Algoritmo de Segmentación de Habla Independiente de Texto en Uno y Dos Niveles

por

Ricardo Sánchez Jurado

Tesis sometida como requisito parcial
para obtener el grado de

MAESTRO EN CIENCIAS EN EL ÁREA DE CIENCIAS COMPUTACIONALES

en el

Instituto Nacional de Astrofísica, Óptica y Electrónica
Noviembre de 2008
Tonantzintla, Puebla

Supervisada por:

Dr. Carlos Alberto Reyes García
Investigador Titular del INAOE
Dra. María del Pilar Gómez Gil
Investigadora Asociada del INAOE

Resumen

El éxito en procesos como el reconocimiento automático del habla depende en gran manera de la segmentación del habla y su etiquetado, siendo la segmentación un factor muy importante. Existen diferentes esquemas para realizar la segmentación, algunos con restricciones (de texto o hablante) y otros sin restricciones (independiente de texto), además de tener diferentes unidades en que se segmenta el habla (palabras, sílabas, fonemas), dentro de las cuales, la unidad más común son los fonemas. En las técnicas sin restricciones, solo se usan características acústicas de la señal para obtener límites fonéticos sin tener alguna información adicional de ésta. Para realizar el proceso de segmentación, se divide la señal en pequeños fragmentos (frames) que puedan ser manejables, a los cuales se les extraen características usando métodos de codificación de la señal como son los Bancos de Filtros en la escala Mel y usando la Transformada *Wavelet* Estacionaria. Además por cada una de las características se obtienen valores de membresía a los conjuntos difusos Alto, Medio y Bajo, lo que permite detectar transiciones entre fonemas que no son muy claras. En esta tesis se trabajó en un algoritmo de segmentación de habla independiente de texto con diversas características, además se propone una nueva forma de calcular distancias entre características de cuatro frames adyacentes utilizando medidas de distancia como la Euclidiana o la Chebyshev. El análisis de estas distancias permite obtener las instancias de tiempo en las cuales existe un límite fonético, por lo que se definieron nuevas estrategias de selección de límites candidatos realizando la segmentación en uno y dos niveles. Para la segmentación en dos niveles se usaron Algoritmos Genéticos a fin de optimizar los parámetros del algoritmo. En este trabajo se utilizaron dos corpus, uno en inglés y otro en español, logrando 80.28 % de detección correcta en el primer corpus y 82.58 % en el segundo, este desempeño es comparado con trabajos similares de segmentación del idioma inglés.

Abstract

Success in the performance of automatic speech recognition depends, among other issues, from an accurate segmentation of the input signal. Such signal may be divided by words, vowels or phonemes, the last being the most popular. Segmentation may be achieved using different techniques, some restricted by text or speaker and others free of restrictions.

In this research we present a text and speaker-independent algorithm to obtain phonetic boundaries of a speech signal, using only acoustic features. The signal is divided into segments, called frames, small enough to be handled by coding algorithms as Mel Filter Banks or stationary wavelet transforms. Each feature is converted to a fuzzy representation in order to detect transitions among phonemes that, in other way, could not be clearly identified. In addition, we propose a modification in Euclidian and Chebishev distances to calculate feature distances using four adjacent frames. New strategies to select candidates for boundaries in one and two levels are also presented and analyzed. Genetic algorithms are used to optimize some parameters in the proposed algorithm. The algorithm was tested using two different corpuses, one in English and one in Spanish language. A correct segmentation of 80.28 % was obtained for English and 82.58 % for Spanish. This performance is similar to results obtained by other research works using English language.

Agradecimientos

Agradezco todo el apoyo brindado por la Coordinación de Ciencias Computacionales y las facilidades otorgadas para la realización de mi trabajo de tesis.

Gracias a cada uno de mis sinodales: al doctor Leopoldo Altamirano y al doctor Miguel Octavio Arias por sus observaciones; al doctor Luis Villaseñor por sus consejos y observaciones durante las revisiones que me permitieron detectar mejoras para mi trabajo.

Especial agradecimiento a mis asesores: al doctor Carlos Alberto Reyes por transmitirme su conocimiento e interés en el área, por sus consejos y aliento durante todo el desarrollo de mi trabajo; a la doctora María del Pilar Gómez por su incorporación a mi trabajo, por su apoyo durante los tiempos más difíciles y su esfuerzo para sacar adelante este proyecto.

A mi papá por su apoyo incondicional y a mi mamá por su amor y desvelos junto conmigo. A mi hermano Jorge por su fe en mí, a mi hermana Claudia que siempre está conmigo y a Ive por su amor, paciencia y confianza.

Y finalmente agradezco a mi Dios ya que sin Él nada podría hacer, por darme la oportunidad de estudiar una maestría y la sabiduría para realizar las cosas.

"*Todo lo puedo en Cristo que me fortalece*" Fil. 4:13

Dedicatoria

A mis padres, porque siempre han tenido fe en mí y cada logro obtenido es de ellos; cada alegría al igual que cada tristeza la viven junto a mí, por eso les dedico este trabajo con todo mi amor, devolviéndoles un poco de lo mucho que me han dado.

Índice general

1. Introducción **1**

1.1. Problemática 2

1.2. Objetivos 3

1.2.1. Objetivo general 3

1.2.2. Objetivos específicos 3

1.3. Justificación 4

1.4. Descripción del documento 5

2. Conceptos básicos sobre Segmentación y Reconocimiento de Habla **7**

2.1. Producción y percepción del habla 7

2.1.1. ¿Cómo se produce el habla? 8

2.1.2. ¿Cómo se percibe el habla? 11

2.2. Reconocimiento automático del habla 13

2.2.1. Importancia del RAH 14

2.2.2. Dificultades 14

2.2.3. ¿Cómo se hace el RAH? 15

2.3. Segmentación y etiquetado del habla 16

2.3.1. Enfoques de segmentación y etiquetado 17

2.3.2. Problemas en la segmentación independiente de texto 18

2.3.3. Pre-proceso y segmentación 19
2.3.4. Detección de límites 19

3. Análisis del habla y extracción de características 21

3.1. Imitando al oído humano 21
3.2. Análisis usando *Melbanks* 22
3.2.1. Transformada Fourier 22
3.2.2. *Melbanks* 23
3.3. Análisis usando *Wavelets* 24
3.3.1. Transformada *Wavelet* 25
3.3.2. Transformada *Wavelet* Estacionaria 27
3.4. Extracción de características 27

4. Trabajos relacionados 31

4.1. Evaluación del desempeño. 31
4.2. Segmentación de habla con independencia de texto para reconocimiento fonético 32
4.2.1. Algoritmo con características físicas 32
4.2.2. Algoritmo con características espectrales 33
4.2.3. Experimentos y resultados 34
4.3. Método para la segmentación de habla independiente de texto . 34
4.3.1. Implementación 35
4.3.2. Experimentos y resultados 36
4.4. Análisis Multiresolución aplicado en la segmentación fonética independiente de texto 36
4.4.1. Implementación 36
4.4.2. Experimentos y resultados 37

4.5. Segmentación de fonemas no supervisada basada en métodos kernel de máximo margen . . . 37
4.5.1. Implementación . . . 37
4.5.2. Experimentos y resultados . . . 39
4.6. Resumen . . . 39

5. Algoritmo propuesto de segmentación independiente de texto 41
5.1. Lenguaje de programación . . . 42
5.2. Bases de datos utilizadas . . . 42
5.2.1. Corpus DIMEx100 . . . 43
5.2.2. Corpus TIMIT . . . 43
5.2.3. Datos experimentales . . . 44
5.3. Evaluación de desempeño . . . 44
5.4. Descripción general del algoritmo . . . 45
5.5. Pre-procesamiento de la señal . . . 46
5.6. Obtención de características . . . 46
5.6.1. Características usando Bancos de Filtros en la escala Mel . . . 46
5.6.2. Características usando *Wavelets* . . . 47
5.6.3. Características difusas . . . 48
5.7. Cálculo de Distancias . . . 50
5.7.1. Distancia Euclidiana . . . 50
5.7.2. Distancia Chebyshev . . . 51
5.7.3. Distancia Chebyshev modificada. . . . 51
5.7.4. Medidas utilizadas para obtener variaciones entre *frames* 51
5.8. Selección de límites candidatos . . . 54
5.8.1. Detección de límites en un nivel . . . 54

5.8.2. Detección de límites en dos niveles 57
5.8.3. Obtención de parámetros del algoritmo en dos niveles . . 59

6. Resultados **61**

6.1. Análisis de Resultados 62
6.2. Comparación con otros trabajos 66

7. Conclusiones y trabajo futuro **69**

7.1. Conclusiones 69
7.2. Aportaciones 70
7.3. Trabajo futuro 70

Índice de figuras

1.1. El proceso de Reconocimiento Automático del habla [1] 1

2.1. Aparato fonador [10] 9
2.2. Zonas bucales [10] 9
2.3. Partes del oído [11] 12
2.4. Pasos del RAH [1] 16

3.1. Distribución de frecuencias en la cóclea [10] 22
3.2. Escala Mel. 24
3.3. Obtención de características usando *Melbanks*. 24
3.4. Proceso de descomposición y reconstrucción de la señal usando *Wavelets* [15] 25
3.5. *Wavelet* madre 'db4'. 26
3.6. Comparación de resolución obtenida con Fourier y *Wavelets* [16] 26
3.7. Mecanismo de establecimiento de ventanas [10] 28
3.8. Extracción de características usando la SWT 29

4.1. Diagrama del algoritmo de segmentación con características espectrales. 33
4.2. Algoritmo de segmentación con características espectrales [1] . 34
4.3. Representación del agrupamiento por ventanas, en cada ventana vertical las diferencias de color indican la clasificación realizada. En el eje *x* se muestra el comienzo de cada fonema con su nombre[4] 38

4.4. Representación de las distancias euclidianas, y los puntos de segmentación obtenidos (máximos locales) [4] 38

5.1. Esquema general de algoritmos de segmentación independiente de texto. 45

5.2. Obtención de características usando filtros triangulares en la escala Mel para el corpus TIMIT. 47

5.3. Obtención de características usando *Wavelets*. 48

5.4. Conjuntos difusos utilizados. 49

5.5. Características obtenidas con *Melbanks* o *Wavelets*, y características difusas. 50

5.6. Distancia entre dos *frames* adyacentes 52

5.7. Ejemplo de cálculo de distancias usando 4 *frames*. 53

5.8. Condiciones para aceptar límites fonéticos candidatos tomando un punto máximo. 55

5.9. Condiciones para aceptar límites, tomando como referencia dos puntos máximos. 56

5.10. Ejemplo de dos máximos que se pueden tomar como límites (*frame* 5 y 7). 56

5.11. Reducción a un solo máximo local. 57

5.12. Diseño del Algoritmo Genético y los parámetros a optimizar. . . 60

5.13. Codificación del individuo. 60

6.1. Características principales del algoritmo de segmentación 62

6.2. Puntos reales de segmentación (azul), puntos detectados con características difusas (rojo) y puntos detectados con características normales (verde) 65

Índice de Tablas

6.1. Resultados obtenidos en el corpus TIMIT 63
6.2. Resultados obtenidos en el corpus DIMEx100 64
6.3. Resultados del mejor caso para el corpus TIMIT. 65
6.4. Resultados del mejor caso DIMEx100 66
6.5. Comparación con otros trabajos sobre el corpus TIMIT. 66
6.6. Comparación con trabajo [1] sobre el corpus DIMEx100. 67

Capítulo 1

Introducción

La segmentación y el etiquetado de una señal de habla en palabras o subpalabras, son tareas fundamentales para la generación de corpus de habla que permitan entrenar y probar sistemas de reconocimiento automático del habla (RAH), además de que estos procesos son de gran importancia para llevar a cabo el proceso completo de RAH (ver Figura 1.1)

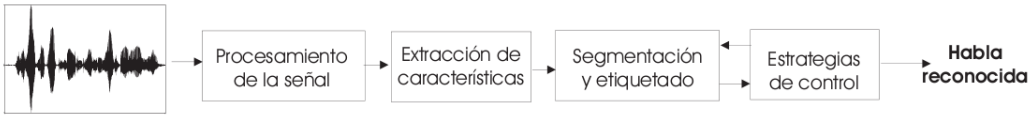


Figura 1.1: El proceso de Reconocimiento Automático del habla [1]

El proceso de segmentación y etiquetado consiste en dividir la señal en unidades, a través de la identificación automática del tiempo inicial y final de cada unidad, y posteriormente la asignación de un identificador a cada una de éstas. Las unidades en que se puede dividir o segmentar la señal, son principalmente: palabras, sílabas o fonemas, siendo estos últimos los más usados.

Muchos de los trabajos realizados en la segmentación del habla presentan restricciones, por ejemplo los trabajos [4] y [3] en los cuales se puede ver la dependencia de hablante (se trabaja con uno o pocos hablantes) o la dependencia de texto (conocimiento previo de la transcripción de la señal), entre otras. Estas restricciones ayudan a que el proceso de segmentación se lleve a cabo con porcentajes correctos de segmentación mayores al 90 %. Además de las restricciones comentadas, existen otras que son heredadas

del proceso completo de RAH, las cuales se explican con mayor detalle en el capítulo 2.

Un enfoque para realizar la segmentación sin restricciones, es utilizando solo características de la señal de habla, sin tener otro conocimiento adicional. Esto provee ciertas ventajas, como lo son la independencia de texto (solo se usa la señal hablada) y de hablante (se puede trabajar con señales de muchos hablantes), pero a su vez presenta problemas para la fase posterior al segmentado, el etiquetado de los fonemas, los cuales se detallarán en el siguiente capítulo. Los algoritmos de segmentación del habla independientes de texto pueden tener uno o más niveles, esto es, realizar la segmentación una vez o realizarla dos o más veces. Esto se hace con el propósito de mejorar límites detectados o incluir límites que fueron excluidos en niveles anteriores del algoritmo.

Dentro de esta tesis se implementó un algoritmo de segmentación independiente de texto mediante un nivel y dos niveles, usando dos formas de codificación para trabajar con la señal de habla. Por otra parte, cabe mencionar que existen trabajos de segmentación como [2], en donde se implementa un algoritmo que permite obtener el valor de los parámetros que regulan su desempeño usando funciones de aproximación, dejando algunos de sus parámetros fijos y viendo el comportamiento del valor de los otros para tratar de ajustarlos. En esta tesis se usaron Algoritmos Genéticos (AG) debido a que en el algoritmo de segmentación propuesto, requerimos saber los mejores valores de los parámetros para el primer y segundo nivel.

1.1. Problemática

Dentro de la segmentación del habla independiente de texto, se tienen problemas como la sobre-segmentación, lo cual implica detectar más límites de los que realmente existen. Otro problema que se tiene es que no se detectan algunos límites válidos (sub-segmentación). En algoritmos de segmentación multinivel, se busca detectar límites que en pasos anteriores no fueron detectados, con el riesgo que también puedan ser aceptados como válidos

límites que en realidad no lo son, aumentando así el porcentaje de sobre-segmentación.

Ligado a los problemas de segmentación, también se tienen problemas para llevar a cabo el etiquetado fonético. Para realizar el etiquetado fonético se necesitan los límites inicial y final de un segmento. Cuando se tiene sobre-segmentación, puede detectarse un límite inválido entre un límite inicial y uno final de un segmento, lo que puede provocar que se intente clasificar como si fueran dos segmentos y no uno. Otro problema se presenta cuando se detecta un límite inicial válido, pero no se detecta su correspondiente límite final, provocando que se intente etiquetar un segmento grande donde realmente se tenían que etiquetar dos o más segmentos.

Otro reto que se presenta al hacer el etiquetado fonético, es cuando se quiere reconocer fonemas usando señales provenientes de muchos hablantes ya que la forma de pronunciar los distintos fonemas puede variar dependiendo de la persona.

1.2. Objetivos

1.2.1. Objetivo general

Esta tesis tiene como objetivo desarrollar un algoritmo de segmentación de habla independiente de texto usando distintos esquemas de codificación.

1.2.2. Objetivos específicos

- Desarrollar un algoritmo de segmentación de habla independiente de texto, independiente de hablante y que trabaje con señales de habla continua.
- Explorar el uso de esquemas de codificación de la señal como *MelBank* (Bancos de filtros *Mel*) y *Wavelets* para analizar la señal.
- Utilizar distintas medidas de distancia que ayuden a generar transiciones prominentes que puedan representar límites fonéticos.

- Definir una forma de seleccionar límites válidos que permita reducir el número de inserciones.
- Usar uno y dos niveles de segmentación.

1.3. Justificación

La segmentación y etiquetado de señales de habla en unidades es de gran importancia en sistemas de reconocimiento automático del habla (RAH), y de mucha ayuda para la generación de corpus que permitan entrenar este tipo de sistemas. De ahí la importancia de que el proceso de segmentación sea preciso, ya que errores en la detección de los límites de las unidades provocará un desempeño pobre al tratar de realizar el etiquetado de éstas y afectará al reconocimiento del habla. Para hacer la división de la señal en unidades se optó por la división en fonemas, ya que actualmente se están utilizando en varios sistemas de RAH y en trabajos de segmentación de habla. Además los fonemas proveen ventajas como por ejemplo, al tratar de reconocerlos, la búsqueda en un diccionario de fonemas no es muy grande, pues dependiendo del lenguaje que se trabaje, el número de fonemas es pequeño. Si ésto se compara por ejemplo a tratar de reconocer sílabas o incluso palabras completas, donde el número de unidades puede ser muy grande, trabajar con fonemas reduce el número de unidades a las que el reconocedor se debe de entrenar.

La segmentación independiente de texto es una alternativa usada para hacer reconocimiento de habla que también provee ventajas, comparado con sistemas que necesitan realizar un modelado de las unidades para hacer la segmentación [3] y que necesitan una gran cantidad de datos de entrenamiento que sean de buena calidad, pues de esto depende su éxito. Otra ventaja se ve en el apoyo a generación de corpus, donde no se necesita conocer de antemano la transcripción fonética de la señal. La segmentación independiente de texto también permite trabajar con diversos hablantes, pues no se necesita una fase de entrenamiento para poder segmentar las unidades, ya que al ser independiente de texto y de hablante, solamente se obtendrán las instancias de tiempo donde existe una transición entre unidades o fonemas.

Aún con las ventajas que este enfoque provee, existe dificultad al momento de detectar la instancia exacta donde se da la transición, pues depende de un método (Melbanks o Wavelets) que pueda extraer características de frames de la señal, las cuales puedan ser analizadas en busqueda de transiciones abruptas que indiquen un límite fonético. Ademas algunas transiciones no están claramente delimitadas, por lo que para tratar de resolver el problema de detección de límites se usa lógica difusa.

1.4. Descripción del documento

Este documento se encuentra organizado de la siguiente manera: en el capítulo 2 se presenta una descripción detallada de conceptos básicos sobre cómo se genera el habla y como escucha el ser humano, además se explica en qué consiste el reconocimiento y segmentación automática del habla. En el capítulo 3 se presentan las formas de codificación de la señal de habla. En capítulo 4 se habla sobre trabajos relacionados a la segmentación de habla independiente de texto. En el capítulo 5 se detalla la implementación del algoritmo propuesto de segmentación independiente de texto y en el capítulo 6 se da un reporte y análisis de los resultados obtenidos. Finalmente en el capítulo 7 se presentan las conclusiones y el trabajo futuro.

Capítulo 2

Conceptos básicos sobre Segmentación y Reconocimiento de Habla

En este capítulo se habla sobre reconocimiento y segmentación automática del habla. Para esto, primero se introducen conceptos básicos de producción y generación de habla. El enfoque tratado en el trabajo propuesto es sobre segmentación fonética del habla, por lo que las unidades a segmentar son fonemas, los cuales son las unidades lingüísticas mínimas [10].

2.1. Producción y percepción del habla

Antes de entrar en detalle con las formas de hacer Reconocimiento Automático del Habla, es de utilidad saber cómo se produce el habla y también como se percibe por el ser humano. El análisis de la lengua se puede realizar en tres niveles:

- *Nivel fonológico*: Se estudian los fonemas.
- *Nivel morfosintáctico*: Se estudian las reglas para producir oraciones.
- *Nivel Semántico*: Se estudia el significado de las frases y su coherencia.

Como se comentó al principio de este capítulo, el interés de este trabajo se enfoca en segmentación del habla en fonemas, por lo que a continuación se habla brevemente sobre fonética. Dentro de la fonética se distinguen la articulatoria y la acústica. La primera está relacionada con el estudio del movimiento de los órganos fonadores que permite formar y emitir sonido, lo cual se detalla en la sección 2.1.1. Con respecto a la fonética acústica, se trata de estudiar las características de la onda sonora y su percepción, lo cual se describe brevemente en la sección 2.1.2.

2.1.1. ¿Cómo se produce el habla?

El habla es producida dentro del proceso de respiración en la parte de exhalación, cuando el aire contenido en los pulmones es expulsado a través de los bronquios y la tráquea. En la producción del habla se tienen en cuenta varias partes que van desde el diafragma hasta la boca, y en conjunto a estas partes se les conoce como aparato fonador.

Aparato fonador

El aparato fonador se divide en tres partes: las cavidades infraglóticas, la cavidad glótica y las cavidades supraglóticas; cada una de ellas son fundamentales para la fonación o producción del habla. En la figura 2.1 se muestran las partes que componen al aparato fonador.

Cavidades infraglóticas

Están compuestas por el diafragma, los pulmones, los bronquios y la tráquea. Estos son los órganos de respiración encargados de proporcionar la corriente de aire espirada para producir el sonido.

Cavidad glótica

Está formada por la laringe. En la laringe se encuentran las cuerdas vocales que son las encargadas de producir la vibración básica para generar la voz.

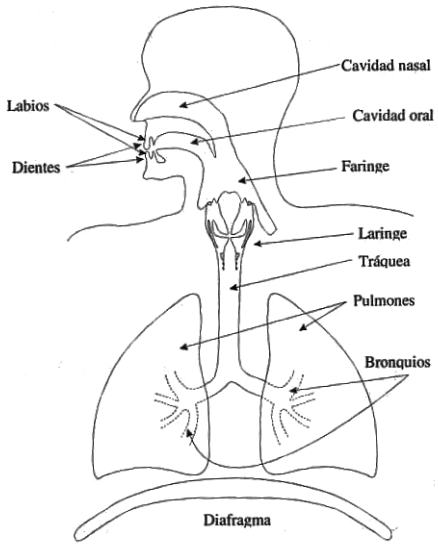


Figura 2.1: Aparato fonador [10]

Cavidades supraglóticas

Las cavidades supraglóticas son la faríngea, nasal, bucal y labial, y en éstas se encuentran órganos de articulación como son el paladar, la lengua, dientes y labios. En la figura 2.2 se muestran las partes que forman las cavidades supraglóticas.

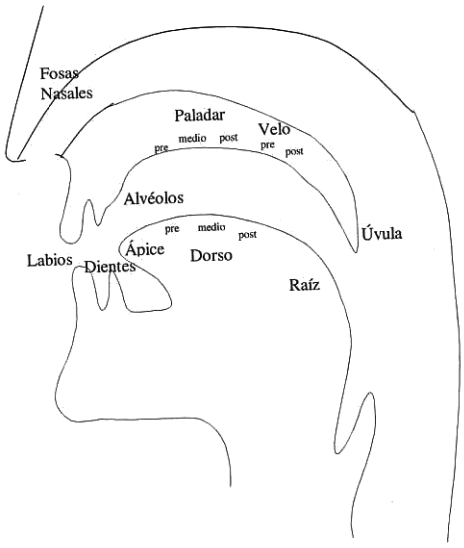


Figura 2.2: Zonas bucales [10]

Producción del habla

Hay cuatro elementos que intervienen en la producción del habla [10]:

1. Una fuente de energía: ésta es proporcionada por el aire expulsado durante la espiración.
2. Un órgano vibratorio: las cuerdas vocales.
3. Una caja de resonancia: formada por las fosas nasales, la cavidad bucal y la faringe.
4. Un sistema de articulación del sonido: formado por órganos como la lengua, labios, dientes y úvula.

Luego de ver estos elementos, procedemos a explicar la producción de habla. En primer lugar, el diafragma empuja los pulmones haciendo que se expulse aire, el cual circula por la tráquea y la laringe, pasando por las cuerdas vocales y haciendo que éstas vibren con un tono fundamental. Este tono fundamental pasa a través de la laringe a la caja de resonancia que forman las fosas nasales y la cavidad bucal. Dependiendo si la úvula está pegada a la pared faríngea o colgada, el aire saldrá por la boca en el primer caso o por la cavidad nasal en el segundo caso. Finalmente sale al exterior la voz.

Clasificación de fonemas

De acuerdo a lo visto en el punto anterior, existen varios elementos que influyen en la producción del habla, y específicamente desde nuestro punto de interés, en los fonemas. Por esto, los fonemas se pueden clasificar en varios grupos:

Por punto de articulación

Indica el lugar de las cavidades supraglóticas donde se articula el fonema. Por ejemplo, en las consonantes se tiene las bilabiales (fonemas como /p/ o /b/), en los cuales se contactan los labios superiores e inferiores.

Por el modo de articulación

Aquí depende la posición que tomen los órganos durante el habla. Por ejemplo en las consonantes oclusivas, donde hay un cierre completo de los órganos articulatorios y el aire sale de forma explosiva después de la interrupción (/p/, /t/, /d/). Lo mismo sucede en el caso de las vocales, por ejemplo las abiertas, dónde la lengua se encuentra totalmente separada del paladar (/a/).

Por la vibración de las cuerdas vocales

Aquí se considera si existe vibración de las cuerdas vocales (sonoras: /g/, /m/) o si no existe vibración (sordas: /p/, /t/).

Por la acción del velo en el paladar

Están por ejemplo los fonemas nasales, donde el velo del paladar está separado de la pared faríngea (/m/, /n/) y los fonemas orales, donde el velo se encuentra pegado a la pared faríngea y el aire solo pasa por la cavidad bucal (/t/, /f/, /d/).

2.1.2. ¿Cómo se percibe el habla?

Para saber como se percibe el habla, necesitamos saber como está compuesto el oído y cuales son sus capacidades básicas.

El aparato auditivo

El oído tiene la función de recibir las ondas sonoras para convertirlas en impulsos nerviosos que finalmente son mandados al cerebro. El oído se divide en tres partes que son: oído externo, oído medio y oído interno. Las partes que componen al oído se pueden ver en la figura 2.3. A continuación se describe como están compuestas las tres partes del oído.

Oído externo.

Está formado por el pabellón auditivo (la oreja) y por el conducto auditivo externo. Los sonidos son recibidos por la oreja y transmitidos al conducto auditivo externo donde se pueden amplificar o atenuar. Esto permite cuidar

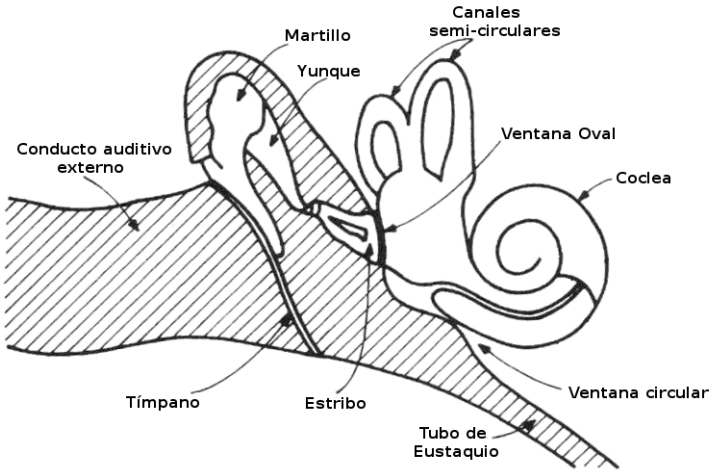


Figura 2.3: Partes del oído [11]

al oído medio e interno.

Oído medio

Está compuesto por el tímpano, el tubo de Eustaquio y tres pequeños huesos llamados martillo, yunque y estribo. Aquí las ondas llegan al tímpano que se conecta al oído interno con el martillo, el yunque y el estribo a través de la ventana oval y la ventana redonda. El tubo de Eustaquio es un canal que se comunica con la faringe. El oído medio transmite los sonidos desde el oído externo al interno realizando una adaptación de las impedancias acústicas [10]. Esto lo hace para proteger al oído interno.

Oído interno

Está formado por la cóclea y el órgano vestibular. En la figura 2.3 se pueden ver partes del órgano vestibular como los canales semi-circulares. Este órgano interviene en el equilibrio del ser humano. La cóclea es la encargada de percibir las frecuencias de las vibraciones sonoras recibidas del oído medio para convertirlas en impulsos nerviosos que se envían al cerebro.

Percepción del habla.

Teniendo el conocimiento básico del oído, podemos ver de forma resumida como se percibe el habla. En primer lugar, los sonidos llegan al oído externo, pasando por el conducto auditivo externo y llegando al tímpano. Estos sonidos hacen vibrar al tímpano. Las vibraciones de éste pasan al oído interno al hacer vibrar los tres pequeños huesos (martillo, yunque y estribo). Finalmente en el oído interno, la cóclea percibe las frecuencias de las vibraciones y son transformadas en señales eléctricas enviadas al cerebro.

Cabe resaltar que el oído humano es capaz de percibir un rango de frecuencias entre los 20 Hz y 20,000 Hz [11]. Debido a que el rango de frecuencias es muy grande, la percepción en la variación de estas frecuencias generalmente sigue una escala logarítmica [10]. Técnicas como los Bancos de Filtros en la escala Mel (*Melbanks*) hacen un análisis de las frecuencias siguiendo una escala no lineal para tratar de imitar al oído. Esto se trata con mayor detalle en el capítulo 3.

2.2. Reconocimiento automático del habla

El Reconocimiento Automático de Habla (RAH) es el proceso mediante el cual una computadora identifica palabras sin la intervención humana. Puede verse como el intento de hacer que las computadoras tengan la habilidad para comunicarse con el ser humano y viceversa, lo cual proveería de muchos beneficios en diversas áreas de la vida cotidiana, en medicina o la industria. Para realizar este proceso, la señal de audio es procesada para su uso, dividida en segmentos (*frames*) de tamaño manejable, de los cuales se extraen características que son usadas en una etapa de segmentado y etiquetado de unidades (por ejemplo fonemas), los cuales formarán las palabras reconocidas. En las siguientes secciones veremos la importancia del RAH, las aplicaciones que se le puede dar, y dificultades que se presentan al tratar de realiza esta tarea.

2.2.1. Importancia del RAH

El RAH puede tener varias utilidades, desde aplicaciones simples como el dictado de un documento hasta llegar a tener una máquina con la cual podamos comunicarnos. A continuación se presenta una lista de áreas dónde se puede usar un sistema de RAH, mostrando la importancia de esta tarea.

Dictado: El uso más común de sistemas de RAH es el de generar una transcripción del habla, por ejemplo, cuando un médico hace una transcripción, dictados en una oficina o en una declaración, o haciendo uso de un procesador de texto. Este último caso no solo requiere que se haga una transcripción de lo que se habla, si no que se necesita controlar al procesador de textos.

Comandos y control: Se pueden desarrollar sistemas que sean controlados por medio de comandos que le indiquen hacer cierta acción especifica.

Telefonía: Los sistemas de RAH pueden evitar el uso de menús jerárquicos utilizados por empresas que dan servicios por teléfono.

Manos libres: Muchos dispositivos de uso común como un reproductor portátil de música o un teléfono celular pueden ser controlados con la voz, evitando distracciones y permitiendo el uso de las manos para otras tareas.

Personas discapacitadas: El uso de sistemas de RAH puede mejorar el estilo de vida de personas con una discapacidad, por ejemplo, alguien que no tienen manos puede controlar sus objetos de uso común usando la voz, o personas sordas pueden comunicarse con otros por medio de un aparato que les transcriba lo que les hablan.

2.2.2. Dificultades

El uso de sistemas de RAH puede ser de mucha ayuda en diversas áreas, pero también es importante saber que no siempre es fácil realizar esta tarea. Existen varias complicaciones al hacer RAH que tienen que ver con el tipo de reconocimiento que se desea hacer. A continuación se muestran las complicaciones que se pueden tener al tratar de hacer el reconocimiento.

Palabras aisladas: En este caso se trata de reconocer una sola palabra, o en ciertos casos varias palabras pero que están claramente delimitadas por una pausa. Como consecuencia de esto, la persona tiene que hablar de forma pausada.

Habla continua: Comúnmente al hablar no se hacen pausas muy marcadas entre palabras, por lo que el problema radica en tratar de delimitar las palabras para que tenga sentido lo que se quiere reconocer.

Habla espontánea: Al hablar de forma natural, las personas comúnmente utilizan algunos sonidos o muletillas mientras tratan de expresarse (ehhh, mmm, este, etc.), y los cuales no tienen sentido o no son necesarios para la tarea requerida.

Dependencia de hablante: En algunos casos se requiere que el sistema de RAH tenga la capacidad de reconocer palabras de varios hablantes. Esto tiene el problema de que existen diferencias en la forma de hablar de cada persona, o si se trata de un hombre o mujer, incluso el estado de ánimo de la persona modifica su forma de hablar. Este tipo de sistemas necesita ser entrenado para reconocer el habla de cada persona que lo va a ocupar, o ser entrenado con un número grande de personas para que sea capaz de reconocer alguna nueva.

Tamaño del vocabulario: Dependiendo del uso que vaya a tener el reconocedor de habla, se necesita entrenar usando un vocabulario. Si el vocabulario es pequeño puede que no se tengan muchos errores en el reconocimiento, al contrario de vocabularios grandes, donde puede confundirse entre varias palabras. Otro problema que se tiene es cuando alguna palabra detectada por el sistema de RAH no existe en el vocabulario, lo que lleve a la decisión de tomarla como nueva palabra, desecharla o tratar de reconocerla nuevamente.

Calidad de la señal de habla: Un factor importante para tener éxito en el RAH es que la señal de entrada no tenga problemas, tales como ruido de fondo o fallas en el equipo de grabación o la calidad de éstos.

2.2.3. ¿Cómo se hace el RAH?

La figura 2.4 muestra el proceso común de un sistema de RAH.

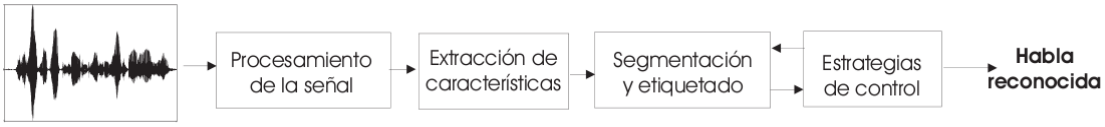


Figura 2.4: Pasos del RAH [1]

El primer paso es obtener la señal de voz, ya sea de una grabación previa o haciéndolo al momento. Posteriormente se hace un pre-procesamiento de la señal (pre-énfasis, normalizado, eliminación de ruido, filtrados, etc) para tenerla con la mejor calidad posible. Para poder trabajar con la señal, es necesario tenerla de alguna forma que sea manejable, por lo que se hace un ventaneo, dividiendo la señal *frames* a los cuales se extraen características usando algún esquema de codificación. Luego de obtener las características de los *frames*, se utiliza un método para detectar las unidades deseadas en la secuencia de *frames*, por ejemplo en fonemas, y etiquetarlas de tal forma que en la siguiente etapa el sistema de RAH sea capaz de construir palabras con los fonemas detectados. Al final se tiene el habla reconocida, ya sea para realizar una acción o simplemente para tener la transcripción de la señal de entrada.

2.3. Segmentación y etiquetado del habla

La segmentación y etiquetado del habla en unidades son partes fundamentales del RAH, ya que del resultado de éstas se forman las palabras reconocidas y depende en gran medida el éxito del proceso de reconocimiento. En este trabajo se opto por segmentar y etiquetar la señal de habla en fonemas, debido a que el número de éstos es pequeño en comparación con el número de sílabas, palabras o frases que se pueden tener en un vocabulario. En las siguientes sub-secciones se describen los enfoques de segmentación que se pueden utilizar, el pre-procesamiento necesario para trabajar con la señal de habla y finalmente se explica en forma sencilla en que consiste la detección de límites.

2.3.1. Enfoques de segmentación y etiquetado

La segmentación y etiquetado del habla son problemas ampliamente estudiados, sobre todo a nivel fonético. Se pueden distinguir dos enfoques en la segmentación y etiquetado, los cuales son:

Top-Down

Estiman la similaridad de hipótesis lingüísticas de alto nivel, modeladas como procesos estocásticos, en base a la secuencia de *frames* del habla observada. Ejemplo de éstos son los Modelos Ocultos de Markov, los cuales generan modelos acústicos de los fonemas y suponen conocida la transcripción de la señal durante la fase de entrenamiento. Este tipo de sistemas realizan un alineamiento forzado de la señal con los modelos correspondientes para realizar la segmentación [3]. Los sistemas basados en este enfoque, además de ser usados para reconocimiento de habla, se utilizan como apoyo para la generación de corpus de habla, los cuales sirven para entrenar otros sistemas de RAH.

La segmentación en este tipo de enfoque, trata de reconocer las unidades a segmentar además de sus límites fonéticos, por lo que se tienen tanto principio y fin de cada unidad segmentada. A este enfoque se le conoce como segmentación dependiente de texto.

Bottom-Up

Se hace segmentación independiente de texto, la cual se realiza únicamente analizando la señal acústica sin tener otro tipo de conocimiento previo. Posterior a la segmentación, se hace el etiquetado de los segmentos obtenidos. Este enfoque es útil cuando no se dispone de la transcripción de la señal y puede ser usado para aplicaciones multilenguaje. Cabe señalar que la segmentación independiente de texto no necesariamente significa independiente del lenguaje ya que las características de cada idioma pueden

ser usadas, por ejemplo, al realizar el etiquetado de fonemas. Este enfoque es el estudiado para realizar este trabajo de tesis.

2.3.2. Problemas en la segmentación independiente de texto

En este enfoque, el objetivo de la segmentación es obtener los límites fonéticos que existen en la señal. Debido a que con la tecnología actual aún no se logra obtener resultados de 100 % de detección de estos límites, puede haber ocasiones en que sólo se detecte el límite inicial o el final de un fonema. Por otro lado se tiene el problema de que durante la segmentación se detecten más límites fonéticos de los que son en realidad. A este fenómeno se le conoce como sobre-segmentación. Los problemas anteriores pueden provocar que entre dos límites fonéticos válidos (inicial y final), se detecte un límite ficticio, lo que implica un error en la clasificación de ese segmento ya que posiblemente se clasificará como dos fonemas y no uno. Otro caso ocurre cuando un límite válido es omitido, por ejemplo omitir el límite entre dos fonemas, lo cual trae como consecuencia que se intente clasificar como si fuera solo un segmento. Para este último caso, se puede pensar que si el segmento detectado es muy grande significa que probablemente se trata de dos o más segmentos y se intente volverlo a segmentar, aunque es difícil saberlo solo tomando como referencia el tamaño. Un estudio estadístico de las duraciones de fonemas en corpus de idioma español realizado en [12], concluyó que las duraciones entre fonemas de una misma clase varían mucho, inclusive que en las duraciones mínima y máxima de un fonema existen diferencias considerables. Todos estos problemas traen como consecuencia que no se tengan resultados satisfactorios en el etiquetado a partir de la segmentación independiente de texto.

Nótese que el etiquetado y segmentación como parte del proceso de RAH heredan los problemas de éste y se le conocen como restricciones. Para el primer enfoque nombrado en ésta sección, se dice que tiene la restricción de ser dependiente de texto, al contrario del segundo que es independiente de texto.

2.3.3. Pre-proceso y segmentación

Para trabajar con la señal de habla es necesario hacerle un procesamiento previo, que puede incluir pre-enfatizado, eliminación de ruido, normalizado, entre otros, de tal forma el resultado del proceso de segmentación tenga mayor éxito. Además se divide señal en segmentos pequeños que puedan ser manejables y de los cuales se extrae información usando algún método de codificación como los expuestos en el capítulo 3 (*Melbanks* y *Wavelets*).

2.3.4. Detección de límites

Para realizar la detección de límites fonéticos, se hace una análisis de las variaciones entre características de *frames* contiguos. La siguiente fórmula muestra una forma sencilla de obtener estas variaciones entre las características de los frames f_1 y f_2:

$$D(f_1, f_2) = \sum_{i=1}^{n} |f_1(i) - f_2(i)| \tag{2.1}$$

dónde n es el número de características extraídas del *frame*.

En la fórmula (2.1) se muestra una resta simple elemento a elemento como la operación para obtener la variación de un *frame* a otro, pero esta operación puede variar, incluyendo distintas operaciones y tomando en cuenta 2 o más *frames* contiguos en el análisis.

Capítulo 3

Análisis del habla y extracción de características

Este capítulo tiene como objetivo explicar la forma de analizar las señales de habla de tal manera que sea posible extraer características que se puedan usar en el proceso de RAH o para la segmentación independiente de texto. Para poder realizar ésto, se trabaja con técnicas que tratan de imitar el comportamiento del oído humano.

3.1. Imitando al oído humano

El humano está perfectamente adaptado para analizar la voz, aún con problemas debidos al número de distintos hablantes y su forma de expresión, incluso si el medio de transmisión tiene ruido, le es posible aislar la señal de interés con cierta facilidad. Para realizar el análisis de forma automática, se trata de imitar algunas características del oído humano de las cuales ya se tiene información sobre su funcionamiento. Por ejemplo, se sabe que el oído es capaz de percibir un rango de frecuencias que van desde los 20 Hz hasta los 20,000 Hz por medio del caracol o cóclea (ver figura 3.1). La percepción de las frecuencias y la sensación de diferencia entre éstas no se percibe con la misma sensibilidad la cual sigue una escala logarítmica. Esto ha permitido concluir que el proceso de audición se fundamenta en la descomposición en

frecuencias de la señal sonora [10], haciendo una análisis de las frecuencias del sonido mediante un banco de filtros.

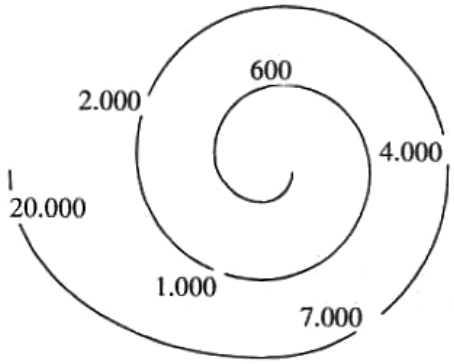


Figura 3.1: Distribución de frecuencias en la cóclea [10]

Las características vistas anteriormente permiten realizar una análisis de la señal usando Bancos de filtros en la escala Mel (*MelBanks*) y *Wavelets*, lo cual se presenta a continuación.

3.2. Análisis usando *Melbanks*

Para realizar este tipo de análisis o codificación, la señal de habla es procesada antes de pasar por un banco de filtros. Este proceso se hace con la Transformada Rápida de Fourier de Tiempo Corto. A continuación se presenta de forma breve en que consiste la transformada de Fourier y posteriormente se explica el análisis mediante *Melbanks*.

3.2.1. Transformada Fourier

La transformada de Fourier (TF) es una herramienta de análisis muy utilizada en el campo científico, por ejemplo en la acústica, métodos numéricos, sonar, electromagnetismo, comunicaciones, teoría de probabilidad, entre otras. Lo que hace es transformar una señal representada en el dominio del tiempo al dominio de la frecuencia sin alterar su contenido de información, por lo que solo

es una forma diferente de representación. De acuerdo a esta transformada, se dice que una señal, por muy compleja que sea, puede ser descompuesta en un conjunto de funciones base de senos y cosenos con diferentes amplitudes y fases. Estas amplitudes representan los coeficientes de Fourier. Además es posible regresar del dominio de la frecuencia al del tiempo reconstruyendo la señal original a partir de los coeficientes.

La transformada de Fourier es usada para trabajar con señales analógicas, pero para trabajar con señales discretas como las que se manejan en la computadora, se requiere usar la Transformada Discreta de Fourier (DFT por sus siglas en inglés *Discrete Fourier Transform*). La DFT para una señal $f(t)$ muestreada en N intervalos iguales se define de la siguiente forma [17]:

$$f\left(n\right) = \sum_{k=0}^{n-1} f\left(k\right) e^{-j2\pi k\left(\frac{n}{N}\right)} \tag{3.1}$$

Una restricción que tiene este tipo de análisis es que se aplica a señales estacionarias, y las señales de habla no son de este tipo. Por esto, se utiliza otra transformada llamada Transformada de Fourier de Tiempo Corto (STFT por sus siglas en inglés *Short-Time Fourier Transform*), en la cual se hace un ventaneo de la señal para dividirla en segmentos, los cuales pueden considerarse estacionarios y entonces aplicar la transformada de Fourier. El ventaneo puede hacerse con traslape, y se usa una venta Hamming para evitar distorsiones espectrales (Figura 3.7).

3.2.2. *Melbanks*

El objetivo del Banco de Filtros en la escala Mel es emular las bandas críticas de percepción del oído. Para esto, la señal es pasada por un conjunto de filtros, los cuales están centrados sobre los ejes de frecuencias dispuestas en la escala no lineal de Mel. Esta escala fue propuesta por Stevens y Volkman [14]. Los filtros pueden ser de forma triangular o Hamming y cada uno de éstos representa una sub-banda. Una relación entre frecuencias Mel y frecuencias lineales se puede ver en la figura 3.2 cuya fórmula es:

$$frecuenciaMel = 2595 * log_{10}(1 + \frac{frecuenciaLineal}{700}) \tag{3.2}$$

Al pasar la señal por los filtros, se calcula el logaritmo de la energía en cada sub-banda y se obtienen las características de ésta. El número de características obtenidas es igual al número de filtros. Esto proceso se visualiza en la figura 3.3.

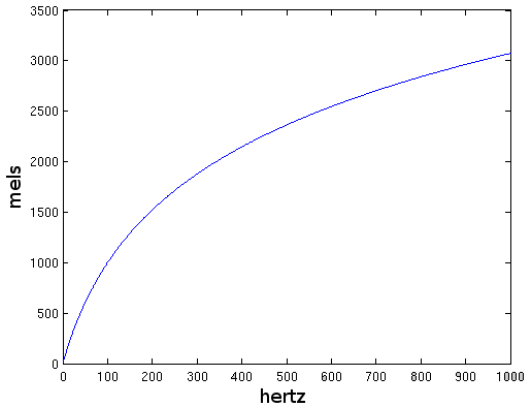


Figura 3.2: Escala Mel.

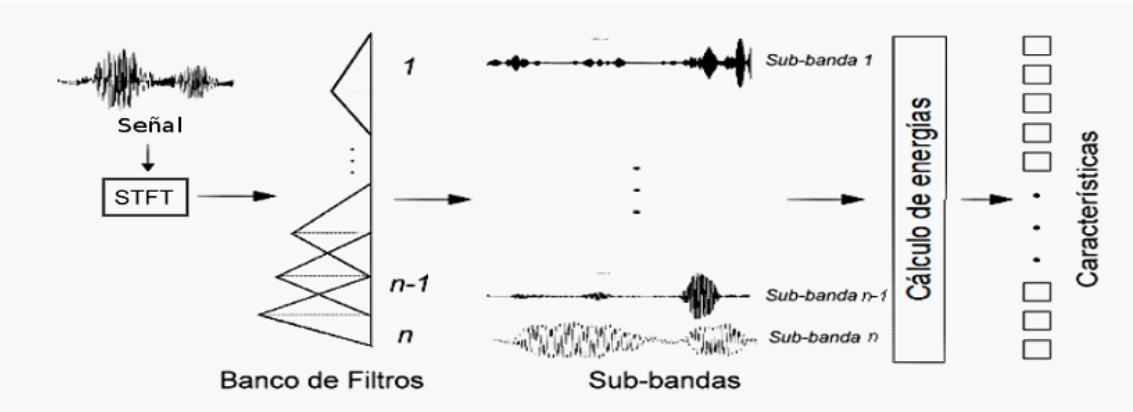


Figura 3.3: Obtención de características usando *Melbanks*.

3.3. Análisis usando *Wavelets*

El análisis con *Wavelets* ha sido usado en lugar de la TF [13] debido a que presenta ciertas ventajas con respecto a ésta, dentro de las cuales la principal es que se puede hacer un análisis multi-resolución de la señal a través de una ventana variable, a diferencia de Fourier donde la ventana de análisis es fija.

3.3.1. Transformada *Wavelet*

La transformada *Wavelet* permite hacer un análisis multi-resolución por medio de un banco de filtros que se compone de un filtro pasa-altas y uno pasa-bajas. Al colocar los bancos de filtros en cascada se obtienen niveles de descomposición como se puede ver en la figura 3.4

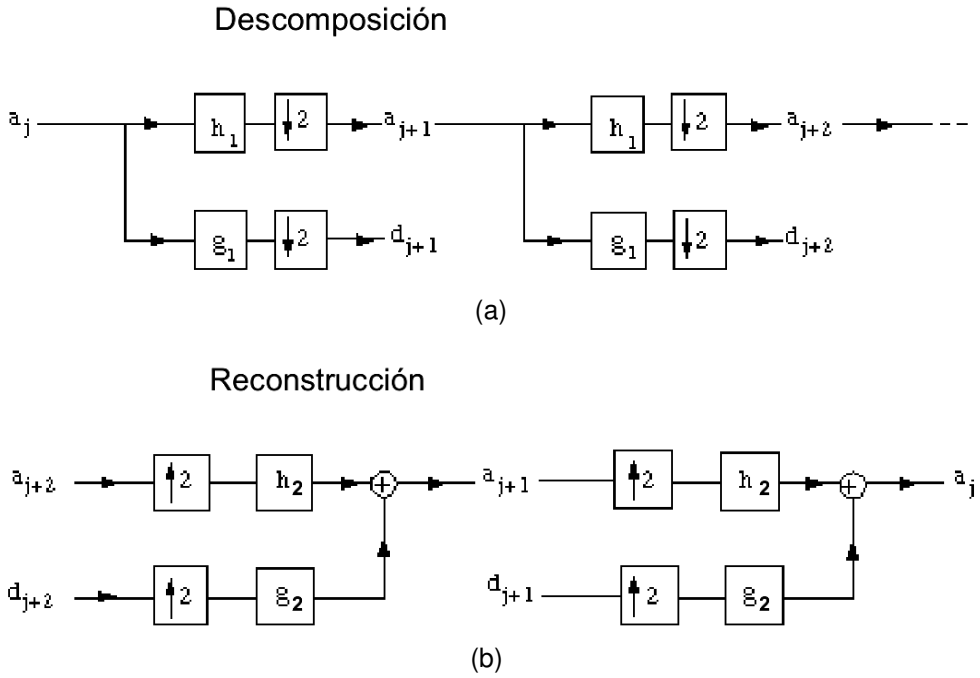


Figura 3.4: Proceso de descomposición y reconstrucción de la señal usando *Wavelets* [15]

En la figura se puede ver el filtro pasa bajas h_1 y el filtro pasa altas g_1, además a_{j+1} que representa la aproximación correspondiente a las bajas frecuencias y d_{j+1} que representa el detalle correspondiente a las altas frecuencias. A estos niveles de descomposición se le conoce como escalas. Cabe señalar que la señal de entrada va siendo sub-muestreada conforme va pasando por los filtros.

Para realizar análisis de una señal continua se usa la Transformada *Wavelet* Continua (CWT por sus siglas en inglés *Continuous Wavelet Transform*), la cual intenta expresar una señal en el tiempo mediante una expansión de términos o coeficientes que se obtienen del producto interno de la señal y diferentes versiones escaladas y trasladadas de una función prototipo conocida como

wavelet madre [15]. Esta *wavelet* madre puede ser de distintas formas, por ejemplo el grupo descubierto por Daubechies denominados “dbN”, donde N es el número de coeficientes usados. En la figura 3.5 se muestra la forma de la *wavelet* madre ’db4’ que es la usada en este trabajo de tesis.

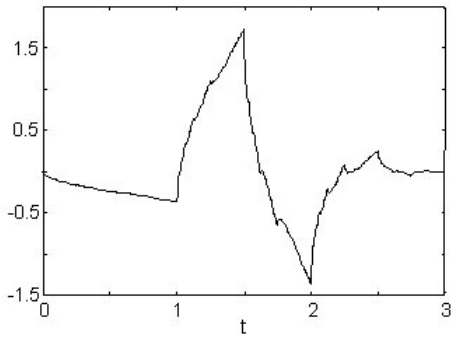
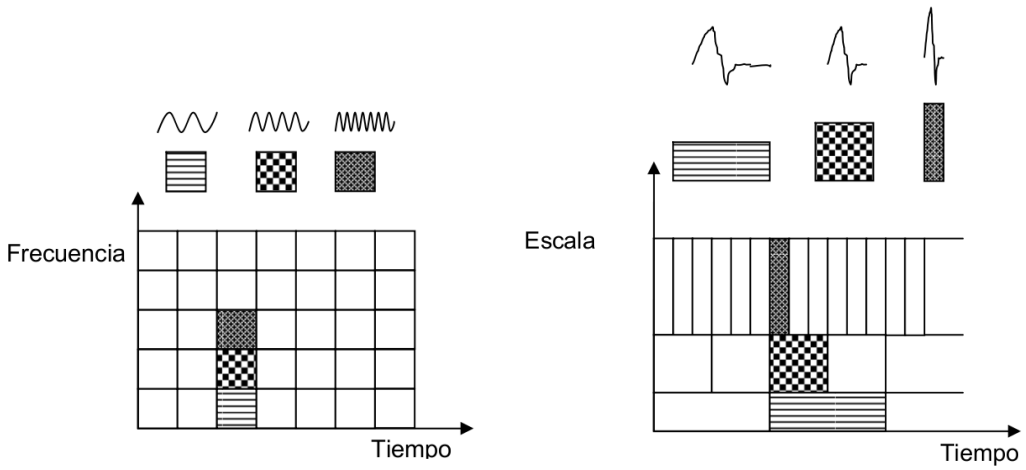


Figura 3.5: *Wavelet* madre ’db4’.

El escalamiento y traslación de la *wavelet* madre a través de la señal bajo análisis, permite obtener un análisis mult-resolución, debido a que cuando la variable de escala es pequeña, se obtiene una buena resolución en el tiempo, y cuando la variable de escala es grande se obtiene una buena resolución en frecuencia. Esto se puede ver reflejado en la figura 3.6 en la que se compara el nivel de resolución obtenido con la CWT y STFT.



(a) Fourier (b) Wavelets

Figura 3.6: Comparación de resolución obtenida con Fourier y *Wavelets* [16]

La CWT permite hacer análisis de señales continuas, por lo que para trabajar

con señales digitales se necesita la versión discreta. La Transformada *Wavelet* Discreta (DWT por sus siglas en ingles *Discrete Wavelets Transform*) permite hacer análisis de señales no estacionarias como las de habla. Un problema con esta transformada es que no es invariante en el tiempo. Esto significa que la DWT de una versión trasladada en el tiempo de una señal X, en general, no es la versión trasladada de la DWT de X. Para evitar esto, en este trabajo de tesis se usa la transformada Wavalet Estacionaria.

3.3.2. Transformada *Wavelet* Estacionaria

La Transformada *Wavelet* Estacionaria (SWT por sus siglas en ingles *Stationary Wavelet Transform*) es derivada de la DWT. La diferencia radica que en la DWT al pasar la señal por los filtros pasa baja y pasa altas, se va haciendo un sub-muestreo, es decir, se van quitando muestras de la señal. En la SWT, la salida de los filtros (a_{j+1}, d_{j+1}) es del mismo tamaño que la señal de entrada. De forma general, esto lo realiza por medio de introducir ceros entre los valores obtenidos. Esta transformada es utilizada para analizar señales de habla, principalmente para eliminación de ruido, ya que permite detectar variaciones en la señal, por lo que se usa en este trabajo para tratar de detectar las variaciones que puedan indicar límites fonéticos.

3.4. Extracción de características

Para analizar la señal de habla y obtener características que sirvan para desarrollar el proceso de segmentación del habla, es necesario realizar un análisis que consiste en tomar pequeños intervalos de la señal con cierta duración, por ejemplo 20 milisegundos, y realizar la extracción de características usando *Melbanks* o *Wavelets* a cada uno de estos segmentos. Estos fragmentos de la señal pueden tener un traslape y es recomendado usar una ventana como la Hamming para evitar las distorsiones de los extremos, poniendo énfasis en la parte central del segmento (ver figura 3.7)

Como ya se vio en la sección 3.2, las características obtenidas con *Melbanks* corresponden a las concentraciones de energía por sub-banda, para el caso

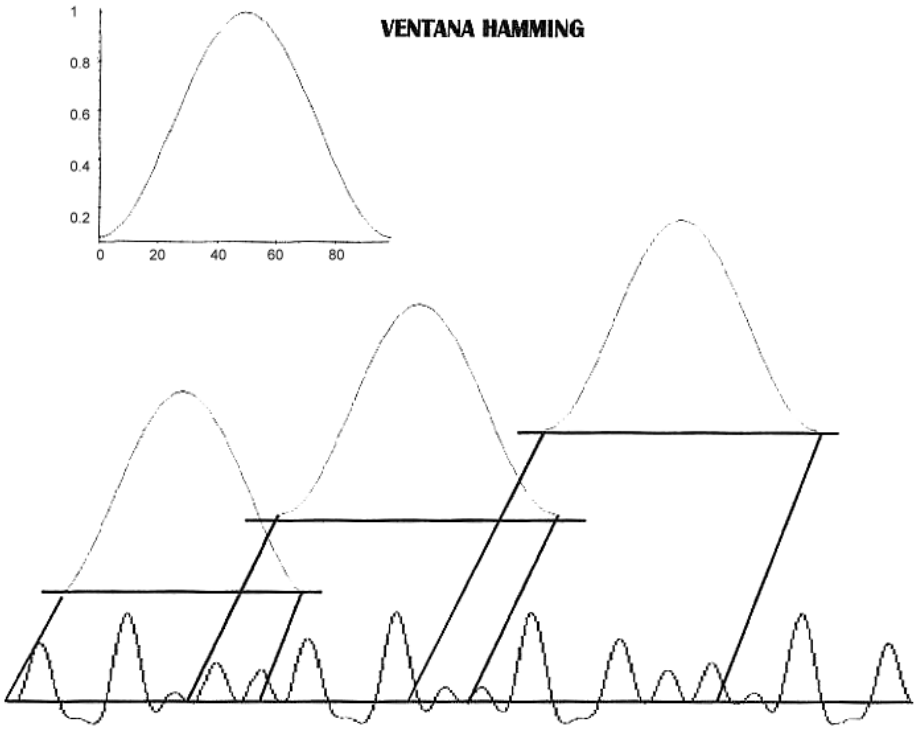


Figura 3.7: Mecanismo de establecimiento de ventanas [10]

de la SWT, las características corresponden a las concentraciones de energía por escala (ver figura 3.8)

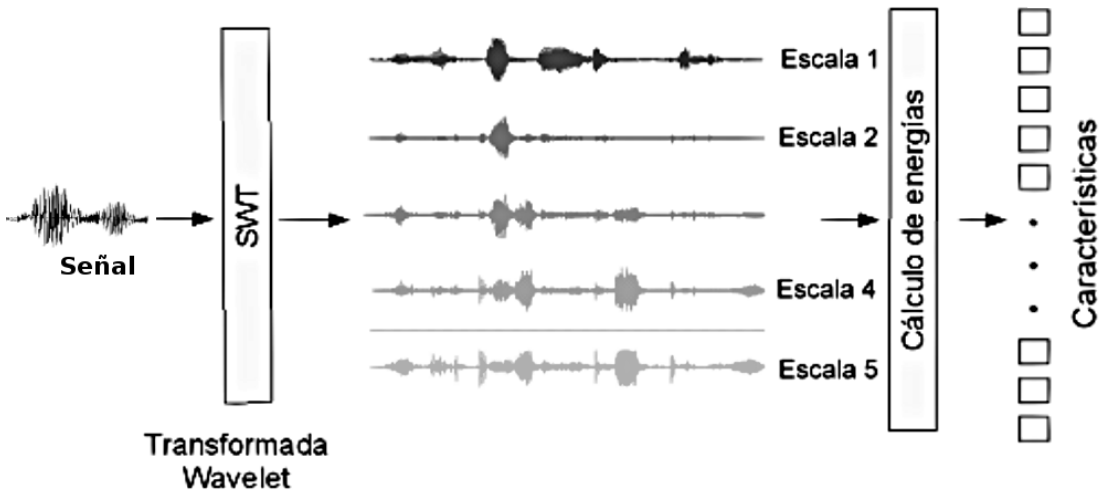


Figura 3.8: Extracción de características usando la SWT

Capítulo 4

Trabajos relacionados

El objetivo de este capítulo es mostrar una breve revisión de algunos trabajos que realizan segmentación independiente de texto, explicando cuales son sus principales características de implementación y los resultados que obtuvieron.

4.1. Evaluación del desempeño.

Se tienen dos medidas de evaluación que son:

1. Porcentaje correcto de segmentación, que indica los límites fonéticos detectados correctamente por el algoritmo con una tolerancia de ±20ms con respecto a los reales:

$$P_c = 100 * \left(\frac{S_c}{S_t} \right) \tag{4.1}$$

   donde S_c es el número de límites detectados correctamente y S_t es el número de límites reales.

2. Porcentaje de sobre-segmentación, que toma en cuenta el número de límites que se insertaron de forma incorrecta:

$$P_i = 100 * \left(\frac{S_d}{S_t} - 1 \right) \tag{4.2}$$

   donde S_d es el total de límites detectados por el algoritmo.

4.2. Segmentación de habla con independencia de texto para reconocimiento fonético

En este trabajo realizado por Huerta [1], se toma la señal de habla sin ningún conocimiento previo, como el texto, y se trata de segmentar fonéticamente de acuerdo a características de la señal. Huerta desarrolló dos algoritmos de segmentación, uno que toma medidas físicas de la señal (intensidad, amplitud, energía), y otro que usa características espectrales, probado con distintas codificaciones del sonido como MFCC (*Mel-frequency cepstral coefficient*), *MelBank* (Bancos de filtros en la escala Mel) y Filtros Bark.

4.2.1. Algoritmo con características físicas

Huerta presenta un algoritmo en el que la señal es descompuesta en *frames* de 20ms con traslape de 10ms y se obtienen vectores con características dependiendo del algoritmo usado (energía, amplitud o intensidad).

Este algoritmo emplea tres parámetros para regular el desempeño de la segmentación que se describen a continuación:

a: Define el número de *frames* previos y posteriores con respecto al *frame* bajo análisis, para obtener las diferencias absolutas de sus respectivas medias.

b: Indica la cantidad de puntos antes y después a los que un pico candidato a límite fonético debe ser mayor.

c: Define un umbral para aceptar o rechazar picos candidatos.

En la figura 4.1 se muestra como intervienen estos parámetros en la selección de los límites fonéticos.

El objetivo de los algoritmos es encontrar las instancias de tiempo donde exista una transición entre fonemas, con una tolerancia de ±20 milisegundos. La estrategia se enfoca en la detección de cambios de las medidas físicas del habla en *frames* consecutivos. Con el propósito de obtener un mayor detalle

de transiciones vagas entre fonemas, en este trabajo se hace uso de tres membresías difusas (Alto, Medio y Bajo) para las diferentes características usadas. Para saber si existen cambios entre *frames* (de características físicas o espectrales) consecutivos, se hace uso de algunas medidas de distancia como son: Euclidiana, Manhattan, Correlación Pearson y Chebyshev.

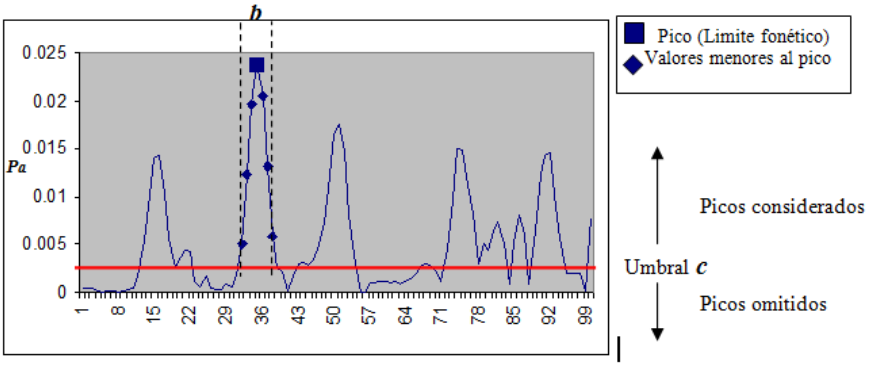


Figura 4.1: Diagrama del algoritmo de segmentación con características espectrales.

4.2.2. Algoritmo con características espectrales

En este algoritmo, Huerta trabaja un enfoque similar al anterior tratando de encontrar instancias de tiempo donde exista una transición entre fonemas, pero en este caso las características extraídas a los *frames* de la señal de habla es por medio de MFCC, Filtros Bark y Filtros Mel. También hace uso de membresías difusas para obtener mayor detalle en las características de los *frames* y lograr detectar mejor los cambios entre *frames*. En este algoritmo la distancia se calcula entre dos *frames* separados por uno intermedio usando la distancia euclidiana y se obtienen puntos (representados por V_t) como los de la figura 4.1, solo que en este caso para que el máximo local sea aceptado como límite fonético debe superar dos condiciones:

1) $V_t > V_{t-1}\ y\ V_t > V_{t+1}$

2) $V_t > \phi$

En la figura 4.2 se muestra un diagrama con los componentes principales del algoritmo.

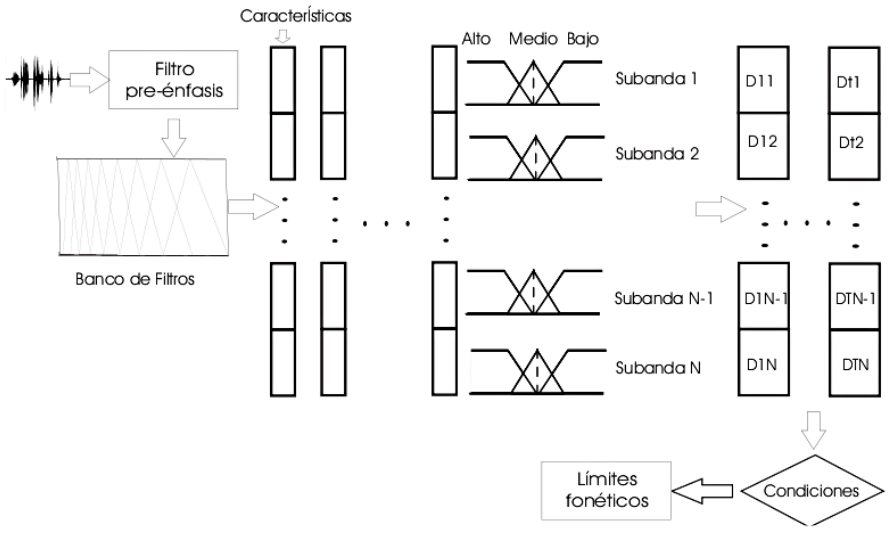


Figura 4.2: Algoritmo de segmentación con características espectrales [1]

4.2.3. Experimentos y resultados

Las pruebas se realizaron con dos corpus, DARPA TIMIT (inglés: 630 hablantes) tomando señales de 68 hablantes (34 hombres y 34 mujeres), y DIMEx100 (español: 100 hablantes) tomando señales de 30 hablantes (15 hombres y 15 mujeres), por lo que el método es independiente de hablante e independiente de texto.

Los mejores resultados obtenidos fueron usando 8 características espectrales obtenidas con Bancos de Filtros Mel, donde se obtuvo un 76.50 % de detección correcta de límites fonéticos en el corpus TIMIT, y 79.89 % en el corpus DIMEx100, manteniendo la sobre-segmentación cercana al 0 %. Los resultados usando características físicas fueron un poco menores.

4.3. Método para la segmentación de habla independiente de texto

En este trabajo realizado por Esposito y Aversano [2], se hace un análisis de la señal para obtener características variantes en el tiempo. Hace uso de

codificaciones del habla como MFCC y *MelBank* para generar 8 parámetros de cada *frame* que se analiza.

4.3.1. Implementación

El proceso de segmentación trata de encontrar transiciones (límites fonéticos) en la evolución de los parámetros obtenidos de cada *frame*, es decir, instancias de tiempo donde se tiene un cambio rápido y significativo de estos parámetros. Para distintos límites fonéticos encontrados que son muy cercanos se usa un procedimiento llamado *fitting procedure* que trata de colocar el límite en el centro de éstos. Su desempeño se basa en tres parámetros, los cuales se describen a continuación:

a: Se usa para determinar cuántos *frames* (en la evolución temporal de la señal) se necesitan para la altura (o intensidad) de un cambio abrupto.

b: Umbral que deben superar la transiciones detectadas para ser tomadas en cuenta.

c: Es el ancho del intervalo donde se buscarán los límites cercanos.

Debido a que el desempeño de la segmentación depende de estos tres parámetros, Esposito y Aversano desarrollaron un algoritmo que trata de obtener los mejores valores para éstos dejando dos fijos y jugando con el valor de otro. El funcionamiento de este algoritmo básicamente es un ciclo en el cual se asignan distintos valores al parámetro libre y se ejecutan pocas pruebas para obtener puntos de ejemplo, los cuales se utilizan para aproximar funciones que se ajusten a esos puntos. Una vez que se tienen las funciones, se calculan sus ceros analíticos y se toma el cero que pertenezca a una región que decrece en forma monótona. Ese valor se toma y se ejecuta una prueba de segmentación, y si el valor del porcentaje de sobre-segmentación es distinto de cero se repite el ciclo.

4.3.2. Experimentos y resultados

Los experimentos realizados en este trabajo fueron usando dos corpus, el DARPA TIMIT y el NTIMIT (versión telefónica de TIMIT con ruido). Los resultados obtenidos se evaluaron de acuerdo al porcentaje correcto de segmentación y sobre-segentación, obteniendo 82 % de detección correcta de límites fonéticos para el corpus TIMIT y 80 % para el NTIMIT, y ambos mantuvieron un porcentaje de sobre-segmentación cercano al 0 %.

4.4. Análisis Multiresolución aplicado en la segmentación fonética independiente de texto

El trabajo realizado por Cherniz *et al.* [13] está basado en el trabajo de Esposito y Aversano visto en la sección 4.2, aplicando una transformada Wavelet a la señal para después realizar el ventaneo de 20 ms con traslape de 10 ms. Para la extracción de características se uso CME (*Continuous Multiresolution Entropy*) y CMD (*Continuous Multiresolution Divergence*).

4.4.1. Implementación

Para la realización de este trabajo, se tomó como base el algoritmo propuesto en [2] y se cambió la forma de obtener características que se hacia con *Melbanks*, para hacerlo basado en CME usando la entropía de Shannon, y CMD usando la distancia de Kullback-Leibler. Lo importante del enfoque usado en este trabajo es que no se hace un ventaneo de la señal de entrada antes del procesamiento como se hace con *Melbanks*, sino que lo primero que realizan es aplicar una transformada *Wavelet* Casi-Continua a la señal y posteriormente se hace el ventaneo.

Para realizar la parametrización basada en CME, después de aplicar la transformada *Wavelet* a la señal, se hace el ventaneo y por cada *frame* se calculan unas probabilidades para las ventanas obteniendo una matriz de nxm

donde n es el número de *frames* y m es el número de escalas. Posteriormente se obtiene la entropía de Shannon de la matriz obteniendo así la matriz CME. Finalmente se aplica PCA (*Principal Component Analysis*) para obtener 8 parámetros (características) por *frame*. Un procesamiento similar se sigue para obtener la matriz CMD, solo que en la matriz de probabilidades obtenida, se calcula la distancia Kullback-Leibler de los *frames* consecutivos.

4.4.2. Experimentos y resultados

Las pruebas se hicieron sobre un subconjunto de 600 oraciones pertenecientes al corpus Albayzin [18] de habla en español. El porcentaje correcto de segmentación reportado es de 86.91 % y el porcentaje de sobre-segmentación es de 17.16 %.

4.5. Segmentación de fonemas no supervisada basada en métodos kernel de máximo margen

Este trabajo realizado por Adell *et al.* [3], desarrolla un método automático no supervisado de segmentación de fonemas independiente de texto, en el cual se hace uso de un algoritmo llamado algoritmo de “agrupamiento de máximo margen” (MMC por sus siglas en inglés *Maximum Margin Clustering*). Para evaluar el desempeño se tienen dos medidas: Porcentaje de detección correcta de fronteras o límites y porcentaje de detección de falsas fronteras.

4.5.1. Implementación

A partir de la señal de voz, se extraen una serie de vectores de parámetros que contendrán sus principales características. La codificación usada para obtener estos parámetros fue MFCC, tomando la señal en segmentos de 15ms con solapamiento de 10ms.

Los vectores obtenidos son agrupados usando MMC en dos clases, las cuales forman una matriz donde las columnas son la clasificación de cada uno de los vectores (figura 4.3). Una vez que se tiene este conjunto de vectores se procede a obtener la distancia euclidiana entre ellos, de tal forma que se puedan ver resultados como los de la figura 4.4, en donde aplicando un algoritmo de detección de máximos locales, se obtiene la segmentación automática.

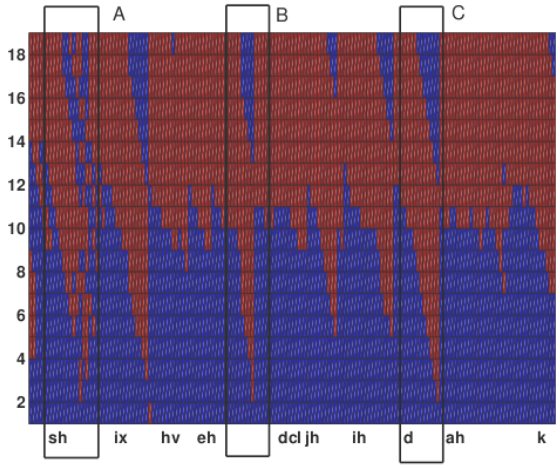


Figura 4.3: Representación del agrupamiento por ventanas, en cada ventana vertical las diferencias de color indican la clasificación realizada. En el eje *x* se muestra el comienzo de cada fonema con su nombre[4]

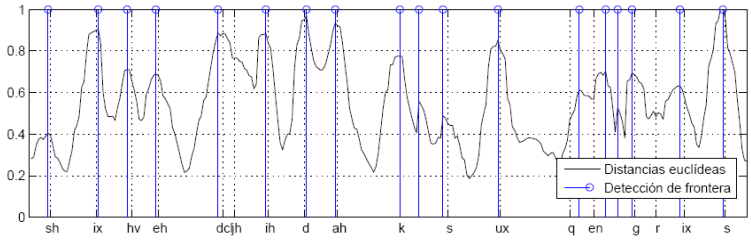


Figura 4.4: Representación de las distancias euclidianas, y los puntos de segmentación obtenidos (máximos locales) [4]

4.5.2. Experimentos y resultados

Las pruebas realizadas en este trabajo fueron con 4 frases pertenecientes a 4 locutores (dos hombres y dos mujeres), tomadas del corpus DARPA TIMIT. Los resultados que obtiene son de 87 % de detección de fronteras o límites correctos, pero el porcentaje de detección de falsas fronteras es 32.61 %.

4.6. Resumen

Los métodos presentados en este capitulo siguen un enfoque similar, dividiendo la señal en frames tratando de detectar variaciones en características de éstos, de las cuales las variaciones más grandes son indicativos de límites fonéticos. Uno de los métodos estudiados propone el uso de *Wavelets* como método de extracción de características, aplicando una transformada *Wavelet* para posteriormente realizar el ventaneo de la señal. En este trabajo de tesis se propone utilizar la transformada Wavelet haciendo el ventaneo y aplicando la transformada a cada frame para ver que resultado se obtiene.

Este tipo de métodos de segmentación tienen problemas al momento de detectar límites, debido a que la forma de calcular las distancias entre frames adyacentes puede generar transiciones pequeñas que no ayuden al proceso de detección. Además de este problema, se tiene otro cuando la medida de distancia genera demasiadas transiciones abruptas muy cercanas, lo cual provoca aumentar las inserciones del algoritmo si no se emplea una forma de detección de límites que pueda afrontar estos problemas. Para ayudar a mejorar estos aspectos problemáticos en la segmentación independiente de texto, se proponen nuevas formas de calcular las distancias entre frames adyancentes que permitan reducir el número de transiciones abruptas muy cercanas. Al momento de detectar límites, se proponen nuevas formas de detección que ayudan a reducir las inserciones permitiendo incrementar el porcentaje correcto de detección.

Capítulo 5

Algoritmo propuesto de segmentación independiente de texto

En este capítulo se hace una descripción detallada del algoritmo de segmentación de habla propuesto, el cual está basado en el trabajo de [1], modificado en los componentes de obtención de características, cálculo de distancias y selección de límites candidatos (ver Figura 5.1). El algoritmo propuesto utiliza *Melbanks* y *Wavelets* como esquemas de codificación del sonido, con los que se obtienen características de los segmentos de la señal o *frames*. Se usan medidas de distancia como la Euclidiana o la Chebyshev, y se propone una modificación a esta última para mejorar los resultados. Además se definen nuevas estrategias para la selección de límites candidatos.

El objetivo del algoritmo de segmentación es obtener las instancias de tiempo en donde exista una transición entre fonemas. Para ésto, se tratan de encontrar variaciones en el tiempo entre las características de *frames* adyacentes utilizando medidas de distancia. Debido a que las transiciones entre fonemas no están claramente definidas, se usan membresías difusas para tener mayor detalle de las características obtenidas y así poder definir la instancia de tiempo donde ocurre la transición entre fonemas. Las instancias de tiempo encontradas por el algoritmo propuesto, tienen una tolerancia de ±20 milisegundos con respecto al tiempo correcto, como se ha tratado en trabajos

como [1, 2, 3, 4, 13].

El algoritmo propuesto es independiente de texto ya que al realizar la segmentación de las señales de habla, no se tiene ningún conocimiento de las características de la señal o de la transcripción fonética. Además, el número de hablantes tomado para realizar las pruebas no estuvo limitado a solo unos cuantos, sino que se utilizó una variedad de señales de 30 hablantes para el idioma español y de 60 en inglés. Aun así, el algoritmo no es considerado independiente de idioma, pues se aplicaron ciertas diferencias en el algoritmo al trabajar con las bases de datos de inglés y español, las cuales se describen en secciones posteriores de este capítulo.

5.1. Lenguaje de programación

Para realizar la programación del algoritmo, inicialmente se trabajó con la herramienta PRAAT [5], la cual fue desarrollada para trabajar específicamente con habla. Esta herramienta tiene funcionalidades para analizar señales de audio y obtener características físicas como amplitud y energía, y también características espectrales por medio de análisis con bancos de filtros, por ejemplo *Melbanks*.

Posteriormente se migró el programa a la herramienta MATLAB [6], debido a que su interfaz gráfica de desarrollo y depuración permite un mejor análisis de los resultados, además de contar con cajas de herramientas, llamadas *toolbox*, de las cuales se usó el *Wavelet Toolbox* para obtener características usando la transformada *Wavelet*. Para hacer análisis de señales de habla con bancos de filtros, se utilizó el *toolbox* llamado VoiceBox [7].

5.2. Bases de datos utilizadas

Para evaluar el desempeño del algoritmo, se utilizaron grabaciones de dos bases de datos o corpus. Uno de estos corpus es el Dimex100 [8], el cual tiene grabaciones en idioma español de México. Para probar el algoritmo en otro idioma distinto al español y tener una base de comparación con otros

trabajos relacionados, se utilizó el corpus TIMIT [9]. Esta base de datos contiene grabaciones en idioma inglés de Estados Unidos, y es usada en trabajos de RAH y segmentación automática. A continuación se dará una breve descripción de los corpus utilizados y del número de grabaciones tomadas para realizar los experimentos de segmentación.

5.2.1. Corpus DIMEx100

El corpus DIMEx100 [8] fue creado en el departamento de Ciencias de la Computación del Instituto de Investigaciones en Matemáticas Aplicadas y en Sistemas (IIMAS), UNAM, con el objetivo de construir modelos acústicos y diccionarios de pronunciación para la creación de sistemas computacionales para el reconocimiento del español hablado en México. Ésta contiene grabaciones de 100 hablantes, donde cada uno grabó 50 frases individuales (distintas para cada hablante) y 10 en común, teniendo un total de 6,000 frases.

La base de datos fue grabada en un estudio de sonido del Centro de Ciencias Aplicadas y Desarrollo Tecnológico (CCADET), UNAM, y tiene un formato de muestreo mono a 16 bits y una tasa de muestreo de 44.1 KHz. Las grabaciones de este corpus no han sido transcritas en su totalidad, solo se cuenta con las transcripciones de 35 hablantes.

5.2.2. Corpus TIMIT

El corpus TIMIT de habla leída, creado por Lamel y Garofolo [9], fue diseñado para proveer datos de habla para estudios fonético-acústicos y para el desarrollo y evaluación de sistemas de reconocimiento automático de habla. deEste corpus contiene grabaciones de 630 hablantes de los ocho principales dialectos del inglés Americano. Las regiones dialécticas incluidas son: New England, Northern, North Midland, South Midland, Southern, New York City, Western y Army Brat.

Cada hablante grabó 10 oraciones para hacer un total de 6,300 oraciones en la base de datos. Las grabaciones tienen una tasa de muestreo de 16 KHz y para cada grabación se tiene un archivo con la transcripción fonética.

5.2.3. Datos experimentales

Para hacer las pruebas con el corpus DIMEx100, se tomó una porción formada por 8 oraciones de 30 hablantes (18 hombres y 12 mujeres), haciendo un total de 240 oraciones. El total de límites fonéticos reales de este conjunto de muestras es de 11,278.

Para las pruebas con el corpus TIMIT, se tomó una porción de 8 señales pertenecientes a 60 hablantes (30 hombres y 30 mujeres), que hacen un total de 480 oraciones, y de las cuales se obtienen 18,162 límites fonéticos reales.

5.3. Evaluación de desempeño

El algoritmo desarrollado se evaluó en base a porcentajes de detección correcta y de sobre-segmentación, que es la manera común de evaluar estos algoritmos en la literatura. Una instancia de tiempo detectada (límite fonético) es correcta, si tiene una diferencia de ±20 milisegundos con respecto al límite fonético real obtenido de la transcripción fonética de la base de datos. Esta tolerancia es común en la literatura y es la que utilizan los trabajos con los que éste se compara. Para calcular el porcentaje correcto de segmentación, se utiliza la siguiente fórmula:

$$P_c = 100 * \left(\frac{S_c}{S_t} \right) \tag{5.1}$$

donde P_c es el porcentaje de segmentación correcta, S_c son el total de límites detectados correctamente por el algoritmo, y S_t es el número total de límites reales de segmentación.

Cuando el límite fonético detectado no está dentro de la tolerancia de los ±20 milisegundos, se tiene una inserción. Estos puntos insertados junto con los puntos correctamente detectados, se consideran para calcular el porcentaje de inserción o también llamado sobre-segmentación. Para calcular este porcentaje se utiliza la siguiente fórmula:

$$P_i = 100 * \left(\frac{S_d}{S_t} - 1 \right) \tag{5.2}$$

donde P_i es el porcentaje de sobre-segmentación, y S_d es el total de límites detectados por el algoritmo.

Ambas medidas se requieren, pues P_i por sí solo no refleja un buen desempeño del algoritmo ya que se pueden detectar muchos más límites de los reales, aumentando la probabilidad de que algunos de éstos sean tomados como correctos. La forma de cuantificar puntos que se detectan de más es por medio del porcentaje de sobre-segmentación P_i, el cual de manera ideal debe estar cercano al 0 %, lo que indica que al menos el número de puntos detectados es cercano o igual al número de límites reales. De esta forma, los valores ideales de ambas medidas son $P_c = 100\,\%$ y $Pi = 0\,\%$

5.4. Descripción general del algoritmo

Como se comentó al principio de este capítulo, el objetivo del algoritmo de segmentación se enfoca en obtener las instancias de tiempo en donde existe un límite fonético. Para realizar esta tarea, se tomó como base el trabajo [1], el cual, como algunos trabajos de la literatura, sigue un esquema de trabajo similar al que se muestra en la figura 5.1.

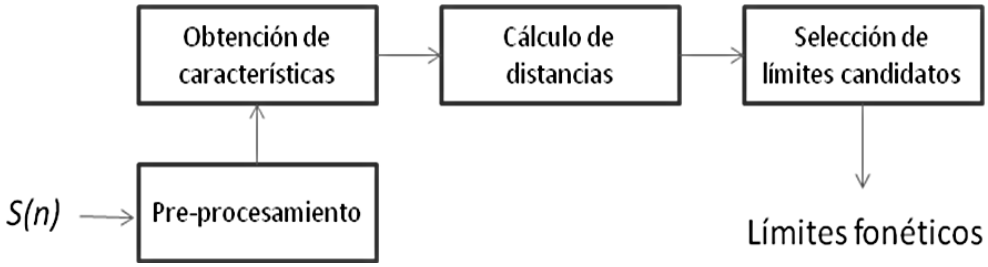


Figura 5.1: Esquema general de algoritmos de segmentación independiente de texto.

Siguiendo este enfoque, se dividió la señal en pequeños segmentos o *frames*, de los cuales se obtuvieron características usando *Melbanks* y *Wavelets*. Posteriormente se calculó la diferencia entre características de *frames* contiguos con ciertas medidas de distancia, para generar puntos en donde existan variaciones prominentes que sean indicativos de un límite fonético. La última parte del algoritmo consistió en decidir cuales de los

puntos detectados son aceptados como candidatos a límites fonéticos, y cuales rechazados de acuerdo a ciertos criterios.

En las siguientes secciones se describen los detalles de implementación de cada uno de los bloques de la figura 5.1.

5.5. Pre-procesamiento de la señal

Para poder realizar la segmentación de la señal, todas las señales son descompuestas en una secuencia de *frames* de 20 milisegundos, con traslape de 10 milisegundos. La duración de estos *frames* está dada por la duración mínima que puede tener un fonema, y que comúnmente es de 20 milisegundos [10]. Para evitar distorsiones espectrales al hacer esta división, a cada *frame* se le aplica una ventana Hamming [1, 2, 3].

5.6. Obtención de características

5.6.1. Características usando Bancos de Filtros en la escala Mel

Como se explicó en el capítulo 3, para obtener las características de una señal de habla, ésta es pasada por un banco de filtros, en este caso, filtros en la escala Mel. Dado que ya tenemos la señal segmentada en *frames*, cada uno de éstos es pasado por el banco de filtros, de los cuales obtenemos el *frame* filtrado por sub-bandas. Posteriormente se calcula la energía de cada sub-banda, y como resultado tenemos un vector de características de longitud igual al número de filtros del banco.

También se describió anteriormente en el capítulo 3, que los filtros pueden tener diferentes formas, lo que se puede aprovechar para obtener diferentes resultados y elegir los mejores. En el caso del corpus DIMEx100, las señales fueron filtradas usando 12 filtros tipo Hamming, y para el caso del corpus TIMIT el número de filtros fue 8 de forma triangular. La selección del número

de filtros para cada caso fue de acuerdo a los mejores resultados obtenidos en experimentos realizados. De aplicar este procedimiento a cada uno de los *frames* de la señal, se obtiene una matriz C de nxm, dónde n es el número de sub-bandas y m es el número de *frames* (figura 5.2).

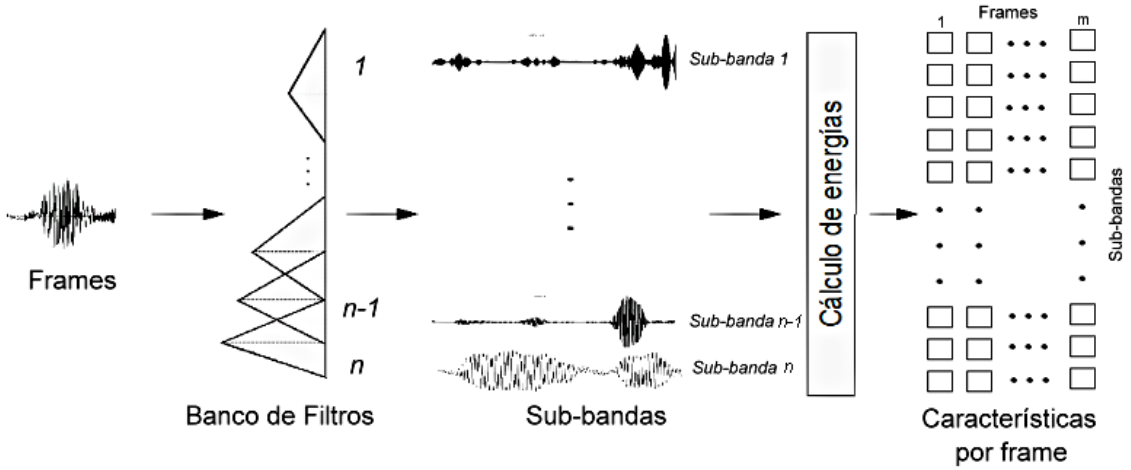


Figura 5.2: Obtención de características usando filtros triangulares en la escala Mel para el corpus TIMIT.

5.6.2. Características usando *Wavelets*

En el caso de *Wavelets*, se tienen distintas opciones a escoger para obtener las características: el tipo de transformada *Wavelet* a usar, y la *Wavelet* madre utilizada en la transformada.

El algoritmo de segmentación implementado utiliza la *Stationary Wavelet Transform* y la *Wavelet* madre 'db4', recordando del capítulo 3 que en el caso de *Wavelets* la señal es descompuesta en escalas.

Para ambos corpus se siguió el mismo procedimiento. A cada *frame* de la señal bajo análisis se le aplicó la transformada *Wavelet* con 5 escalas, que fue la que mejor resultados dio en una prueba experimental que incluyó 3, 8 y 12 escalas. Cada escala obtenida se tomó como un vector S_i y se le calculó la entropía "log energy" utilizando la fórmula:

$$E(S) = \sum_i log\left(S_i^2\right) \tag{5.3}$$

obteniendo 5 características por *frame*. De igual forma que en caso anterior, al aplicar la transformada a cada uno de los *frames*, se obtiene una matriz C de nxm, donde n son las escalas y m son los *frames* de la señal. La figura 5.3 muestra este procedimiento.

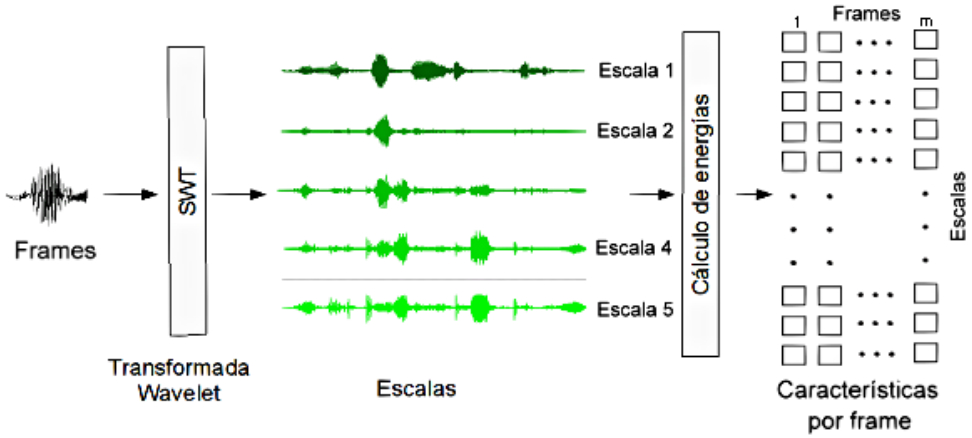


Figura 5.3: Obtención de características usando *Wavelets*.

5.6.3. Características difusas

La implementación de este bloque se hizo según se describe en [1]. Debido a que las transiciones entre *frames* adyacentes pueden no estar claramente definidas, Huerta decidió asignarle a cada característica de los *frames* valores de membresía de ciertos conjuntos difusos, de manera que al comparar características de frames con valores númericos (valores reales) similares, los valores difusos permitan generar distancias prominentes usando las medidas de distancia que se presentan en la sección 5.7. Los conjuntos difusos utilizados fueron tres: Alto, Medio y Bajo, representados por funciones triangulares traslapadas entre sí 50 % (figura 5.4). El espacio difuso para estos conjuntos se define para cada sub-banda, calculando el valor máximo y mínimo de cada una. Los valores de membresía obtenidos están en el rango de [0 1].

En este trabajo de tesis se realizaron pruebas con distintos tipos de funciones de membresía y con distintos traslapes entre ellas, de los cuales el traslape de 50 % dio los mejores resultados. En algunas pruebas realizadas, el uso de funciones de membresía trapezoidales mostró buenos resultados, pero en la

mayoría de casos el uso de funciones triangulares fue mejor.

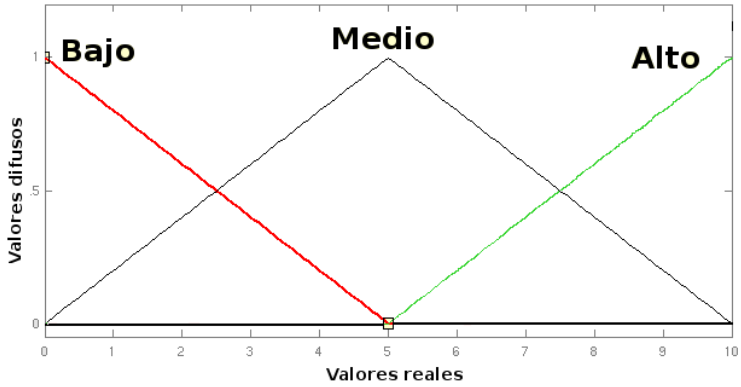


Figura 5.4: Conjuntos difusos utilizados.

Dados los valores máximo (max), medio (med) y mínimo (min) correspondientes a una sub-banda o escala, los valores de membresía de una característica del *frame* se calcula de la siguiente forma [19]:

$$\mu_A(x) = \begin{cases} 0 & x \leq med \\ \frac{x-med}{max-med} & med < x < max \\ 1 & x \geq max \end{cases} \tag{5.4}$$

$$\mu_M(x) = \begin{cases} 0 & x \leq min \ \vee \ x \geq max \\ \frac{x-min}{med-min} & min < x \leq med \\ \frac{max-x}{max-med} & med < x < max \end{cases} \tag{5.5}$$

$$\mu_B(x) = \begin{cases} 1 & x \leq min \\ 0 & x \geq med \\ \frac{med-x}{med-min} & med < x < max \end{cases} \tag{5.6}$$

Donde $\mu_A(x)$ es el valor de membresía de x al conjunto *Alto*, $\mu_M(x)$ es valor de membresía de x al conjunto *Medio* y $\mu_B(x)$ es el valor de membresía de x al conjunto *Bajo*. Una vez que se han calculado los valores de membresía de cada característica de los *frames*, se obtiene una matriz C de nxm donde n es el número de sub-bandas o escalas, y m son los *frames*, y para cada elemento de la matriz se tienen tres valores, correspondientes a *Alto*, *Medio* y

Bajo respectivamente. La figura 5.5 muestra este procedimiento.

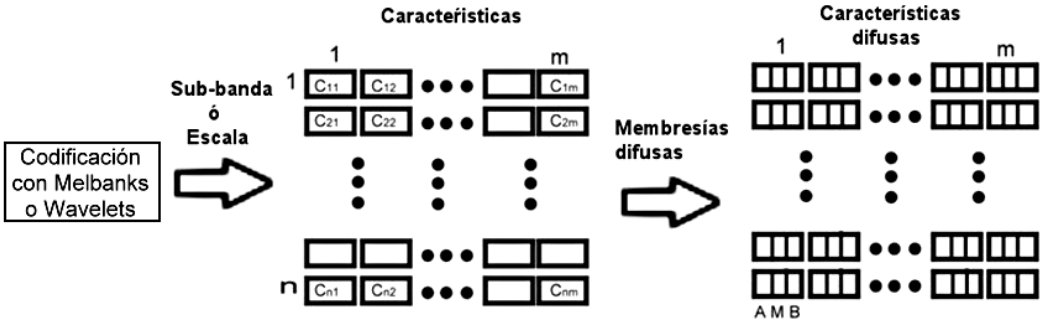


Figura 5.5: Características obtenidas con *Melbanks* o *Wavelets*, y características difusas.

5.7. Cálculo de Distancias

Para que el algoritmo de segmentación propuesto pueda detectar transiciones entre *frames* adyacentes, es necesario utilizar medidas de distancia que sean capaces de generar una diferencia considerable entre valores de las características de los *frames*, aún cuando estas diferencias no sean muy prominentes. A continuación se describen las medidas de distancia utilizadas.

5.7.1. Distancia Euclidiana

Mide la distancia entre dos puntos $X(x_1, x_2, ..., x_n)$ y $Y(y_1, y_2, ..., y_n)$. Para el algoritmo de segmentación, los puntos X y Y representan características de dos *frames* y la fórmula para calcular la distancia entre ellos se expresa de la siguiente forma:

$$d\,(X,Y) = \left(\sum_{i=1}^{n}(x_i - y_i)^2\right)^{\frac{1}{2}} \tag{5.7}$$

Para el caso de características difusas, la fórmula queda expresada así:

$$d_F\,(X,Y) = \left(\sum\left(\mu_F(X) - \mu_F(Y)\right)\right)^{\frac{1}{2}} \tag{5.8}$$

$donde\ F = \{Alto, Medio, Bajo\}$

5.7.2. Distancia Chebyshev

Está definida como el valor máximo de distancia entre atributos individuales:

$$d\left(X,Y\right) = max_{i=1\ldots n}\left\{|x_i - y_i|\right\} \tag{5.9}$$

Para este trabajo, la distancia corresponde al valor absoluto de la máxima distancia entre características de *frames*.

Para el caso de características difusas, la expresión queda como:

$$d_F\left(X,Y\right) = max_{i=1\ldots n}\left\{|\mu_F(x_i) - \mu_F(y_i)|\right\} \tag{5.10}$$

$donde\ F = \{Alto, Medio, Bajo\}$

5.7.3. Distancia Chebyshev modificada.

En este trabajo se propone una modificación a la distancia Chebyshev, elevando al cuadrado el valor máximo obtenido de la diferencia de atributos individuales. Esto se hace para que la diferencia sea más prominente. Las distancias quedan expresadas de la siguiente forma:

$$d\left(X,Y\right) = max_{i=1\ldots n}\left\{\left(x_i - y_i\right)^2\right\} \tag{5.11}$$

y para el caso de características difusas:

$$d_F\left(X,Y\right) = max_{i=1\ldots n}\left\{\left(\mu_F(x_i) - \mu_F(y_i)\right)^2\right\} \tag{5.12}$$

$donde\ F = \{Alto, Medio, Bajo\}$

5.7.4. Medidas utilizadas para obtener variaciones entre *frames*

Una vez definidas las medidas de distancias entre *frames*, se puede decidir entre cuantos *frames* adyacentes se buscará la transición o límite fonético. El

propósito principal de estas medidas es generar valores de distancia entre *frames* contiguos, los cuales representan posibles límites fonéticos. Estas medidas pueden tomar en cuenta dos o más *frames* contiguos para el cálculo de la distancia; en el caso de este trabajo, se hicieron pruebas tomando dos y cuatro *frames*.

Variación entre dos *frames*

En este primer caso, se va analizando la secuencia de *frames* de la señal y se toman 2 *frames* a la vez para calcular su diferencia utilizando cualquiera de las medidas descritas anteriormente. En la figura 5.6 se visualiza este procedimiento, donde las distancias D_1, D_2, D_{m-1} y D_m solo se indican y se muestran la distancia D_4 con los frames f_3 y f_4 que intervienen en su cálculo. También en la misma figura se pueden ver líneas que conectan a dos *frames* las cuales representan que entre éstos se calcula la distancia.

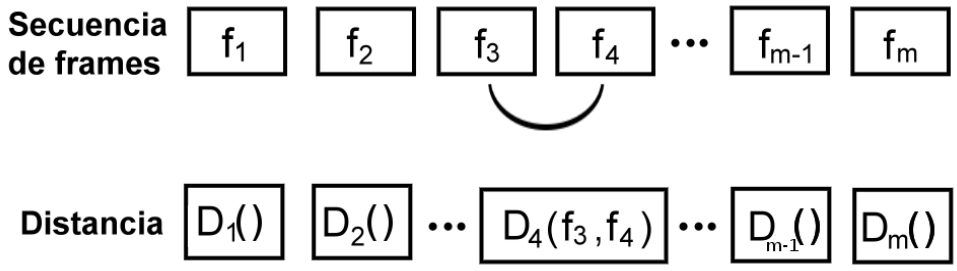


Figura 5.6: Distancia entre dos *frames* adyacentes

De esta forma se puede obtener buena resolución en el tiempo al generar transiciones prominentes, pero se tiene el problema que existen puntos en donde se generan dos o tres transiciones en lugar de solo una, por lo que se propone utilizar más *frames* al buscar la transición.

Variación entre cuatro *frames*

Para el segundo caso, se toman en cuenta cuatro *frames* para definir la distancia de los dos *frames* centrales. La figura 5.7 ejemplifica el cálculo de

esta distancia, donde solo se muestra gráficamente el desarrollo de la distancia D_3 en la que intervienen los frames f_2, f_3, f_4 y f_5.

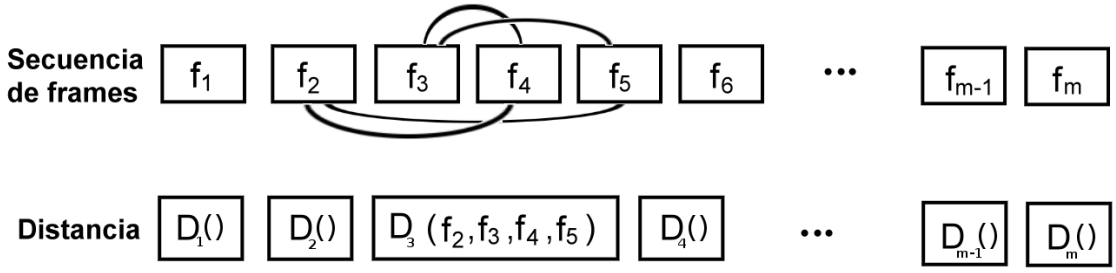


Figura 5.7: Ejemplo de cálculo de distancias usando 4 *frames*.

La fórmula para calcular la distancia euclidiana entre cuatro *frames* contiguos utilizando características difusas se expresa de la siguiente forma:

$$D(f_m, \dots, f_{m+3}) = \sqrt{\sum_{i=m}^{m+1} \sum_{j=m+2}^{m+3} \sum_{k=1}^{n} (A_{i,k} - A_{j,k})^2 + (M_{i,k} - M_{j,k})^2 + (B_{i,k} - B_{j,k})^2} \tag{5.13}$$

donde $\{A, M, B\}$ representan los valores de membresía para cada una de las n características del *frame* en cuestión.

Para el caso de la distancia Chebyshev modificada la fórmula se expresa así:

$$D(f_m, \dots, f_{m+3}) = \sum_{i=m}^{m+1} \sum_{j=m+2}^{m+3} \sum_{k=1}^{n} max((A_{i,k} - A_{j,k})^2, (M_{i,k} - M_{j,k})^2, (B_{i,k} - B_{j,k})^2) \tag{5.14}$$

Al calcular la distancia de esta forma, se evita generar transiciones que están muy juntas y en realidad hacen referencia a una sola, lo que puede incrementar el porcentaje de sobre-segmentación.

5.8. Selección de límites candidatos

Una vez que se han calculado las distancias entre *frames* adyacentes, éstas se analizan para determinar si los valores máximos indican una transición. Las distancias obtenidas pueden verse como puntos de una gráfica, de los cuales, los máximos locales pueden ser considerados como transiciones entre fonemas o límites fonéticos candidatos [1, 2]. Para que el máximo local pueda ser considerado como límite fonético, debe cumplir ciertas condiciones que se describen a continuación.

5.8.1. Detección de límites en un nivel

Las condiciones para aceptar un límite candidato tomando en cuenta un punto máximo son:

1) $D_{t-1} < D_t < D_{t+1}$

2) $D_t > \phi$

donde ϕ es un umbral y D_t es el valor de distancia calculado en las subsecciones vistas antes, donde t representa el instante de tiempo de ese valor de distancia, considerando que en el ventaneo realizado a la señal, cada instante *t* representa una duración de 10 milisegundos.

Para la condición 1, el valor de distancia del máximo local en el instante t, debe ser mayor a su antecesor y sucesor. Para la segunda condición, el valor de distancia del máximo local, debe ser mayor a cierto umbral ϕ. El propósito del umbral es evitar que aquellos puntos que tienen un valor de diferencia pequeño no se consideren, aunque pueden existir puntos menores al umbral que en realidad representaban un límite, los cuales no serán detectados. Estas condiciones se pueden visualizar en la figura 5.8, donde la línea horizontal representa el umbral y los círculos oscuros en los puntos máximos son los límites fonéticos candidatos.

Sin embargo, con las condiciones definidas de esta forma existe la posibilidad de detectar dos o más límites candidatos cercanos, que en realidad hacen referencia a uno solo, por lo que las condiciones descritas pueden ser

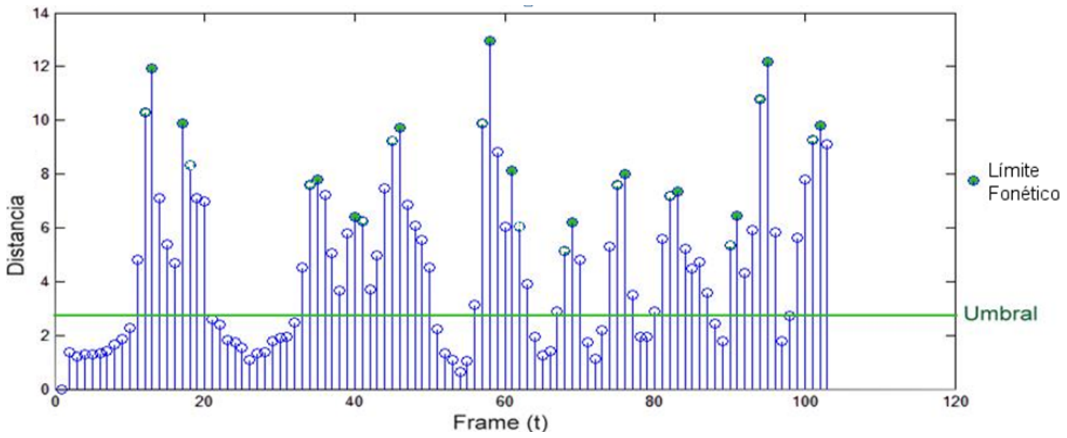


Figura 5.8: Condiciones para aceptar límites fonéticos candidatos tomando un punto máximo.

modificadas para solucionar parte del problema, definiéndolas de la siguiente forma:

1) $(D_{t-1} < D_t) \wedge (D_{t+1} < D_{t+2})$

2) $D_t,\ D_{t+1} > \phi_1$

Al hacer estas modificaciones, se evita tener varios puntos máximos cercanos que hagan referencia a un único límite fonético, evitando así incrementar la sobre-segmentación, además de que esto permite que el valor del umbral ϕ_1 pueda relajarse ligeramente, permitiendo le detección de más límites candidatos.

El comportamiento de la obtención de límites al utilizar las condiciones modificadas se ve reflejado en la figura 5.9, donde los círculos oscuros que se encuentran en los máximos locales representan los valores de distancia contemplados para hacer la selección de límites candidatos.

A pesar de que las modificaciones anteriores ayudan a disminuir la sobre-segmentación incrementando el porcentaje correcto de detección, cuando se trabaja con el corpus en español o usando la codificación con *Wavelets* se realiza un procesamiento adicional al vector de distancias D. Este procedimiento consiste en tomar dos valores contiguos de distancia a la vez para promediarlos. En la figura 5.10 se muestran los valores de distancia de varios *frames*, de los cuales es posible detectar dos límites cercanos que hacen referencia a uno solo. En la figura 5.11 se muestra como ejemplo los

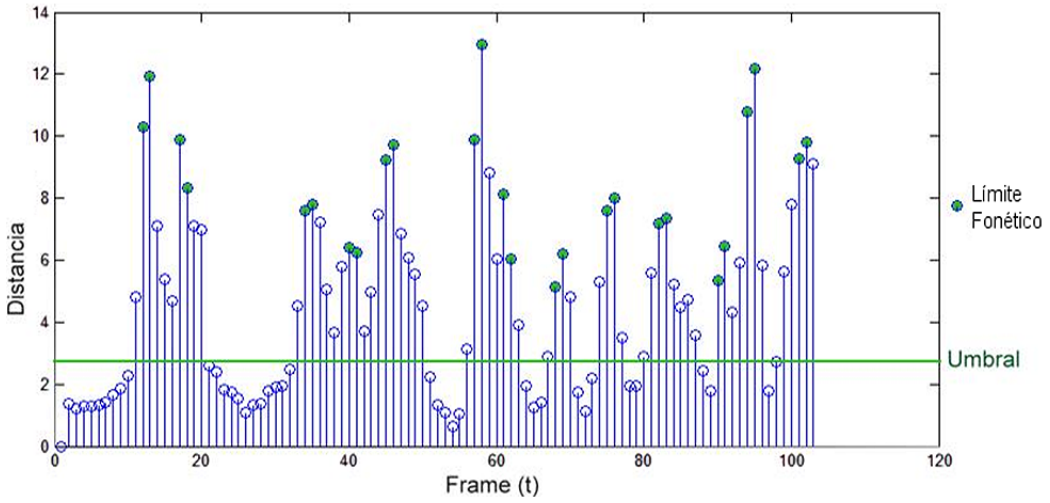


Figura 5.9: Condiciones para aceptar límites, tomando como referencia dos puntos máximos.

resultados de haber cambiado los valores del vector D, obteniendo un solo máximo centrado en el *frame* 5.

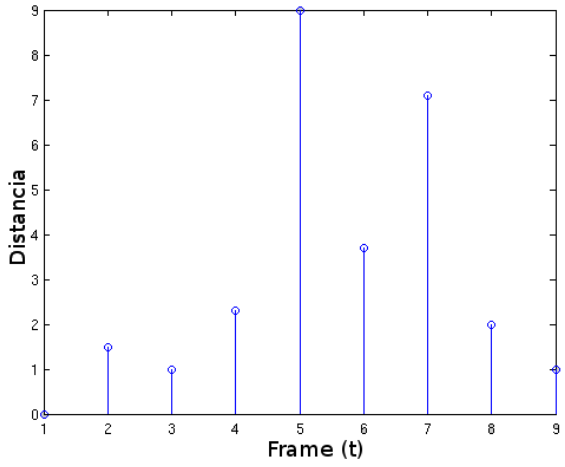


Figura 5.10: Ejemplo de dos máximos que se pueden tomar como límites (*frame* 5 y 7).

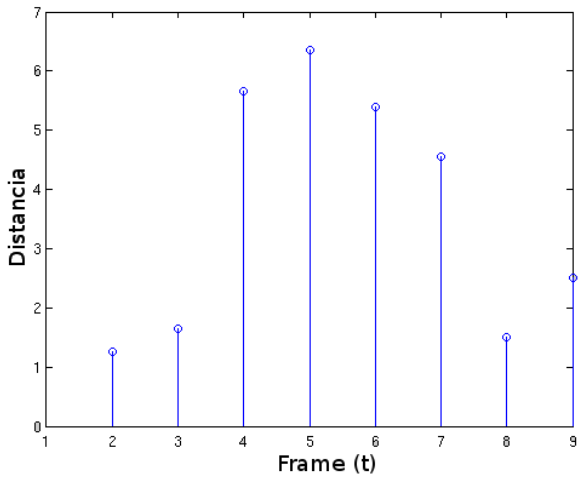


Figura 5.11: Reducción a un solo máximo local.

5.8.2. Detección de límites en dos niveles

La idea anterior se puede extender para detectar límites en dos niveles, tratando de detectar límites en un primer nivel usando un valor de umbral alto, y en el segundo nivel usar otro valor de umbral que permita detectar límites adicionales. Esto se hizo ya que en las pruebas realizadas los valores de distancia muy grandes generalmente indicaban que el límite fonético realmente existía en ese punto, por lo que ese límite se puede contemplar como correcto y tomarlo como base para el segundo nivel. Con estos límites detectados inicialmente, se tiene un conjunto de “límites base”, los cuales se usarán en el siguiente nivel como referencia para aceptar nuevos límites candidatos, siempre y cuando superen ciertas condiciones para cada nivel. En pruebas realizadas con un solo nivel de segmentación, se usaron características difusas y no difusas, obteniendo mejores resultados con las primeras pero logrando resultados cercanos con las segundas. Por esto, los valores de distancia tomados en cuenta para el primer nivel, se calculan usando las características difusas de los *frames.* Con el propósito de obtener límites diferentes a los obtenidos en el primer nivel, en el segundo nivel se toman los valores no difusos para calcular las distancias, y se introducen nuevas condiciones para aceptarlos como límites candidatos. Una primera aproximación en las condiciones para realizar la detección en dos niveles y

tomando un punto máximo se lleva a cabo de la siguiente forma:

Nivel 1:

1. $D_{t-1} < D_t < D_{t+1}$
2. $D_t > \phi_1$

Nivel 2:

1. $D_{t-1} < D_t < D_{t+1}$
2. $D_t > \phi_2$
3. *Dados l_i y l_{i+1} límites fonéticos detectados en el nivel 1, donde i=1 ... total de límites del nivel 1; un límite detectado en el nivel 2 (D_t) es aceptado como candidato si cumple:* $(|D_t - l_i|) > \delta_1 \ \wedge \ (|l_{i+1} - D_t|) > \delta_2$*, donde $\delta_1$es la distancia que separa al límite candidato D_t del límite l_i, y δ_2 es la distancia que debe haber entre l_{i+1} y D_t.*

La tercera condición evita que se vuelvan a detectar límites cercanos a los del primer nivel. Su propósito es detectar límites en segmentos grandes donde no se detectaron en el primer nivel. Los valores de ϕ_1, ϕ_2, δ_1 y δ_2 se calcularon usando un Algoritmo Genético. Cuando se toman en cuenta 2 puntos máximos, las condiciones para los dos niveles son:

Nivel 1

1. $(D_{t-1} < D_t) \ \wedge \ (D_{t+1} < D_{t+2})$
2. $D_t, \ D_{t+1} > \phi_1$

Nivel 2

1. $(D_{t-1} < D_t) \ \wedge \ (D_{t+1} < D_{t+2})$
2. $D_t, \ D_{t+1} > \phi_2$

3. *Dados l_i y l_{i+1} límites fonéticos detectados en el nivel 1, donde i=1 ... total de límites del nivel 1; un límite detectado en el nivel 2 centrado entre D_t y D_{t+1} (D_{t^*}) es aceptado como candidato si cumple: $(|D_{t^*} - l_i|) > \delta_1 \ \wedge \ (|l_{i+1} - D_{t^*}|) > \delta_2$, donde $\delta_1$es la distancia que separa al límite candidato D_{t^*} del límite l_i, y δ_2 es la distancia ó diferencia que debe haber entre l_{i+1} y D_{t^*}.*

Nótese que las condiciones del nivel 1 son iguales al caso de detección en un nivel. Para el nivel 2 se introduce un nuevo valor de umbral, permitiendo que se contemplen puntos máximos adicionales que puedan ser candidatos a límites fonéticos. Los valores de ϕ_1, ϕ_2, δ_1y δ_2 regulan el desempeño del algoritmo de segmentación en dos niveles, por lo que se usa un algoritmo genético para obtener los valores de los parámetros que den los mejores resultados.

5.8.3. Obtención de parámetros del algoritmo en dos niveles

Como se ha comentado antes, los parámetros que se utilizan en la segmentación en dos niveles se obtienen usando un Algoritmo Genético. Además de los parámetros ϕ_1, ϕ_2, δ_1y δ_2, se introdujo un parámetro adicional que indicaba el tipo de función de membresía difusa, de las cuales podían ser triangulares, trapezoidales o de campana. En forma general, el funcionamiento del AG consiste en realizar pruebas con valores aleatorios de los parámetros (individuos) para evaluar su nivel de aptitud, posteriormente se realiza una cruza entre individuos siguiendo diferentes criterios para obtener una nueva generación. A los nuevos individuos obtenidos se les calcula su nivel de aptitud y se repite el ciclo hasta llegar a un criterio de paro, que puede ser llegar al resultado esperado o haber tenido cierto número de ciclos completados. El diseño del AG se puede visualizar en la figura 5.12 y la forma de codificar los individuos se muestra en la figura 5.13. Las principales características del AG son:

- Se genera una población inicial de *n* individuos donde el *n-ésimo* es una solución conocida.

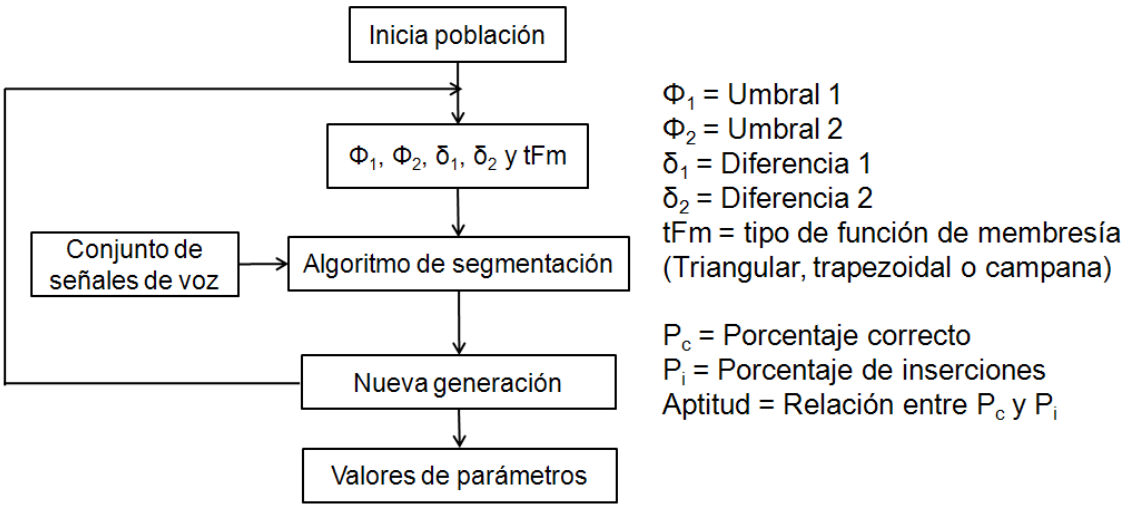


Figura 5.12: Diseño del Algoritmo Genético y los parámetros a optimizar.

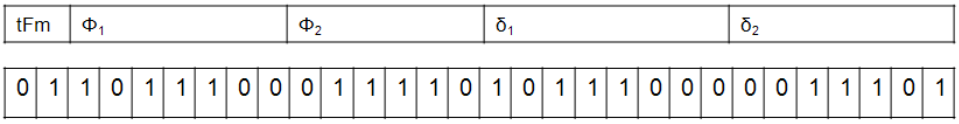


Figura 5.13: Codificación del individuo.

- El nivel de aptitud esta dado por $F = P_c * P_i$.
- Se utiliza el método de la ruleta para realizar la cruza con una probabilidad de 8 %.
- Se tiene una probabilidad de mutación del 4 % inicialmente y se incrementa 0.2 % por generación.
- Se maneja elitismo

Estos parámetros fueron escogidos en base a experimentación, y no fueron tomados de alguna otra referencia por lo que no se garantiza que funcionen para algún otro tipo de señales.

Las pruebas realizadas fueron con poblaciones de 10 individuos, logrando obtener resultados aceptables en menos de 20 generaciones.

Capítulo 6

Resultados

En este capítulo se muestran los resultados obtenidos para diversos casos probados del algoritmo propuesto y para ambas bases de datos utilizadas. Como se explicó en el capítulo anterior, al implementar el algoritmo se tuvieron distintas opciones en cada etapa del desarrollo, y haciendo combinaciones de ellas se obtuvieron diferentes resultados, de los cuales se hace un análisis para saber que tanto ayudaron las mejoras propuestas al algoritmo. También se hace una comparación con el trabajo de Huerta [1], el cual utiliza las bases de datos DIMEx100 y TIMIT, y con el trabajo realizado por Esposito y Aversano [2] que utilizan el corpus TIMIT. Cabe mencionar que la comparación solo se hace usando obtención de características con *Melbanks*, que es la forma que utilizaron estos trabajos.

A manera de resumen, en la figura 6.1 se muestra un diagrama con las características del algoritmo de segmentación propuesto haciendo referencia a la sección del capítulo 5 donde se explicó su implementación.

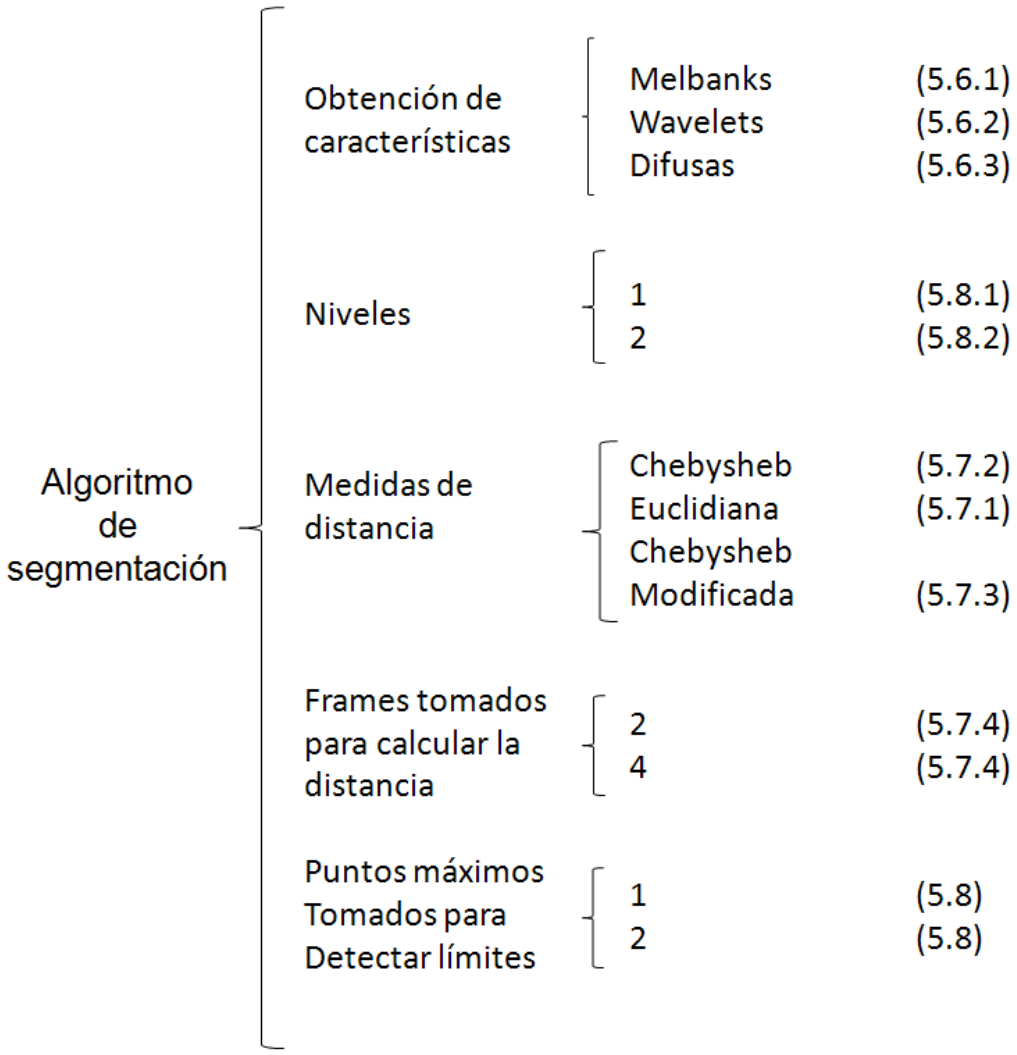


Figura 6.1: Características principales del algoritmo de segmentación

6.1. Análisis de Resultados

Durante la fase de implementación, se evaluaron las distintas variaciones que se probarían conforme se iban obteniendo resultados, por lo que no se tienen datos de algunas variaciones especificas que se consideraron innecesarias. En la tabla 6.1 se muestran todos los resultados obtenidos al ejecutar el algoritmo sobre el corpus TIMIT. Los resultados presentados en las tablas 6.1 y 6.2 son un promedio de los porcentajes obtenidos por el algoritmo para cada archivo de las bases de datos. La primera columna indica el método de codificación de

la señal que puede ser *Melbanks* o *Wavelets*. La segunda columna (Difusas), indica si se obtuvieron valores de membresía para cada característica o se trabajó con las características no difusas.

Tabla 6.1: Resultados obtenidos en el corpus TIMIT

	Método de Codificación	Difusas	Medida de distancia	# niveles	# frames en distancia	# máximos tomados	P_c	P_i
1	Melbanks	no	Euclidiana	1	2	1	77.62%	0.02%
2	Melbanks	no	Euclidiana	1	4	2	79.17%	0.07%
3	Melbanks	si	Euclidiana	1	2	1	77.81%	0.01%
4	Melbanks	si	Euclidiana	1	2	2	79.48%	0.03%
5	Melbanks	si	Euclidiana	1	4	1	79.07%	0.01%
6	Melbanks	si	Euclidiana	1	4	2	79.57%	0.02%
7	Melbanks	si	Euclidiana	2	2	1	78.27%	0.01%
8	Melbanks	si	Euclidiana	2	4	2	79.71%	0.00%
9	Melbanks	si	Chebyshev	1	4	2	79.88%	0.01%
10	Melbanks	si	Chebyshev	2	4	2	79.70%	0.01%
11	Melbanks	si	Chebyshev Mod	1	4	2	**80.28**%	**0.00**%
12	Melbanks	si	Chebyshev Mod	2	4	2	79.93%	0.02%
13	Wavelets	si	Euclidiana	1	4	2	76.98%	0.03%
14	Wavelets	si	Chebyshev	1	4	2	77.28%	0.04%
15	Wavelets	si	Chebyshev Mod	1	4	2	77.04%	0.04%

En los renglones 1 y 2 de la tabla 6.1 se muestran el peor y mejor resultado obtenidos sin usar características difusas en el algoritmo, o las mejoras propuestas en este trabajo. Como se puede observar en el renglón 11 de la tabla anterior, los mejores resultados se obtuvieron con características difusas y aplicando las mejoras propuestas, es decir, calcular la distancia entre 4 *frames* adyacentes utilizando la distancia Chebyshev modificada y tomando 2 valores de distancia máximos al seleccionar límites fonéticos candidatos. De igual forma se obtuvo una mejora con respecto a los primeros casos de la tabla al aplicar dos niveles de segmentación, aunque en la mayoría de los casos su desempeño fue inferior a solo usar un nivel con las modificaciones ya mencionadas. También se puede observar que el algoritmo con características obtenidas con *Wavelets* no pudo obtener resultados tan buenos como en el caso de *Melbanks*. Cabe recordar que el objetivo es obtener el mayor P_c manteniendo P_i muy cercano a 0 %.

En la tabla 6.2 se muestran los resultados más significativos obtenidos para el

corpus DIMEx100.

Tabla 6.2: Resultados obtenidos en el corpus DIMEx100

	Método de Codificación	Fuzzy	Medida de distancia	# niveles	# frames en distancia	# máximos tomados	P_c	P_i
1	Melbanks	no	Euclidiana	1	2	1	76.80%	0.04%
2	Melbanks	no	Euclidiana	1	4	2	80.96%	0.06%
3	Melbanks	si	Euclidiana	1	2	1	79.78%	0.04%
4	Melbanks	si	Euclidiana	1	4	2	81.98%	0.08%
5	Melbanks	si	Euclidiana	2	4	2	**82.58%**	**0.00%**
6	Melbanks	si	Chebyshev	1	4	2	81.91%	0.09%
7	Melbanks	si	Chebyshev Mod	1	4	2	81.64%	0.05%
8	Wavelets	si	Euclidiana	1	4	2	77.71%	0.07%
9	Wavelets	si	Chebyshev	1	4	2	77.29%	0.04%
10	Wavelets	si	Chebyshev Mod	1	4	2	76.90%	0.03%

Se puede observar del renglón 5 en la tabla 6.2 que los mejores resultados se obtienen aplicando las modificaciones propuestas, y en este caso, el uso de dos niveles fue mejor a usar solo uno. Para el primer caso (renglón 1) que no ocupa características difusas ni alguna otra modificación propuesta, se puede ver que el desempeño está muy por debajo de los demás.

Por otra parte, se observa que el algoritmo con características usando *Wavelets* no dio buenos resultados. Esto probablemente se debe a que al seguir el mismo enfoque de obtención de características, es decir, segmentar la señal en *frames*, descomponer cada *frame* en escalas y calcular la energía de cada una de éstas, se desaprovecha la propiedad de multi-resolución de las *Wavelets*.

Para ver en qué forma ayuda obtener características difusas, en la figura 6.2 se muestran los resultados obtenidos sobre un archivo de audio del corpus TIMIT aplicando el algoritmo con características difusas (FUZZY) y con características no difusas (NO FUZZY). En el eje vertical se muestra el tiempo en que fue detectado el límite fonético correspondiente mostrado sobre el eje horizontal. Por cada elemento del eje horizontal se muestran una, dos o tres barras, las cuales representan si se detectó un límite en la instancia de tiempo correspondiente. En la figura 6.2 también se pueden ver los límites insertados incorrectamente por el algoritmo con características no difusas en los puntos 5, 10 y 16, de los cuales el algoritmo con características difusas solo

es responsable del insertado en el punto 10, contribuyendo así a disminuir el porcentaje de sobre-segmentación, además de que en este caso detectó correctamente 16 de los 17 límites reales.

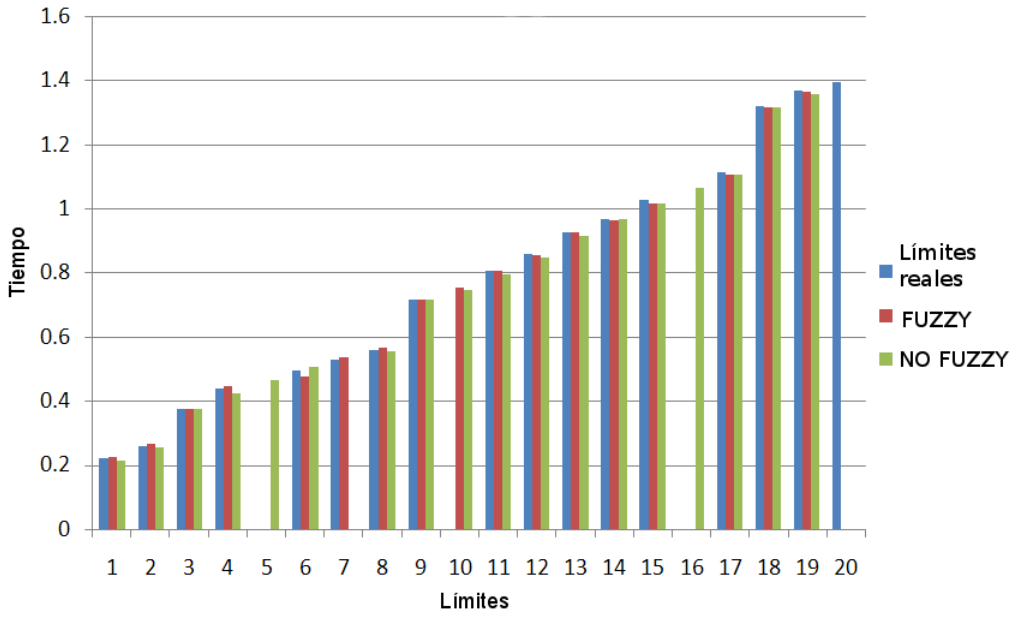


Figura 6.2: Puntos reales de segmentación (azul), puntos detectados con características difusas (rojo) y puntos detectados con características normales (verde)

Los porcentajes individuales no necesariamente reflejan el desempeño global. Para ver un ejemplo de esto, en la tabla 6.3 se muestran los resultados del mejor caso para el corpus TIMIT, y también su mejor y peor caso individual.

Tabla 6.3: Resultados del mejor caso para el corpus TIMIT.

Archivos	Limites reales	Detectados	Correctos	Inserciones	P_c	P_i
Todos	18162	18165	14571	3591	80.28%	0.00%
Mejor individual	17	17	16	1	94.12%	0.00%
peor individual	27	19	13	6	48.15%	-29.63%

De igual forma para la base de datos en español, los resultados varían entre archivos individuales. En la tabla 6.4 se ven los resultados para el mejor caso DIMEx100 y su peor y mejor caso individual.

Tabla 6.4: Resultados del mejor caso DIMEx100

Archivos	Limites reales	Detectados	Correctos	Inserciones	P_c	P_i
Todos	11046	11047	9122	1925	82.58%	0.00%
Mejor individual	50	50	46	4	92.00%	0.00%
peor individual	40	30	26	4	65.00%	-25.00%

6.2. Comparación con otros trabajos

Para hacer una comparación con trabajos relacionados, se deben tomar en cuenta características similares entre ellos, como lo son la base de datos utilizada, la forma de codificar la señal para extraer características y las medidas de desempeño. Los trabajos [1, 2] cumplen estas características, además de que en forma general, se pueden ver características similares en su implementación, por ejemplo: el número de *frames* tomados en cuenta para calcular la distancia, y el número de *frames* tomados en cuenta para seleccionar límites candidatos. En la tabla 6.5 se muestra una comparación con los trabajos [1, 2] realizando pruebas sobre el corpus TIMIT.

Tabla 6.5: Comparación con otros trabajos sobre el corpus TIMIT.

Algoritmo	**# *frames* en cálculo de distancia**	**# *frames* para obtener el límite**	**P_c**	**P_i**
Segmentación de habla con independencia de texto para reconocimiento fonético [1]	2	3	76.5%	-0.08%
Algoritmo para segmentación de habla independiente de texto [2]	4	5	82.00%	~0.00%
Algoritmo propuesto	**4**	**4**	**80.28%**	**0%**

Para los resultados con el corpus DIMEx100, se hace una comparación con el trabajo [1], la cual se ve en la siguiente tabla.

Tabla 6.6: Comparación con trabajo [1] sobre el corpus DIMEx100.

Algoritmo	**# *frames* en cálculo de distancia**	**# *frames* para obtener el límite**	**P_c**	**P_i**
Segmentación de habla con independencia de texto para reconocimiento fonético [1]	2	3	79.89%	0.08%
Algoritmo propuesto	**4**	**4**	**82.58%**	**0.00%**

El desempeño del algoritmo propuesto comparado al trabajo [1] es mejor debido a los cambios realizados en la forma de calcular las distancias entre frames y la selección de límites candidatos. Estos cambios permitieron reducir el número de puntos cercanos que hacen referencia a un solo límite, lo que permitió relajar el umbral y aceptar un mayor número de límites, incrementando así el porcentaje de detección correcta y manteniendo el porcentaje de sobre-segmentación cercano a 0 %. En el trabajo realizado en [2], su distancia propuesta calculada entre 4 *frames* y combinado con el método que proponen para seleccionar límites candidatos, les permitió incrementar el porcentaje de detección correcta manteniendo el P_i cercano a 0 %. Se puede concluir que este factor de resolución descrito en la sección 5.7.4 tuvo mucha importancia, pues en la forma de tomar la distancia entre 4 *frames* adyacentes, se pueden perder transiciones que estaban muy cercanas y que hacían referencia a diferentes límites fonéticos.

Capítulo 7

Conclusiones y trabajo futuro

7.1. Conclusiones

Los objetivos de este trabajo se cumplieron, logrando implementar un algoritmo de segmentación de habla independiente de texto, independiente de hablante y además manejando habla continua, con todas las dificultades que ésta tiene. De igual forma se trabajó con dos idiomas distintos obteniendo resultados similares, aunque ésto no indica que se pueda considerar independiente de idioma debido a que en la implementación del algoritmo hubo pequeñas diferencias al trabajar con cada uno de los idiomas, aún así, los elementos fundamentales del algoritmo fueron los mismo en ambos casos. Con respecto a las mejoras propuestas en este trabajo, se obtuvo un aumento de casi el 4 % de detección correcta de límites fonéticos con respecto al trabajo base [1] y se estuvo cerca de los resultados obtenidos en el trabajo [2], confirmando que el proceso de obtención de variaciones entre características de *frames* contiguos y el número de *frames* tomados en cuenta para realizar el cálculo de distancias es un factor importante en la mejora de los resultados. Además de este proceso, el procedimiento de selección de límites fonéticos es otro factor de gran importancia que interviene en el éxito de la segmentación. También se pudo probar el proceso de segmentación en dos niveles, el cual obtuvo mejores resultados en el corpus de idioma español, y logrando resultados cercanos a los obtenidos usando un solo nivel de segmentación en el idioma inglés. Probablemente no se ha explotado en su totalidad el uso de los dos niveles ya

que los resultados solo fueron mejores en un caso, por lo que se deben buscar alternativas que permitan aprovechar el uso de una segmentación multi-nivel.

Otro aspecto importante tratado en esta tesis fue el uso de una forma distinta de codificación de la señal alternativa al uso de *Melbanks*, los *Wavelets*. Aunque el uso de *Wavelets* no superó a ninguno de los resultados obtenidos con *Melbanks* (debido probablemente a usar el mismo enfoque de segmentar la señal antes de aplicar la transformada) sigue siendo una alternativa prometedora y que vale la pena seguir investigando, pues hay trabajos como [13] que ha logrado resultados competitivos al uso de *Melbanks*.

7.2. Aportaciones

Este trabajo de tesis aportó:

1. Un algoritmo de segmentación independiente de texto, que trabaja con habla continua y además es independiente de hablante.

2. Una nueva forma de calcular distancias entre características de cuatro *frames* contiguos que permitió generar transiciones abruptas, evitando que se generaran muchas de éstas muy cercanas. Además se definió una modificación a la distancia Chebyshev.

3. Nuevas condiciones para la selección de límites candidatos que involucra a cuatro frames contiguos, tomando como referencia dos máximos locales. Esto permitió tener un algoritmo que no requiere un post-procesamiento de límites detectados.

7.3. Trabajo futuro

Como trabajo futuro se plantea seguir probando con distintas formas de obtener variaciones entre *frames* y buscar formas de selección de límites correctos evitando las inserciones.

También se propone usar distintas formas de detección de límites para cada nivel de segmentación y después unir los resultados, no solo utilizando distintas características o medidas de distancia diferentes, si no tal vez usar un ventaneo diferente para cada nivel.

Se propone aplicar un proceso de refinamiento de límites detectados con el objetivo de que la segmentación sea más precisa y ser mas exigentes con la tolerancia actual ($\pm 20ms$). Una vez que se tiene detectado el límite fonético, se puede definir una ventana de 40 ms centrada en el límite detactado y volver a segmentar ese segmento para determinar la posición final del límite.

Así mismo se puede incursionar en el uso de umbrales adaptativos, determinando las condiciones en las características que permita dicha adaptación. Se deben buscar formas de controlar estos umbrales a través de características de la señal evitando perder la independencia de texto.

Otro tema interesante es tratar de usar la codificación de la señal mediante *Wavelets*, investigando que tipo de transformada se adapta mejor al problema de segmentación, por ejemplo la transformada *Wavelet* Casi-Continua.

Finalmente se propone trabajar con otros corpus, principalmente algunos que tengan señales contaminadas con ruido para saber que tan robusto es el algoritmo propuesto en ese tipo de ambientes, y también trabajar con corpus de otros idiomas para ver si se mantienen los resultados obtenidos.

Bibliografía

[1] L.D. Huerta. *Segmentación del habla con independencia de texto*. Tesis de Maestría en Ciencias Computacionales. Instituto Nacional de Astrofísica Óptica y Electrónica. Tonantzintla, Puebla, México, 2006.

[2] A. Esposito, G. Aversano. *Text independent Methods for Speech Segmentation*. *In: Proc. Lecture Notes in Computer Science, vol. 3445, pp. 261-290*, 2005

[3] J. Adell, A. Bonafonte, J. A. Gómez, M. J. Castro. *Análisis de la Segmentación Automática de Fonemas para la Síntesis de Voz*. III Jornadas en Tecnología del Habla, Universidad Politécnica de Cataluña (UPC), 2004.

[4] Y. P. Estevan, V. Wan, O. Scharenborg, A. Gallardo. *Segmentación de fonemas no supervisada basada en métodos kernel de máximo margen*. *Proceedings of IV Jornadas en Tecnología del Habla*. pp. 383-388, Zaragoza, Spain, 2006.

[5] P. Boersma, D. Weenink. *"Praat: doing phonetics by computer"*. Fecha de consulta: Junio de 2007. URL:www.praat.org

[6] The MathWorks, Inc. MATLAB. Fecha de consulta: Enero de 2008. URL: http://www.mathworks.com/

[7] M. Brookes, *"VOICEBOX: Speech Processing Toolbox for MATLAB"*. Fecha de consulta: Enero de 2008. URL:http://www.ee.ic.ac.uk/hp/staff/dmb/voicebox/voicebox.html

[8] L.A Pineda,L. Villaseñor-Pineda, J. Cuétara, H. Castellanos, I. López DIMEx100: *A New Phonetic and Speech corpus for Mexican Spanish.*

Proceedings of the 9th Ibero-American Conference on AI, (IBERAMIA), Puebla, Mexico, November 22-25, 2004. Lecture Notes in Artificial Intelligence, Vol. 3315, pp. 974-983, Springer 2004.

[9] L. Lamel, J. Garofolo. *Darpa timit acoutic-phonetic continuous speech corpus*. Technical report, U.S Departament of Commerce. 1993

[10] J. Bernal, J. Bobadilla, P. Gómez. *Reconocimiento de Voz y Fonética Acústica*. Editorial RA-MA, Madrid, España. 2000.

[11] Manfred R. Schroeder. *Computer Speech Recognition, Compression, Synthesis*. Springer Series in Information Sciences , Vol. 35 . 2nd ed., 2004, 375 p.

[12] G. Sandoval, C.A. Reyes. *Segmentación fonética de los corpus DIMEx100 y Tlatoa, y obtención de datos estadísticos*. Reporte interno. Instituto Nacional de Astrofísica Óptica y Electrónica. Tonantzintla, Puebla, México, 2007.

[13] A.S. Cherniz, M.E. Torres, H.L. Rufiner, A. Esposito. *Multiresolution analysis applied to text-independent phone segmentation*. *16th Argentine Bioengineering Congress and the 5th Conference of Clinical Engineering*. IOP Publishing. Journal of Physics: Conference Series 90 012083 (7pp), 2007

[14] Volkman J. Stevens S. *The relation of pitch to frequency*. *American Journal of Psychology 1940; 53: 329-353*

[15] A.L Reyes. *Un método para la Identificación Automática del Lenguaje Hablado Basado en Características Suprasegmentales*. Tesis de Doctorado en Ciencias Computacionales. Instituto Nacional de Astrofísica Óptica y Electrónica. Tonantzintla, Puebla, México, 2007.

[16] Amara Graps. *An introduction to wavelets. IEEE Computational Science and Engineering, Summer 1995, vol. 2, num. 2, pp. 50-61,* 1995.

[17] Foo-tim Chau, Yi-zeng Liang, Junbin Gao, Xue-guang Shao. *Chemometrics: From Basics to Wavelet Transform*. John Wiley & Sons, Inc., Hoboken, New Jersey, 2004.

[18] A. Moreno, D. Poch, A. Bonafonte, E. Lleida, J. Llisterri, J. B Mariño, C. Nadeu. *ALBAYZÍN Speech Database: Design of the Phonetic Corpus*. *3rd European Conference on Speech Communication and Technology*. Berlin, Germany, 21- 23 September 1993. Vol. 1, pp. 175-178.

[19] Carlos A. Reyes García. “Sistemas difusos: fundamentos y aplicaciones”. Notas de curso. Tonantzintla, Puebla, 2006.