UNIVERSIDAD AUTÓNOMA DE MADRID

ESCUELA POLITÉCNICA SUPERIOR

Grado en Ingeniería de Tecnologías y Servicios de Telecomunicación

TRABAJO FIN DE GRADO

SEGMENTACIÓN DE AUDIO BROADCAST

Benjamín García Naranjo
Tutor: Joaquín González Rodríguez

Enero 2016

SEGMENTACIÓN DE AUDIO BROADCAST

AUTOR: Benjamín García Naranjo
TUTOR: Joaquín González Rodríguez

Biometric Recognition Group - ATVS
Departamento de Tecnología Electrónica y de las Comunicaciones
Escuela Politécnica Superior
Universidad Autónoma de Madrid
Enero 2016

Resumen

La segmentación de audio tiene un indudable interés de investigación, ya que es un paso esencial en el pre-procesado de multitud de aplicaciones de procesado de audio debido a que proporciona una notable mejoría de rendimiento en muchos ámbitos de las tecnologías del habla. De ahí el interés de este TFG.

El desarrollo del mismo comenzará con un estudio del estado del arte en segmentación de audio, y posteriormente se analizará el sistema que se ha diseñado e implementado para este proyecto. En concreto, detallaremos un sencillo e intuitivo algoritmo útil para detectar señal de voz de una forma muy precisa en señales de audio broadcast.

La base de dicho algoritmo es que si comparamos los espectrogramas de señales de voz, música y ruido, podemos observar que las señales de voz suelen mostrar patrones relativos a la presencia de varios armónicos, que son influenciados por la forma del tracto vocal y que en señales de música y ruido no aparecen. De esta forma, proponemos capturar esas trayectorias de armónicos que en contraste con las notas musicales, varían de frecuencia, y así detectar en qué zonas de la señal hay presencia de voz. Posteriormente utilizaremos otras características de la señal de voz para mejorar este algoritmo inicial, como por ejemplo la información de la frecuencia fundamental (pitch) para introducir nuevos datos que nos permitan mejorar la detección de voz.

Finalmente, los resultados ofrecidos por este algoritmo se evaluarán mostrando la tasa de acierto y de error tras la aplicación del sistema diseñado sobre la base de datos creada por el Área de Tratamiento de Voz y Señales (ATVS). Esta base de datos contiene 20 horas audio etiquetado de 4 programas de radio reales que contienen música, publicidad, tertulias… y cuya implementación también forma parte de este proyecto.

Palabras clave

Audio, segmentación, voz, música, ruido, procesado, broadcast, algoritmo, armónicos, frecuencia fundamental, trayectorias, tasa de acierto, tasa de error, radio, base de datos, ATVS.

Abstract

Audio segmentation is undoubtedly an interest of investigation, as it is an essential step in the preprocessing of many audio processing applications, due to its contribution of a significant effectiveness improvement in many areas of technology speech. Therefore the interest of this TFG.

Its development will begin with a study of the state of the art audio segmentation. Later on, the system designed and implemented for this project will be analysed. In detail, we will present a simple and intuitive spectral feature for detecting the presence of spoken speech into audio broadcast signal.

The basis of the mentioned algorithm is that if we compare the spectrograms of speech signals, music and noise, can we observe that speech signals usually display patterns relating to the presence of several harmonics, and that in music and noise signals do not appear. In this way, we propose to capture sustained harmonics´ trajectories which –in contrast to the partials of a note played on a musical instrument– vary in frequency, and so detect in which areas of the signal, the voice is present. After that, we will use other features of the speech signal to improve the initial algorithm. For example, the information of the fundamental frequency (pitch) to enter new data which allow us to improve the speech detection.

Finally, the results offered by this algorithm will be evaluated showing the success and error rates, after having applied the designed system on database created by Area de Tratamiento de Voz y Señales (ATVS). This database contains 20 labelled audio hours of 4 programs which contain music advertising, social gatherings… and whose implementation is also part of this project.

Keywords

Audio, segmentation, voice, music, noise, processing, broadcast, algorithm, harmonic, fundamental frequency, trajectories, success rate, error rate, radio, database, ATVS.

Agradecimientos

A mis padres, Benjamín y Teresa,

que me dieron la vida,

y me enseñaron que es el esfuerzo,

lo que da sentido al camino.

En primer lugar, me gustaría dar las gracias a mi tu tutor, Joaquín González, por darme la oportunidad de llevar a cabo este proyecto, por su inestimable ayuda y por su asesoramiento y compañía a lo largo de estos meses. Así mismo, agradecer también a todos los compañeros del ATVS que han colaborado en este proyecto, con la creación de la base de datos.

Llegado el final de esta etapa de mi vida (supongo), no puedo olvidar a todas esas personas que han hecho que haya sido tan especial.

A mi familia, a mi padre, a mi madre y a mi hermano, por confiar en mí y ayudarme desde el primer día: gracias, soy quien soy gracias a vosotros.

A mis amigos. A los que ya estaban, por acompañarme en este trayecto (en especial gracias a Samu). A los que he podido conocer durante estos cuatro años, por compartir tantos momentos que llevaré siempre conmigo. Y por último a esas personas que he conocido durante este TFG, esperando que pueda compartir con ellos el siguiente trayecto (gracias Marta).

ÍNDICE DE CONTENIDOS

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

1 Introducción

1.1 Motivación

Aunque el procesado de voz es un campo muy amplio que se orienta hacia aplicaciones en situaciones reales, la segmentación de audio en un entorno como es la radio, no ha sido especialmente investigado. Como veremos más adelante, los detectores de voz clásicos se basan en extracción de características de la señal que permiten clasificar la señal en voz o silencio pero la mayoría de estos algoritmos no serían útiles en programas de radio ya que en este entorno, la música y la voz suelen ir solapadas y estos algoritmos serían incapaces de diferenciar voz de voz cantada sobre música. Por lo tanto, nuestro proyecto se centrará en crear un nuevo algoritmo que teniendo en cuenta todos estos factores, sea capaz de realizar correctamente la detección de voz.

Por otro lado, dicho entorno presenta una dificultad añadida: las grabaciones de audio en las que aparece voz, música, ruido, silencios… no son grabaciones claras, suelen contener un gran nivel de ruido, y a menudo aparecen solapamientos de voz hablada sobre música, voz cantada sobre música, pitidos, voces solapadas en tertulias... un reto y un desafío motivador que, junto con mi director de TFG, hemos decidido acometer.

Por otro lado, la participación del grupo ATVS-UAM en numerosas competiciones a nivel mundial en este ámbito, como las participaciones en las evaluaciones Albayzin 2010 y 2014 ("ATVS-UAM System Description for the Audio Segmentation and Speaker Diarization Albayzin 2010 Evaluation" [2] y "Albayzín-2014 evaluation: audio segmentation and classification in broadcast news domains" [3]), y los buenos resultados obtenidos, han servido como acicate para abordar un TFG de estas características.

1.2 Objetivos

El objetivo que persigue este proyecto es conseguir una segmentación de audio broadcast, en este caso de programas de radio, lo más precisa posible para que posteriormente puedan realizarse algoritmos y aplicaciones sobre dicho audio ya segmentado, como pueden ser la diarización de locutores, detección de publicidad, detección de palabras clave…

Así, el objetivo principal de este TFG, es el de diseñar, e implementar desde cero un algoritmo de segmentación de audio, encargado de reconocer, de una forma muy precisa, cuando existe señal de voz en señales de audio extraídas de programas de radio.

Un objetivo secundario de este proyecto, ha sido el de la creación y posterior etiquetado de una base de datos que permita comprobar cuáles son los resultados reales arrojados por el algoritmo que hemos implementado. Esta base de datos creada y etiquetada por miembros del ATVS Biometric Group, presenta cualidades diversas en cuanto a calidad de grabación, ruidos, pitidos, ecos... por lo que se analizarán los resultados teniendo en cuenta estos parámetros.

Un tercer objetivo es el de comprobar la bondad de nuestro algoritmo, realizando un estudio de los resultados obtenidos al probarlo sobre la base de datos; y finalmente ajustar algunos parámetros de dicho algoritmo.

Finalmente, también es objeto de este TFG, el realizar un estudio detallado sobre el estado del arte, lo que nos ha permitido conocer qué características de la señal son las más utilizadas en detección de voz. También se ha investigado si estas características son igual de eficaces en señales de audio en general, como en señales de radio en particular, teniendo en cuenta tanto los resultados obtenidos, como el coste computacional empleado.

1.3 Organización de la memoria

La memoria de este TFG está estructurada de la siguiente forma:

- **Capítulo 1. Introducción.**
  En este primer capítulo se explica cuál ha sido la motivación para la realización de este TFG así como los objetivos que se pretenden conseguir con la realización del mismo y una breve explicación de la estructura que tiene el trabajo.

- **Capítulo 2. Estado del arte.**
  Aquí se analiza el estado del arte sobre la segmentación de audio: se estudian diferentes métodos para llevar a cabo esta segmentación, se examinan algunas de las características más importantes que se extraen de la señal de audio y que nos permiten la clasificación del mismo.

- **Capítulo 3. Diseño y desarrollo.**
  Se describe cuál ha sido el proceso de creación del algoritmo. Se explica cómo se ha llevado a cabo la creación y etiquetado de la base de datos utilizada para determinar la bondad del algoritmo diseñado. Por último se presentan los métodos utilizados para obtener los resultados del siguiente capítulo.

- **Capítulo 4. Pruebas y resultados.**
  Se aportan datos sobre las pruebas llevadas a cabo y que nos permiten hacernos idea de la precisión del algoritmo diseñado.

- **Capítulo 5. Conclusiones y trabajo futuro.**
  Se analiza brevemente el resultado final del algoritmo y se detallan las posibles futuras líneas de avance tras este TFG.

2 Estado del arte

La segmentación de audio tiene numerosas aplicaciones y juega un papel muy importante en el pre procesado de audio. Es por eso que a pesar de ser un problema complejo, ha sido y sigue siendo un ámbito en el que se lleva a cabo gran cantidad de investigación y desarrollo.

Son muchos los métodos utilizados para llevar a cabo la segmentación de audio. Gran parte de ellos se basan en la extracción de determinadas características que permitan clasificar el audio analizado, pero como se verá más tarde, algunos de estos algoritmos no son realmente útiles en determinados entornos como es el que nos atañe, la radio.

2.1 Extracción de características del audio

Una primera aproximación de Detectores de Actividad de Voz clásicos, se basa en características de la señal como la energía, la periodicidad, las tasas de cruce por cero apoyadas en técnicas de discriminación basadas en modelos heurísticos.

Los VAD (Detectores de Actividad de Voz) más sofisticados, utilizan estas características, pero basan su discriminación en modelos estadísticos. Estos modelos estadísticos típicos están apoyados en clasificadores que asumen diversas distribuciones (ejemplo gaussianas) para describir las características de ruido y voz. Además de un buen rendimiento y consistencia a través del funcionamiento con varios tipos de ruido, las características deseadas de un VAD incorporan baja complejidad computacional y adaptación rápida a los tipos de ruido cambiantes.

La base de dichos algoritmos es por tanto, la extracción de características de la señal de voz de forma que permiten diferenciar la señal de voz frente al ruido o al silencio.

La extracción de características es el proceso que se encarga de convertir la señal de audio en una secuencia de vectores que tienen información sobre la señal analizada. Estos vectores se utilizan como base de varios tipos de algoritmos de análisis de audio.

La señal de voz es un proceso aleatorio y no estacionario, lo que supone una dificultad a la hora de analizar la señal. La voz es pseudo-estacionaria sólo a corto plazo (decenas de ms.) por tanto, para aplicar técnicas de análisis y procesado, debemos limitar el segmento a procesar a este orden de magnitud. Ello obligará al uso de tramas de audio de una duración determinada.

El mecanismo que nos permite realizar un análisis localizado mediante el uso de tramas se denomina enventanado de la señal. El enventanado consiste en la multiplicación sobre la señal completa de una función limitada en el tiempo (ventana), lo que producirá una nueva señal de voz cuyo valor fuera del intervalo definido por la ventana es nulo.

Podemos expresar el enventanado como $x(n) = s(n).w(n-m)$ donde $s(n)$ es la señal original, $w(n)$ es la ventana temporal y $x(n)$ la trama de señal enventanada. Esto es equivalente a la convolución del espectro deseado de la señal con la transformada de Fourier de la ventana correspondiente.

Una vez se tiene la señal divida en tramas, se procede a la extracción de características. Algunas de las características más utilizadas en la mayoría de algoritmos de detección de voz son la energía y la tasa de cruces por cero.

2.1.1 Energía

La Energía E de una señal discreta s(n) se define con la expresión

$$E = \sum_{n=-\infty}^{\infty} s(n)^2$$

La energía de la señal de voz será por tanto

$$E_{s(m)} = \sum_{n=-\infty}^{\infty} [s(n).w(n-m)]^2 = \sum_{n=m-N+1}^{m} s^2(n).w^2(n-m)$$

Y si hacemos $w^2(n) = h(n)$ tendremos

$$E_s(m) = \sum_{n=m-N+1}^{m} s^2(n).h(n-m)$$

La energía a corto plazo se utiliza en diferentes problemas de clasificación. En las señales de voz nos permite distinguir entre tramas de voz sorda y tramas de voz sonora. En el caso de habla con una alta calidad se utiliza para distinguir habla de silencio.

2.1.2 Tasa de cruces por cero

Se denomina cruce por cero al hecho de que exista una diferencia de signo entre muestras consecutivas. La tasa de cruces por cero se define matemáticamente como

$$Z_s(m) = \frac{1}{N} \sum_{n=m-N+1}^{m} \frac{|sgn\{s(n)\} - sgn\{s(n-1)\}|}{2} w(n-m)$$

Donde la función sgn queda definida como

$$sgn\{s(n)\} = \begin{cases} 1, s(n) > 0 \\ -1, s(n) < 0 \end{cases}$$

La tasa de cruces por cero es útil para la caracterización de diferentes señales de audio y se ha utilizado popularmente en problemas de clasificación voz/música. En algunos algoritmos se sugiere una variación de la ZCR y se usa la HZCRR que es más discriminativa.

2.2 Problemas para la detección de voz en señales broadcast de radio

Los programas de radio analizados están compuestos en su gran mayoría por noticias, tertulias, música y publicidad. Por otro lado, la mayoría de los anuncios suelen estar compuestos por música y voz solapadas de forma contigua, o con voz hablada sobre una melodía, por lo que no podremos utilizar un detector de voz clásico basado en características como las nombradas anteriormente en un entorno como éste.

Los algoritmos de detección de voz clásicos basados en la extracción de características de la señal, funcionan de forma eficiente en situaciones en la que solo existe voz, como el que se muestra en el ejemplo de la figura 2-1. Para este ejemplo, el detector de voz funciona correctamente y consigue clasificar la señal de forma precisa en voz y no voz.

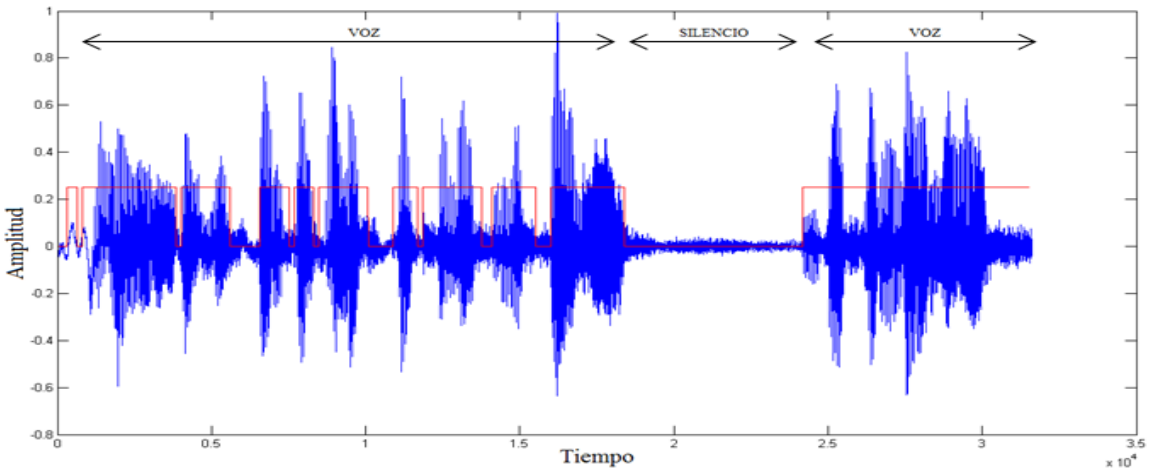

Figura 2-1: Ejemplo de señal de audio en la que aparecen únicamente voz y silencios. En rojo el resultado de aplicar el detector de voz clásico (1 cuando hay voz, 0 cuando no).

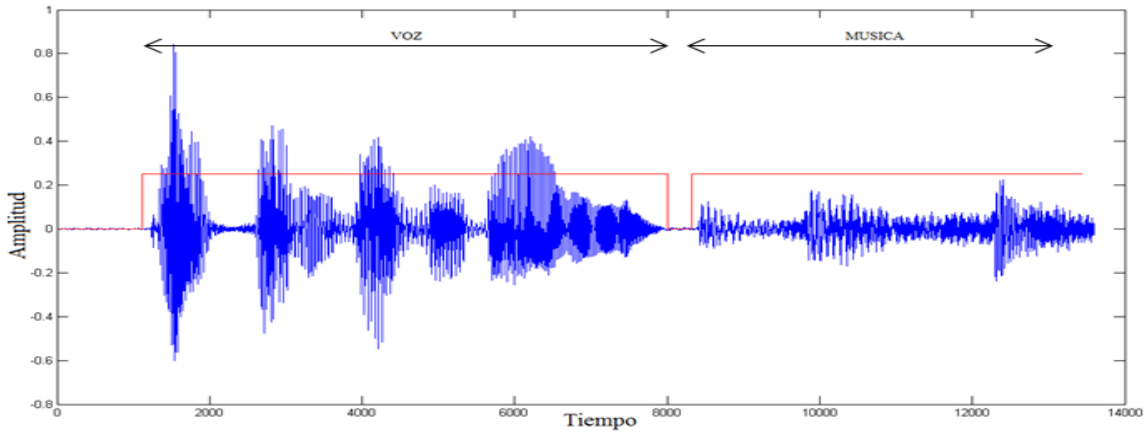

Figura 2-2: Ejemplo de señal de audio en la que aparecen voz y música. En rojo el resultado de aplicar el detector de voz clásico (1 cuando hay voz, 0 cuando no).

Sin embargo, en el ejemplo de la figura 2-2, en el que aparece voz y después música, los resultados del VAD serían incorrectos ya que está clasificando la música (parte final) como voz.

2.3 Detección de voz y música a partir de características espectrales

Estos errores en la detección de voz hacen replantearse que las características que se extraen de la señal no son realmente útiles para el caso que nos ocupa. Así aparece la necesidad de utilizar otros algoritmos de detección de voz basados en características distintas a las nombradas anteriormente.

En [1], que es la base de este TFG, se define una característica espectral simple y útil para detección de voz en base a la representación en escala logarítmica de la STFT. Esta característica está motivada por algunas observaciones simples relativas a las variaciones espectro-temporales de la señal de voz. Con este tipo de representación, al comparar el espectrograma de una señal de voz, se observan varias características específicas que lo distinguen de los espectrogramas de música o ruido (figura 2-3).

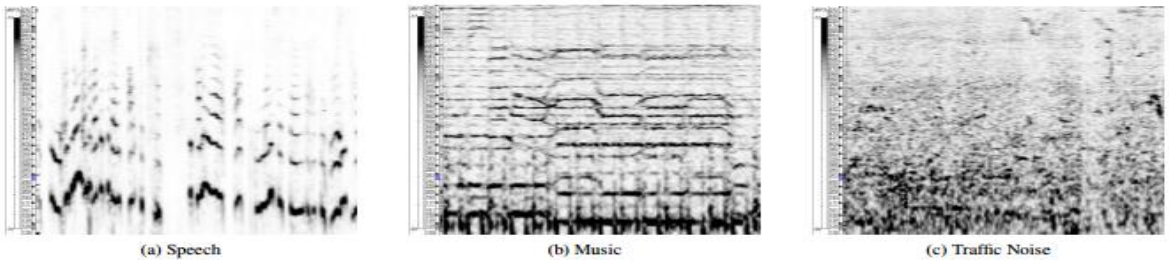

Figura 2-3: Comparación de los log-espectrogramas para (a) habla, (b) música y (c) ruido de tráfico. Cada sección corresponde con un tramo de 7 segundos y un rango de frecuencias de 100 Hz a 3000 Hz en escala logarítmica.

Así por ejemplo, podemos comprobar que en dicha representación (figura 2-3) las señales de voz suelen mostrar patrones relativos a la presencia de varios armónicos, que son influenciados por la forma del tracto vocal. Una segunda observación importante, es que los armónicos son sostenidos durante un corto periodo de tiempo en el cual tienden a variar en frecuencia. De esta forma, se visualizan claramente las trayectorias curvas que presenta el espectrograma de la señal de voz.

En la misma figura 2-3, por el contrario, el espectro de la señal de música se caracteriza por tener trayectorias mayoritariamente horizontales en las zonas de baja frecuencia del espectro, y en el espectro de la señal de ruido no se identifica ningún patrón de interés. Así podemos discriminar de forma efectiva la señal de voz respecto al ruido o a la música.

La idea básica que subyace detrás de esta característica que acabamos de definir, es capturar las trayectorias de los armónicos que, en contraste con una nota de algún instrumento musical, varían en frecuencia. Este fenómeno da lugar a una alta correlación al comparar el espectro de dos tramas cercanas. Por lo tanto cada trama X_t se comparará con una trama posterior $X_{t+offset}$. Sin embargo, para poder permitir las trayectorias curvas de los armónicos, han de tenerse en cuenta las variaciones en frecuencia. Esto se consigue calculando la correlación cruzada entre X_t y $X_{t+offset}$. La correlación cruzada estima el grado de correlación entre versiones desplazadas de estos vectores. El grado de desplazamiento viene dado por el valor de lo que conocemos como lags, l. Dados dos vectores x e y de longitud N, la correlación cruzada para valores de lags $l \in [-N\,, N]$ queda definida como

$$R_{xy}(l) = \sum_i x_i y_{i+l}$$

En nuestro caso, los vectores de entrada son tramas en el tiempo y el lag corresponde a un desplazamiento en el eje de frecuencias. Podemos definir r_{xcorr} como la máxima correlación cruzada en un rango de lags

$$r_{xcorr}(X_t, X_{t+offset}) = \max R_{X_t, X_{t+offset}}\ (l)$$

Donde $l \in [-l_{max}, l_{max}]$ denota el desplazamiento en el eje de frecuencias. También definimos r como un caso especial de la correlación cruzada comentada previamente en la cual lag $l = 0$ y por tanto

$$r(X_t, X_{t+offset}) = R_{X_t, X_{t+offset}}(0)$$

Para obtener un indicador de la presencia de voz definimos el valor de $r_{xcorr} - r$, de forma que para señales en las que solo aparece música "ideal", la correlación cruzada va a tener el máximo en lag 0 y por lo tanto el valor de $r_{xcorr} - r = R(0) - r = r - r = 0$, mientras que para señales de voz que presentan armónicos, $R(l)$ va a tener el valor máximo en algún $l \neq 0$ y por consiguiente el valor de $r_{xcorr} - r$ será un valor positivo.

Ajustando determinados parámetros como frecuencia de muestreo, rango de frecuencias de la escala logarítmica, solapamiento en el enventanado, duración de las ventanas, número de muestras de la fft, máximo lag, máximo offset… y teniendo en cuenta que los archivos donde se lleva a cabo estas pruebas son "ideales", se pueden obtener resultados tan óptimos como los que se presentan a continuación.

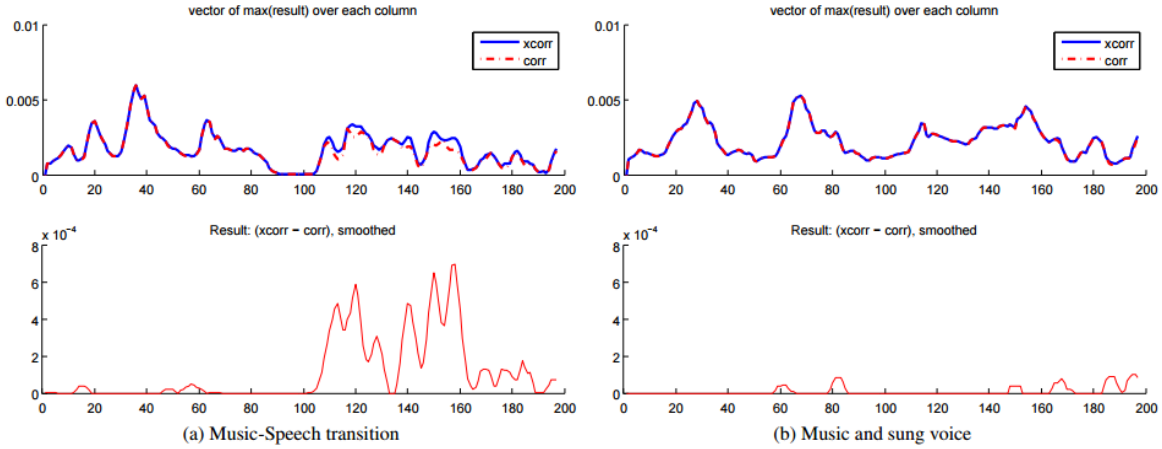

Figura 2-4: Comparación de las características para las señales (a) música y después habla y (b) música y después voz cantada. La primera fila compara los valores de $R(0)$ (línea punteada) con r_{xcorr} (línea sólida). La segunda fila muestra el valor de $r_{xcorr} - r$.

Este será el elemento base fundamental de este TFG, a continuación mostraremos como llevar a cabo dicho algoritmo, estudiaremos cuáles son los valores de los parámetros que mejor resultados ofrecen, y propondremos una mejora para dicho algoritmo. Finalmente realizaremos un estudio minucioso de las trayectorias armónicas de la señal de voz y se incluirán nuevas características que permitan detectar voz de una manera aún más precisa.

3 Diseño y desarrollo

3.1 Base de datos

3.1.1 Creación de la base de datos

La primera fase de este proyecto consistió en la creación de una base de datos en la que poder comprobar y mejorar la eficiencia del algoritmo diseñado. Este proceso se llevó a cabo por un conjunto de 4 personas del grupo de investigación ATVS-UAM.

Inicialmente se propusieron dos medios de los que extraer las señales a analizar: la radio y la televisión. Aunque la televisión contara con la ventaja de poder apoyarse en la imagen para llevar a cabo el etiquetado, nos decantamos por la señal de radio ya que la señal de audio de los programas de radio era más fácil de extraer. Además, la mayoría de programas de radio podían ser descargados de forma sencilla desde plataformas como "ivoox" o directamente desde la página web de la emisora correspondiente.

Tomada la decisión de utilizar programas de radio, surgió la pregunta: ¿Qué tipo de programas de radio escoger? Necesitábamos programas que fueran lo más completos posibles, es decir, que tuvieran tanto música, como tertulias, publicidad, llamadas, entrevistas… Y por consenso entre los miembros del grupo los programas quedaron limitados al ámbito nacional.

De esta forma, los programas elegidos, así como los horarios de emisión, grabación y etiquetado para los diferentes espacios, fueron los siguientes:

PROGRAMA	CADENA	HORARIO		
		Emisión	Grabación	Etiquetado
Julia en la Onda	ONDA CERO	16:00 – 19:00	18:00 – 19:00	18:00 – 18:30
Hoy por hoy	CADENA SER	06:00 – 12:00	09:00 – 10:00	09:30 – 10:00
Más de uno	ONDA CERO	06:00 – 12:00	09:00 – 10:00	09:00 – 09:30
La mañana	COPE	06:00 – 12:00	10:00 – 11:00	10:00 – 10:30

Tabla 3-1: Distribución y horarios de los programas de la base de datos.

De cada uno de los programas se eligió la media hora más completa, y sobre ella se realizó el etiquetado. Dicho etiquetado fue realizado sobre 10 días consecutivos (desde el 25 de

Mayo de 2015 al 5 de Junio de 2015), consiguiendo así que durante esos días la publicidad fuera más o menos la misma y que los locutores de las tertulias no cambiaran debido a las vacaciones. La base de datos consta por tanto de un total de 20 horas de señal de radio etiquetadas (cada persona del grupo etiquetó 10 audios) aunque en estos momentos la base de datos está siendo ampliada por otros compañero. A continuación se muestra un resumen del contenido de la base de datos elaborada.

Tabla 3-2: Resumen de la base de datos.

	Archivos de audio	Porcentaje de voz	Porcentaje de no voz	Duración (minutos)
JULIA EN LA ONDA	JO_20150525.wav	97,88%	2,11%	30:01
	JO_20150526.wav	98,10%	1,89%	30:02
	JO_20150527.wav	98,31%	1,68%	30:27
	JO_20150528.wav	98,85%	1,64%	28:38
	JO_20150529.wav	97,09%	2,91%	30:12
	JO_20150601.wav	97,11%	2,90%	30:39
	JO_20150602.wav	97,62%	2,37%	29:35
	JO_20150603.wav	97,25%	2,75%	29:46
	JO_20150604.wav	97,89%	2,10%	29:25
	JO_20150605.wav	98,19%	1,78%	28:30
HOY POR HOY	HH_20150525.wav	96,64%	3,35%	29:13
	HH_20150526.wav	97,51%	2,48%	29:52
	HH_20150527.wav	96,80%	3,20%	28:38
	HH_20150528.wav	96,88%	3,11%	29:05
	HH_20150529.wav	96,62%	3,37%	28:27
	HH_20150601.wav	96,84%	3,15%	29:17
	HH_20150602.wav	96,75%	3,24%	30:26
	HH_20150603.wav	96,61%	3,39%	29:39
	HH_20150604.wav	96,30%	3,69%	30:05
	HH_20150605.wav	96,10%	3,89%	29:30
LA MAÑANA	LM_20150525.wav	96,91%	3,08%	29:99
	LM_20150526.wav	94,94%	5,05%	30:01
	LM_20150527.wav	94,36%	5,63%	30:03
	LM_20150528.wav	94,49%	5,38%	29:9
	LM_20150529.wav	93,62%	6,25%	30:00
	LM_20150601.wav	94,81%	5,18%	30:54
	LM_20150602.wav	96,21%	3,79%	30:09
	LM_20150603.wav	93,80%	6,12%	30:03
	LM_20150604.wav	96,15%	4,85%	30:04
	LM_20150605.wav	95,14%	4,86%	30:06
MÁS DE UNO	MU_20150525.wav	95,36%	4,63%	30:01
	MU_20150526.wav	95,30%	4,69%	30:04
	MU_20150527.wav	94,16%	5,83%	29:34
	MU_20150528.wav	91,84%	8,15%	29:59
	MU_20150529.wav	95,12%	4,87%	29:56
	MU_20150601.wav	89,72%	10,27%	30:11
	MU_20150602.wav	90,61%	9,38%	30:00
	MU_20150603.wav	88,36%	11,63%	30:23
	MU_20150604.wav	74,86%	25,13%	30:02
	MU_20150605.wav	93,36%	6,63%	29:55

3.1.2 Etiquetado de la base de datos

El etiquetado se realizó con la herramienta "Wavesurfer" en base a una serie de instrucciones de etiquetado que se crearon con la ayuda y guía de los profesores Joaquín González y Doroteo Torre (ejemplo en la figura 3-1).

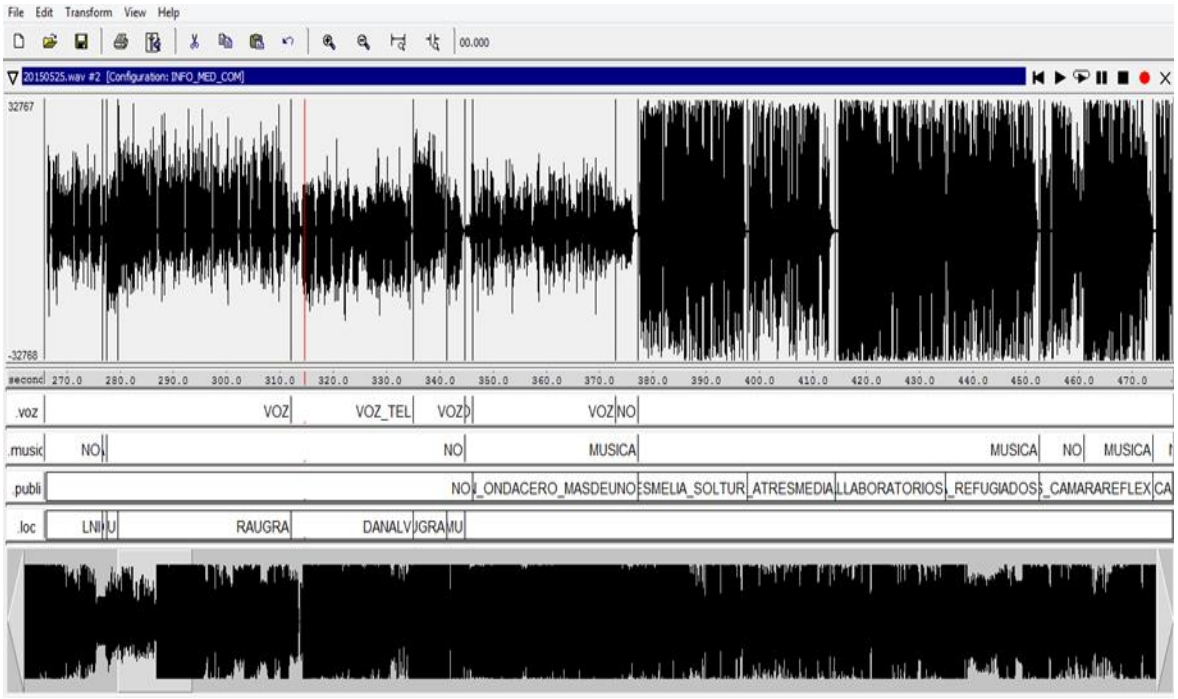

Figura 3-1: Ejemplo de etiquetado de audio con la aplicación "Wavesurfer".

ETIQUETADO DEL TIPO DE AUDIO

Las primeras etiquetas son .voz que contiene las clases "NO"/"VOZ"/"VOZ_TEL" (voz telefónica) y .music que contiene las clases "MUSICA"/"NO".

Dos de estas clases pueden darse simultáneamente y por tanto, es importante indicar si ninguna, una o las dos clases se dan en un instante de tiempo dado.

Es frecuente encontrarse por ejemplo, con música que da paso a cada sección del programa: ese fragmento será etiquetado como música sin voz ("MUSICA" y "NO"). Seguidamente, el locutor comienza a hablar a la vez que la música sigue sonando de fondo, ahora se estarán dando simultáneamente dos clases: voz y música ("MUSICA" y "VOZ").

Para este etiquetado se considera voz todo aquello que una persona diga, y por música cualquier variedad de la misma. En este sentido, el canto dentro de una canción no se considera como voz, sino como música.

ETIQUETADO DE PUBLICIDAD

El etiquetado de la publicidad es un poco más complejo. Se trata de indicar donde comienza y termina un anuncio "enlatado" ("AN_<MARCA>_<PRODUCTO>") y donde comienza y termina una mención publicitaria realizada por los locutores del programa ("ME_<MARCA>_<_PRODUCTO>"). La no existencia de publicidad se indicará con la etiqueta "NO".

ETIQUETADO DE LOCUTORES

Los locutores se deben etiquetar de dos formas distintas:

- Si se trata de locutores habituales, con "NNNAAA", donde NNN son las tres primeras letras del nombre y AAA las primeras tres letras del apellido.
- Si son locutores ocasionales con "L1" a "L9".
- En caso de que haya voz solapada se etiqueta con "SOLAP".
- En caso de que no haya locutores se etiqueta con "NO".

3.1.3 Preparación de los datos

Cada programa de radio proporcionado por cada una de las webs tenía diferente formato y frecuencia de muestreo. Por lo tanto, después de realizar un estudio de la calidad necesaria frente al coste computacional, se llegó a la conclusión de unificar los archivos a 16 KHz de frecuencia de muestreo, 16 bits/muestra, con un solo canal de grabación (mono) y formato "wav".

Este proceso de conversión se ha realizado con la herramienta "ffmpeg", por ejemplo utilizando el comando

ffmpeg -i archivoantiguo.mp4 -ac 1 -ar 16000 nuevoarchivo.wav

convertimos el *archivoantiguo.mp4* en el *nuevoarchivo.wav* en modo monocanal (*-ac 1*) y a una frecuencia de muestreo de 16000 (*-ar 16000*).

3.2 Normalización de ganancia

Como podemos observar en la figura 3-1, la energía de una señal tan larga es muy poco uniforme, lo que complicaría en gran medida el proceso de detección de voz. De esta forma, y para mejorar la efectividad de nuestro algoritmo, se procedió a la normalización de la ganancia de la señal.

La normalización de la ganancia la hemos realizado multiplicando la señal por una serie de ventanas triangulares de 5 segundos solapadas un 50%. Estas ventanas, tienen cada una, una ganancia distinta que hacen que al multiplicar la señal por estas ventanas, consigamos una señal más uniforme.

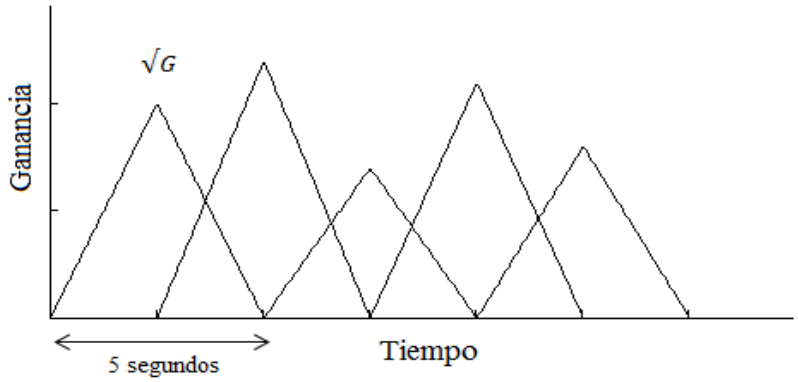

Figura 3-2: Ventanas triangulares que multiplican a la señal original para obtener la normalización de la ganancia.

La forma de calcular la ganancia de cada ventana triangular es la siguiente:

- Enventanamos esta señal de 5 segundos en ventanas de 30 ms con un solapamiento de 10 ms.

- Para cada ventana de 30 ms calculamos su energía logarítmica de la forma

$$E(k) = \frac{1}{N_{muestras}} \sum x^2$$

$$logE(k) = 10 \log\big(E(k)\big)$$

- Calculamos la $dinamica$ de $logE$

    - Si $dinamica > 20\ dB$ consideramos silencio todo lo sea menor que el 30% de la dinámica (desde el mínimo).
    - Si $dinamica < 20\ dB$ consideramos que todo es señal de no silencio.

- Calculamos la energía logarítmica de la señal que no es silencio x_{nosil}

$$E_{nosil} = \frac{1}{N_{muestrasxnosil}} \sum E(k)_{nosil}$$

- Procedemos entonces a calcular la ganancia G como

$$G = \sqrt{\frac{1}{E_{nosil}}}$$

En la figura 3-3 se puede apreciar la señal original y la señal con la ganancia normalizada, obteniendo así una señal más uniforme que hará que aumente la eficacia de nuestro algoritmo.

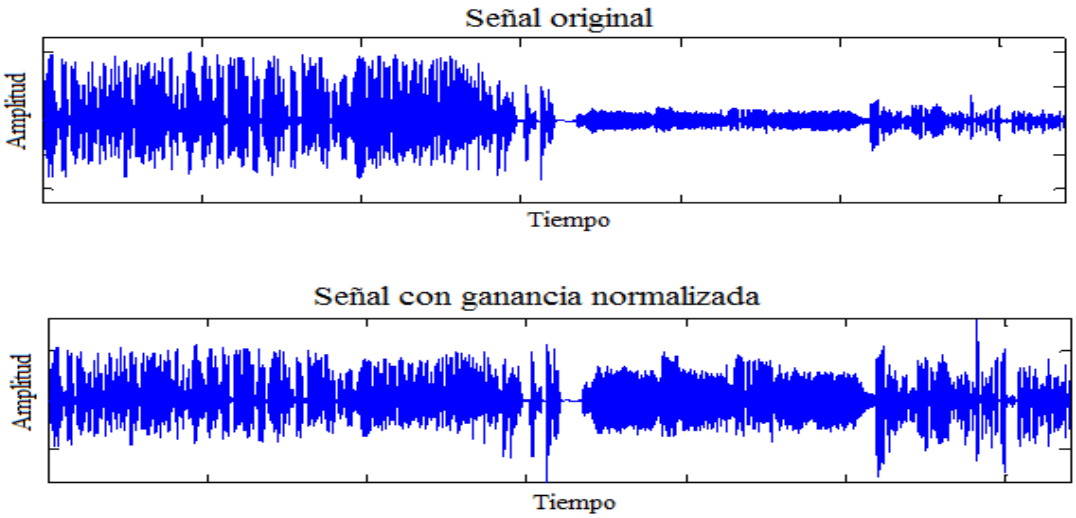

Figura 3-3: Señal original y señal obtenida después de aplicar la normalización de la ganancia (duración total = 25 segundos).

3.3 Enventanado de la señal

Una vez tenemos la señal normalizada en ganancia se procede al enventanado de la señal. Para ello hemos utilizado ventanas de 30 ms con un solapamiento de 10 ms. Se ha utilizado una ventana de tipo hamming cuya estructura temporal está definida de la siguiente forma

$$w(n) = \begin{cases} 0.54 - 0.46\cos\left[\frac{2\pi n}{N-1}\right], 0 \le n \le N-1 \\ 0, \qquad en\ caso\ contrario \end{cases}$$

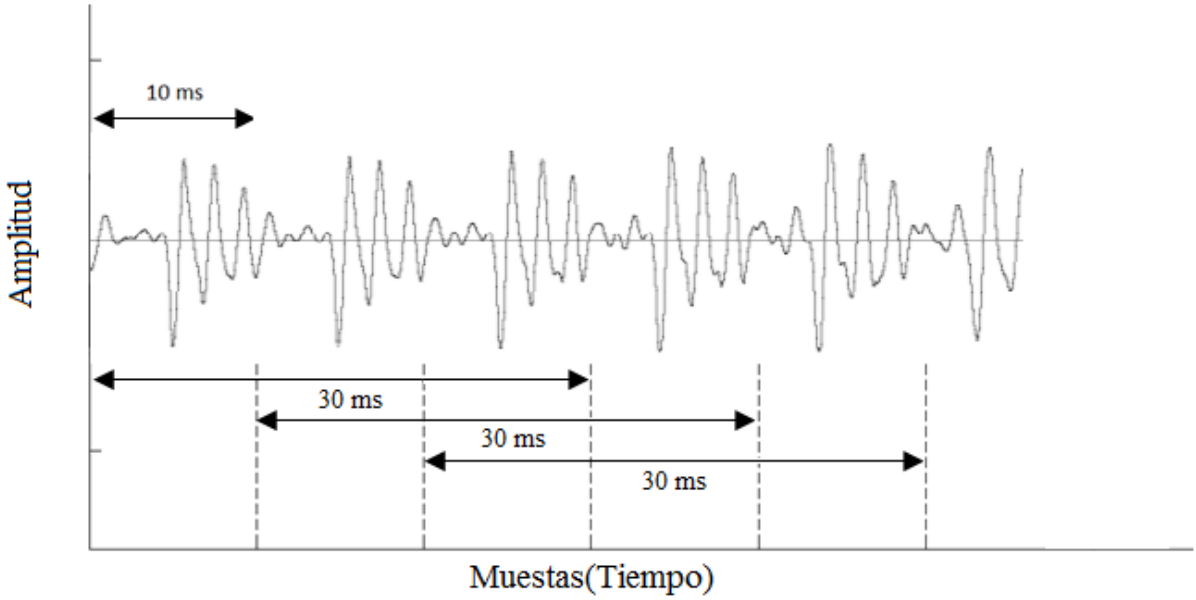

Figura 3-4: Enventanado y solapamiento utilizado. Ventanas de 30 mS con un desplazamiento de 10 mS.

3.4 Representación del "Melgrama"

El espectrograma nos permite una representación en tiempo, frecuencia y nivel espectral a la vez de la señal, pero teniendo en cuenta que nuestro objetivo es identificar trayectorias paralelas, esta solución dista bastante de ser una representación óptima.

Como se puede apreciar en la figura 3-5, con el melgrama obtenemos una representación en la que se aprecian perfectamente las trayectorias paralelas, mientras que si utilizamos el espectrograma, las trayectorias tienden a subir hacia arriba y a no ser paralelas. Por lo tanto el uso del melgrama nos facilita en gran medida la detección de voz.

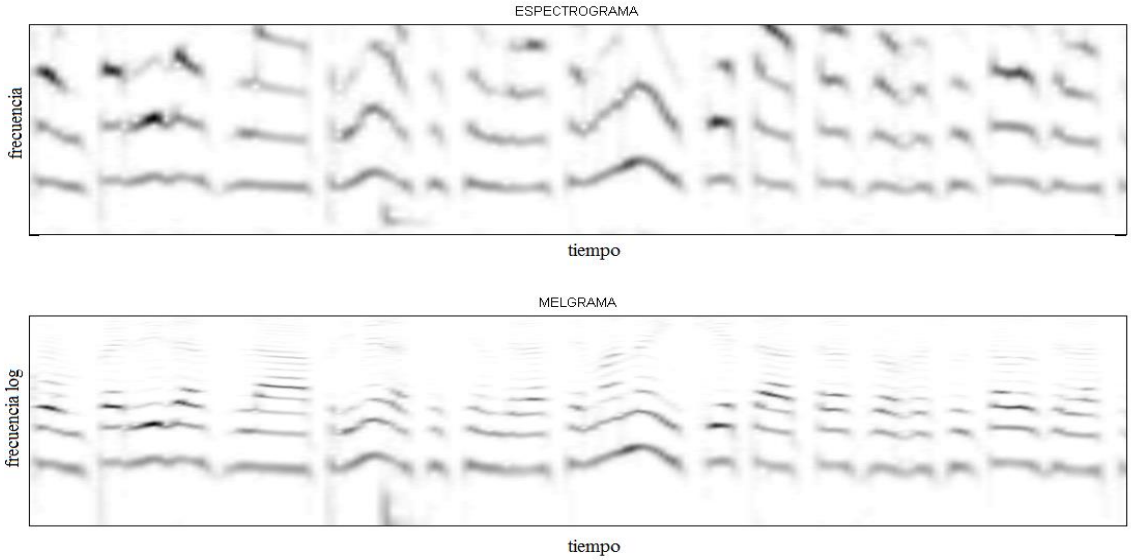

Figura 3-5 Melgrama vs Espectrograma del mismo segmento de audio.

El proceso para el cálculo del melgrama ha sido el siguiente. Una vez tenemos la señal enventanada, se calcula la fft de las muestras de cada ventana de análisis (en nuestro caso de 4096 muestras). Posteriormente, cada una de las tramas que se obtienen del cálculo de la fft se indexan en una matriz, de forma que dicha matriz representará por lo tanto la variación del espectro de la señal y la energía en función del tiempo.

Para el cálculo del melgrama se definieron inicialmente 5 octavas de 40 bins cada una que varían entre (25 Hz–50 Hz), (50 Hz–100 Hz), (100 Hz-200 Hz), (200 Hz- 400Hz) y (400 Hz-800 Hz).

El estudio de los resultados arrojados por este método anterior en el que se utilizan 5 octavas, nos hizo replantearnos la idea inicial, y pensamos que convendría realizar un diezmado por 2 consiguiendo así una frecuencia de muestreo de 8 KHz (el primer armónico aparece muy borroso por falta de resolución).

Por otro lado, también modificamos las octavas, pensando que sería mejor no calcular la primera octava (25 Hz-50 Hz) e incluir las octavas (800 Hz-1600 Hz) y (1600-3200) necesarias para voz de mujer y música.

De esta forma, obtuvimos una matriz final $melgrama$ de tantas columnas como ventanas en las que se divide la señal original y de 240 filas que corresponden a las 6 octavas de 40 bins cada una.

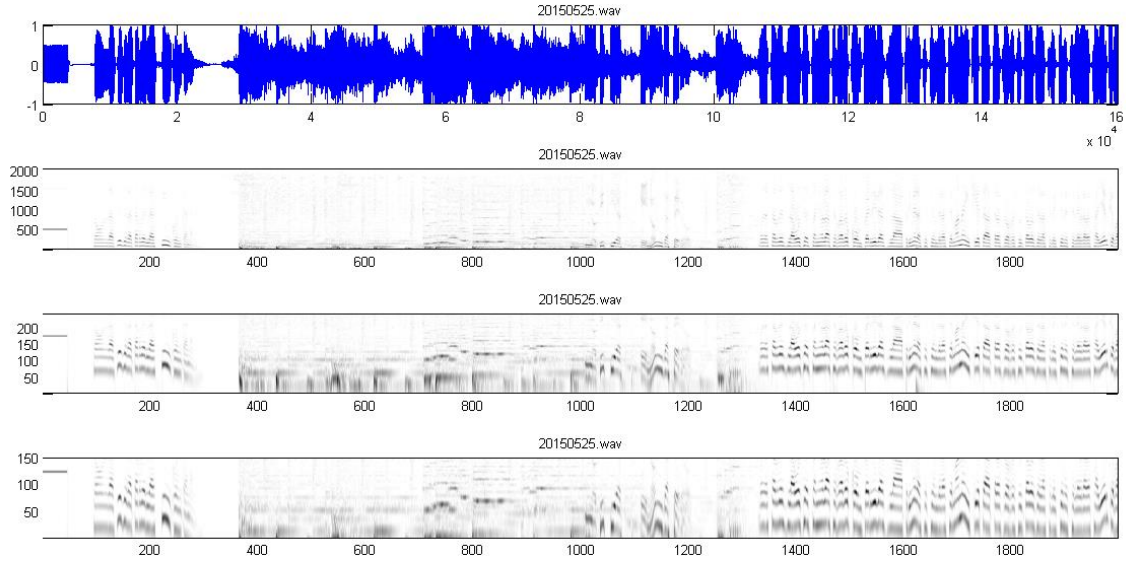

Figura 3-6: Representación alineada en el tiempo de la señal de ejemplo y de los diferentes melgramas estudiados.

3.5 Cálculo de las correlaciones

Como ya hemos comentado anteriormente, la idea en la que se basa este TFG es la captura de las trayectorias armónicas de la señal de voz, ya que estas varían en frecuencia. Este fenómeno resulta en una alta correlación cuando comparamos los patrones espectrales de dos tramas de audio sucesivas y en esto nos basaremos para la clasificación de la señal en voz o música.

Este proceso se va a llevar a cabo calculando la correlación cruzada de cada una de las columnas de la matriz $melgrama$ con sus columnas cercanas. Además, al calcular la correlación cruzada de dos vectores, estos se desplazarán un numero de posiciones definidos por valor del lag.

El valor de r_{xcorr} se define como la máxima correlación cruzada en un rango de lags

$$r_{xcorr}(X_t, X_{t+offset}) = \max R_{X_t, X_{t+offset}}\ (l)$$

En el caso de nuestro algoritmo, la correlación de un determinado vector se calcula con los 4 vectores de su derecha y los 4 vectores su izquierda. Siendo así el valor de $offset$ = [-4 -3 -2 -1 0 1 2 3 4]. Además se tendrá en cuenta un desplazamiento de los vectores de 10 muestras hacia arriba y 10 muestras hacia abajo y por eso se tomará un valor de lag = [$-maxlag : maxlag$] donde $maxlag = 10$.

El resultado del cálculo de la correlación de cada vector con sus vecinos resulta en una matriz $correlación$ de 9 columnas y 21 filas y por lo tanto el vector r_{xcorr} será el máximo de cada columna.

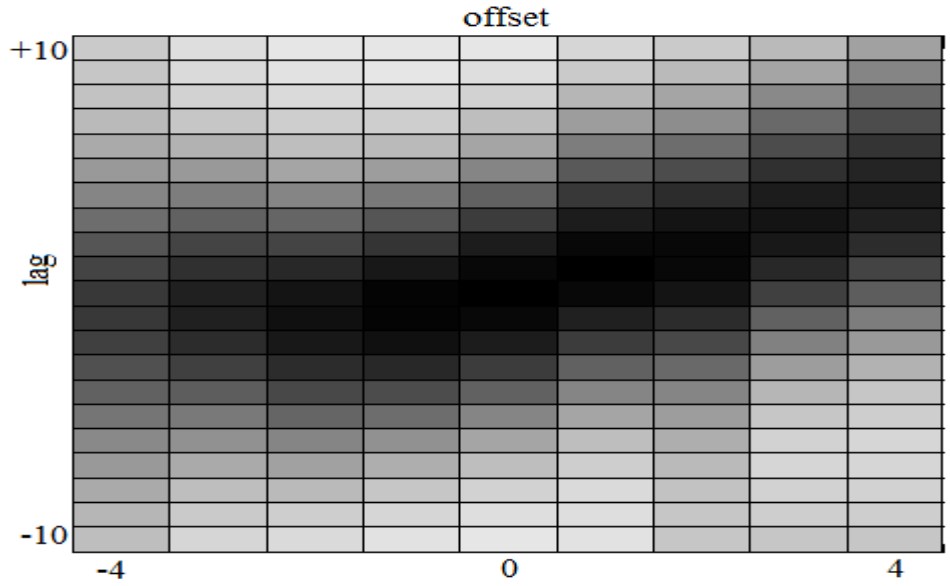

Figura 3-7: Matriz correlación resultante de calcular la correlación a un determinado vector de la matriz melgrama.

Además también se calcula

$$r(X_t, X_{t+offset}) = R_{X_t, X_{t+offset}}(0)$$

Como hemos visto anteriormente (sección 2.3), el valor de $r_{xcorr} - r$ es un indicador de la presencia de voz. Para señales de música, la correlación cruzada va a tener el máximo en lag 0 por lo tanto el valor de $r_{xcorr} - r = R(0) - r = r - r = 0$ y para señales de voz (l) va a tener el valor máximo en algún $l \neq 0$ y por consiguiente el valor de $r_{xcorr} - r$ será un valor positivo.

3.6 Estudio de las trayectorias y score

El cálculo del valor de $r_{xcorr} - r$ arroja buenos resultados y nos permite la detección de voz frente a música de una manera satisfactoria. Pero en algunas situaciones (figura 3-8) se presentan valores incorrectos, como por ejemplo valores de Score altos en zonas de música o valores de Score bajos en zonas donde hay voz.

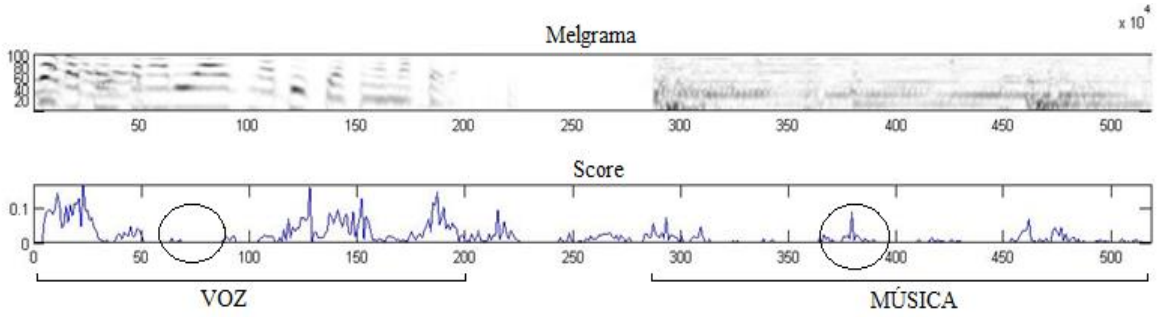

Figura 3-8: Fallos en el Score obtenido. Valores alto en zonas de música y valores bajos en zonas de voz.

Se llevó a cabo un análisis inicial a las trayectorias y se apreció que para zonas donde había voz, y se obtenía un Score alto (Detección correcta), las trayectorias tenían forma como las de la figura 3-9:

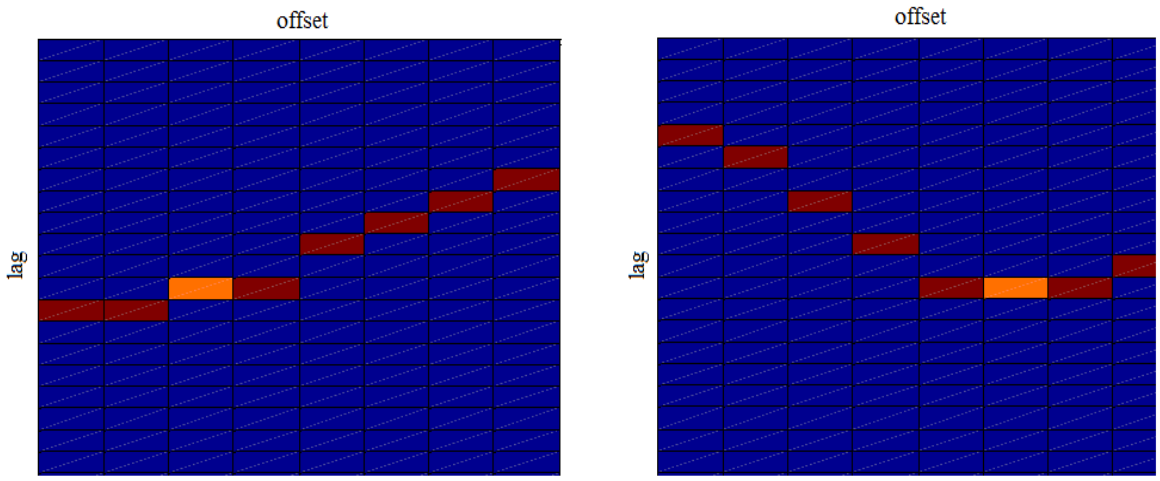

Figura 3-9: Dos ejemplos de trayectorias para tramos de voz.

Sin embargo para tramos de la señal donde no había voz y se obtenía un valor del Score alta, las trayectorias eran como las de la figura 3-10:

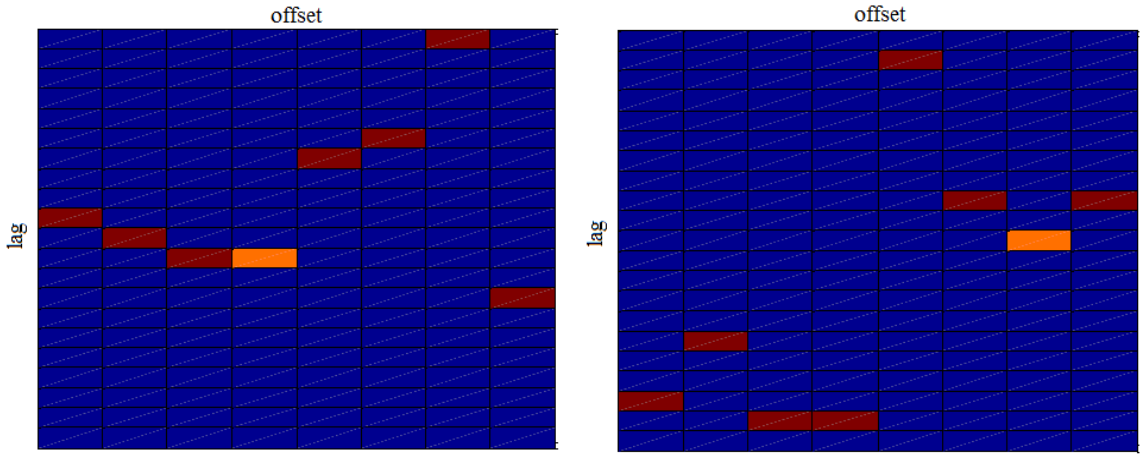

Figura 3-10: Dos ejemplos de trayectorias para tramos de no voz con un Score alto.

Estas dos observaciones de las trayectorias para voz y música eran exactamente lo que buscábamos, y confirmaban la hipótesis de continuidad de trayectorias si hay voz y discontinuidad cuando no la hay.

Por este motivo hemos llevado a cabo un estudio de la matriz de correlaciones que nos permita eliminar situaciones como las descritas anteriormente y por consiguiente mejorar el rendimiento de nuestro algoritmo. Nuestra trayectoria candidata (la que vamos a estudiar) es la secuencia de máximos de la matriz correlación, es decir, el vector r_{xcorr}.

Para realizar el estudio de las trayectorias hemos seguido estos pasos:

Paso 1. Descartar valores que son menores que el 30% de la dinámica.

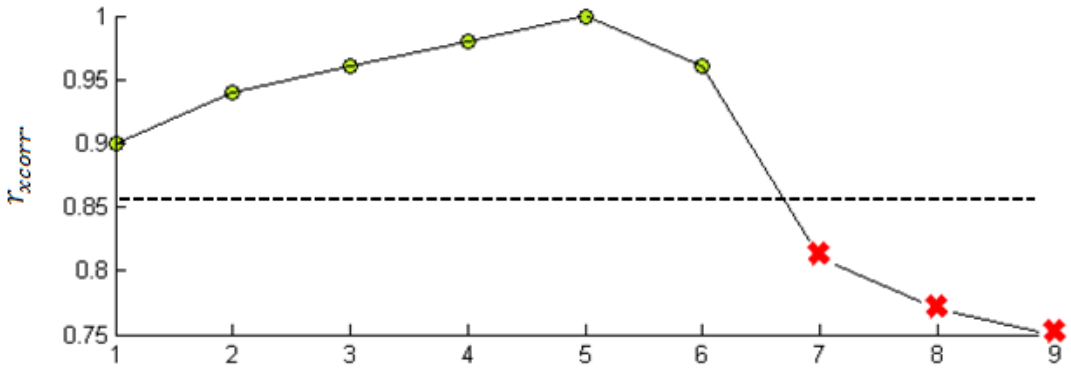

Figura 3-11: Descarte de valores menores que el 30% de la dinámica.

En el caso de la figura 3-11 descartaríamos los 3 últimos valores y continuaríamos el estudio de trayectorias con los valores restantes.

Paso 2. Descartar valores que cambian más del 20% del valor anterior. Para esto, localizamos el valor máximo y vamos recorriendo el vector hacia la derecha y hacia la izquierda viendo si se produce un cambio > 20%.

Paso 3. Cálculo del Score.

Si el número de puntos restantes es ≤ 4

$$Score = \frac{\sum Puntos}{4}$$

Si el número de puntos > 4

$$Score = \frac{\sum Puntos}{Número\ de\ puntos}$$

Paso 4. Pasamos el valor del score a escala logarítmica.

$$Score_{log} = log10(Score)$$

3.7 Detector de pitch

Para mejorar la detección de voz, se pretende añadir al algoritmo inicial dos nuevas características.

La primera característica es el pitch, que es la frecuencia fundamental a la que vibran las cuerdas vocales y es uno de los parámetros que mejor caracterizan la voz de un locutor. Y la segunda, que será determinante para la detección de voz, es lo que hemos denominado "fiabilidad del pitch".

3.7.1 Estimación del pitch y fiabilidad

La estimación del pitch se realiza mediante el cálculo de la correlación. Primero se calcula la auto correlación de cada tramo de la señal, y posteriormente se obtiene el máximo local (excluyendo el máximo global) de esta. El cálculo de dicho máximo local se ha realizado mediante la función $findpeaks$ de Matlab.

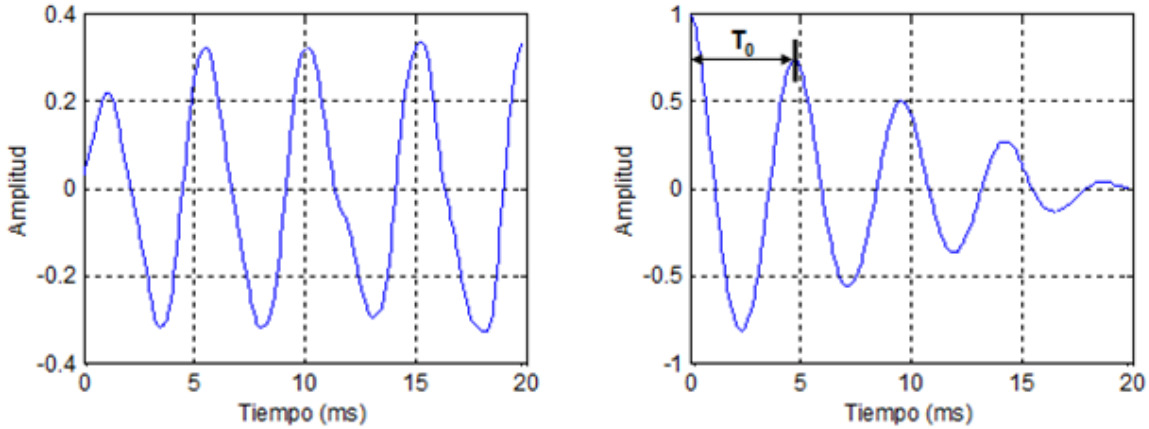

Figura 3-12: Método de estimación de pitch por correlación.

Una vez identificado el valor de ese máximo, que coincide con el valor de T_0 , podemos calcular, el valor de la frecuencia fundamental f_0 o pitch de la forma $f_0 = \frac{1}{T_0}$.

La "fiabilidad del pitch" se obtiene como el valor del primer pico distinto de cero.

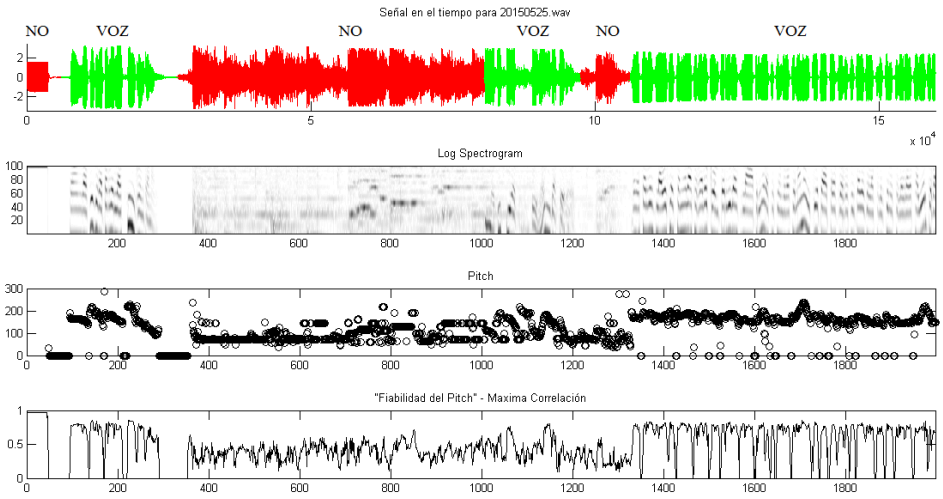

Figura 3-13: Representación del pitch y de su "fiabilidad" para el audio de ejemplo.

En la figura 3-13 podemos observar la señal en el tiempo (con valores en verde para voz y valores en rojo para no voz), seguido de la representación del melgrama. La tercera gráfica muestra el resultado del detector de pitch, seguido de la "fiabilidad" del pitch (máxima correlación).

3.7.2 Corrección por pitch constante

Una vez obtenidos los valores del pitch y de la fiabilidad del pitch, se probó el algoritmo diseñado hasta el momento con una base de datos más pequeña creada por mí, que contenía 30 ejemplos de programas de radio más cortos obtenidos de la base de datos inicial.

Al analizar los resultados obtenidos para estos ejemplos, constatamos la aparición de valores de pitch de aproximadamente 900 Hz. Analizando el audio, se verificó que estos valores correspondían con los pitidos que se suelen utilizar para marcar las horas en punto en los programas de radio (figura 3-14).

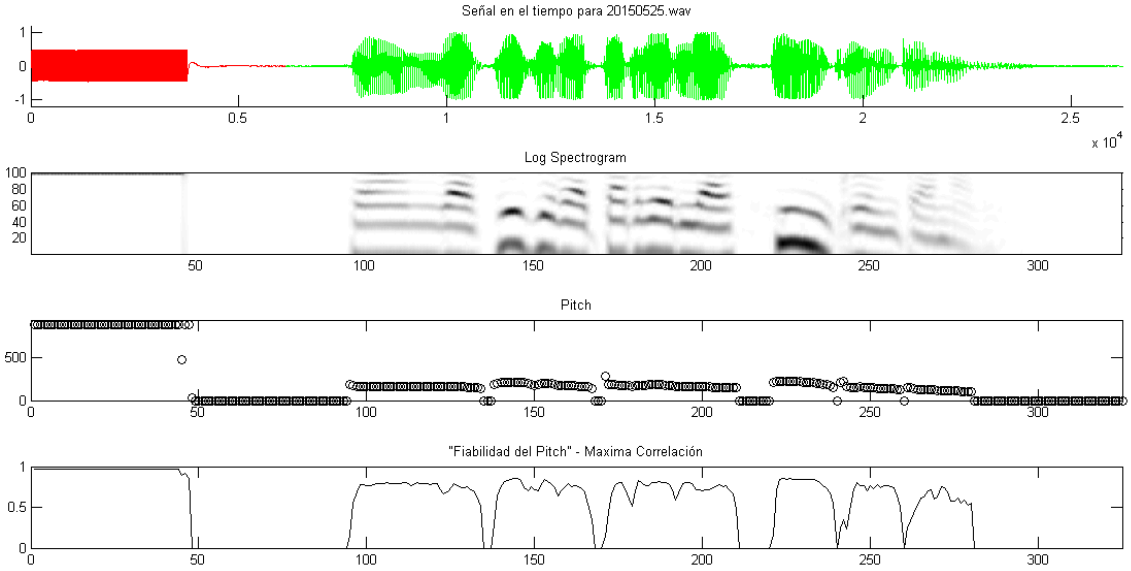

Figura 3-14: Pitch y fiabilidad del pitch para un audio en el que aparece un pitido

Procedimos entonces a eliminar estos valores de pitch "anormales" y que podían ser confusos para el diseño del algoritmo. Para ello, sabiendo que la distribución de la frecuencia fundamental de los hombres varía entre 50 y 200 Hz y las mujeres entre 150 y 400 Hz (figura 3-15), hicimos algo tan sencillo como colocar un umbral (en este caso, 500 Hz), a partir del cual el tramo corresponderá con algún tipo de ruido como el pitido que estamos estudiando ya que ningún ser humano puede llegar a producir voz a esos niveles tan altos de pitch.

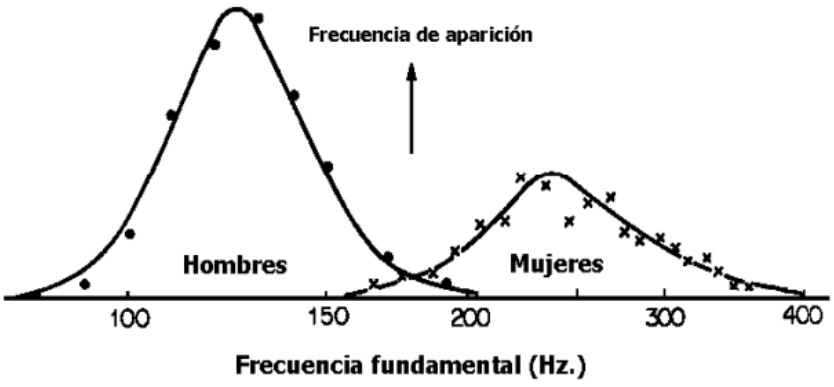

Figura 3-15 Distribución de la frecuencia fundamental de hombres y mujeres.

Por lo tanto y como se aprecia en la figura 3-16, nuestro detector de pitch, anularía los valores de pitch mayores de 500 Hz, dejando únicamente valores de pitch que pueden darse en voz humana.

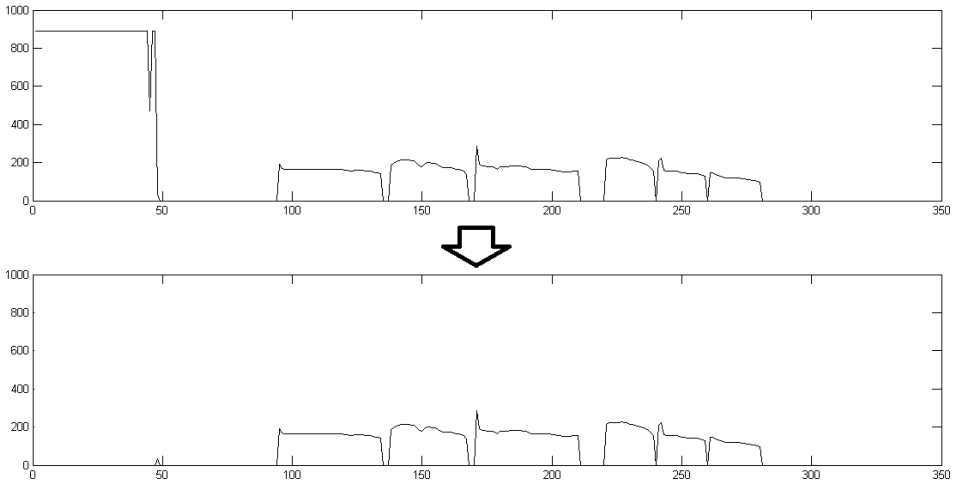

Figura 3-16: Eliminación del pitido para el audio de ejemplo de la Figura 3-14 de ejemplo.

Otra observación, aún más importante, fue que se encontraron muchos valores bajos de Score para zonas donde había voz clara y por lo tanto se detectaban como no voz. Como se muestra en la figura 3-17, en el tramo de la señal seleccionado hay voz, la fiabilidad el pitch es muy alta, y sin embargo el Score obtenido es muy bajo.

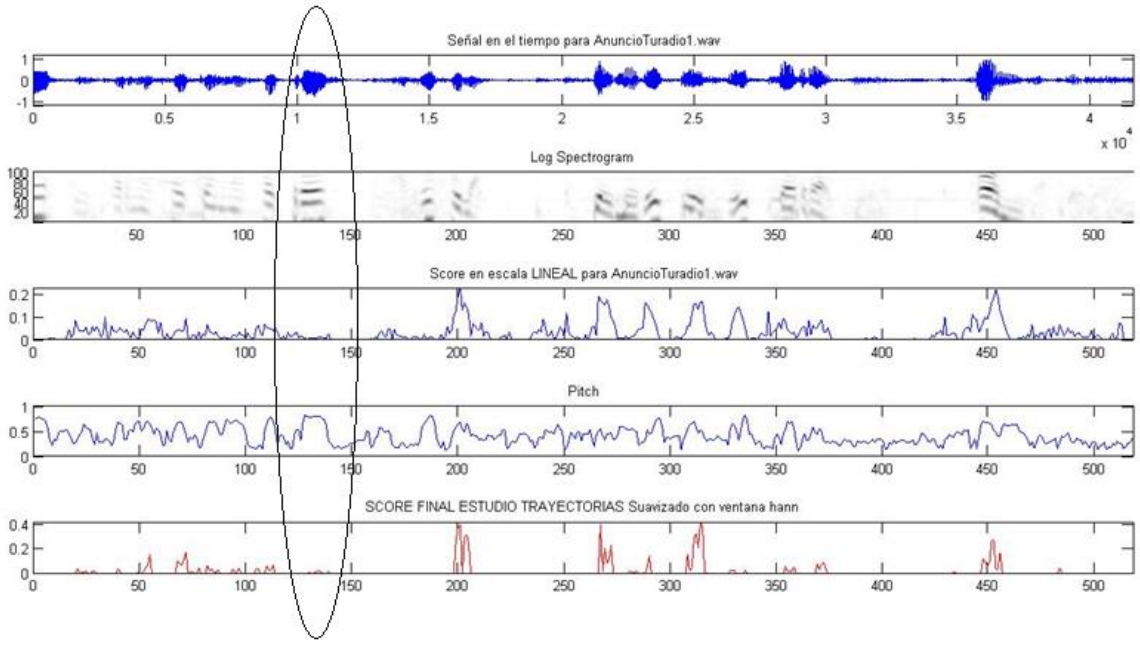

Figura 3-17: Ejemplo de un tramo de la señal donde claramente hay voz, la fiabilidad el pitch es muy alta, pero el Score obtenido es bajo.

Después de realizar una selección de ejemplos donde ocurría esto, se observó que la mayoría de ellos era debido a que el tramo tenía un pitch que variaba muy poco, el pitch

era casi constante. A partir de eso, se procedió al diseño del algoritmo de corrección por pitch constante.

La idea que se implementó fue la de detectar segmentos de al menos 5 tramas donde el pitch varía poco (menos del 10%) en todo el segmento y además la fiabilidad del pitch en todos ellos fuera mayor que un determinado umbral. Esos tramos, fueron marcados como VOZ, independientemente de lo que indicase el Score de trayectorias.

Los resultados arrojados por el algoritmo de detección de pitch constante quedan reflejados en el ejemplo mostrado en la figura 3-18.

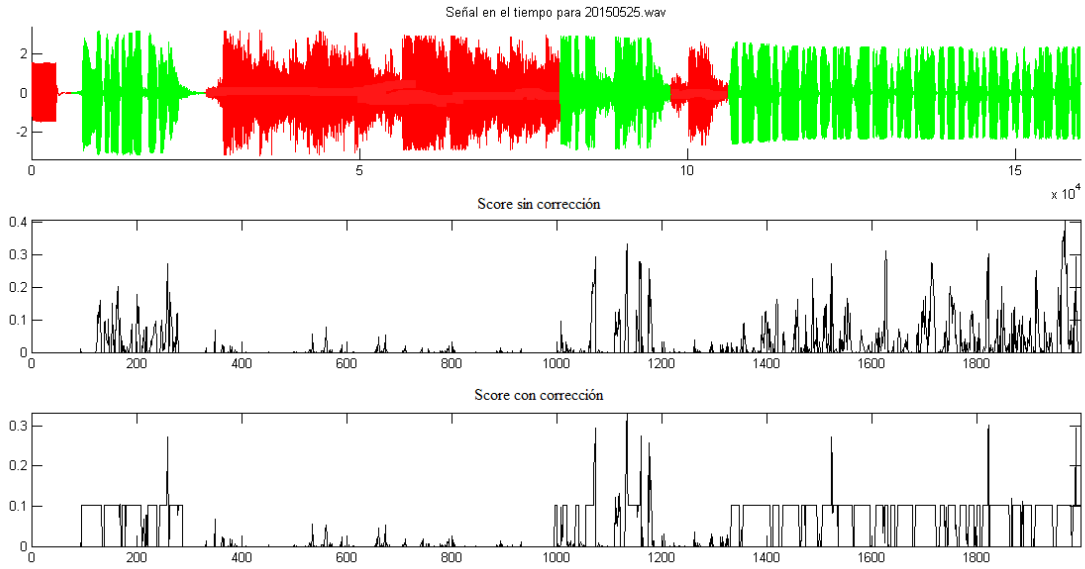

Figura 3-18: Score instantáneo antes de aplicar la corrección por pitch constante y Score instantáneo resultante de aplicar la corrección por pitch constante en el audio de ejemplo.

3.8 Score Final

3.8.1 Score Instantáneo

Una vez tenemos el Score instantáneo, el objetivo es obtener un vector de decisión que tome valores 1 (cuando se determine que hay voz) y 0 (cuando se determine que no hay voz) y que nos permita comparar esa decisión con las etiquetas de la base de datos.

Para calcular ese vector de decisión, necesitamos obtener el valor de un umbral. Los valores del Score instantáneo que estén por encima de ese umbral serán tomados como voz y los que estén por debajo serán tomados como no voz. Para llegar a obtener el valor de ese umbral se han llevado a cabo las siguientes operaciones:

En primer lugar, calculamos el histograma del Score instantáneo, y seguidamente, calculamos la suma acumulada de los valores de dicho histograma (figura 3-19).

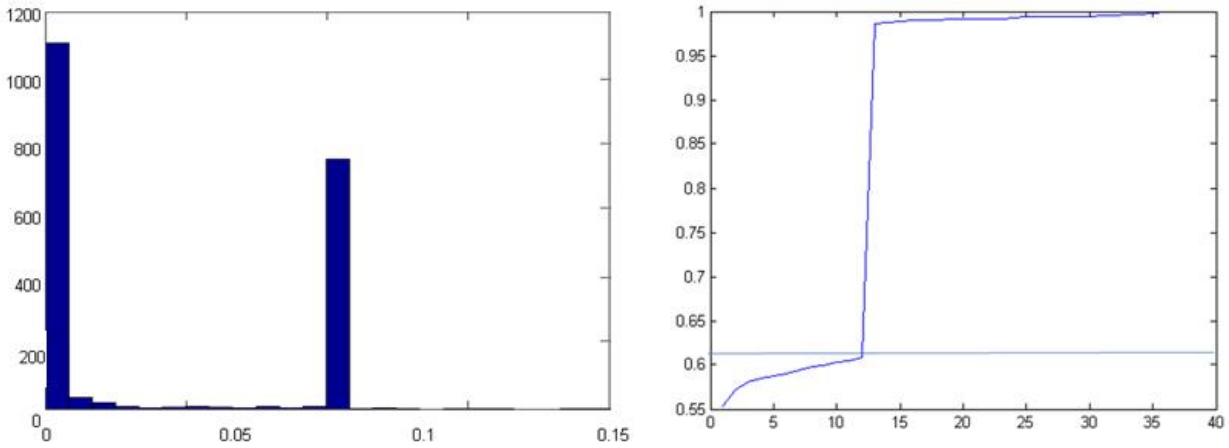

Figura 3-19: Histograma del Score instantáneo (izquierda) y su suma acumulada (derecha) para el audio de ejemplo.

Y finalmente, calculamos el valor que corresponde al 10% de esa suma acumulada y comprobamos a qué bin del histograma corresponde. Este valor del bin del histograma será el umbral que nos permitirá clasificar la señal en voz y no voz.

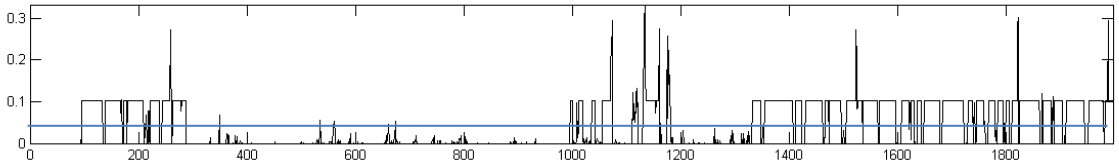

Figura 3-20: Umbral del Score que clasificará entre voz y no voz para el audio de ejemplo.

Así obtenemos (figura 3-21) un vector de unos y ceros. Todo lo que esté por encima de ese umbral lo tomaremos como voz (1) y lo que esté por debajo como no voz (0).

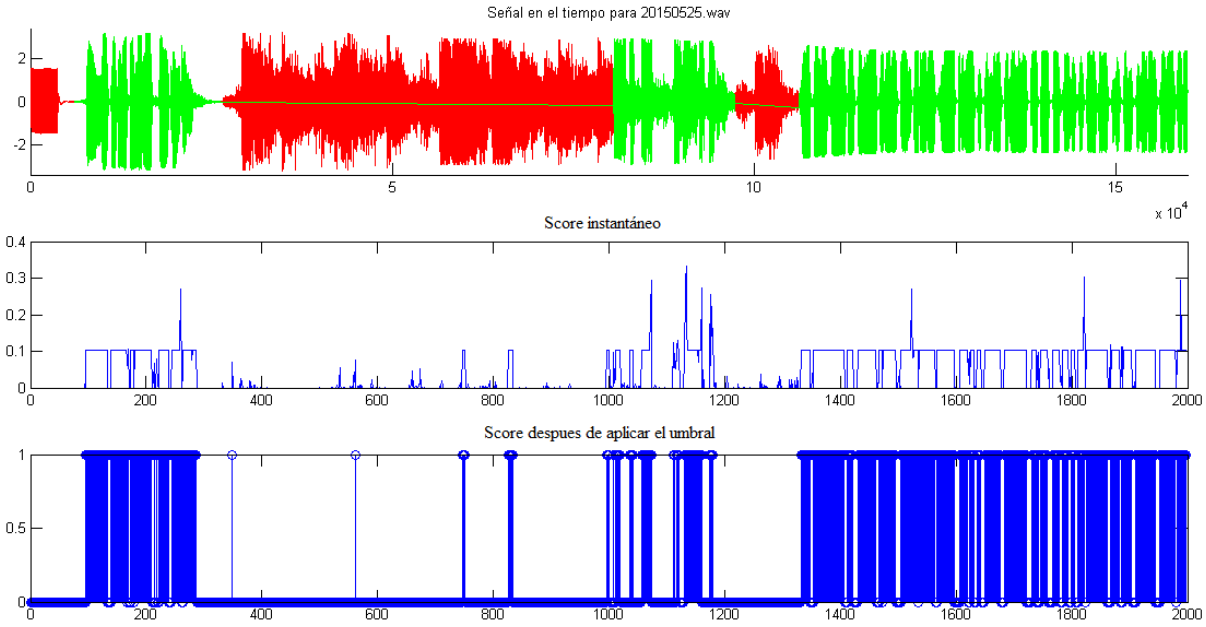

Figura 3-21: Score obtenido después de aplicar el umbral calculado anteriormente (1 para voz y 0 para no voz).

3.8.2 Decisión final

Como se puede apreciar en la figura 3-21 hay zonas donde se ve claramente que hay voz, pero "aparecen incrustados" valores clasificados como no voz. Para solventar este problema, vamos a tener en cuenta el entorno de cada trama, ya que por ejemplo no puede darse una única trama de voz entre tramas de no voz.

Para tener en cuenta el entorno de cada trama calculamos, para cada valor del vector, la suma de los 100 valores vecinos (50 a la derecha y 50 a la izquierda). Para reducir el coste computacional, el cálculo de esta suma se puede realizar como:

$$S = Suma - ultimo + anterior$$

De esta forma se obtiene una función como la de la figura 3-22, en la que, como se puede apreciar, si hacemos una correcta elección del umbral, obtenemos una clasificación de la señal en voz y no voz muy precisa.

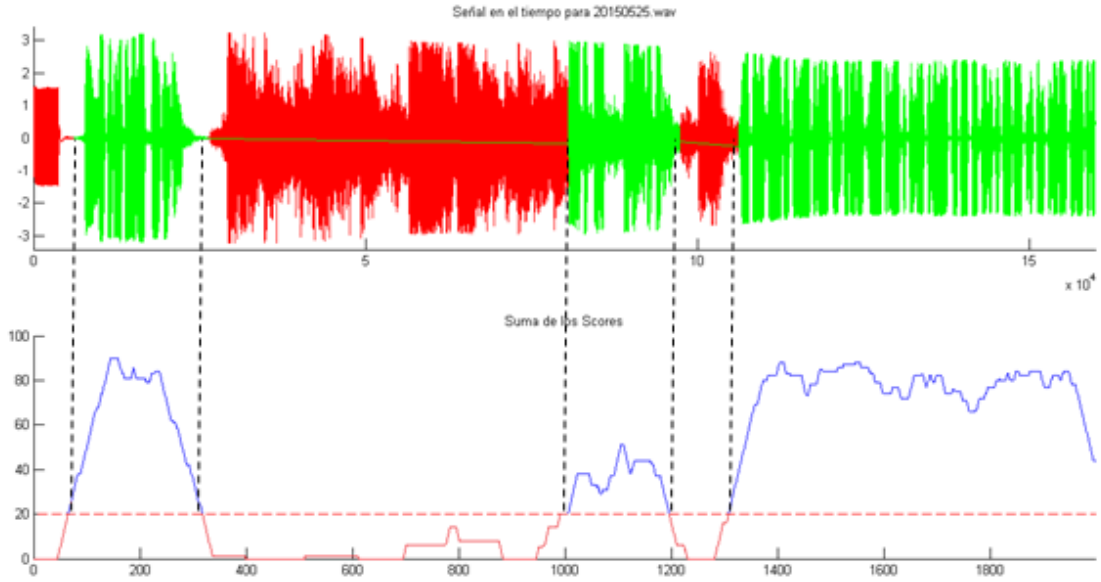

Figura 3-22: Resultado del cálculo del vector suma (suma de los valores vecinos (50 a la derecha y 50 a la izquierda) de cada valor del Score de 1s y 0s).

La figura 3-23 muestra un resumen de todo el proceso seguido:

- Señal en el tiempo (gráfica superior).
- Score instantáneo (segunda gráfica).
- Vector suma de los valores del Score instantáneo y el umbral que determinara que es voz (tercera gráfica).
- Decisión FINAL después de aplicar el umbral a la suma (penúltima gráfica).
- Etiquetas reales obtenidas de la base de datos (gráfica inferior).

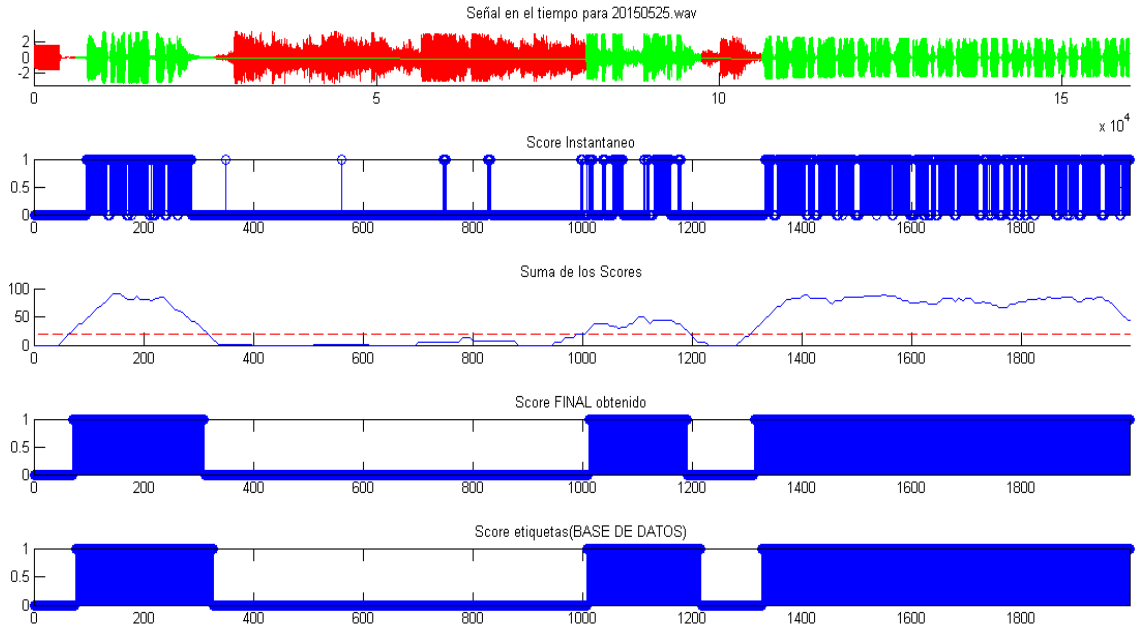

Figura 3-23: Resumen del cálculo del Score final.

4 Pruebas y resultados

Ya hemos definido anteriormente cómo es la estructura de nuestra base de datos y cómo calculamos el Score que nos permitirá comparar con las etiquetas de la misma.

Las pruebas que vamos a realizar van a consistir en la comparación de dos vectores, el vector de etiquetas .voz de la base de datos, con el vector de decisión final obtenido mediante nuestro algoritmo.

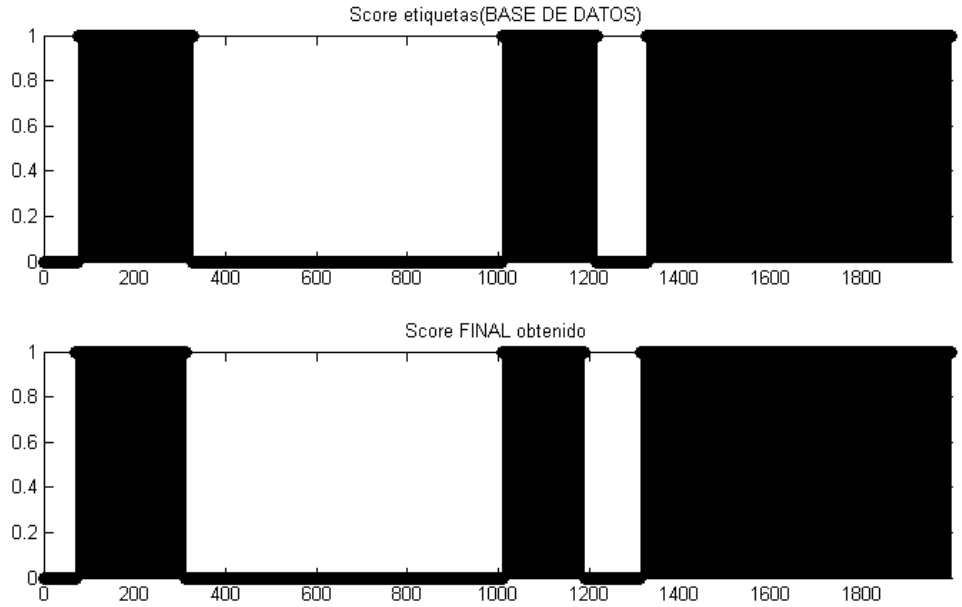

Figura 4-1: Score obtenido de la base de datos VS Score obtenido mediante nuestro algoritmo para un determinado audio de ejemplo.

Dicha comparación, será medida con los siguientes parámetros:

- Porcentaje de acierto total
- Porcentaje de acierto en voz
- Porcentaje de acierto en no voz

Se considerará acierto si el valor de la etiqueta de la base de datos coincide con el valor de la decisión final que determina el algoritmo diseñado.

A continuación se muestran los resultados gráficos de cuatro audios de ejemplo (figuras 4-2, 4-3, 4-4 y 4-5) y los porcentajes de acierto para cada uno de los cuatro programas de radio objeto de análisis (tabla 4-1).

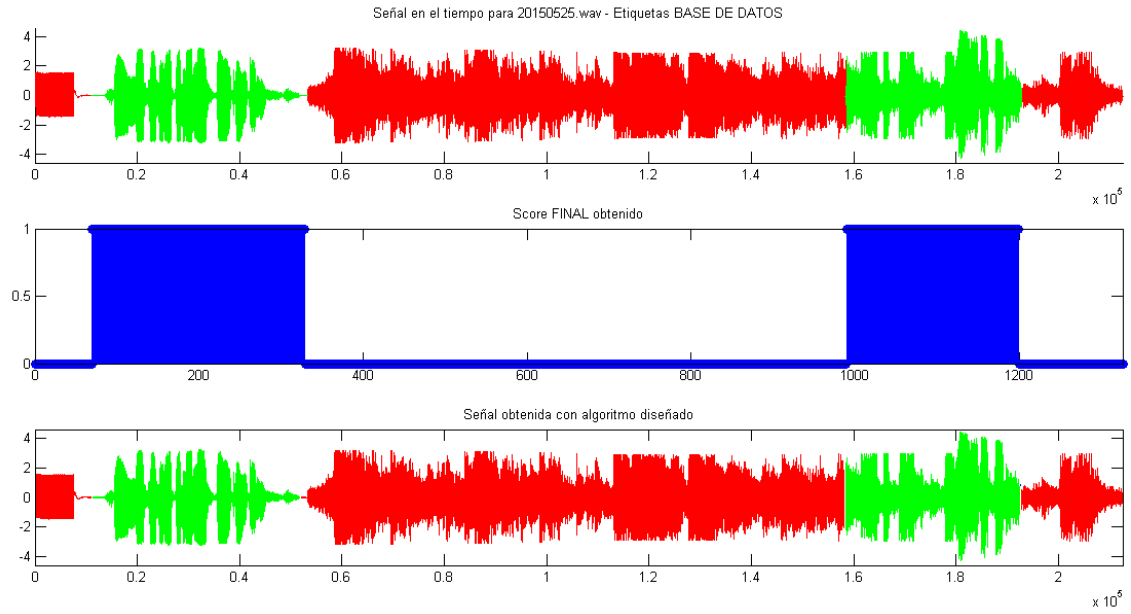

Figura 4-2: Resultados de nuestro algoritmo para el audio de ejemplo 1.

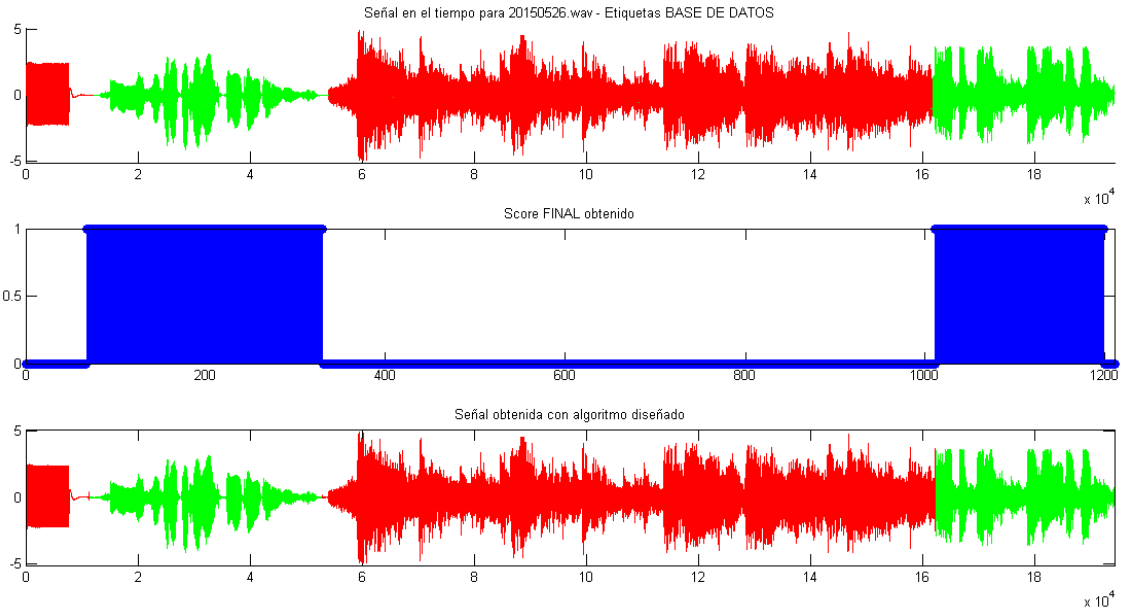

Figura 4-3: Resultados de nuestro algoritmo para el audio de ejemplo 2.

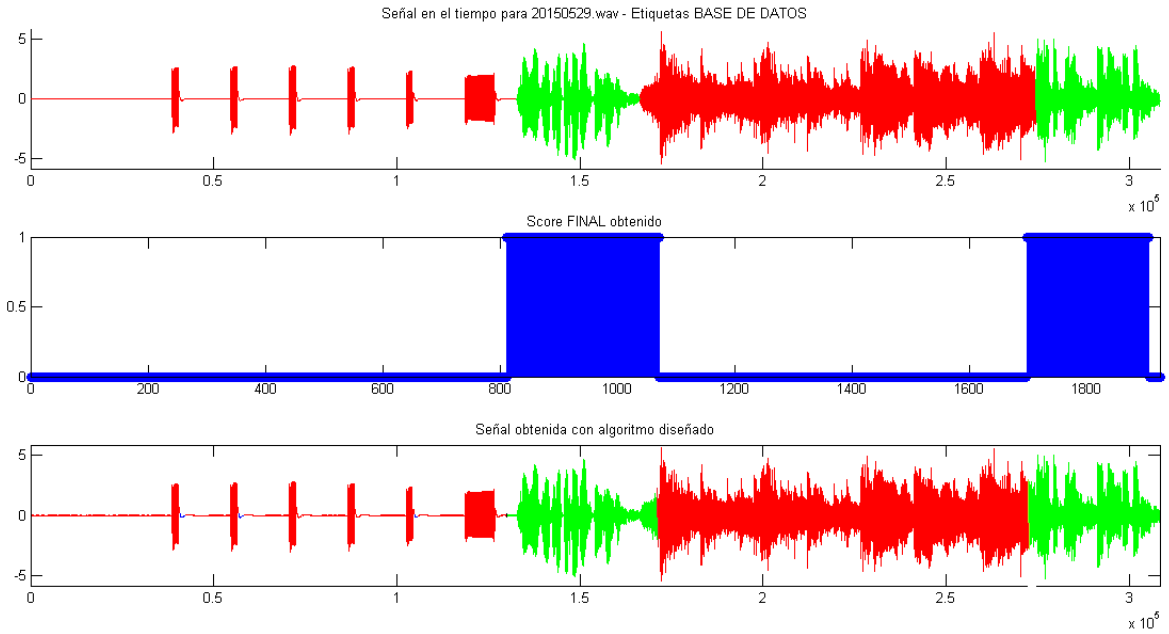

Figura 4-4: Resultados de nuestro algoritmo para el audio de ejemplo 3.

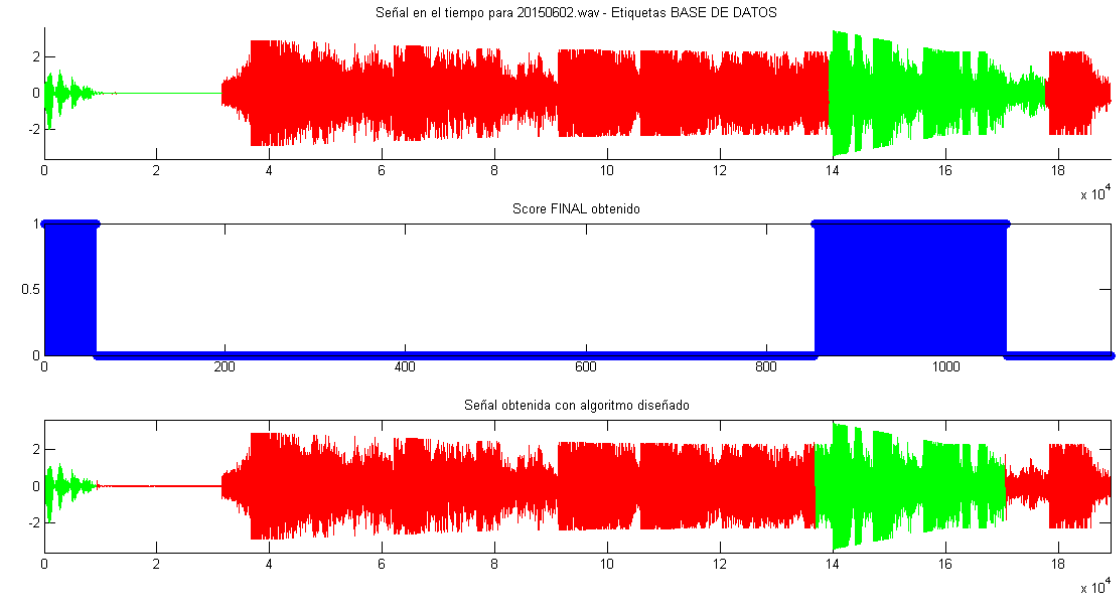

Figura 4-5: Resultados de nuestro algoritmo para el audio de ejemplo 4.

Tabla 4-1: Resultados para los distintos programas

	Archivos de audio	Porcentaje de acierto en voz	Porcentaje de acierto en no voz	Porcentaje de acierto total
JULIA EN LA ONDA	JO_20150525.wav	92.94%	98.34%	94.96%
	JO_20150526.wav	93.17%	96.24%	93.25%
	JO_20150527.wav	91.18%	97.76%	92.21%
	JO_20150528.wav	94.78%	98.12%	95.01%
	JO_20150529.wav	94.23%	96.76%	95.46%
	JO_20150601.wav	92.31%	95.67%	93.45%
	JO_20150602.wav	94.1%	97.79%	94.99%
	JO_20150603.wav	91.89%	98.11%	92.69%
	JO_20150604.wav	91.23%	95.43%	91.99%
	JO_20150605.wav	93.39%	97.54%	94.03%
	Total	**92.92%**	**97.18%**	**93.8%**
HOY POR HOY	HH_20150525.wav	91.53%	97.13%	92.54%
	HH_20150526.wav	94.18%	99.02%	95.77%
	HH_20150527.wav	92.54%	98.52%	93.09%
	HH_20150528.wav	94.63%	98.73%	95.67%
	HH_20150529.wav	91.99%	95.77%	93.25%
	HH_20150601.wav	93.88%	96.03%	94.98%
	HH_20150602.wav	91.45%	97.7%	92.73%
	HH_20150603.wav	93.2%	96.92%	94.28%
	HH_20150604.wav	92.5%	96.21%	93.48%
	HH_20150605.wav	92.91%	96.28%	93.13%
	Total	**92.98%**	**97.23%**	**94.04%**
LA MAÑANA	LM_20150525.wav	90.38%	97.56%	91.43%
	LM_20150526.wav	92.56%	95.81%	92.65%
	LM_20150527.wav	91.93%	96.65%	92.58%
	LM_20150528.wav	94.22%	98.04%	95.23%
	LM_20150529.wav	92.7%	97.19%	94.88%
	LM_20150601.wav	93.82%	96.12%	94.19%
	LM_20150602.wav	94.78%	97.8%	95.52%
	LM_20150603.wav	92.45%	96.82%	93.32%
	LM_20150604.wav	92.71%	95.44%	93.55%
	LM_20150605.wav	94.06%	96.94%	94.99%
	Total	**92.96%**	**96.84%**	**93.88%**
MÁS DE UNO	MU_20150525.wav	93.48%	98.12%	94.16%
	MU_20150526.wav	92.95%	97.77%	93.40%
	MU_20150527.wav	91.83%	95.39%	92.74%
	MU_20150528.wav	94.22%	98.02%	95.67%
	MU_20150529.wav	92.29%	96.3%	93.93%
	MU_20150601.wav	91.99%	95.97%	93.04%
	MU_20150602.wav	92.84%	96.9%	93.49%
	MU_20150603.wav	92.13%	97.74%	93.18%
	MU_20150604.wav	92.44%	96.19%	93.82%
	MU_20150605.wav	90.87%	96.89%	92.11%
	Total	**92.5%**	**96.93%**	**93.43%**

5 Conclusiones y trabajo futuro

5.1 Conclusiones

Este trabajo tenía como objetivos tanto el diseño y desarrollo de un algoritmo de detección de voz como la creación de una base de datos que nos permitiera conocer la bondad de este algoritmo.

- Hemos diseñado un algoritmo de detección de voz que funciona de forma eficiente. Es un algoritmo sencillo e intuitivo que puede ser útil en multitud de aplicaciones y además tiene un coste computacional bastante bajo en comparación a otros detectores de voz.

- Dicho detector de voz, es realmente eficiente en situaciones en las que hay que identificar la voz cantada como música, algo que la mayoría de los detectores de voz clásicos no hacen.

- Hemos creado un módulo de normalización de ganancia desde cero. Su funcionamiento es asombroso, y de hecho, va a ser utilizado para otras líneas de investigación dentro del grupo ATVS.

- Hemos implementado una función para la representación espectral de los melgramas que puede ser útil en otras aplicaciones de procesado de voz y también un detector de pitch que además tiene un medidor de "fiabilidad del pitch". Ambas han sido "creadas desde cero".

- La base de datos de audio broadcast creada para este TFG puede ser útil para futuras investigaciones. Su etiquetado, no ha sido lo suficientemente preciso ya que las transiciones entre voz y música estaban etiquetadas de forma distinta por los miembros del grupo, lo que ha producido una merma de la eficiencia del algoritmo.

5.2 Trabajo futuro

- Un primer aspecto a tener en cuenta, sería el ampliar la base de datos utilizada. Además podría ser interesante obtener audios de otros medios como la televisión para poder así, ver y estudiar la diferencia entre señales de audio de radio y señales de audio de televisión. También se podrían extraer audios en diferentes idiomas para evaluar la robustez del algoritmo en un idioma diferente al castellano.

- Mejorar la precisión en el etiquetado de esta base datos. Además sería recomendable una evaluación de la bondad del algoritmo que no sea tan restrictiva en las transiciones entre voz y no voz. Por ejemplo, dando un margen de error de unos 20 mS ya que el etiquetado del audio es un proceso realizado por personas y no es del todo preciso.

- También podría ser de utilidad implementar un detector música. Sería de gran utilidad que funcionase en paralelo a este algoritmo de detección de voz y que permitiera una segmentación de audio óptima. Para ello se propone el estudio de algunas características de la señal de música como son la melodía, armonía y ritmo entre otras. En las partes de nuestro algoritmo donde se detecta no voz, se utilizaría el detector de música para así diferenciar entre zonas con música y ruido consiguiendo así un clasificador de MÚSICA/VOZ/RUIDO eficaz.

- Finalmente, se podría pensar en una mejora del algoritmo mediante el uso de HMMs (Modelos Ocultos de Markov). Los HMMs son utilizados en la actualidad en sistemas donde el modelado tiene una dependencia del tiempo, como pueden ser los sistemas de reconocimiento fonético y del habla. Este proceso requiere un proceso de entrenamiento, por lo tanto, iría ligado a la ampliación de la base de datos.

Referencias

[1] Reinhard Sonnleitner, Bernhard Niedermayer, "A simple and effective spectral feature for speech detection in mixed audio signals", in *Proc. Of the 15th Int. Conference on Digital Audio Effects (DAFx-12),* York, UK, September 17-21, 2012.

[2] Javier Franco Pedroso, Ignacio López-Moreno, Doroteo T. Toledano, and Joaquín González-Rodriguez, "ATVS-UAM System Description for the Audio Segmentation and Speaker Diarization Albayzin 2010 Evaluation", in *FALA 2010 , VI Jornadas en Tecnología del Habla and II Iberian SLTech Workshop,* 2010.

[3] Javier Franco-Pedroso, Elena Gomez Rincon, Daniel Ramos and Joaquín González-Rodriguez, "ATVS-UAM System Description for the Albayzin 2014 Audio Segmentation Evaluation", in *VII Jornadas en Tecnología del Habla and III Iberian SLTech Workshop*, November 19-21, 2014.

[4] Diego Castán, Alfonso Ortega, Antonio Miguel and Eduardo LLeia, "Audio segmentation-by-classification approach based on factor analysis in broadcast news domain", in *EURASIP Journal on Audio, Speech, and Music Processing*, 2014.

[5] C. Liu, L. Xie, and H. Meng, "Classification of musicand speech in mandarin news broadcasts," in *9th National Conference on Man-Machine Speech Communication(NCMMSC),* China, 2007.

[6] J. Pohjalainen, T. Raitio, and P. Alku, "Detection of shouted speech in the presence of ambient noise," in *Proceedings 12th Annual Conf. Int. Speech Communication Association*, Italy, 2011.

[7] B. Schuller, G. Rigoll, and K. Lang M. "Discrimination of speech and monophonic singing in continuous audio streams applying multi-layer support vector machines," in *International Conference on Multimedia Computing and Systems,* 2004, pp. 1655–1658.

[8] K. Seyerlehner, T. Pohle, M. Schedl, and G. Widmer, "Automatic music detection in television productions," in *Proceedings of the Int. Conf. on Digital Audio Effects*, Bordeaux,France, 2007.

[9] Lee Ngee Tan, Bengt J. Borgstrom and Abeer Alwan, "voice activity detection using harmonic frequency componets in likelihood ratio test", *ICASSP 2010.*

[10] J. Ramírez, J. M. Górriz and J. C. Segura, "Voice Activity Detection. Fundamentals and Speech Recognition System Robustness", in Bentham *Science Publishers*, pp. 46-59, 2010.

[11] Javier Ortega García. Asignatura "Tratamiento de Señales de Voz y Audio", Escuela Politécnica Superior, Universidad Autónoma de Madrid, 2015.

[12] Benjamin Elizalde, Gerald Friedland, "Lost in segmentation: Three approaches for speech/non-speech detection in consumer-produced video", in *Multimedia and Expo (ICME), 2013 IEEE International Conference*, San Jose, 2013.

[13] Ananya Misra, "Speech/Nonspeech Segmentation in Web Videos," in *Proceedings of Interspeech*, 2012.

[14] T. Hain and P. C. Woodland, "Segmentation and classification of broadcast news audio," in *Proceedings of ICSLP*, 1998, pp. 2727–2730.

[15] M. Graciarena, A. Alwan, D. Ellis, H.Franco, L. Ferrer, J. Hansen, A. Janin, B.-S. Lee, Y. Lei, V. Mitra, N. Morgan, S. O. Sadjadi, T.J. Tsai, N. Scheffer, L. N. Tan, B. Williams, "All for One: Feature Combination for Highly Channel-Degraded Speech Activity Detection", *Interspeech*, Lyon, 2013, pp. 709-713.

[16] J. Bach, J. Anemueller, and B Kollmeier, "Robust speech detection in real acoustic backgrounds with perceptually motivated features," *Speech Communication*, vol. 53, no. 5, pp. 690–706, 2011.

[17] B. Schuller, B. Schmitt B. J. D. Arsic, S. Reiter, K. Lang M. and G. Rigoll, "Feature selection and stacking for robust discrimination of speech, monophonic singing, and polyphonic music," in *IEEE International Conference on Multimedia & Expo*, 2005, pp. 840–843.