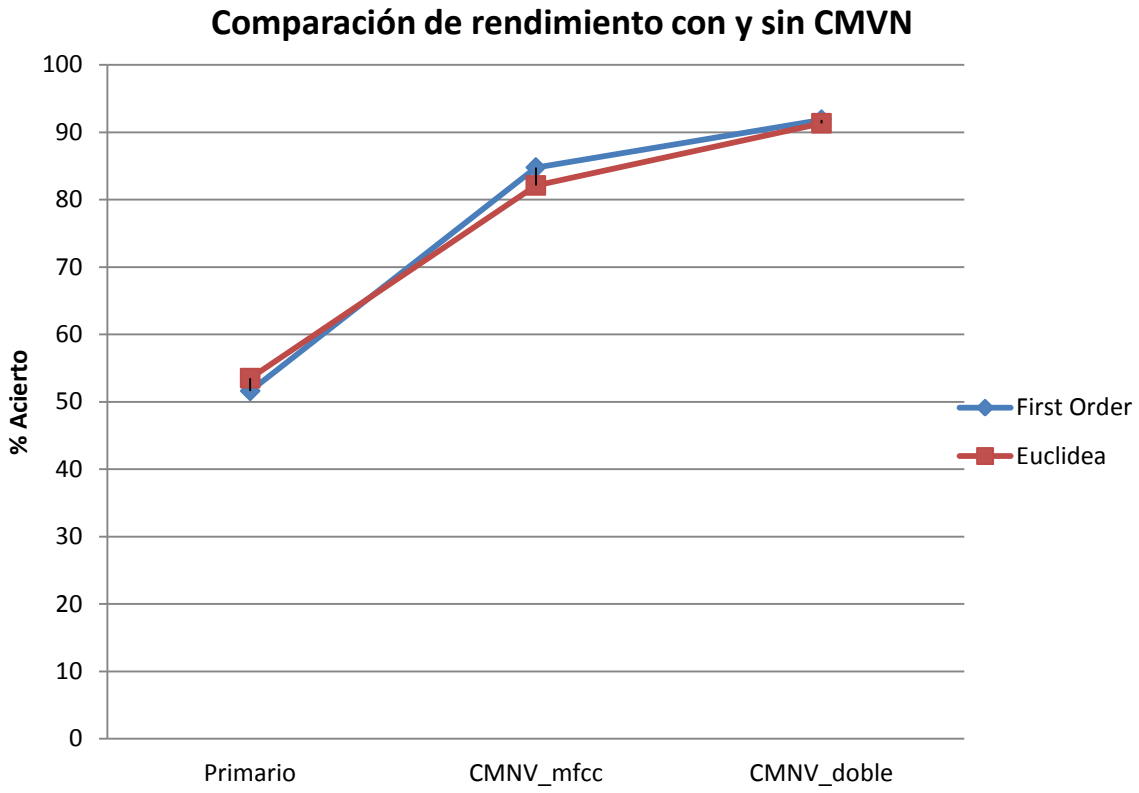


Figura 4.4: Evolución del rendimiento con las mejoras (% acierto en detección).

- **Error medio en detección correcta (en segundos)**

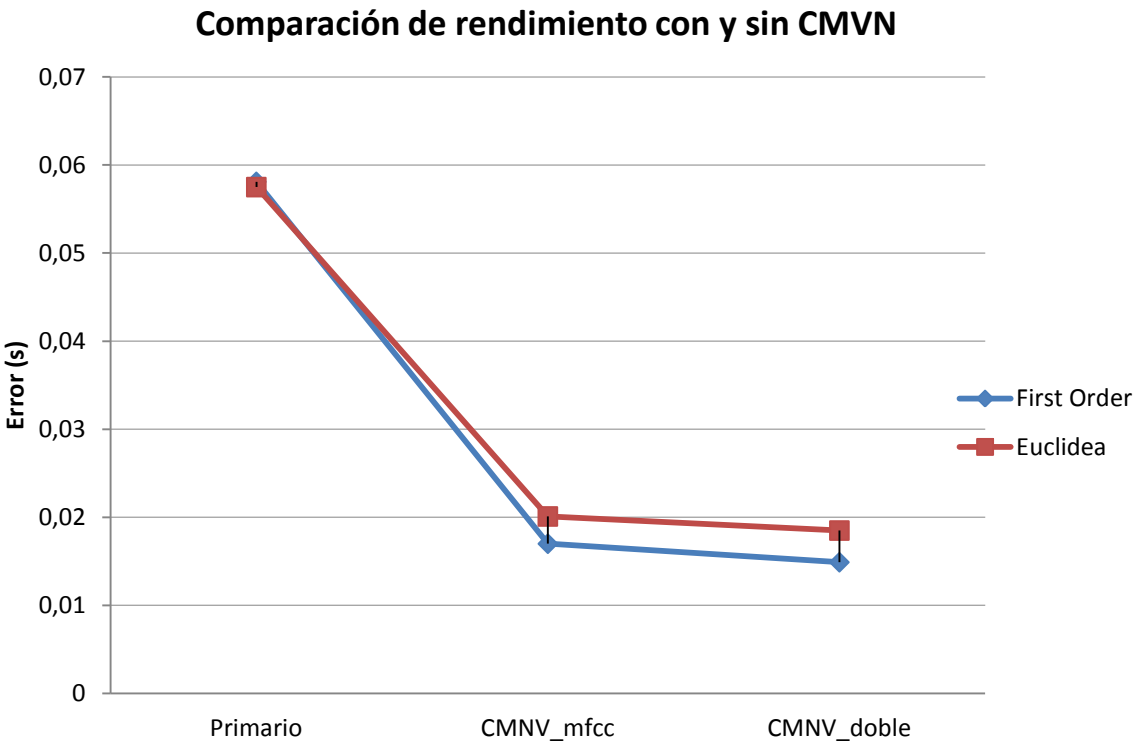


Figura 4.5: Evolución del rendimiento con las mejoras (error medio en segundos).

Análisis de la mejora CMNV.

Después de la implantación de la mejora, se evalúan las mejoras obtenidas para cada tipo de muestras con las que ha sido testada:

- Para muestras grabadas en condiciones normales de uso del programa (Las muestras de Regazo) el resultado obtenido es del 100% de detecciones correctas para las muestras grabadas con los 2 teléfonos (al principio rondaba el 80%).

- Para grabaciones semi-ruidosas (bolsillo), hay una diferencia notable entre la muestra grabada con iPhone y con HTC:

    - La grabada con iPhone tiene un ruido inicial importante de 5 segundos aproximadamente debido al hecho de introducir el móvil en el bolsillo. Sin embargo el porcentaje de error que se obtiene es del 1% que equivale a 0.5 segundos de grabación mientras que al principio el porcentaje era

del 25% que equivalía a en torno a 13 segundos. Por tanto con CMNV, en este caso se obtiene una mejora muy importante.

- En la muestra grabada con HTC, no solo existe el ruido propio de introducir un teléfono en el bolsillo, sino que se escucha una voz humana, que enmascara por completo la señal que estamos grabando. La combinación de estos ruidos tiene una longitud de unos 6 segundos, que sin embargo se corrigen gracias al algoritmo CMNV. Sin embargo, en esta muestra se produce un fenómeno diferente que será importante caso de estudio más adelante. La grabación contiene una melodía de piano que repite dos compases prácticamente sin variaciones, por lo que el programa se confunde e indica que estamos en una zona que no corresponde. Esto será un problema importante que no había sido previsto al inicio del proyecto y que será necesario abordar más adelante. Sería de gran importancia para el caso en que se quisiera ampliar el uso de este programa al análisis de pistas musicales, debido a que estas suelen tener patrones o melodías que se repiten.

  La diferencia respecto al porcentaje de acierto con respecto al iPhone, se da por tanto en este hecho, puesto que el patrón que se repite tiene una duración de unas 350 detecciones que equivalen al 11% de la muestra aproximadamente.

- Para muestras ruidosas (Bolso), la mejora en ambos casos es más que evidente, pasando del 18% y 11% al 85% y 75% respectivamente. En este caso también existen diferencias significativas en cuanto a los resultados obtenidos para los 2 teléfonos. La muestra grabada con el iPhone tiene mucho ruido al principio, debido a que se introduce en el bolso, se cierra la cremallera varias veces etc. Esto tiene una duración de unos 10 segundos y anula prácticamente la señal original. Es por ello que el 15% del error que se comete venga de esta parte. Por otro lado, la muestra grabada con el HTC, además del ruido generado por introducir el teléfono en el bolso, también tiene una parte en la que se escucha una voz que impide por completo escuchar el audio de la película. Por tanto, aquí la detección se hace prácticamente imposible de realizar y de ahí viene el 25% de detecciones erróneas. Para este tipo de casos, se planteará la solución de, en caso de que se detecten estas situaciones y sea necesario, grabar un trozo más largo de audio con el fin de abarcar una zona de menos ruido para tratar de retrasar la respuesta del sistema en caso de duda, en lugar de dar una detección errónea.

Próximos objetivos.

Tras el análisis de la mejora realizada con la implantación del módulo CMNV tanto en MFCC como en *fingerprint*, se plantean los siguientes objetivos:

- Mejorar los resultados obtenidos tanto en porcentaje de acierto como en error medio, mediante el análisis M-Best de las M mejores posibles detecciones y las distancias de cada una a través de ventanas deslizantes que abarquen diferentes trayectorias.

- Tratar de identificar cuando una detección es incorrecta, para así poder indicar al programa que se debe grabar más audio para obtener la detección correcta.

- Abordar el caso de patrones repetidos en una grabación que pueden hacer que el programa confunda regiones del audio. Para ello, será necesario un preprocesado del audio original.

4.3 Segunda mejora: Distancias mínimas y trayectorias

El funcionamiento del algoritmo de búsqueda se basa en el cálculo de la distancia mínima entre cada par de vectores de coeficientes del audio original y de la muestra a comparar. Al analizarse toda la muestra linealmente, existen una serie de detecciones más probables, cuyas distancias son las menores llamadas M-Best. En el siguiente apartado, veremos cómo se obtienen y que pueden aportar para mejorar el sistema.

4.3.1 Obtención de M-BEST

El primer experimento de cálculo de M-Best, consistirá en la obtención de las 10-Best detecciones posibles para todo el banco de pruebas, evaluando si entre esas 10 mejores detecciones están las detecciones correctas y midiendo los errores eligiendo de entre las 10-best detecciones la mejor.

Euclidea			
		% acierto	% acierto 10-Best
iPhone	Regazo	100	100
	Bolsillo	99,09	100
	Bolso	84,32	89,35
HTC	Regazo	100	100
	Bolsillo	90,53	100
	Bolso	74,04	79,6

Tabla 4.11: Rendimiento alcanzable utilizando 10-Best con distancia Euclídea.

La más clara de las mejoras se da en la muestra del bolsillo grabada con HTC, aquella que detectaba la zona donde se repetía la melodía, se observa que dentro de las 10 detecciones más probables, se encuentran todas las correctas. También se observan mejoras de entorno al 5% para las muestras del bolso tanto de iPhone como del HTC. Los resultados obtenidos utilizando distancia First Order son los siguientes:

First Order			
		% acierto	% acierto 10-Best
iPhone	Regazo	100	100
	Bolsillo	98,53	99,53
	Bolso	86,63	92,03
HTC	Regazo	100	100
	Bolsillo	88,97	98,53
	Bolso	76,01	81,47

Tabla 4.12: Rendimiento alcanzable utilizando 10-Best con distancia Fisrst Order.

	Ganancia media con 10-BEST
First Order	5,19%
Euclídea	5,24%

Tabla 4.13: Ganancia media de rendimiento con 10-Best.

Las conclusiones después de haber hecho el análisis 10-Best, son que en ambos casos, distancia First Order y distancia Euclidea, se obtiene alrededor de un 5% más de detecciones correctas que hecho el análisis sin M-best. También cabe destacar que la muestra del bolsillo del HTC es la que más aumenta en el número de detecciones correctas, debido a que la zona de error se debe a la parte de la melodía que se repite y no a un fuerte ruido que hiciera imposible la detección. Como diferencia más notable entre ambas distancias, se puede decir que la distancia Euclidea funciona mejor en muestras semi-ruidosas (Bolsillos), y sin embargo, la distancia First Order, ofrece mejores resultados para muestras más ruidosas (Bolsos).

4.3.2 Análisis de trayectorias

A partir de la obtención de las M-Best detecciones, se puede poner en funcionamiento el algoritmo de análisis de trayectorias. Su incorporación aporta una mejora en el rendimiento del programa, tanto en porcentaje de acierto, como en error medio. Esto es debido a que se evalúa la solución más coherente en función de una serie de detecciones posteriores alineadas en el tiempo. Las tablas de rendimiento para este algoritmo son las siguientes:

Distancia First Order				
	(%acierto)	sin	con	diff
HTC	Bolsillo	88,97	89,4	0,43
	Bolso	76,01	75,49	-0,52
iPhone	Bolsillo	98,53	98,9	0,37
	Bolso	86,63	86,83	0,2

Tabla 4.14: Análisis de Trayectorias para distancia First Order

Distancia Euclidea				
	(%acierto)	sin	con	diff
HTC	Bolsillo	90,53	90,43	-0,1
	Bolso	74,04	74,5	0,46
iPhone	Bolsillo	99,09	99,94	0,85
	Bolso	84,32	85,34	1,02

Tabla 4.15: Análisis de Trayectorias para distancia Euclídea

mejora global del algoritmo:	0,45%

Un primer aspecto a destacar es que después de muchas pruebas, el hecho de utilizar una distancia u otra, no es crítico ningún aspecto fundamental. Por lo tanto, a partir de ahora, todas las pruebas se llevaran a cabo utilizando distancia First Order, porque es la que menor coste computacional tiene.

Respecto a las tablas de resultados, en 6 de los 8 casos, se consigue una mejora utilizando el algoritmo. Si bien estas mejoras no son muy significativas en cuanto a porcentaje (apenas un 0.45 %), hay que tener en cuenta que cuanto más cercano es éste al 100%, más difícil es pulirlo para aumentar el rendimiento. No obstante, a parte de esta ligera mejora, la verdadera potencia de este algoritmo reside en los parámetros característicos que se derivan de su utilización. Dichos parámetros son los siguientes:

1. El número de trayectorias que sobreviven al análisis.
2. La distancia entre las trayectorias supervivientes.

El primer parámetro nos servirá como factor de calidad de la detección, esto es, si sobreviven muchas trayectorias, es más probable que tengamos una detección fiable. Por otro lado, el segundo puede aportarnos información sobre aspectos interesantes del audio a analizar, ya sea una canción, una banda sonora de una película, etc. Concretamente nos referimos a patrones que se repiten, como puede ser un estribillo en una canción o una melodía que se escucha en varias ocasiones a lo largo de una película (como sucede en las pruebas de este proyecto). Si tenemos varias trayectorias supervivientes distanciadas entre ellas, es probable que estemos ante uno de estos casos. Por tanto, contrastando los valores de estos parámetros con la información que tenemos a priori, seremos capaces de fijar unos umbrales que permitan establecer si la detección es suficientemente fiable y en caso contrario analizar una muestra de tamaño superior con el fin de determinar patrones que sincronicen ambos audios inequívocamente.

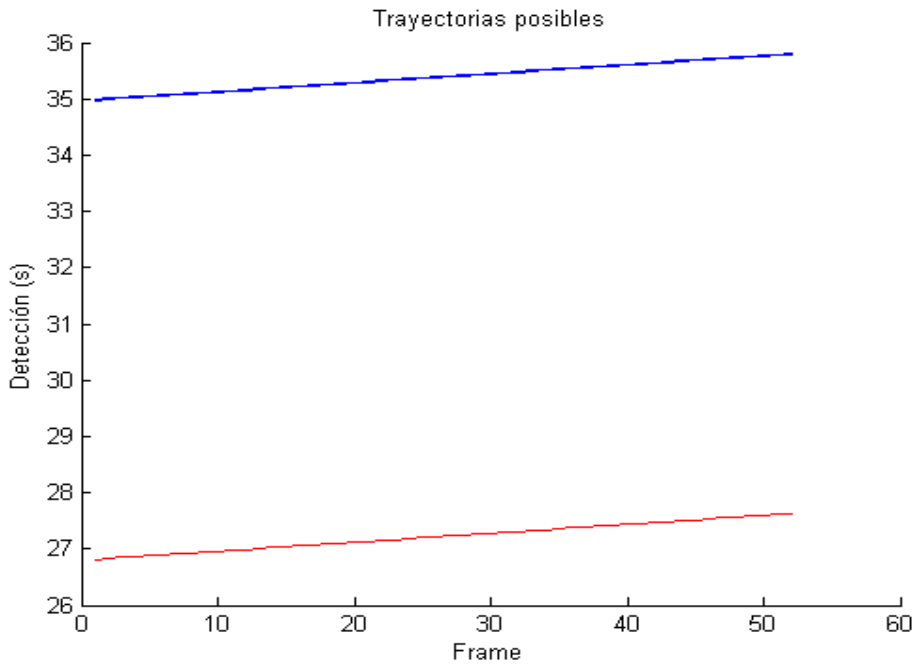


Figura 4.6: Análisis de trayectorias: Patrones repetidos.

El gráfico representa las trayectorias supervivientes en una parte musical de la pista de audio. En ella existe una melodía de piano de una duración aproximada de 8 segundos que se repite. Como se puede observar, se obtienen varias trayectorias supervivientes separadas en el tiempo.

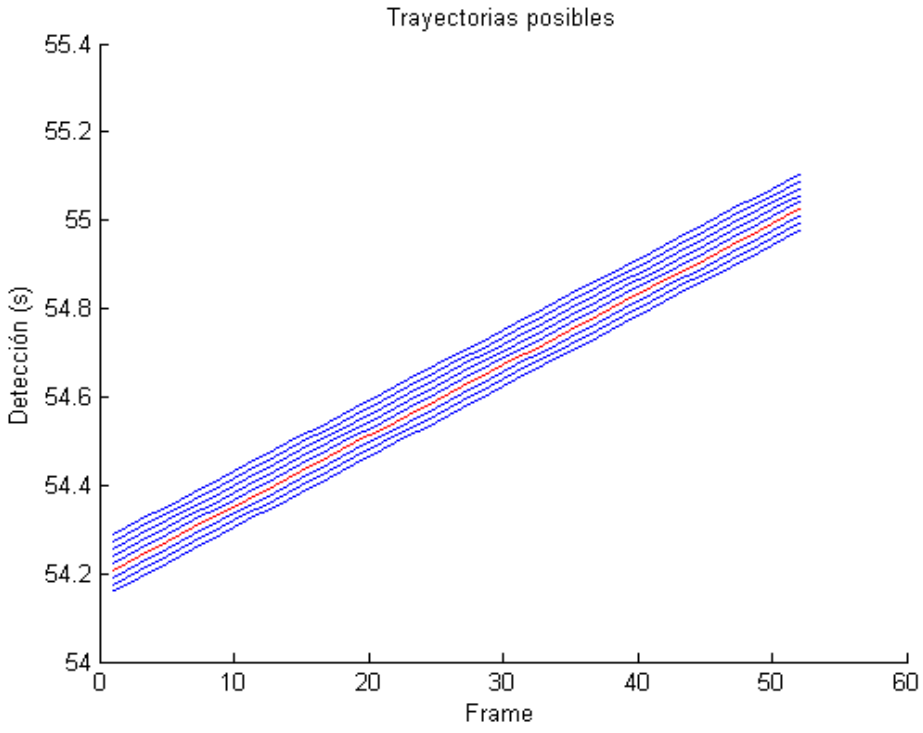


Figura 4.7: Análisis de trayectorias: Detección fiable.

4.3.2.1 Análisis de trayectorias: Parámetros característicos

La cuestión que queremos abordar es, analizando las pruebas hechas, obtener unos valores umbrales óptimos para estos parámetros, que nos identifiquen lo más certeramente posible si la detección ha sido correcta o no. Antes de continuar, es necesario comentar la forma de extracción de estos parámetros. El primero de ellos (número de trayectorias supervivientes) es bastante obvio, pero el segundo puede no resultarlo tanto. La distancia entre trayectorias supervivientes se calcula de la siguiente manera:

1. Se calcula la media de los puntos finales de las trayectorias supervivientes
2. Se calcula la diferencia entre esa media y el punto final de la trayectoria cuya distancia es la mínima.

$$Distancia\colon\ D_t(i) = T_{minima,last}(i)\ \ - \left[\frac{1}{N}\sum_{n=1}^{N=tray.sup.} T_{n,last}(i)\right]$$

De este modo, si tenemos varias trayectorias separadas, el valor de este parámetro deberá ser grande. Por ejemplo en el análisis de la muestra del bolsillo grabada con el HTC, nos encontramos con un patrón de melodía que se repite tras 8 segundos (tiempo aproximado de un compás para piano). Al hacer el análisis de trayectorias justo en esa parte de la muestra, se obtienen 3 trayectorias que sobreviven que corresponden a 34.8s aprox. y otra que lo hacen para un tiempo de 26.6s aprox. (Trayectoria A en la siguiente Tabla):

Trayectoria	A	B	C
1	0	50,864	0
2	34,83	50,8	57,28
3	0	50,928	57,216
4	0	50,816	57,248
5	34,81	50,88	57,232
6	26,67	50,912	57,344
7	0	50,848	57,296
8	0	50,832	57,312
9	34,84	0	57,264
10	0	50,896	57,328
Distancia media entre trayectorias supervivientes	**2,04**	**0**	**-0,064**

Tabla 4.16: Varios ejemplos de trayectorias detectadas: El resaltado indica la trayectoria cuya distancia es la menor.

Este caso, aun siendo uno de los peores casos (que se repitiera el mismo patrón tan seguido en el tiempo, y que hubiera más supervivientes de una trayectoria que de otra), se puede observar una clara diferencia en el valor de distancia obtenido, respecto a valores obtenidos para trayectorias sin este peculiar problema.

Para el establecimiento de los umbrales óptimos, se hará un análisis de los valores de los dos parámetros a lo largo de la todas las muestras disponibles, con el fin de poder caracterizar tanto zonas de error como zonas de acierto.

Figura 4.8: Parámetros obtenidos con análisis de trayectorias para la muestra bolso HTC.

La figura representa una media del número de trayectorias supervivientes (medias de 30 frames, 0.48 segundos) en azul, y las zonas marcadas en verde, son aquellas en las cuales el parámetro distancia es mayor que la unidad. Se ha destacado en rojo, la zona donde la detección ha sido errónea. Para esta muestra se trata de una zona bastante grande puesto que al principio de la grabación se produce gran cantidad de ruido al introducir el teléfono en el bolso. Se observa claramente como en esta zona el número de trayectorias supervivientes es mucho menor que para la zona donde la detección es correcta.

El parámetro Distancia media entre trayectorias supervivientes, ha sido truncado por motivos de legibilidad de la figura. Corresponde con el color verde, para distancias mayores que la unidad (distancia considerable, como se vio anteriormente). Se observa cómo este parámetro también define bastante bien la zona de error, ya que en prácticamente toda ella, el parámetro distancia tiene un valor superior a uno. También hay que tener en cuenta que, aunque en menor medida, está también presente en zonas de detección correctas consideradas zonas de riesgo.

Los resultados obtenidos para el análisis de los parámetros principales derivados del algoritmo de análisis de trayectorias para los casos a mejorar son los siguientes:

	Trayectorias en zona de error	Trayectorias en zona de acierto
Bolso HTC	4,1547	8,809
Bolsillo HTC	7,716	8,9702
Bolso IPhone	4,0143	8,922
Bolsillo IPhone	3,7143	9,104
MEDIA:	**4,895**	**8.947**

Tabla 4.17: Número medio de trayectorias supervivientes en zona de error y zona de acierto.

	% Distancia > 1 en error	% Distancia > 1 en acierto
Bolso HTC	75,45%	11,88%
Bolsillo HTC	82,25%	17,43%
Bolso IPhone	54,76%	14,48%
Bolsillo IPhone	60%	6,81%
MEDIA:	**68,11%**	**12.65%**

Tabla 4.18: Parámetro distancia mayor que 1 en zona de error y zona de acierto.

A la vista de las tablas de resultados, parece obvio pensar que estos parámetros nos aportan bastante información acerca de si una detección fue o no correcta ya que en ambos casos, la diferencia es muy notable entre la zona de acierto y la zona de error. Esto por tanto supone un gran avance a la hora de mejorar la precisión en la detección, o bien solicitando más tiempo de grabación si aún no se ha podido realizar una detección con garantías o indicando la detección como errónea.

En el caso del número de trayectorias supervivientes, se observa como para zonas de acierto, en todos los casos analizados el valor se encuentra muy próximo al 9. Para las zonas de error hay que hacer una pequeña observación, y es que para 3 de las 4, el valor obtenido está rondando el 4, mientras que para otra de ellas es considerablemente mayor. Esto se debe a que es la muestra donde se repite la

melodía y se produce detección errónea. Al no ser una detección errónea causada por el ruido, el número de trayectorias supervivientes es bastante grande. Sin embargo sigue siendo inferior al valor para zonas de acierto. Para este tipo de casos es para los que se definió el parámetro distancia.

El parámetro distancia, es un parámetro que puede tener un rango de valores muy amplio, aunque como quedó demostrado anteriormente, no nos importará mucho a partir de un cierto valor que indique que estamos ante una posible zona de detección doble. Por eso, se ha considerado como la unidad el valor de referencia (para zonas de detección única este valor suele ser diez veces menor aproximadamente).

Se observa una clara diferencia de probabilidad de que este valor sea mayor que uno para la zona de error (un 68% de media) y para la zona de acierto (un 12% de media). El siguiente paso será combinar ambos parámetros de tal forma que podamos realizar una detección prácticamente fiable al 100%. Para ello se podrán establecer umbrales más o menos conservadores, dependiendo si se desea un porcentaje de acierto mayor o un tiempo de respuesta menor. Si el umbral es menos conservador, obtendremos una respuesta más temprana pero con mayor riesgo de ser errónea y viceversa. Para evaluar todo esto, se harán diferentes pruebas con diferentes umbrales.

La prueba consiste en aplicar el umbral 1 para el parámetro Distancia combinado con los umbrales 3, 4, 5, 6, 7, 8 y 9 para el parámetro número de trayectorias. De modo que si el la detección no cumple dichos requisitos de calidad, se realiza una nueva detección pasado un tiempo t_0, y así sucesivamente hasta conseguir cumplir los requisitos. Cuando se alcancen estos requisitos, se devolverá el tiempo de sincronismo teniendo en cuenta todos los retardos.

El tiempo t_0, deberá ser suficientemente grande como para evitar múltiples detecciones que no cumplan los requisitos puesto que esto ralentizaría el funcionamiento y suficientemente pequeño para no alargar en exceso el tiempo de grabación requerido. El estudio del valor óptimo de este intervalo de tiempo se hará más adelante. Para la siguiente prueba, se ha determinado un tiempo $t_0 = 150\ frames$, que equivale a 2.4 segundos:

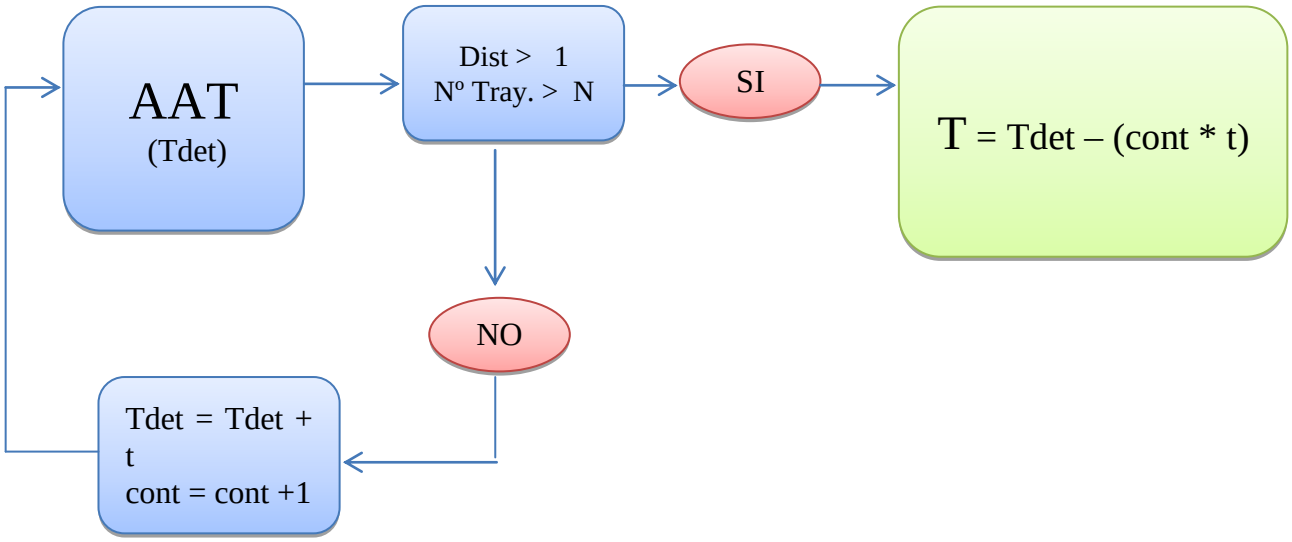


Figura 4.9: Lógica del algoritmo de análisis de trayectorias.

Los resultados son los siguientes:

	Rendimiento del sistema para t=150 frames y distancia = 1						
Umbral	3	4	5	6	7	8	9
Bolso HTC	96,15%	98,95%	100%	100%	100%	100%	100%
Bolsillo HTC	95,99%	96,25%	97,50%	98,03%	99,11%	100%	100%
Bolso Iphone	98,45%	99,44%	99,54%	99,61%	99,90%	100%	100%
Bolsillo Iphone	99,74%	100%	100%	100%	100%	100%	100%
Media	**97,58%**	**98,66%**	**99,26%**	**99,41%**	**99,75%**	**100%**	**100%**

Tabla 4.19: Rendimiento del sistema para diferentes umbrales de trayectorias supervivientes con t = 150 frames.

Utilizando este método podemos asegurar un 100% de detecciones correctas para todo el banco de muestras utilizando un umbral suficientemente alto. La muestra que requiere de un umbral mayor para alcanzar ese porcentaje óptimo de es la muestra del Bolsillo HTC. Esto se debe a que la media de trayectorias supervivientes en zona de error para esta muestra era la más alta (cercana a 8) y obviamente, el margen para obtener detecciones correctas debe ser mayor. Para un umbral de 8 trayectorias supervivientes (combinado con el umbral de distancia > 1) podemos garantizar que todas las detecciones serán correctas.

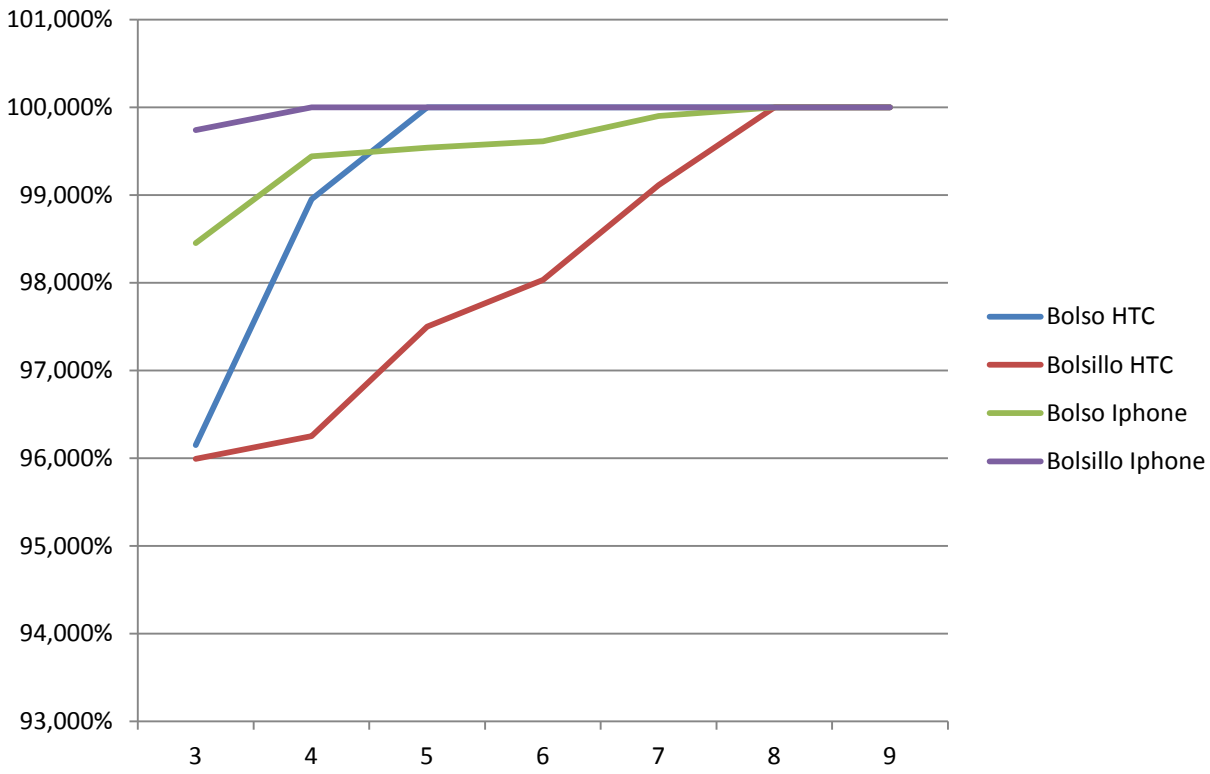


Figura 4.10: Rendimiento del sistema para diferentes umbrales de trayectorias.

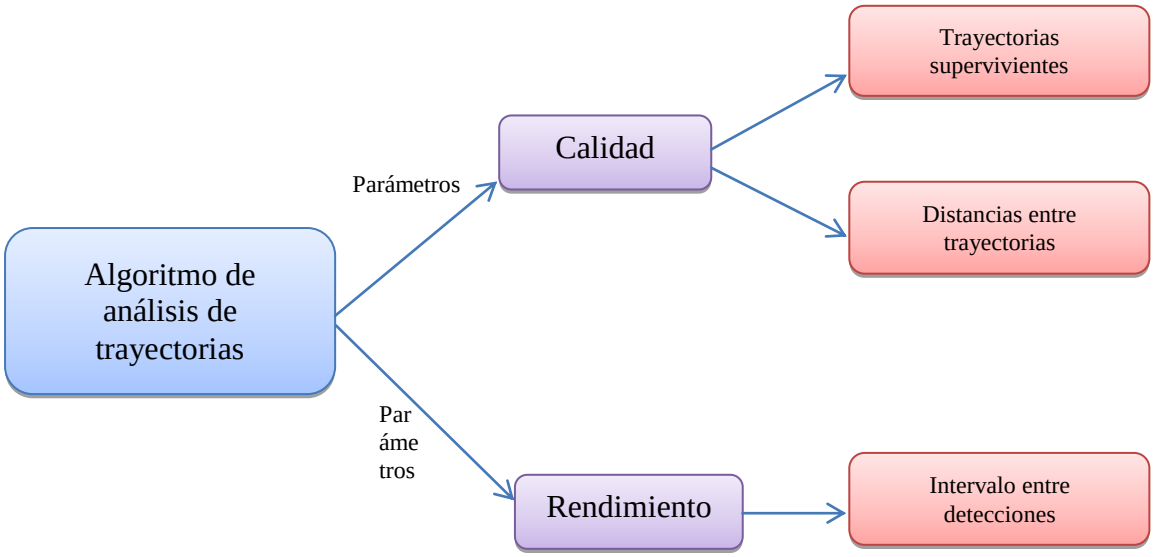
Sin embargo, para lograr esto, en muchos casos necesitamos realizar varias detecciones, lo cual supone un aumento en el tiempo total de respuesta del sistema, o lo que es lo mismo, la necesidad de grabar más audio. En las tablas sucesivas se mostrarán los valores medios de tiempo empleado para cada detección a lo largo de toda la muestra:

Tiempo medio de detección con t=150 frames (s)							
Umbral:	3	4	5	6	7	8	9
Bolso HTC	10,38	10,71	11,13	11,53	11,85	13,31	17,77
Bolsillo HTC	9,7	9,75	9,91	10,14	10,5	10,91	14,91
Bolso Iphone	9,36	9,43	9,45	9,59	9,76	10,14	13,3
Bolsillo Iphone	8,23	8,27	8,32	8,43	8,62	9,23	10,76
Media	**9,41**	**9,54**	**9,70**	**9,92**	**10,18**	**10,89**	**14,18**

Tabla 4.20: Tiempo medio de detección para diferentes umbrales de trayectorias con t = 150 frames.

Se puede intuir la aparición de un nuevo parámetro que afectará al programa. Se trata del intervalo de tiempo que se emplee para obtener las siguientes detecciones en caso de que la anterior no fuera suficientemente satisfactoria. Este parámetro difiere de los anteriores en un aspecto básico, puesto que no se trata de un factor de calidad

de detección, pero sí de rendimiento, puesto que ajustándolo de manera oportuna, podremos reducir el tiempo medio de detección.



Para las pruebas anteriores, se ha utilizado un valor de 150 frames, que equivalía a 2.4 segundos, pensando en que si la detección no fue satisfactoria, es probable que se necesite un margen relativamente grande para poder obtener una detección mejor. Ahora se testeará diferentes valores de este intervalo, para ver de qué forma se comporta el programa:

Tiempo medio de detección con t=100 frames (s)							
Umbral:	3	4	5	6	7	8	9
Bolso HTC	9,95	10,28	10,77	11,25	11,54	12,7	16,74
Bolsillo HTC	9,23	9,29	9,56	9,85	10,35	10,7	13,06
Bolso Iphone	9,2	9,29	9,3	9,41	9,57	9,87	11,87
Bolsillo Iphone	8,19	8,24	8,27	8,34	8,46	9,08	10,32
Media	**9,14**	**9,27**	**9,47**	**9,71**	**9,98**	**10,58**	**12,99**

Tabla 4.21: Tiempo medio de detección para diferentes umbrales de trayectorias con t = 100 frames.

Reduciendo el tiempo del intervalo, vemos como los tiempos medios de detección se reducen. Para este nuevo valor del intervalo, la tabla de rendimiento del sistema será diferente, y por tanto hay que tenerla en cuenta para realizar el estudio. La nueva tabla para un valor de intervalo de 100 frames es la siguiente:

	Rendimiento del sistema para t=100 frames						
Umbral	3	4	5	6	7	8	9
Bolso HTC	94,98%	98,45%	100%	100%	100%	100%	100%
Bolsillo HTC	94,14%	94,53%	97,22%	97,99%	99,22%	100%	100%
Bolso Iphone	98,22%	99,44%	99,54%	99,61%	99,90%	100%	100%
Bolsillo Iphone	99,74%	100,00%	100,00%	100,00%	100,00%	100%	100%
Media	**96,77%**	**98,10%**	**99,18%**	**99,40%**	**99,78%**	**100%**	**100%**

Tabla 4.22: Rendimiento para diferentes umbrales de trayectorias con t = 100 frames.

Para valores a partir de 8, se obtiene un rendimiento óptimo como en el caso de t=150, sin embargo, se produce una disminución de rendimiento para los umbrales más bajos. Por tanto se deduce que una disminución del intervalo de tiempo produce un tiempo medio de detección menor pero un rendimiento más pobre para umbrales bajos.

Es lógico que cuanto menor sea el tiempo del intervalo, el tiempo medio de respuesta sea menor puesto que se realizan más detecciones en menos tiempo y es más probable alcanzar una que satisface los umbrales en un tiempo menor. El hecho del descenso del rendimiento en umbrales bajos se puede explicar de la siguiente manera: Al realizar detecciones más seguidas, si una de ellas fue incorrecta, es más probable que la siguiente vuelva a ser incorrecta cuanto más cercana a ella sea, debido a que seguirá influyendo más el ruido. Esto pone a prueba más al límite al algoritmo para umbrales bajos, y por eso los valores de rendimiento serán menores.

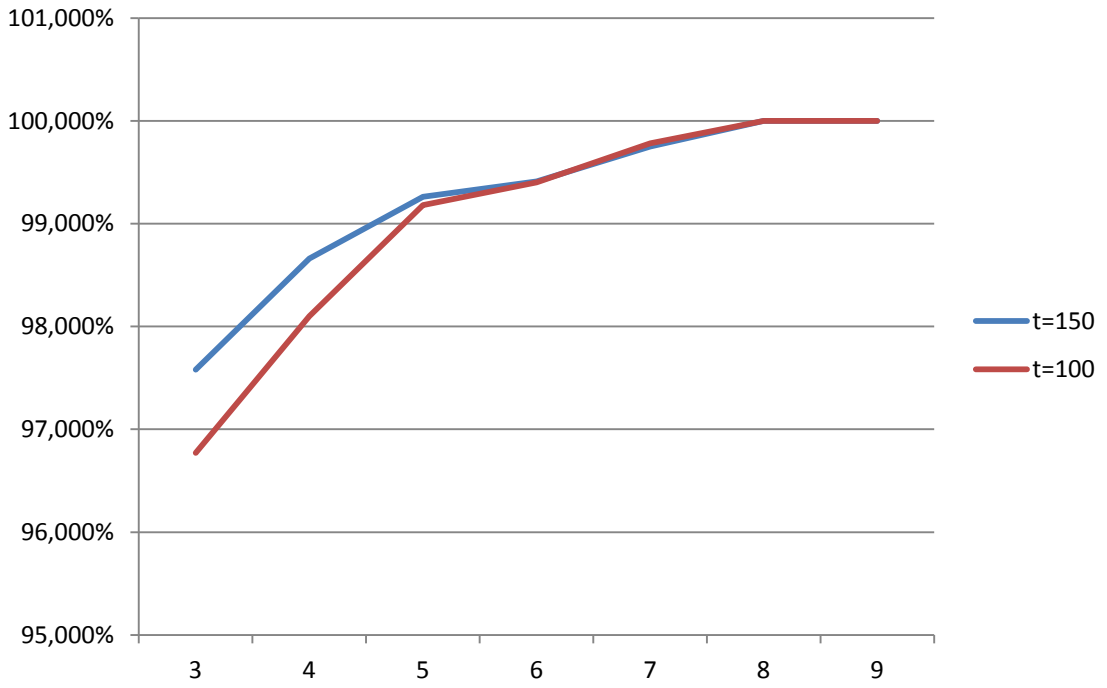


Figura 4.11: Rendimiento del sistema para intervalos de 100 y 150 frames.

También habrá que tener en cuenta que cuanto menor sea el intervalo de tiempo, el programa tendrá que hacer más detecciones, y esto supondrá un coste computacional mayor. Este problema será caso de estudio más adelante. Finalmente, la comparativa de tiempo medio de detección para ambos intervalos es la siguiente:

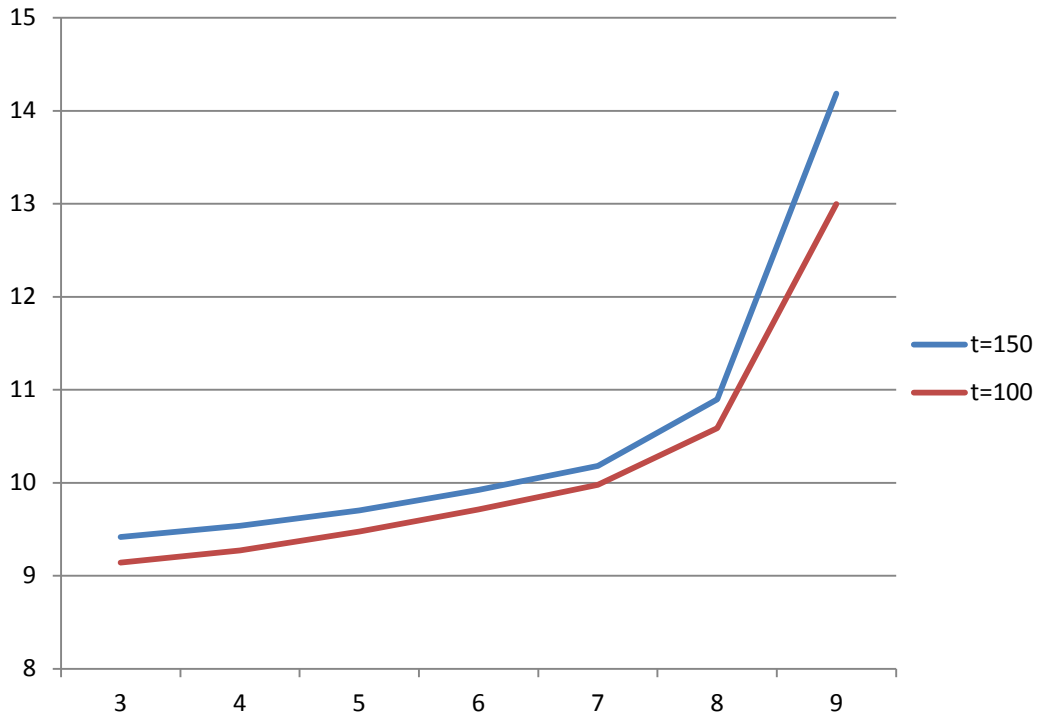


Figura 4.12: Tiempo medio de detección para intervalos de 100 y 150 frames.

Se observa un tiempo superior para todos los umbrales para el intervalo de 150 frames respecto al intervalo de 100 frames. La comparativa del rendimiento del sistema para ambos intervalos es la siguiente:

Para valores de umbral bajos, cuanto mayor es el intervalo de tiempo entre detecciones, se obtiene un mejor rendimiento en porcentaje de detecciones satisfactorias. Sin embargo, para valores de umbral altos, ambas curvas prácticamente se solapan, lo cual hace pensar que lo óptimo, sería elegir un umbral suficientemente elevado y un intervalo de tiempo entre detecciones pequeño que reduzca el tiempo de respuesta. Sin embargo esto llevaría consigo un elevado aumento de coste computacional que a partir de ahora tendremos en cuenta.

A continuación, se mostrarán los gráficos del rendimiento y del tiempo medio de detección, en función de los parámetros umbral de trayectorias supervivientes e intervalo de tiempo entre detecciones insatisfactorias. Se tendrán en cuenta dos

casos distintos de posibles muestras: Muestras afectadas por ruido y repetición de patrones y muestras afectadas únicamente por ruido.

4.3.2.2 Test sobre muestra ruidosa y con repetición de patrones

Esta es la muestra del Bolsillo HTC, que es la que tiene los problemas con la doble detección de parte de la melodía que se repite:

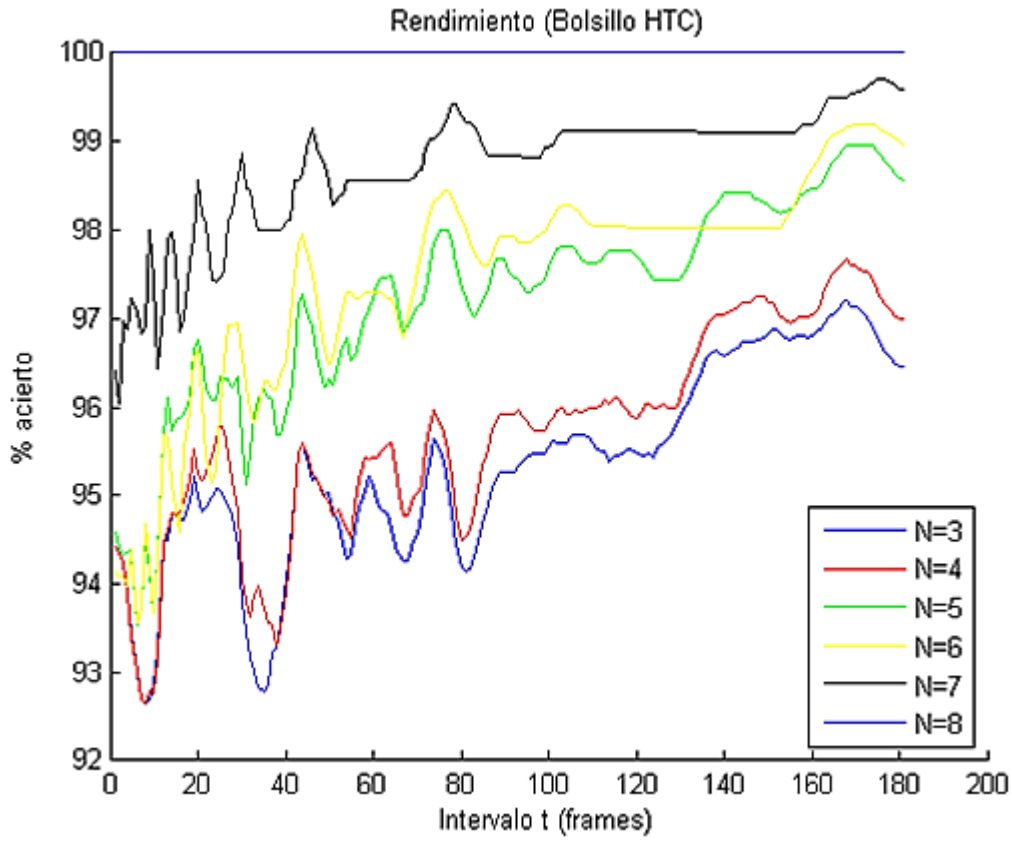


Figura 4.13: Rendimiento del sistema para audio con repetición de patrones.
(N indica el umbral de trayectorias supervivientes)

Se observa cómo afecta esta peculiaridad a valores de umbral bajo combinados con intervalos de detección pequeños. Se manifiesta con continuos vaivenes en la gráfica del rendimiento. Esto se debe a que para intervalos de tiempo pequeños, al realizar muchas detecciones en la zona conflictiva es más probable que si la siguiente detección está muy cercana a la anterior, también siga estando en la zona conflictiva. De este modo está sujeta a una probabilidad mayor de ser incorrecta, y por tanto hacer que el rendimiento sea menor.

Debido a que esta muestra es un tanto especial, se produce esa irregularidad para intervalos de tiempo bajos. Sin embargo, según aumentamos el tamaño del intervalo

entre detecciones, este rendimiento tiende a crecer. Esto se debe a que, al contrario de la explicación anterior, si este intervalo es mayor, las detecciones posteriores tienen menos riesgo de estar en la zona conflictiva. Estas consideraciones se producen para los casos en los que el umbral de trayectorias supervivientes fijado es bajo, lo que nos animará en un futuro a fijar umbrales mayores, para evitar este tipo de riesgos.

Veremos ahora cómo se comporta esta muestra en cuanto al tiempo medio de detección se refiere:

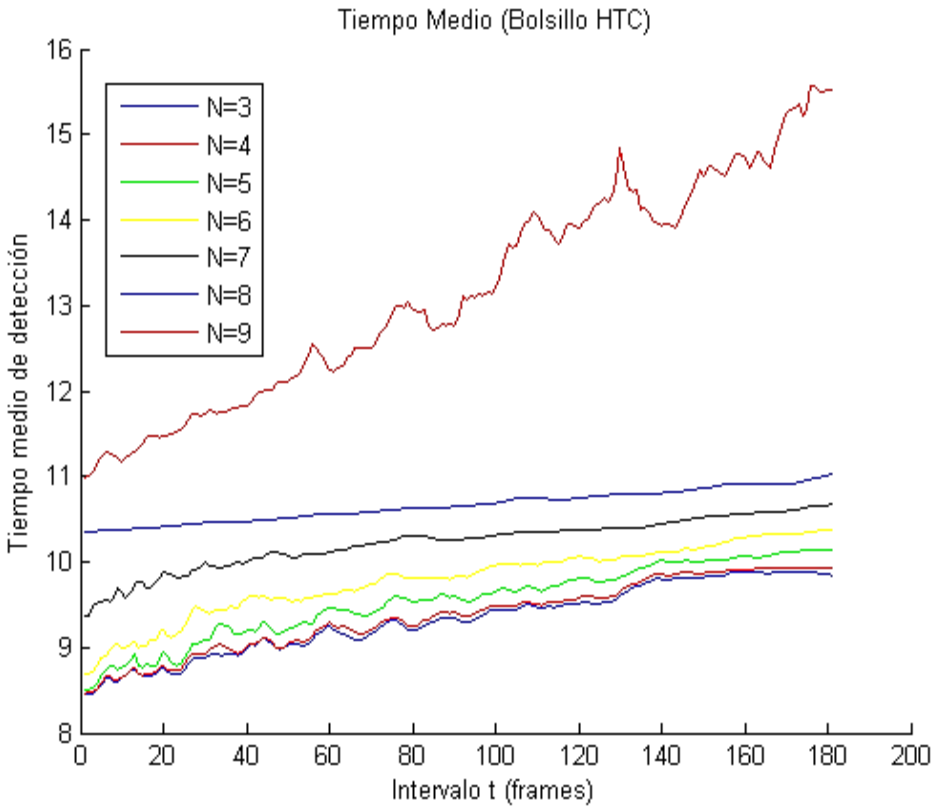


Figura 4.14: Tiempo medio de respuesta para audio con repetición de patrones.

Cuanto menor sea el umbral, menos detecciones se harán (aunque sean incorrectas), y cuanto menor sea el intervalo, más cercanas en el tiempo serán estas detecciones, por tanto el tiempo medio total será mucho menor. Se observa como los tiempos medios se disparan si se utiliza como umbral 9 para el número de trayectorias supervivientes, es decir, 9 o 10 trayectorias.

4.3.2.3 Test sobre muestras ruidosas únicamente

Dentro de este grupo de muestras se encuentran las otras 3 con las que estamos trabajando que tienen niveles de ruido considerables, y que no se han podido corregir usando simplemente CMNV. Las gráficas que se muestran a continuación corresponden con la muestra del Bolso iPhone, que de las tres, es la que peor rendimiento tenía, y por tanto será en la cual se observarán mejor las características que se desean mostrar:

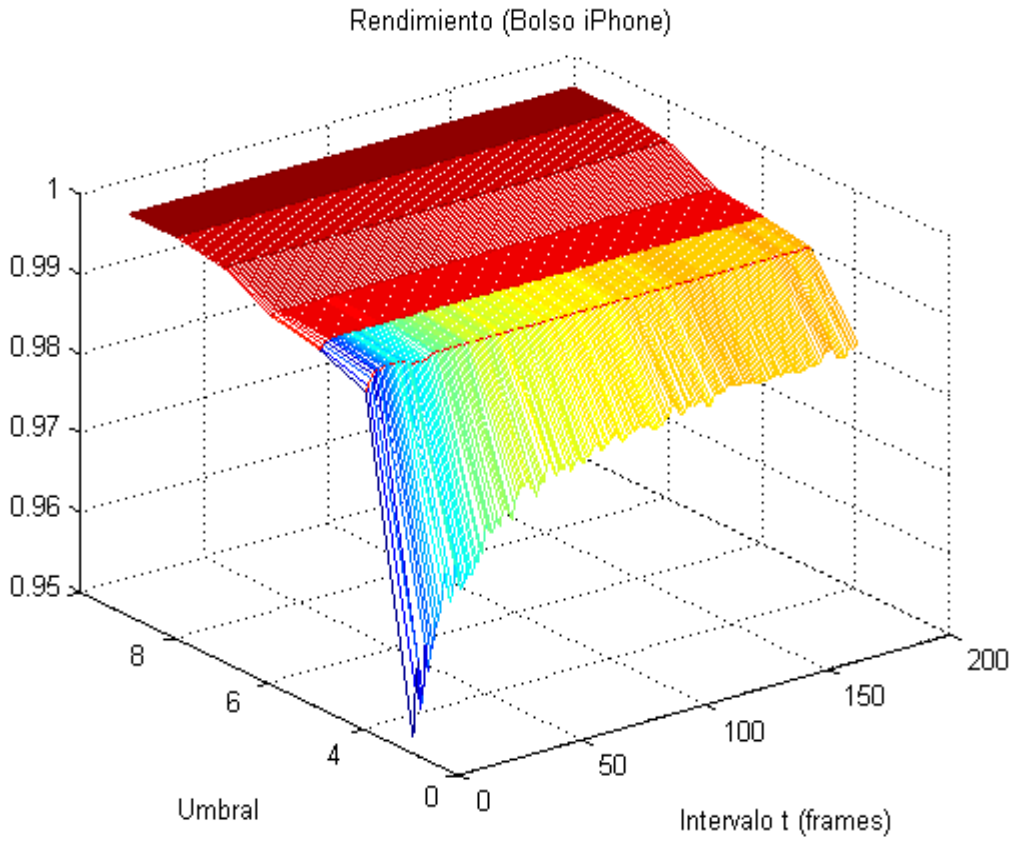


Figura 4.15: Rendimiento del sistema para muestra afectada por ruido únicamente.

En este caso, se ve como el rendimiento sigue una tendencia creciente, al contrario que para la muestra con repetición de patrones, que seguía una trayectoria más caótica. Esta trayectoria ascendente es más significativa para un valor de umbral muy bajo. Sin embargo para valores superiores es casi lineal, esto es, un aumento en el intervalo de tiempo entre detecciones, no supone un aumento significativo en la probabilidad de acierto.

Veremos ahora que ocurre con el tiempo total de respuesta del sistema para este tipo de muestras:

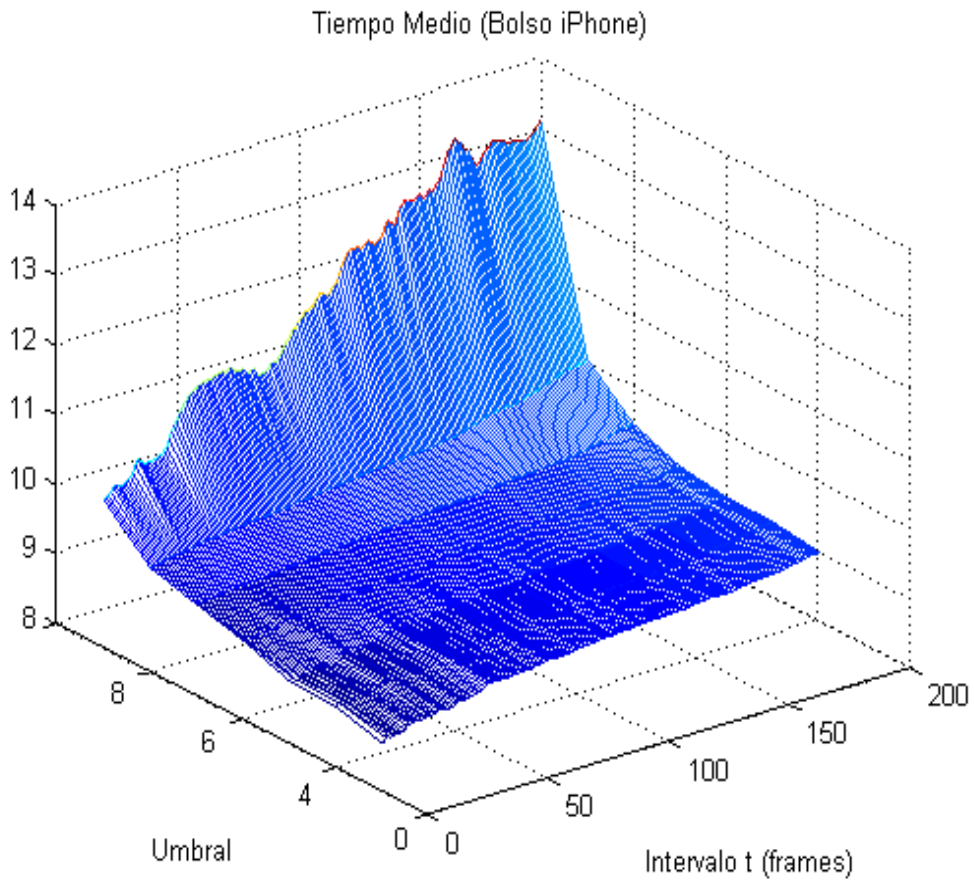


Figura 4.16: Tiempo medio de respuesta del sistema (en segundos) para muestra afectada por ruido únicamente.

El tiempo medio de respuesta del sistema se incrementa al aumentar el umbral y el intervalo de forma casi lineal para todos los umbrales excepto para el umbral 9. A la vista de las pruebas realizadas, se extraerán una serie de conclusiones.

4.3.2.4 Conclusiones respecto a parámetros característicos

1. **Respecto al Umbral de trayectorias supervivientes**

   1.1 Para umbrales bajos, las muestras más vulnerables, es decir, muestras con patrones repetidos o muestras con mucho ruido, sufren un claro descenso del rendimiento.

   1.2 Estableciendo el umbral en 8, todas las muestras, ya sean más o menos ruidosas o con patrones repetidos, alcanzan el rendimiento óptimo.

   1.3 Cuanto mayor es el umbral, mayor es el tiempo de respuesta del sistema.

2. **Respecto al intervalo de tiempo entre detecciones**

    2.1 En muestras muy ruidosas o con patrones repetidos, fijar un intervalo mayor, ayuda a que el rendimiento del sistema mejore.
    2.2 En muestras menos ruidosas, el rendimiento de sistema no mejora significativamente al aumentar dicho intervalo.
    2.3 El tiempo medio de detección siempre es mayor al aumentar el valor del intervalo.

3. **En general**

    3.1 La solución ideal sería pues fijar un umbral suficientemente elevado, que nos garantizara un rendimiento óptimo, y un intervalo de tiempo suficientemente pequeño, que no hiciera aumentar en exceso el tiempo de respuesta.
    3.2 Tendremos que tener en cuenta que el coste computacional, será un factor importante a tener en cuenta puesto que puede ralentizar el sistema, si no se elige un intervalo adecuado.

Así pues, si fijamos el valor del umbral en 8, estos son los tiempos medios de detección para todas las muestras en función del valor del intervalo escogido:

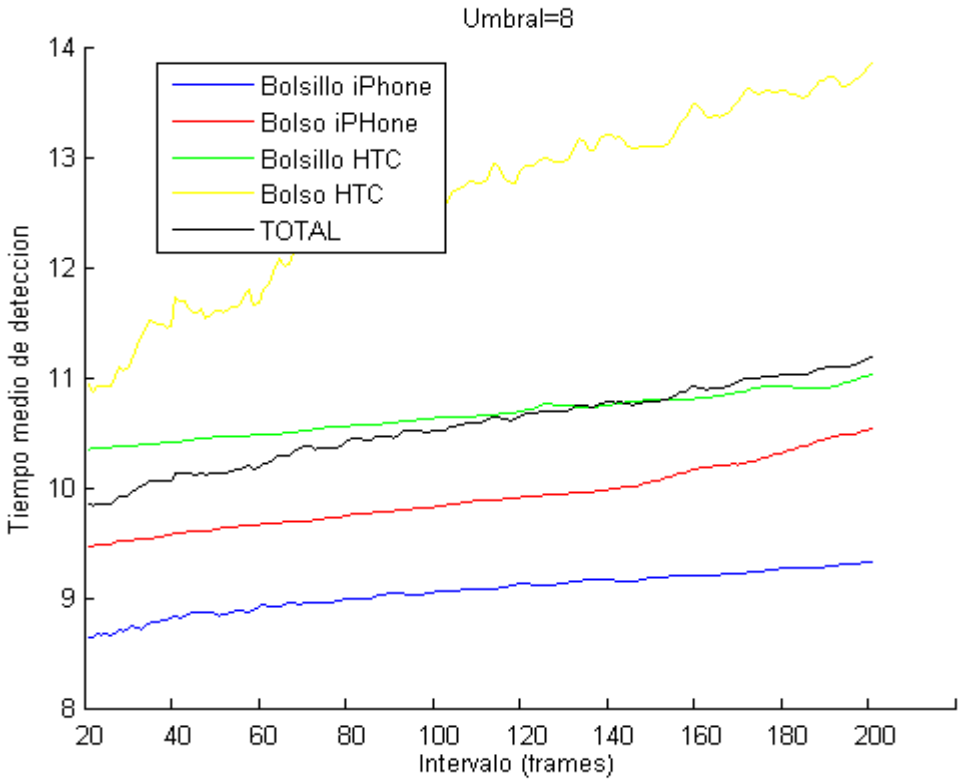


Figura 4.17: Tiempo medio de respuesta para umbral óptimo (8).

La muestra que más tiempo medio de detección requiere para una detección precisa, es la del Bolso HTC, cosa que indica que es la que más ruido tiene, y por tanto, la que requiere de un mayor número de detecciones extra para cumplir los requisitos de calidad.

4.3.2.5 Análisis del coste computacional

El coste computacional se define como la cantidad de operaciones que tiene que hacer el programa para calcular el resultado final. En este caso, este coste siempre irá estrechamente relacionado con el coste de detección. El siguiente gráfico muestra la media de detecciones extra que tiene que hacer el algoritmo de análisis de trayectorias para todo el banco de muestras:

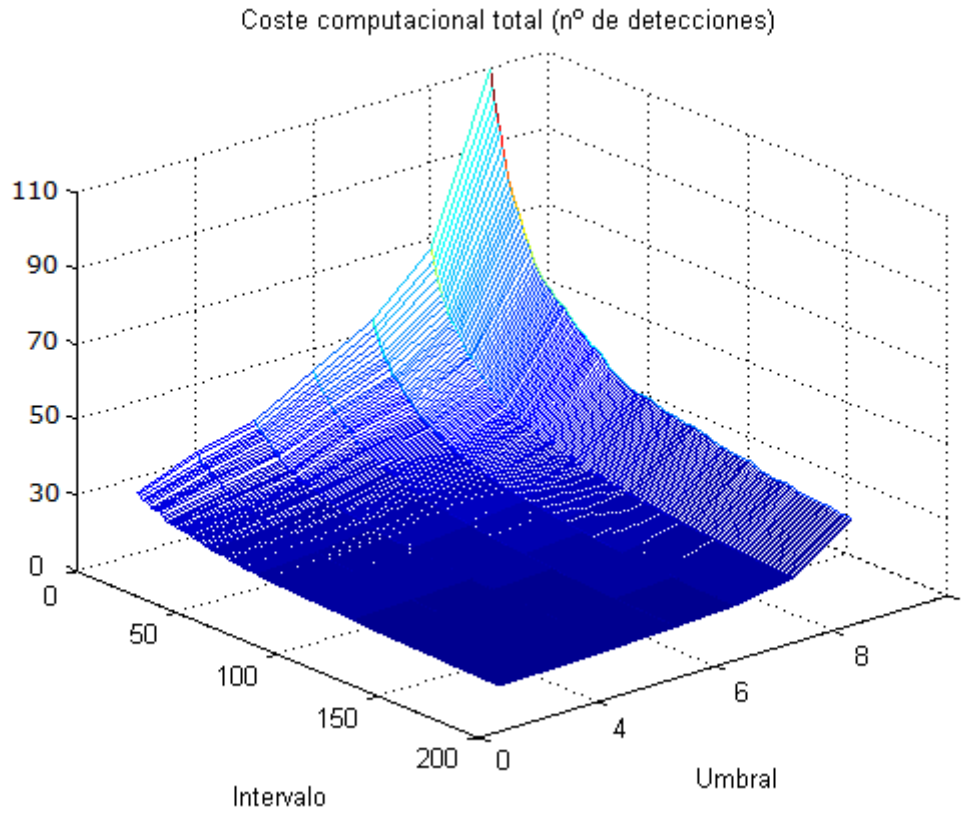


Figura 4.18: Coste computacional del AAT para todas las muestras.

El gráfico pone de manifiesto cómo el coste computacional es mayor para valores de umbral altos (con los que se consigue un mayor rendimiento) y valores de intervalo más pequeños (con los que se consigue un tiempo medio menor). Es decir, al aumentar las prestaciones del sistema, esto se traduce en un aumento del coste computacional. Si se establece como prioridad el rendimiento del sistema , se tendrá que establecer un valor del intervalo de tiempo entre detecciones que sin provocar un

excesivo aumento del tiempo medio de detección, mantenga los niveles de coste computacional relativamente bajos. Por tanto, fijamos el valor del umbral de número de trayectorias en 8 (umbral óptimo) y observamos cómo se comporta el coste computacional.

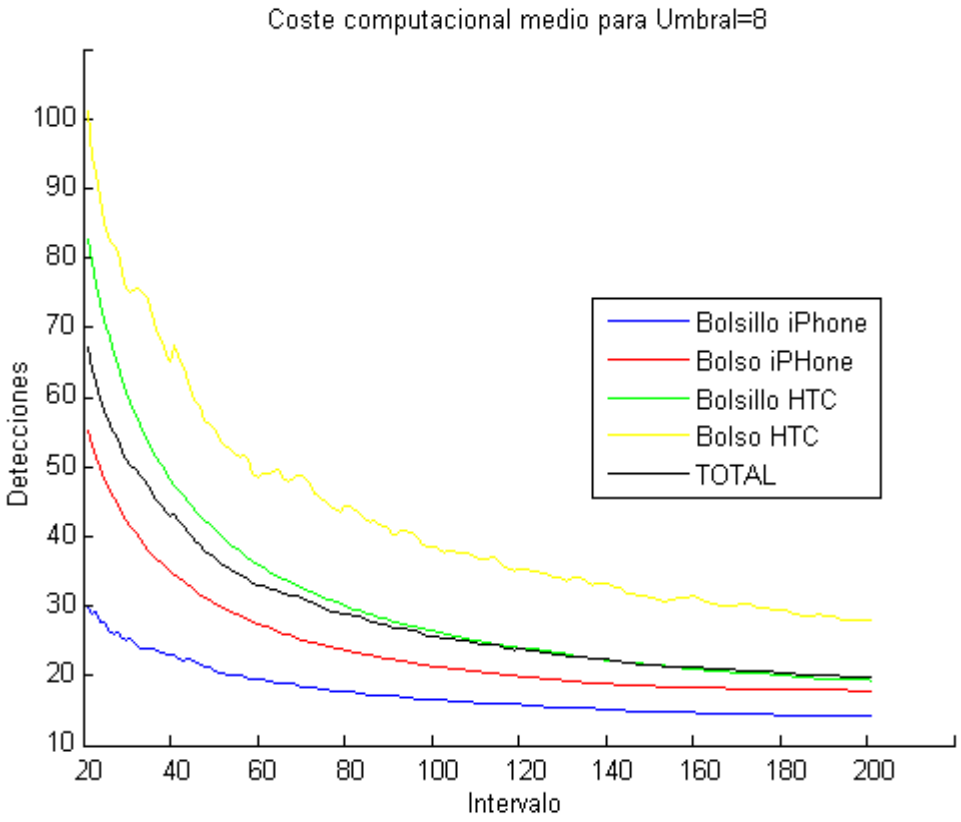


Figura 4.19: Coste computacional para umbral de trayectorias óptimo (8).

La gráfica tiene una forma exponencial decreciente, y para valores superiores a 140 frames, el descenso en el coste computacional es prácticamente nulo. Observando la gráfica del tiempo total de respuesta del sistema para umbral 8 que obtuvimos anteriormente, se puede ver como esta sigue una tendencia siempre ascendente, pero a partir de un intervalo cercano a 140 frames la pendiente aumenta. Esto nos lleva a pensar que fijar un intervalo de 140 es la solución más óptima para este parámetro de rendimiento.

Por tanto recapitulando el trabajo realizado en esta sección, los valores más adecuados tanto en rendimiento como en coste computacional y tiempo de respuesta se obtienen para los siguientes valores de los parámetros característicos:

- Número de trayectorias supervivientes: 8
- Distancia mínima entre trayectorias supervivientes: 1 segundo
- Intervalo de tiempo entre detecciones: 140 frames (será nuevamente evaluado cuando se optimice el algoritmo).

4.3.3 Optimización mediante estimación de ruido

Hasta ahora se ha desarrollado un algoritmo de robustez contra el ruido genérico, es decir, ataca el problema de la misma forma independientemente del nivel de ruido de la muestra. El algoritmo necesita de la obtención de muchas detecciones para devolver un valor fiable, y eso, en coste computacional es caro. Su utilización debería ser más necesario para las muestras más ruidosas ya que como se demostró anteriormente, para muestras poco ruidosas (regazo) se obtenían resultados óptimos sin utilizar el algoritmo. La cuestión ahora tratará de cómo saber si una muestra es más o menos ruidosa para decidir si es útil aplicar el algoritmo o si vamos a obtener resultados óptimos sin utilizarlo, o utilizándolo pero no al 100% de su capacidad.

Como se explicó en la sección que describe el funcionamiento del programa básico, las detecciones se calculan como la distancia mínima entre dos vectores de coeficientes MFCC. Este valor de distancia mínima es un indicador de la diferencia entre los dos audios, y por tanto para muestras muy contaminadas por ruido debería ser mayor. Esta idea nos lleva a analizar el comportamiento de este valor a lo largo de la muestra, diferenciando entre zonas de error y zonas de acierto. En la siguiente tabla se muestran tanto las medias de distancias de detección (en zona de acierto y zona de error) así como los extremos conflictivos, distancia mínima en zona de error y distancia máxima en zona de acierto:

	media error	**media acierto**	**min. error**	**max. acierto**
Bolsillo iPhone	230,78	187,43	217,97	231,87
Bolso iPhone	229,89	204,51	209,34	242,83
Bolsillo HTC	208,61	190,46	199,62	233,73
Bolso HTC	231,32	203,62	218,92	238,09
Media	**225,15**	**196,505**		

Tabla 4.23: Distancia de detección media en zona de acierto y zona de error

Se puede observar una clara diferencia entre los valores medios para zonas de acierto y zonas de error. También se observa un valor significativo, el hecho de que el valor mínimo de distancia en zona de error sea mayor que la media en zona de acierto. Esto indica que se podrían haber considerado satisfactorias muchas detecciones sin haber utilizado el algoritmo de análisis de trayectorias. Por ejemplo, toda aquella cuya distancia mínima fuera inferior a 199, que corresponde con el valor mínimo de todas las detecciones erróneas.

Para tener una visión más global de todo esto, se mostrará a continuación unos histogramas que contienen toda la información de distancias mínimas para todas las muestras analizadas, separadas en zona de acierto y zona de error:

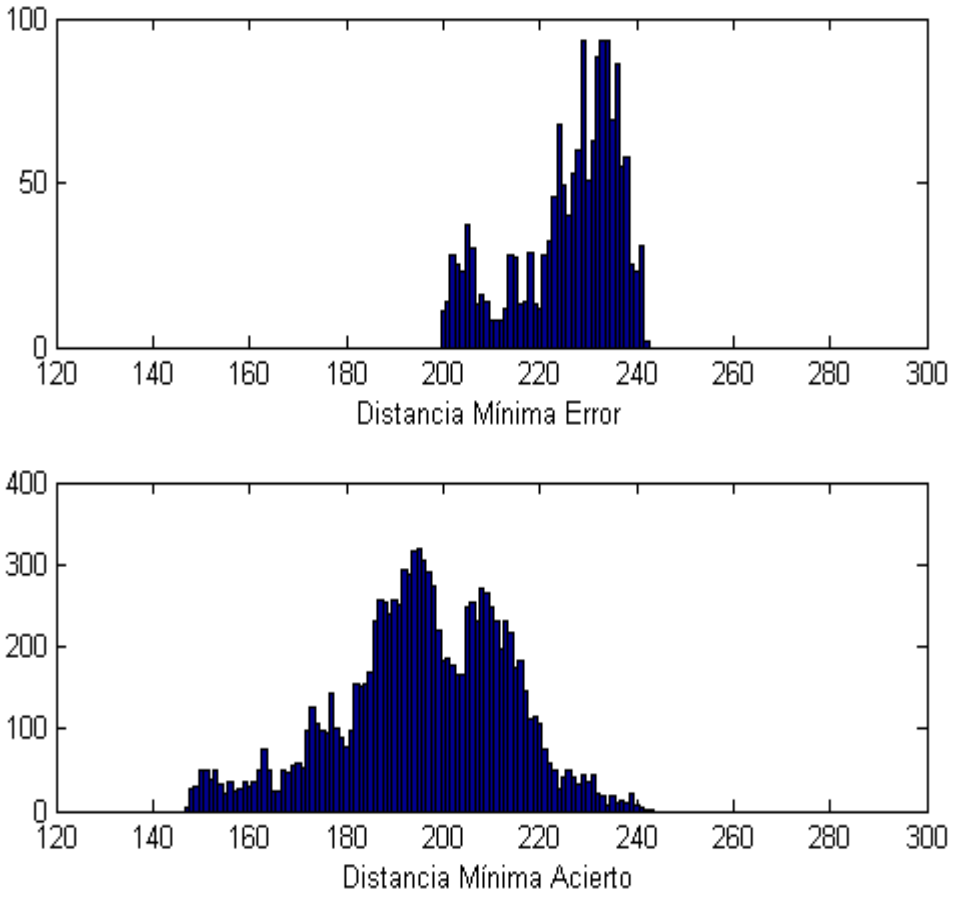


Figura 4.20: Histograma de distancias mínimas en zonas de error y acierto.

La gráfica muestra cómo se distribuye la probabilidad de la distancia para casos de acierto, como para casos de error. Como se ve, hay una gran zona de solapamiento entre ambas que será la zona más compleja de analizar. Para distancias en las que no tenemos ninguna detección errónea, podremos asegurar, al menos para estas muestras, que la detección será correcta sin tener que aplicar el algoritmo de robustez. Para las distancias para las que no hay aciertos y solo errores, dicha detección será considerada errónea y se procederá a realizar otra un intervalo de tiempo posterior. Para evaluar todo esto, es necesario analizar las funciones distribución de probabilidad de ambas variables.

Función Distribución de probabilidad discreta:

$$F(x) = P(X < x) = \sum_{k=-\infty}^{x} f(k)$$

Donde f(x) es la función densidad de probabilidad de la misma, representada mediante los histogramas (sin normalizar).

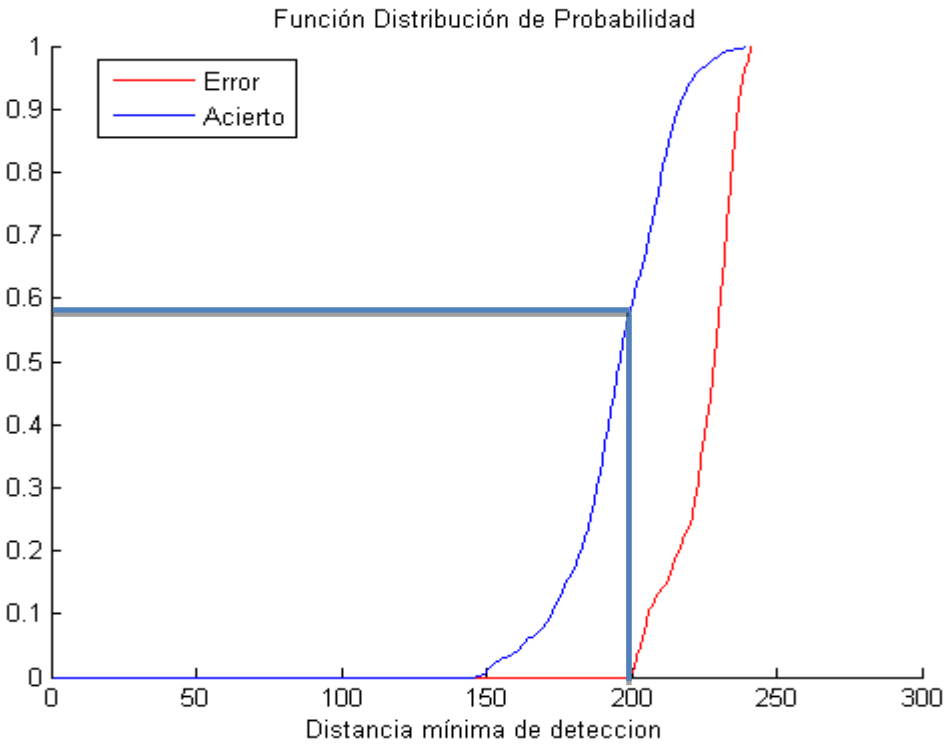


Figura 4.21a: Función Distribución de Probabilidad de distancia de detección.

Como se observa, casi el 60% de las detecciones serían seguras sin utilizar el algoritmo. Ahora la cuestión será, para el resto, calibrar una curva para que mida la intensidad con la que debemos aplicar el algoritmo para funcionar con garantías. Dicha intensidad se definirá como la longitud de la trayectoria a analizar, siendo 10 el máximo y 1, (o la no utilización del mismo) el mínimo. Por ejemplo, no será necesario aplicar el algoritmo con la misma intensidad si obtenemos una detección con distancia 201 que una con distancia 220, puesto que para la primera, la probabilidad de error con respecto a acierto es mínima y viceversa. A continuación se mostrará una gráfica diferente de la función distribución de probabilidad de ambas.

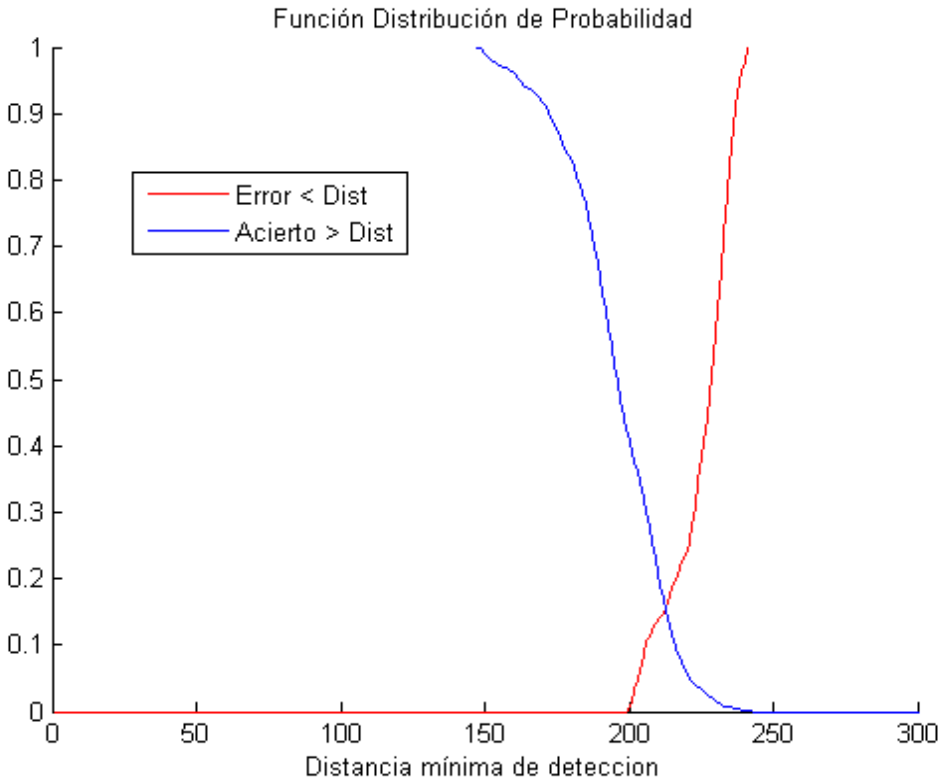


Figura 4.21b: Función Distribución de Probabilidad de distancia de detección (2).

En este caso se ha representado la distribución de probabilidad inversa para la variable aleatoria acierto. Esto quiere decir, que los valores de la curva representan la probabilidad de que se produzca un acierto en la detección para una distancia mayor a x:

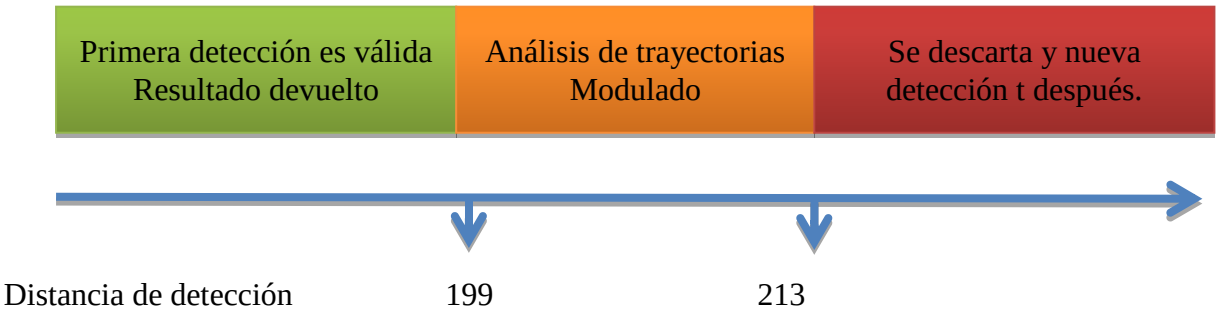
$$F_{acierto,inv}(x) = P(acierto\ |X > x) = 1 - \sum_{k=-\infty}^{x} f(k)$$

De este modo se obtienen dos valores de distancia significativos: distancia de seguridad y distancia de rechazo:

- **Distancia de seguridad**: Corresponde con el valor máximo de distancia para el cual la FDP de la variable error deja de ser cero. Este valor es igual a 199.

- **Distancia de rechazo:** Corresponde con el valor de distancia donde se cruzan ambas FDP y significa que a partir del mismo, la probabilidad de cometer un error supera a la de acertar. Este valor equivale a 213.

Con estos dos valores fijados, nos queda un conjunto de distancias separado en tres intervalos en los cuales se actuará de diferente manera:



4.3.3.1 Modulación de intensidad del algoritmo.

Según ha sido diseñado, se define la intensidad con la que se aplica el algoritmo como la longitud de las trayectorias que se analizan. Es decir, el algoritmo funciona a máxima intensidad cuando las trayectorias analizadas tienen longitud 10 y a mínima intensidad cuando las trayectorias tienen longitud 2. Como no podía ser de otra manera, esta intensidad de uso está asociada al coste computacional, por lo que con esta modulación lo que se pretende es usarlo de forma eficiente, para conseguir resultados óptimos, reduciendo el coste computacional.

El objetivo será, por tanto, aplicar unos umbrales de potencia para la zona en la que las FDPs de acierto y error se superponen (siendo siempre mayor el valor de la FDP de acierto, entre 199 y 213).

La siguiente gráfica, muestra al detalle esta zona de las FDP. Se representa la resta de ambas funciones distribución de tal modo que el valor sea máximo para una distancia de 199, y cero para una distancia de 213. Posteriormente se ha cuantificado la FDP resultante en los niveles de intensidad que aplicará el algoritmo:

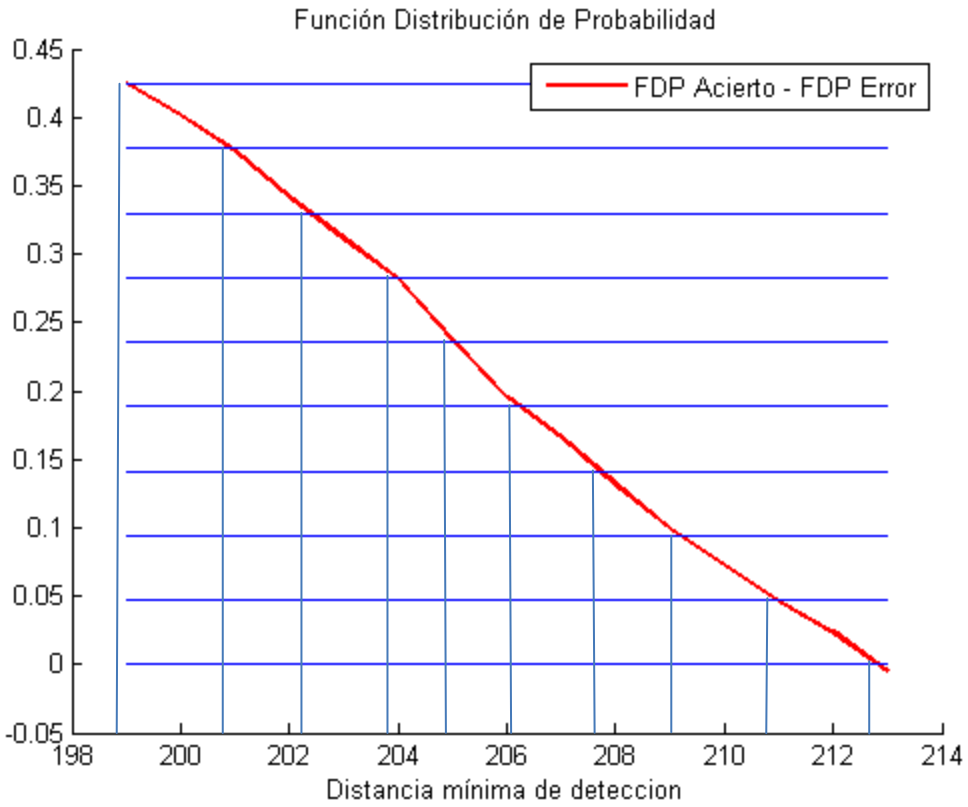


Figura 4.22: Modulación de la intensidad del algoritmo.

De la gráfica se obtienen los siguientes valores para los umbrales de potencia:

Potencia del algoritmo y distancia Umbral									
Nº Tray.	2	3	4	5	6	7	8	9	10
Umbral	199	200.8	202.27	203.82	204.9	206.12	207.65	209.06	210.8
	200.8	202.27	203.82	204.9	206.12	207.65	209.06	210.8	213

Tabla 4.24: Umbrales del AAT modulado.

El funcionamiento del algoritmo con el módulo de estimación de ruido funcionará según el siguiente esquema:

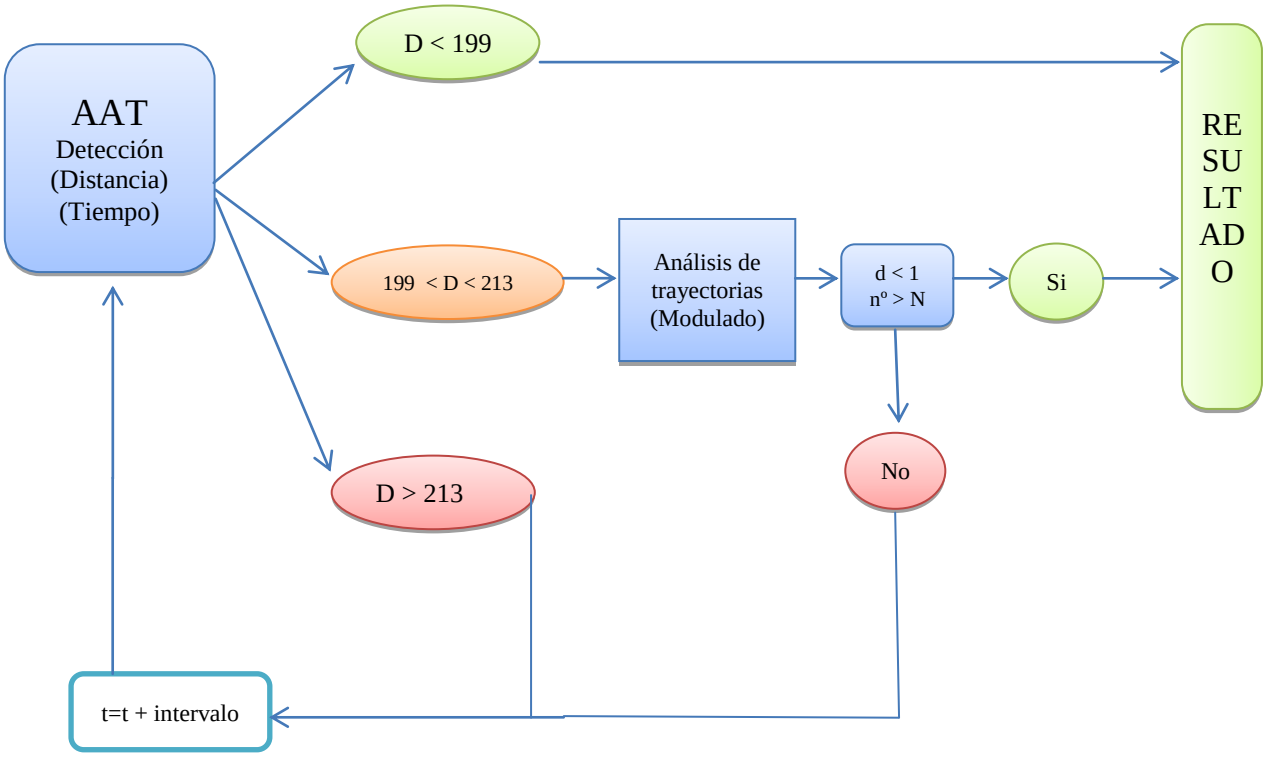


Figura 4.23: Lógica del AAT modulado.

Esta nueva funcionalidad, hace que el algoritmo de análisis de trayectorias se comporte ligeramente diferente a como lo hacía antes. Esta diferencia consiste en que si a la salida del mismo, el resultado no es satisfactorio, se vuelve a estimar el nivel de ruido antes de volver a calcular las trayectorias. Anteriormente, por el contrario, el algoritmo ante una salida no satisfactoria, volvía a calcular todas las trayectorias. Esto provocará un descenso mayor del coste computacional, como veremos más adelante.

Se ha hecho un análisis de los tres factores fundamentales: El rendimiento, el coste computacional y el tiempo de respuesta. Para ello, como se hizo anteriormente, se han calculado en función del parámetro margen entre detecciones no satisfactorias. Los valores umbrales del algoritmo análisis de trayectorias utilizados son lo óptimos calculados en la sección anterior. Estos son: 8 para el número de trayectorias supervivientes y 1 segundo para la distancia media entre las trayectorias supervivientes. Con todo esto, los resultados obtenidos en cuanto al rendimiento son los siguientes:

	Bolso iPhone	**Bolsillo iPhone**	**Bolso HTC**	**Bolsillo HTC**
Rendimiento	100%	100%	100%	100%
Modulación [199-213]				
Margen (t) [20:180]frames				

Como muestra la tabla, el rendimiento es óptimo en todos los casos, como ya sucediera sin la estimación de ruido, por lo que podemos decir que el uso de este algoritmo, no reduce la calidad del sistema.

Como advertencia hay que decir que en el desarrollo del sistema, este se ha ajustado completamente a los datos de test (únicos de que disponíamos) por lo que estos resultados esperamos que sean optimistas y queda pendiente analizar la generalización a datos distintos de los aquí analizados. En cualquier caso, las técnicas que se han empleado posiblemente pudiesen ser empleadas, si bien es posible que fuese necesario adaptar los pocos parámetros ajustables de las mismas.

Ahora veremos que sucede con el coste computacional (expresado en número de detecciones necesarias para dar un resultado final satisfactorio al 100%). La siguiente gráfica muestra el número de detecciones medio para toda la muestra en función del intervalo de tiempo que se establezca.

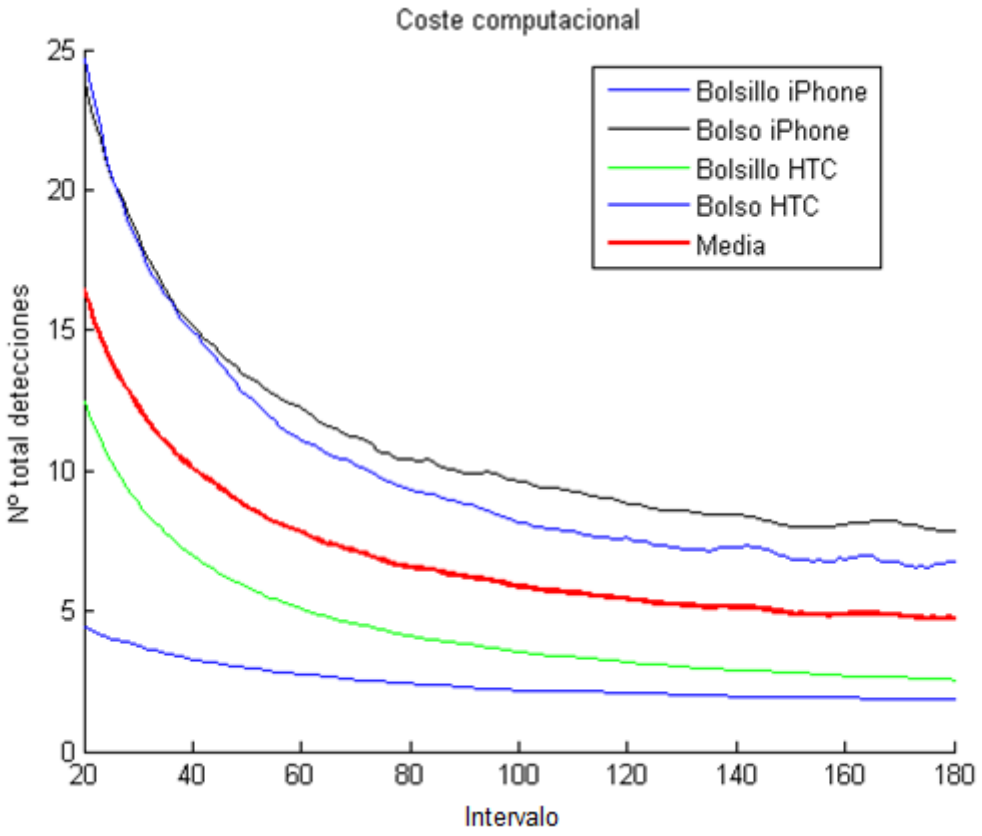


Figura 4.24: Coste computacional con optimización.

La gráfica tiene forma de exponencial descendente, al igual que la que se obtenía anteriormente. La gran diferencia reside en el que el número de detecciones es considerablemente menor, por ejemplo para un margen de 140, ahora estamos

rondando las 5 detecciones necesarias mientras que antes, superábamos claramente las 20. Por tanto, queda demostrado que la implantación de la estimación de ruido inicial, trae consigo una reducción muy importante en cuanto al coste computacional, sin afectar al rendimiento.

Veremos ahora como afecta la implantación del módulo de estimación al tiempo medio necesario para una detección. Se muestra una comparativa entre dicho tiempo con y sin estimación de ruido:

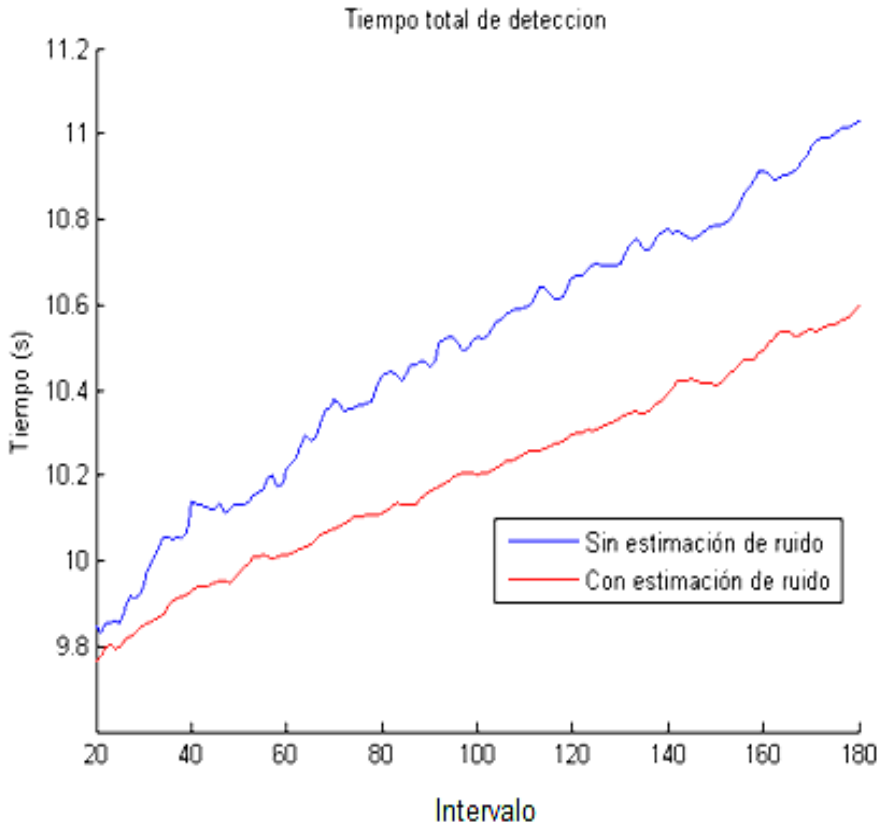


Figura 4.25: Comparativa del tiempo total de detección con y sin estimación de ruido.

Como muestran las gráficas, al aplicar la estimación de ruido, se obtiene un tiempo total de detección ligeramente menor. Esto se explica por las siguientes razones:

- En primer lugar, las detecciones consideradas correctas inicialmente no tienen que ser procesadas por el algoritmo de análisis de trayectorias, y por tanto no se produce el retraso de 10 frames (160 ms).
- Por otro lado, las que si son procesadas, lo hacen durante menos frames-puesto que ahora el algoritmo está modulado, y en el peor de los casos será el mismo tiempo que antes.

- Por último, las detecciones consideradas incorrectas desde un principio, deberían hacer que se incrementara el tiempo total, ya que llevarían consigo detecciones posteriores. Sin embargo esto no supone una diferencia excesiva porque un gran porcentaje de ellas, hubieran sido igualmente desechadas por el algoritmo de análisis de trayectorias. Por tanto, este ligero aumento se ve ampliamente compensado por los otros dos decrementos, por lo que se consigue un resultado global muy satisfactorio.

4.3.3.2 Margen de seguridad en estimación previa

El análisis de las funciones densidad y distribución de probabilidad del parámetro distancia de detección realizado anteriormente, indicaba que para distancias inferiores a 199, no se registraba ningún caso de detección errónea. Estas pruebas han sido realizadas sobre muestras bastante exigentes para el sistema en lo que a nivel de ruido y distorsión se refiere, por lo que estos valores deberían ser altamente significativos. Sin embargo, para un trabajo más conservador, en el que no demos cabida a posibles detecciones incorrectas identificadas como correctas, se estudia el establecimiento de un margen de seguridad. Este margen de seguridad conlleva un decremento en las prestaciones del sistema, tanto en tiempo medio de respuesta como en coste computacional, pero nos aporta mayor seguridad de que ninguna detección errónea escapará al filtro. El proceder es muy sencillo, simplemente se establece el margen inferior en un valor menor y se recalculan los intervalos de actuación del algoritmo de análisis de trayectorias.

En este caso se fija el umbral a partir del cual empieza a funcionar el algoritmo en 190 en lugar de 199, y se hace una modulación lineal hasta 213. Se ha decidido realizar la modulación lineal por simplicidad, y basándonos en la siguiente idea: Estableciendo el margen de seguridad, se modula la intensidad del algoritmo desde una distancia menor, por tanto, para la franja de distancias conflictivas definidas anteriormente, la intensidad empleada ahora será mayor. Si antes con menos intensidad los resultados fueron óptimos, en este caso lo deberían seguir siendo. Por tanto, la modulación lineal que proponemos ahora, en ningún caso influiría negativamente en la precisión del sistema.

Como era de esperar, para esta nueva situación, se repiten los resultados de precisión del apartando anterior, obteniéndose en 100% en todos los casos:

	Bolso iPhone	Bolsillo iPhone	Bolso HTC	Bolsillo HTC

Rendimiento Modulación [190-213] Margen (t) [20:180]frames	100%	100%	100%	100%

Sin embargo, tanto el coste computacional en número de detecciones necesarias, como el tiempo medio total necesario, se verán afectados, puesto que establecer el margen en una distancia inferior, supondrá el procesado de muchas detecciones que antes se consideraban correctas en primera instancia y que ahora deberán ser analizadas por el algoritmo de análisis de trayectorias. A continuación veremos cómo afecta la implantación de este margen de seguridad tanto al coste computacional como al tiempo medio de detección:

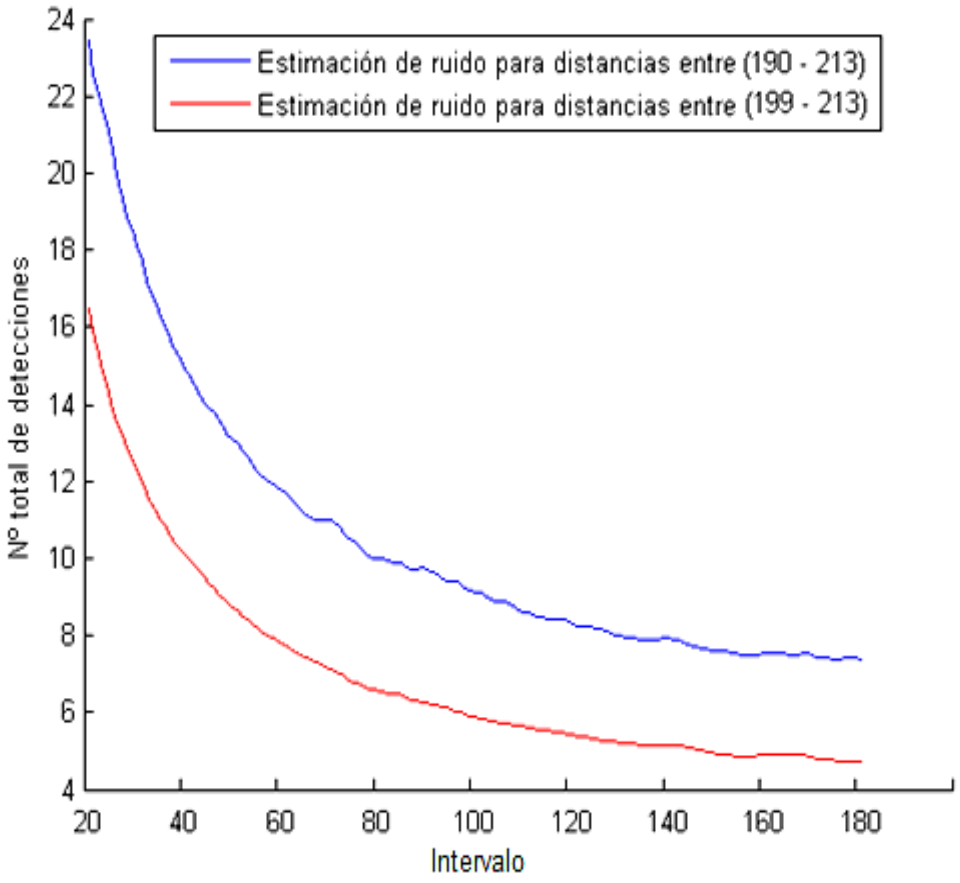


Figura 4.26: Coste computacional con modulación [190-213] vs [199-213].

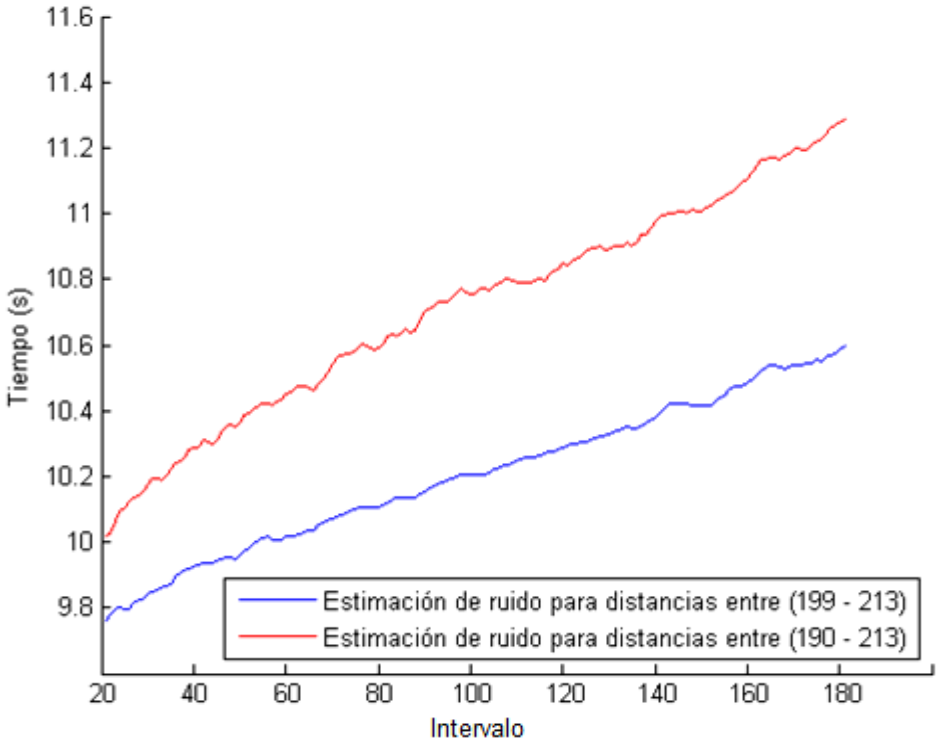


Figura 4.27: Tiempo medio de respuesta con modulación [190-213] vs [199-213].

Para un intervalo de 140 frames (intervalo establecido como óptimo anteriormente), el coste computacional asciende hasta las 7.92 detecciones de media, mientras que antes se situaba por debajo de las 6. Para el tiempo total medio, ahora tenemos valores de 10.97 segundos, mientras que antes se situaba en torno a los 10.30 segundos. Evidentemente, supone un deterioro en cuanto a las prestaciones del sistema, sin embargo la fiabilidad del sistema aumenta para todo tipo de situaciones.

4.3.3.3 Conclusiones respecto a la optimización del algoritmo.

Como ha quedado claramente de manifiesto, la inclusión de este pre-análisis basado en la primera detección, nos otorga beneficios muy importantes tanto en coste computacional, como en tiempo total de detección. Todo ello sin perjudicar al parámetro más importante de todos, la calidad de la detección, la cual sigue siendo óptima para todo el banco de muestras.

Se consideran unos resultados muy satisfactorios para cumplir los objetivos del proyecto, puesto que partíamos de valores muy bajos en cuanto a probabilidad de acierto, y ahora nos encontramos con un rendimiento óptimo para todas las muestras analizadas, muchas de las cuales tienen un nivel de ruido bastante considerable,

siendo poco frecuente en muestras de este tipo. Es por ello que se ha conseguido un nivel de robustez frente al ruido bastante importante.

Aparte de esto, se ha tratado de optimizar el sistema de forma que no sea muy costoso computacionalmente conseguir esta robustez. Es en este ámbito donde más se puede profundizar, aplicando por ejemplo, técnicas de *clustering*, aunque ya no entra en el propósito de este proyecto, cuyo principal objetivo ha sido cumplido con un grado de satisfacción bastante elevado.

Capítulo 5:

Conclusiones y trabajo futuro

5.1 Conclusiones

Los objetivos fijados en un principio consistían en conseguir un sistema que fuera capaz de realizar detecciones con una precisión de decenas de milisegundos y con una tasa de acierto lo más alta posible, teniendo en cuenta que posiblemente fuera imposible realizar detecciones precisas para audios en muy malas condiciones.

El punto de partida hacía pensar que se podrían lograr mejoras significativas, puesto que se partía de rendimientos un tanto pobres para muestras de relativa calidad. Estas muestras contenían partes muy corrompidas pero la mayor parte de ellas tenían niveles de ruido normales, siendo claramente diferenciado el audio de interés en todas ellas.

El salto cualitativo más importante en cuanto a precisión y tasa de acierto en las detecciones vino de la mano de la introducción del algoritmo de normalización de media y varianza para coeficientes espectrales CMVN. Tras su incorporación, el porcentaje de éxito en detecciones aumento muy considerablemente, siendo mayor para aquellas muestras que tenían un rendimiento peor. La implantación de la doble normalización CMVN, que no estaba prevista en un principio, supuso un incremento adicional del rendimiento del sistema para todo el banco de muestras, alcanzando ya niveles óptimos (ausencia de errores) para las muestras poco ruidosas y cercanos para las muestras de peor calidad.

En este punto se finalizó el proceso de incluir mejoras basadas en algoritmos ya estudiados y de eficacia contrastada, y comenzó el análisis más exhaustivo del comportamiento del sistema, para poder desarrollar algoritmos propios que pudieran aportar alguna mejora extra. Se llegó a la conclusión de que los porcentajes de detecciones erróneas que se estaban obteniendo, venían propiciados por muestras totalmente distorsionadas en las que resultaba imposible distinguir el audio que se intentaba sincronizar. El trabajo a partir de este momento estuvo encaminado a saber detectar este tipo de situaciones y poder corregirlas extendiendo el tiempo de grabación hasta conseguir una sincronización segura. Para ello se desarrolló el algoritmo de análisis de trayectorias temporales, con el cual se consiguió establecer unos márgenes de fiabilidad para las detecciones. De este modo, para todo el banco de muestras utilizado, se obtenían niveles óptimos en cuanto a porcentaje de acierto (100%), a costa de un coste computacional y un tiempo de respuesta superior. Debido al incremento de estos dos factores negativos, la parte final del proyecto estuvo encaminada a optimizar el algoritmo de análisis de trayectorias para reducir el coste computacional y el tiempo de respuesta, sin que ello afectara al rendimiento del sistema.

Con todo esto, se consideran ampliamente satisfechos los objetivos fijados para este proyecto, consiguiendo un sistema muy robusto frente al ruido en prácticamente todo tipo de situaciones, y que es capaz de detectar situaciones en las que por la degradación extrema del audio no es posible realizar la sincronización o en las que por la propia naturaleza del audio se incluyen repeticiones que dan lugar a detecciones dudosas. En esos casos se ha optado por extender el tiempo de detección hasta que sea posible realizar una sincronización satisfactoria.

5.2 Trabajo futuro

A pesar de la consecución de los objetivos fijados, y de la fiabilidad del sistema, existen importantes vías de mejora, así como de investigación respecto a varios aspectos fundamentales del proyecto:

Por un lado, respecto a la obtención de vectores de coeficientes característicos, este sistema está basado en coeficientes MFCCs, cuyo uso es muy común en todo tipo de sistemas de procesamiento de audio en tiempo real. Sin embargo, existen otros métodos de obtención de coeficientes característicos más complejos, cuyo rendimiento en condiciones ruidosas puede llegar a ser mejor (SSCH) [22]. El estudio de la implantación de este método podría ser objeto de una futura investigación.

Por otro lado, existen técnicas de normalización de vectores de coeficientes más avanzadas que las usadas en este proyecto (CSN, evolución de la utilizada CMVN) que también podrían ser consideradas en un futuro para mejorar el sistema.

Uno de los aspectos que admite un mayor margen de mejora es sin duda el algoritmo de búsqueda de coeficientes. El utilizado en este proyecto realiza una búsqueda lineal, lo que significa un tiempo de procesamiento elevado al realizar comparaciones entre todos los pares de vectores de coeficientes. La principal mejora vendría propiciada por la implementación de algoritmos de *clustering* que organizaran los vectores de coeficientes mediante la técnica de vecinos cercanos, de modo que la búsqueda fuera mucho más rápida. Este aspecto es esencial si queremos conseguir un algoritmo con buenas condiciones de escalabilidad y cuyos tiempos de búsqueda en los datos no escalen de forma lineal con la longitud de los datos en los que se busca.

Finalmente, hay que hacer mención al algoritmo novedoso desarrollado en este proyecto, el algoritmo de análisis de trayectorias. Se trata de un prototipo de método de estudio de fiabilidad, que ha sido introducido y posteriormente optimizado pero que aún puede ser optimizado mucho más mediante el estudio de sus parámetros característicos.

Referencias

[1] Shazam - *Music Discovery, Charts & Song Lyrics.* [online].[Accessed on 10/Apr/2015]. Available in web: http://www.shazam.com/company

[2] MIREX HOME - *Music Information Retrival Exchange, nema.lis.illinois.edu* [online].[Accessed on 12/Apr/2015]. Available in web: http://www.music-ir.org/mirex/wiki/MIREX_HOME.

[3] David Gerhard, *Audio Signal Classification: History and Current Techniques,* Technical Report *TR-CS 2003-07 November, 2003.* Available in web: http://www2.cs.uregina.ca/~gerhard/publications/TRdbg-Audio.pdf

[4] Tong Zhang and Jay C.-C. Kuo. Hierarchical classification of audio data for acrhiving and retrieving. *In International Conference on Acoustics, Speech and Signal Processing*, volume VI, pages 3001–3004. IEEE, 1999.

[5] Audible Magic - *Automatic Content Recognition (ACR*).[online].[Accessed on 12/Apr/2015]. Available in web: https://www.audiblemagic.com/

[6] Guillermo González Caravaca, *Reducción de ruido en grabaciones de audio*. Grupo ATVS UAM , July 2011.

[7] Streaming Raddio – *Frequencia de muestreo.* [online].[Accessed on 7/Apr/2015]. Available in web: http://streamingraddios.com/frecuencia-de-muestreo

[8] Gema Piñero Sipam, *Análisis espectral*. Universitat Politecnica de Valencia. [online] Available in web: http://gpinyero.webs.upv.es/Tema5.pdf

[9] Julius O. Smith III and Jonathan S. Abel. *"The Bark Frequency Scale",* Available in web: https://ccrma.stanford.edu/~jos/bbt/Bark_Frequency_Scale.html

[10] A. Acero and R.M. Stern, *Environmental robustness in automatic speech recognition.* Proc. ICASSP-90, 1:849-852, 1990.

[11] Department of Electrical Engineering – *Filterbank Analysis*. Columbia University. [online] .[Accessed on 28/March/2015]. Available in web: http://www.ee.columbia.edu/ln/LabROSA/doc/HTKBook21/node54.html

[12] School of computing - *Feature extraction MFCCs*. University of Eastern Finland [online] .[Accessed on 30/March/2015]. Available in web: http://cs.uef.fi/pages/ /STWS2014/lecture_notes/feature_Extraction_MFCCs.pdf

[13] Avery Li-Chun Wang , *An Industrial-Strength Audio Search Algorithm.* Shazam Entertainment, Ltd.

[14] Karsten Kumpf and Robin W. King. *Automatic accent classification of foreign accented austrialian english speech*. In Fourth Internalional Conference on Spoken Language Processing, volume 3, pages 1740–1743, 1996.

[15] John Saunders. *Real-time discrimination of broadcast speech/music*. In International Conference on Acoustics, Speech and Signal Processing, pages 993–996. IEEE, 1996.

[16] Eric Scheirer and Malcolm Slaney. *Construction and evaluation of a robust multifeature speech/music discriminator*. Speech and Signal Processing, volume II, pages 1331–1334. IEEE, 1997.

[17] S. Tiberwala. H. Hermansky. *"Multiband and Adaptation Approaches to Robust Speech Recognition"*, Proc. EUROSPEECH'97, Rhodes, Greece, pp 2619-2622, 199.

[18] Viikki, O. and Laurila, K., *"Cepstral domain segmental feature vector normalization for noise robust speech recognition"*, Speech Communication.

[19] Jun Du, Ren-Hua Wang. *"Cepstral Shape Normalization (CSN) for Robust Speech Recognition"*, University of Science and Technology of China, Hefei, P. R. China, 230027.

[20] J. Chen, J. Benesty, Y. Huang and E.J. Diethorn. *"Fundamentals of Noise Reduction"*. Springer Handbook. s.l. : Springer, 2008.

[21] Philip N. Garner. *Cepstral normalisation and the signal to noise ratio spectrum in automatic speech recognition.* Idiap Research Institute, Centre du Parc, Rue Marconi 19, PO Box 592, 1920 Martigny, Switzerland.

[22] R. Thangarajan and A.M. Natarajan *"A Robust Front-End Processor combining Mel Frequency Cepstral Coefficient and Sub-band Spectral Centroid Histogram methods for Automatic Speech Recognition"*. International Journal of Signal Processing, Image Processing and Pattern Recognition Vol. 2, No. 2, June 2009.

[23] Svein G. Pettersen and Bojana Gajic *"Model Compensation for Features Based on Subband Spectral Centroid Histograms"*. Norwegian University of Science and Technology.

[24] Jelani Nelson, David P. Woodruff, *Fast Manhattan Sketches in Data Streams*. PODS'10, June 6–11, 2010, Indianapolis, Indiana, USA. Copyright 2010 ACM 978-1-4503-0033-9/.

Anexo A

PRESUPUESTO

1) Material

- Compra de ordenador personal (Software incluido)..... ..... 1.500 €
- Material de oficina ..... 200 €
- Total de ejecución material..... 1.700 €

2) Gastos generales

- Ejecución Material ..... 272 €

3) Beneficio Industrial

- 6 % sobre Ejecución Material..... 102 €

4) Honorarios Proyecto

- 750 horas a 17 € / hora ..... 12.750 €

5) Material fungible

- Gastos de impresión ..... 80 €
- Encuadernación ..... 180 €

6) Subtotal del presupuesto

- Subtotal Presupuesto..... 15.084 €

7) I.V.A. aplicable

- 21% Subtotal Presupuesto..... 3167.6 €

8) <u>Total presupuesto</u>

- **Total Presupuesto ..... 18251.6€**

Madrid, Mayo de 2015
El ingeniero Jefe de Proyecto

Fdo.: Andrés Martín López
Ingeniero de Telecomunicación

Anexo B

PLIEGO DE CONDICIONES

Este documento contiene las condiciones legales que guiarán la realización, en este proyecto, de un sistema de búsqueda rápida de audio en audio. En lo que sigue, se supondrá que el proyecto ha sido encargado por una empresa cliente a una empresa consultora con la finalidad de realizar dicho sistema. Dicha empresa ha debido desarrollar una línea de investigación con objeto de elaborar el proyecto. Esta línea de investigación, junto con el posterior desarrollo de los programas está amparada por las condiciones particulares del siguiente pliego.

Supuesto que la utilización industrial de los métodos recogidos en el presente proyecto ha sido decidida por parte de la empresa cliente o de otras, la obra a realizar se regulará por las siguientes:

Condiciones generales

1. La modalidad de contratación será el concurso. La adjudicación se hará, por tanto, a la proposición más favorable sin atender exclusivamente al valor económico, dependiendo de las mayores garantías ofrecidas. La empresa que somete el proyecto a concurso se reserva el derecho a declararlo desierto.

2. El montaje y mecanización completa de los equipos que intervengan será realizado totalmente por la empresa licitadora.

3. En la oferta, se hará constar el precio total por el que se compromete a realizar la obra y el tanto por ciento de baja que supone este precio en relación con un importe límite si este se hubiera fijado.

4. La obra se realizará bajo la dirección técnica de un Ingeniero Superior de Telecomunicación, auxiliado por el número de Ingenieros Técnicos y Programadores que se estime preciso para el desarrollo de la misma.

5. Aparte del Ingeniero Director, el contratista tendrá derecho a contratar al resto del personal, pudiendo ceder esta prerrogativa a favor del Ingeniero Director, quien no estará obligado a aceptarla.

6. El contratista tiene derecho a sacar copias a su costa de los planos, pliego de condiciones y presupuestos. El Ingeniero autor del proyecto autorizará con su firma las copias solicitadas por el contratista después de confrontarlas.

7. Se abonará al contratista la obra que realmente ejecute con sujeción al proyecto que sirvió de base para la contratación, a las modificaciones autorizadas por la superioridad o a las órdenes que con arreglo a sus facultades le hayan comunicado por escrito al Ingeniero Director de obras siempre que dicha obra se haya ajustado a los preceptos de los pliegos de condiciones, con arreglo a los cuales, se harán las modificaciones y la valoración de las diversas unidades sin que el importe total pueda exceder de los presupuestos aprobados. Por consiguiente, el número de unidades que se consignan en el proyecto o en el presupuesto, no podrá servirle de fundamento para entablar reclamaciones de ninguna clase, salvo en los casos de rescisión.

8. Tanto en las certificaciones de obras como en la liquidación final, se abonarán los trabajos realizados por el contratista a los precios de ejecución material que figuran en el presupuesto para cada unidad de la obra.

9. Si excepcionalmente se hubiera ejecutado algún trabajo que no se ajustase a las condiciones de la contrata pero que sin embargo es admisible a juicio del Ingeniero Director de obras, se dará conocimiento a la Dirección, proponiendo a la vez la rebaja de precios que el Ingeniero estime justa y si la Dirección resolviera aceptar la obra, quedará el contratista obligado a conformarse con la rebaja acordada.

10. Cuando se juzgue necesario emplear materiales o ejecutar obras que no figuren en el presupuesto de la contrata, se evaluará su importe a los precios asignados a otras obras o materiales análogos si los hubiere y cuando no, se discutirán entre el Ingeniero Director y el contratista, sometiéndolos a la aprobación de la Dirección. Los nuevos precios convenidos por uno u otro procedimiento, se sujetarán siempre al establecido en el punto anterior.

11. Cuando el contratista, con autorización del Ingeniero Director de obras, emplee materiales de calidad más elevada o de mayores dimensiones de lo estipulado en el proyecto, o sustituya una clase de fabricación por otra que tenga asignado mayor precio o ejecute con mayores dimensiones cualquier otra parte de las obras, o en general, introduzca en ellas cualquier modificación que sea beneficiosa a juicio del Ingeniero Director de obras, no tendrá derecho sin embargo, sino a lo que le correspondería si hubiera realizado la obra con estricta sujeción a lo proyectado y contratado.

12. Las cantidades calculadas para obras accesorias, aunque figuren por partida alzada en el presupuesto final (general), no serán abonadas sino a los precios de la contrata, según las condiciones de la misma y los proyectos particulares que para ellas se formen, o en su defecto, por lo que resulte de su medición final.

13. El contratista queda obligado a abonar al Ingeniero autor del proyecto y director de obras así como a los Ingenieros Técnicos, el importe de sus respectivos honorarios facultativos

por formación del proyecto, dirección técnica y administración en su caso, con arreglo a las tarifas y honorarios vigentes.

14. Concluida la ejecución de la obra, será reconocida por el Ingeniero Director que a tal efecto designe la empresa.

15. La garantía definitiva será del 4% del presupuesto y la provisional del 2%.

16. La forma de pago será por certificaciones mensuales de la obra ejecutada, de acuerdo con los precios del presupuesto, deducida la baja si la hubiera.

17. La fecha de comienzo de las obras será a partir de los 15 días naturales del replanteo oficial de las mismas y la definitiva, al año de haber ejecutado la provisional, procediéndose si no existe reclamación alguna, a la reclamación de la fianza.

18. Si el contratista al efectuar el replanteo, observase algún error en el proyecto, deberá comunicarlo en el plazo de quince días al Ingeniero Director de obras, pues transcurrido ese plazo será responsable de la exactitud del proyecto.

19. El contratista está obligado a designar una persona responsable que se entenderá con el Ingeniero Director de obras, o con el delegado que éste designe, para todo relacionado con ella. Al ser el Ingeniero Director de obras el que interpreta el proyecto, el contratista deberá consultarle cualquier duda que surja en su realización.

20. Durante la realización de la obra, se girarán visitas de inspección por personal facultativo de la empresa cliente, para hacer las comprobaciones que se crean oportunas. Es obligación del contratista, la conservación de la obra ya ejecutada hasta la recepción de la misma, por lo que el deterioro parcial o total de ella, aunque sea por agentes atmosféricos u otras causas, deberá ser reparado o reconstruido por su cuenta.

21. El contratista, deberá realizar la obra en el plazo mencionado a partir de la fecha del contrato, incurriendo en multa, por retraso de la ejecución siempre que éste no sea debido a causas de fuerza mayor. A la terminación de la obra, se hará una recepción provisional previo reconocimiento y examen por la dirección técnica, el depositario de efectos, el interventor y el jefe de servicio o un representante, estampando su conformidad el contratista.

22. Hecha la recepción provisional, se certificará al contratista el resto de la obra, reservándose la administración el importe de los gastos de conservación de la misma hasta su recepción definitiva y la fianza durante el tiempo señalado como plazo de garantía. La recepción definitiva se hará en las mismas condiciones que la provisional, extendiéndose el acta correspondiente. El Director Técnico propondrá a la Junta Económica la devolución de la fianza al contratista de acuerdo con las condiciones económicas legales establecidas.

23. Las tarifas para la determinación de honorarios, reguladas por orden de la Presidencia del Gobierno el 19 de Octubre de 1961, se aplicarán sobre el denominado en la actualidad “Presupuesto de Ejecución de Contrata” y anteriormente llamado ”Presupuesto de Ejecución Material” que hoy designa otro concepto.

Condiciones particulares

La empresa consultora, que ha desarrollado el presente proyecto, lo entregará a la empresa cliente bajo las condiciones generales ya formuladas, debiendo añadirse las siguientes condiciones particulares:

1. La propiedad intelectual de los procesos descritos y analizados en el presente trabajo, pertenece por entero a la empresa consultora representada por el Ingeniero Director del Proyecto.

2. La empresa consultora se reserva el derecho a la utilización total o parcial de los resultados de la investigación realizada para desarrollar el siguiente proyecto, bien para su publicación o bien para su uso en trabajos o proyectos posteriores, para la misma empresa cliente o para otra.

3. Cualquier tipo de reproducción aparte de las reseñadas en las condiciones generales, bien sea para uso particular de la empresa cliente, o para cualquier otra aplicación, contará con autorización expresa y por escrito del Ingeniero Director del Proyecto, que actuará en representación de la empresa consultora.

4. En la autorización se ha de hacer constar la aplicación a que se destinan sus reproducciones así como su cantidad.

5. En todas las reproducciones se indicará su procedencia, explicitando el nombre del proyecto, nombre del Ingeniero Director y de la empresa consultora.

6. Si el proyecto pasa la etapa de desarrollo, cualquier modificación que se realice sobre él, deberá ser notificada al Ingeniero Director del Proyecto y a criterio de éste, la empresa consultora decidirá aceptar o no la modificación propuesta.

7. Si la modificación se acepta, la empresa consultora se hará responsable al mismo nivel que el proyecto inicial del que resulta el añadirla.

8. Si la modificación no es aceptada, por el contrario, la empresa consultora declinará toda responsabilidad que se derive de la aplicación o influencia de la misma.

9. Si la empresa cliente decide desarrollar industrialmente uno o varios productos en los que resulte parcial o totalmente aplicable el estudio de este proyecto, deberá comunicarlo a la empresa consultora.

10. La empresa consultora no se responsabiliza de los efectos laterales que se puedan producir en el momento en que se utilice la herramienta objeto del presente proyecto para la realización de otras aplicaciones.

11. La empresa consultora tendrá prioridad respecto a otras en la elaboración de los proyectos auxiliares que fuese necesario desarrollar para dicha aplicación industrial, siempre que no haga explícita renuncia a este hecho. En este caso, deberá autorizar expresamente los proyectos presentados por otros.

12. El Ingeniero Director del presente proyecto, será el responsable de la dirección de la aplicación industrial siempre que la empresa consultora lo estime oportuno. En caso contrario, la persona designada deberá contar con la autorización del mismo, quien delegará en él las responsabilidades que ostente.