Algoritmo para el Reconocimiento de Comandos de Voz

Andrés F. Soto P.
Universidad Tecnológica Metropolitana
Doctorando en Ciencias de la Ingeniería, Mención Automática
Departamento de Ingeniería Eléctrica. Av. José Pedro Alessandri 1242
Ñuñoa. Santiago, Chile
e-mail: andres.soto@utem.cl

Patricio Olavarrieta S.
Universidad Tecnológica Metropolitana, Ingeniero Civil Electricista
Universidad de Chile, Master DpI U. Jaume I Castellón España
Departamento de Ingeniería Eléctrica, Av. José Pedro Alessandri 1242, Ñuñoa, Santiago de Chile
e-mail: polavarr@utem.cl

Carlos Álvarez G.
Universidad de Santiago de Chile
Doctorando en Ciencias de la Ingeniería, Mención Automática
Departamento de Ingeniería Eléctrica
Ecuador 3769, Estación Central, Santiago de Chile
e-mail: carlos.alvarez@technologies.cl

Lucio Cañete A.
Universidad de Santiago de Chile
Doctor en Ciencias de la Ingeniería, Mención Automática
Departamento de Tecnologías Industriales
Avenida Ecuador 3769, Estación Central, Santiago de Chile
e-mail: lucio.canete@usach.cl

Resumen

Este trabajo tiene como objetivo el reconocimiento electrónico de un conjunto finito de comandos de voz; siendo cada uno de estos comandos una palabra corta. El procedimiento definido para la captura y reconocimiento fue el uso de la transformada Haar wavelet (H-WT); que permite representar una señal digital a través de dos coeficientes independientes y característicos de cada tramo de la señal tratada. Representándose entonces un comando de voz como una función de estos coeficientes en cada tramo en que se mide, resultando esta imagen distinta para cada comando. Se capta la señal bajo especificaciones, se acondiciona en amplitud, número de muestras y silencios para procesar con H-WT. En un sencillo uso de recursos, los resultados para dos comandos tienen como mínimo un 75% de reconocimiento.

Palabras Clave: Señales, Análisis de Señales, Transformada Wavelets, Procesamiento de Señales, Reconocimiento de Palabras.

Abstract

This study aims to address the recognition of a finite set of voice commands, each of these commands one short word. The procedure set out to capture and recognition was using Haar wavelet transform (H-WT), which allows to represent a digital signal through two independent coefficients characteristic of each segment of the treated signal. Thus represents a voice command as a function of these coefficients in each segment is measured, resulting in the different image for each command. He picks up the signal according to specifications, packaged in amplitude, number of samples to process and silences with H-WT. In a simple use of resources, the results for two commands have at least 75% of recognition.

Keywords: Signals, Signal Analysis, Wavelet Transform, Signal Processing and Words Recognitions.

▸ Andrés F. Soto P.
▸ Carlos Álvarez G.
▸ Patricio Olavarrieta S.
▸ Lucio Cañete A.

1 Introducción

El uso de comandos de voz tiene importancia en aplicaciones de campo, donde la instrucción debe ser entregada en forma directa por el usuario final [19] [9]. El objetivo es entonces desarrollar un algoritmo computacional para el reconocimiento de comandos de voz, basado en la transformada Haar wavelet (en adelante transformada Haar), que cumpla con las siguientes características: cada comando de voz es una palabra corta, será utilizado en un dispositivo personal, para un conjunto finito de comandos de voz, que opere en una plataforma de bajo costo y de prestaciones limitadas, en tiempo real y que permita un adecuado porcentaje de éxito en el reconocimiento del comando.

En el área existen una gran variedad de trabajos que desarrollaron métodos para el reconocimiento automático de voz. Estos métodos son aplicados a diferentes entornos que son caracterizados por: el tipo de entrada, tamaño de la población y sus características, idioma, tipos de locutores, género y edades, características ambientales, medio de transmisión, tamaño del vocabulario, formato de la información de entrada y otros. Esta diversidad de características, colocan igual número de exigencias en el algoritmo de reconocimiento de voz.

Los parámetros clásicos para caracterizar una señal de voz incluyen: captar la altura máxima [7][8][23], análisis de Fourier [7][8][23], Coeficientes Ceptrales y su uso en la escala de frecuencias de Mel [7][8][23], Código de Percepción Lineal (LPC) [26], Predicción Lineal Perceptiva (LPP) [7].

Técnicas como Dynamic Time Warping (DTW) para reconocimiento de voz [7] reportan mejoras en alrededor de 20 veces en la simplificación computacional, cuando están basadas en el uso de wavelets [1]. Hay escritos en reconocimiento de voz que utilizan la transformada wavelet para la obtención de plantillas que permiten el análisis de la señal, con aplicación en compresión [11]. También han sido usados para mejorar la robustez del reconocimiento de voz con los coeficientes ceptrales, frente a ruidos y distorsiones en ambientes no controlados [6]. Otros han utilizando la técnica de análisis estocástico de Hidden Markov Model (HMM) [10][20], con redes neuronales para reconocimiento de voz [13] [14][5][28][22]. Estas técnicas con diferentes enfoques han dado buenos resultados, no obstante requieren de un uso extensivo de recursos, en particular en lo que se refiere a procesadores y memoria. En este desarrollo se muestra el reconocimiento de comandos de voz, que presenta mayor simplicidad que las frases antes dichas, se implementa un algoritmo de suyo más simple basado en la transformada de Haar y que como conjunto, es anidable en hardware de prestaciones limitadas.

El presente escrito está organizado de la siguiente manera: sección 2, se revisa la transformada wavelet y se desarrolla en detalle la obtención de la Transformada Haar Wavelets; sección 3, se detalla el procedimiento de captura y reconocimiento del comando de voz; sección 4, describe el algoritmo de captura y reconocimiento; sección 5, se revisan los resultados y discusiones de la aplicación del algoritmo sobre los comandos de voz seleccionados para las pruebas y, finalmente en la sección 6 se presentan las conclusiones del trabajo.

2 La Transformada Wavelet

La transformación de funciones se basa en presentarlas de forma diferente, extrayendo información de la función, que procesada en intervalos obtiene constantes dimensionales. Es el caso de las dimensiones tiempo o frecuencia o ambos en el caso de señales. La transformación se hace utilizando funciones bases conocidas; en la transformada de Fourier las funciones bases son senos y cosenos y se centra en el concepto de frecuencia. La transformada wavelets usa funciones en forma de ondas con extensión limitada, las que deben satisfacer condiciones de admisibilidad [16][2][3], siendo una herramienta matemática para el análisis en los conceptos de tiempo y frecuencia.

Intuitivamente, dado que se trata de una onda de duración limitada y de media cero, su energía es finita. Es una onda oscilante en el eje de las abscisas, que debe estar normalizada y centrada en el entorno de tiempo (t=0). Aplicaciones de la transformada wavelets son diversas: análisis del comportamiento de proceso químicos [18], análisis para diagnóstico cardiaco [4][15][17], compresión de imágenes en telemedicina [27], reconocimiento de patrones [25], búsqueda de características en seña-

les [21], clasificación de participantes en juegos en línea [24], entre otros.

2.1 Transformada Wavelet Continua

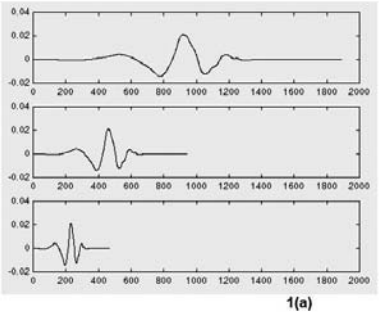
La transformada wavelet constituye una técnica para el análisis del comportamiento de una señal f(t), mediante una función ventana ψ, que encuadra la señal dentro de un intervalo de análisis. La transformada wavelet continua sobre f(t), se define como:

$$TWC_f(a,b) = \frac{1}{\sqrt{|a|}} \int_{-\infty}^{\infty} f(t)\psi\left(\frac{t-b}{a}\right)dt \qquad (1)$$

Donde la función wavelet está definida por:

$$\psi(t,a,b) = \frac{1}{\sqrt{|a|}}\psi\left(\frac{t-b}{a}\right) \qquad (2)$$

Los parámetros a y b son los de escala y traslación respectivamente, lo que genera una familia de funciones escaladas por el parámetro a y otra familia de funciones trasladadas por el parámetro b. En la Figura 1(a) se pueden ver los efectos de aplicar parámetros de escala y en 1(b) de traslación:

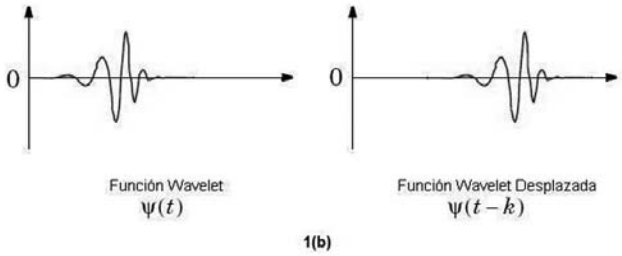


$f(t) = \psi(t) \quad ; \quad a = 1$

$f(t) = \psi(2t) \quad ; \quad a = \frac{1}{2}$

$f(t) = \psi(4t) \quad ; \quad a = \frac{1}{4}$

1(a)



1(b)

Figura 1: Función Wavelets (a) con escala y (b) con traslación

Un ejemplo de funciones wavelets que cumplen con la condición de admisibilidad son: Haar, la familia de wavelets Daubechies que son las más famosas, Sombrero Mexicano y Morlet, entre otras. Como muestra la Figura 2:

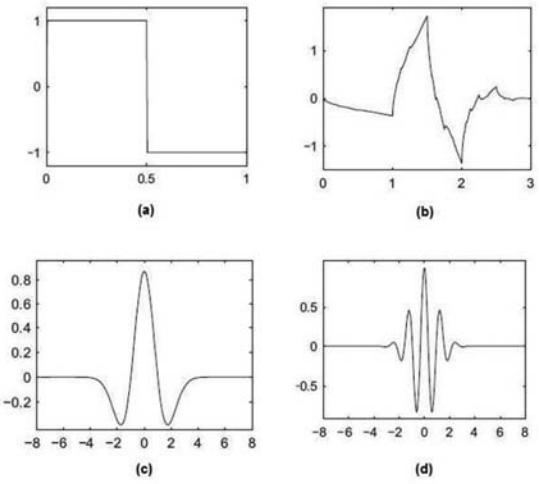


Figura 2: (a) Wavelets Haar (b) Db2 (c) Sombrero Mexicano (d) Morlet.

2.2 Transformada Wavelet Discreta

La forma más común de discretizar los valores de a y b de la ecuación (2) es utilizar una red diádica a = 2-j y b = k2-j (Huang et al., 2000), siendo j y k valores enteros. De esta manera se obtiene:

$$\psi_{j,k}(t) = 2^{\frac{j}{2}}\psi\left(2^j t - k\right) \quad j,k \in Z \qquad (3)$$

- Andrés F. Soto P.
- Carlos Álvarez G.
- Patricio Olavarrieta S.
- Lucio Cañete A.

La ecuación (3), representa la familia de funciones wavelets que se pueden obtener variando los enteros j y k. La función Haar wavelet, en adelante función Haar, se define de la siguiente manera:

$$H(x) = \begin{cases} 1 & \text{if} \quad 0 \le x < \frac{1}{2} \\ -1 & \text{if} \quad \frac{1}{2} \le x < 1 \\ 0 & \text{else.} \end{cases} \qquad (4)$$

Como se muestra en la figura 3:

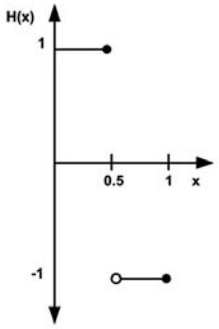


Figura 4: Función de escala Unitaria

Trabajando con la función unitaria, se demostrara que se puede obtener la función Haar, ver Figuras 4 y 5:

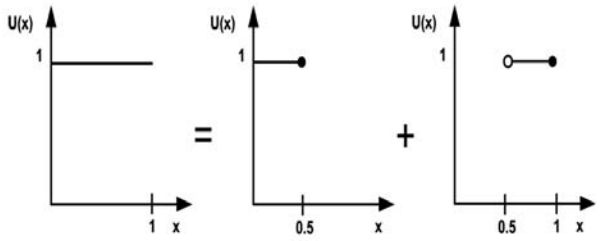


Figura 4: Función de escala Unitaria

Esta se puede descomponer en:

$$\phi[0,1[= \phi[0,1/2[+ \phi[1/2,1[\qquad (5)$$

De la misma forma si hacemos una resta se obtiene la función Haar básica, como se muestra en la Figura 5:

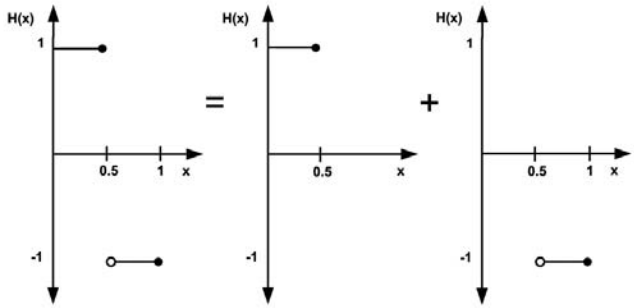


Figura 5: Función Haar a partir de la función Escala Unitaria

Que corresponde a:

$$\psi_H[0,1[= \phi[0,1/2[- \phi[1/2,1[\qquad (6)$$

Si sumamos (5) y (6), se obtiene:

$$\begin{cases} 1/2\,(\phi[0,1[+ \psi[0,1[) = \phi[0,1/2[\\ 1/2\,(\phi[0,1[- \psi[0,1[) = \phi[1/2,1[\end{cases} \qquad (7)$$

Si la aplicamos la ecuación (7) a dos valores consecutivos de una señal, se obtiene una aproximación de la señal Â que se puede escribir de la siguiente forma:

$$\begin{aligned} \tilde{s} &= s_0 \cdot \phi[0,1/2[+ s_1 \cdot \phi[1/2,1[\\ &= s_0 \cdot (1/2)(\phi[0,1[+ \psi[0,1[) + \\ &\quad s_1 \cdot (1/2)(\phi[0,1[- \psi[0,1[) \\ &= \frac{s_0 + s_1}{2}\phi[0,1[+ \frac{s_0 - s_1}{2}\psi[0,1[\end{aligned} \qquad (8)$$

La función de aproximación de la señal queda expresada por dos coeficientes formados a partir del muestreo de ella, denominándose uno coeficiente de escala y el otro coeficiente de wavelet; el valor (so+s1)/2 representa el promedio entre dos valores consecutivos del muestreo de la señal y (so-s1)/2, representa la razón de cambio en dichos valores. Si aplicamos nuevamente la ecuación (8), se obtiene un nuevo nivel de resolución y así sucesivamente como se muestra en la figura 6.

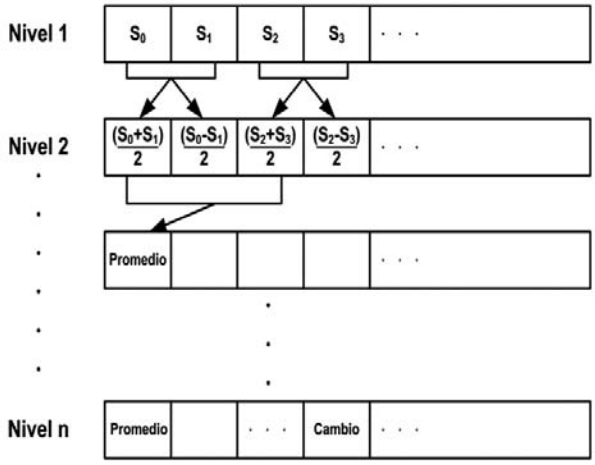


Figura 6: Niveles de Resolución

La expresión general, cuando se toman n pares de muestras identificadas con el índice k, es la ecuación (9).

$$\tilde{s} = \sum_k a_k \phi(k) + \sum_k d_k \psi(k) \quad (9)$$

Esta ecuación nos ratifica los dos conceptos antes mencionados para el uso de esta transformada. Las muestras se van acumulando en la sumatoria de ak (promedio) y en la sumatoria de dk los valores de cambio Se precisa que el largo L del comando de voz sea una potencia de 2 y el número de niveles máximos de resolución N, está dado por:

$$N = \log_2(L) \quad (10)$$

Es interesante ver que en el último nivel de integración se obtienen dos coeficientes que representan al comando de voz, comprimido y sin pérdida de información. Se espera que al aplicar el algoritmo inverso se recaptura la señal de voz original.

El sistema a desarrollar almacenará entonces un número pequeño de coeficientes y no la señal completa, como se muestra en la Figura 7:

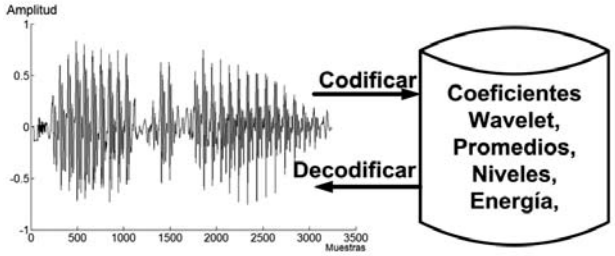


Figura 7: Almacenamiento Comprimido del Comando de Voz

Algunas de las aplicaciones de la transformada wavelet son [16]: análisis de temperatura en procesos, detección de índices financieros, detección de bordes, compresión de datos, reducción de ruido.

3 Proceso De Captura Y Reconocimiento

En general el procesamiento del habla involucra una amplitud de destrezas, sin embargo en comandos de voz las frases o texto que se reducen a una palabra de todas manera nos deja con algunos problemas a solucionar, esencialmente relativos a las variaciones del orador como: género del mismo, edad, potencia del sonido emitido o condiciones ambientales, modulación del comando, entre otras.

El algoritmo contempla dos pasos excluyentes para asegurar que la señal capturada tenga un largo igual a una potencia de 2. Estas son la decimación e interpolación. En la decimación, se busca eliminar muestras de la señal, por ejemplo si la señal es de largo 1048 entonces en este paso la señal será normada a una señal de 1024 (210), por lo cual eliminarán 24 muestras En la interpolación, se busca agregar muestras a la señal para producir el mismo efecto, por ejemplo si la señal es de largo 1000, será necesario agregar 24 muestras para que la señal tenga un largo de 1024.

El algoritmo consta de dos funciones, la primera para la captura, como se muestra en la Figura 8, y la segunda para el reconocimiento, como se muestra en la Figura 9.

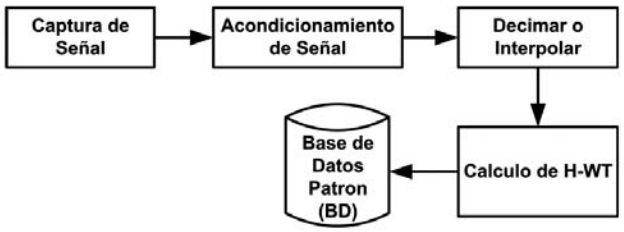


Figura 8: Diagrama Funcional de los Algoritmos

Los pasos en común para ambas funciones son: captura de la señal, acondicionamiento, decimación, interpolación y cálculo de la H-TW.

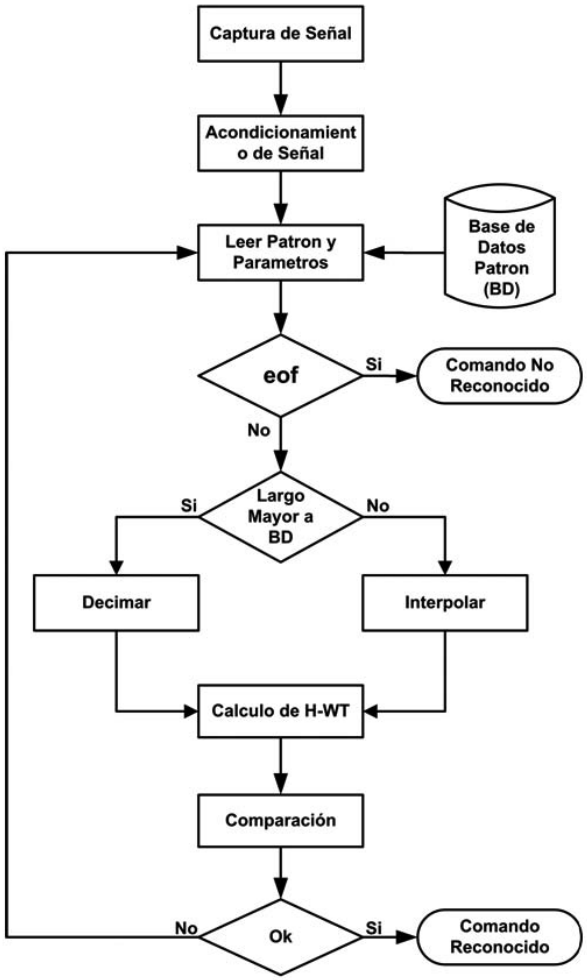


Figura 9: Diagrama Funcional de los Algoritmos

A continuación procederemos a describir cada uno de los pasos que constituyen los algoritmos.

3.1 Captura

Los comandos a capturar son: 'abrir', 'cerrar' y 'grabar'. Estos se capturarán utilizando un micrófono y una tarjeta de sonido que guarda la señal de voz en formato digital WAV [12], el cual es uno de los más utilizados y conocidos para el almacenamiento de sonidos, que incluye un proceso de normalización de amplitud entre 1 y -1. Los parámetros utilizados para la captura del comando son los siguientes: Formato PCM, frecuencia de Muestreo de 11025 [Hz], Mono Stereo de 16 bits y con una duración máxima de 4 segundos.

Para hacer el procesamiento de la señal menos susceptible a truncamientos y para aplanarla espectralmente, se pasa la señal digitalizada de voz a través de un filtro de pre-énfasis de un solo coeficiente constante, que obedece a la ecuación 11, en la cual se utilizó un valor de a igual a 0.95 (Huang et al., 2001):

$$H(z) = 1 - az^{-1}, \qquad 0.9 < a < 1.0 \qquad (11)$$

La captura de estas señales cortas, necesariamente "atrapan silencios al comienzo y al final de la grabación", que no aportan información, pero si alargan el numero de muestras. Se ejecuta un algoritmo para eliminar en forma automática estos silencios de señal, a través del uso de un umbral de energía, consiguiendo un efecto de focalizado en el comando, como se muestra en la Figura 10.

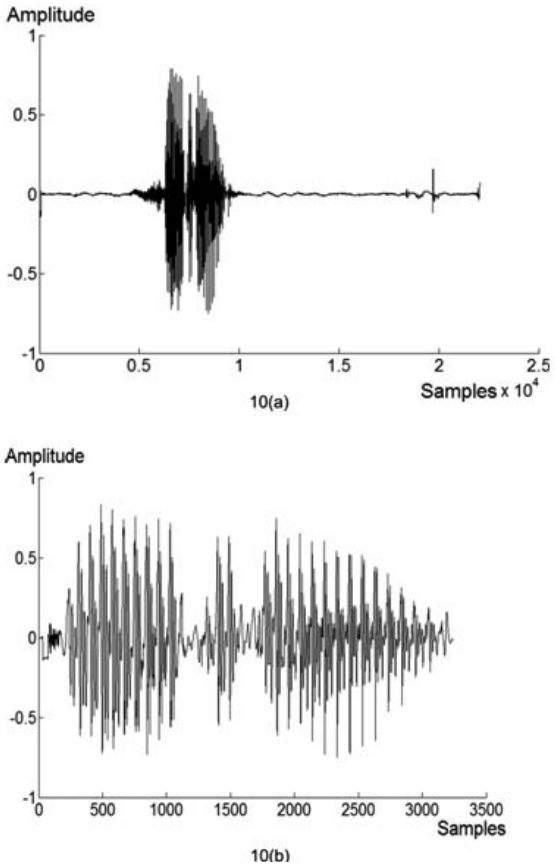


Figura 10: (a) Señal original. (b) Señal luego de Eliminación de Silencios.

3.2 Decimar

El objetivo de la decimación es eliminar muestras del comando de voz. Para esto se tiene el largo del comando que se desea reconocer y el largo almacenado en la base de datos, que será una potencia de 2, entonces, se calcula cada cuantas muestras es necesario eliminar una.

El criterio para seleccionar la que será eliminada, se escogerá aquella muestra que este más cercana al promedio, para esta selección no se consideran los extremos. Finalmente realizando un desplazamiento de la señal a la izquierda e iterando se obtiene la señal del largo deseado.

En la Figura 11, se muestra gráficamente la idea de la eliminación de muestras:

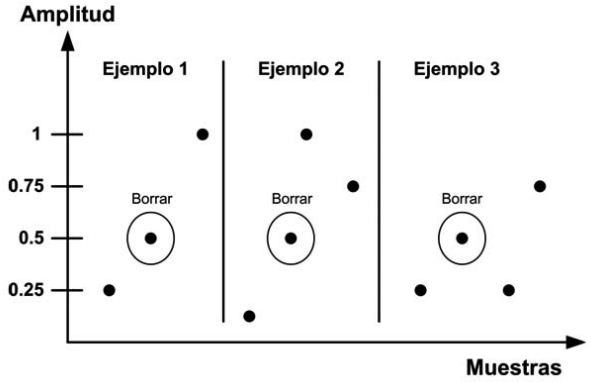


Figura 12: Inserción de Muestras

Este paso se utilizará en dos instancias, al momento de ingresar el comando de voz como un patrón, y al momento en que un comando de voz debe ser reconocido. En la primera instancia debemos obtener el número de niveles exactos para utilizar la H-WT.
La segunda instancia en que este proceso se utiliza, es cuando el comando de voz que se desea reconocer resulta más extenso en el número de muestras que tiene el comando almacenado, con el cual se efectuará la comparación.

3.3 Interpolar

El objetivo de la interpolación es agregar muestras a la señal para lograr que el largo del comando de voz que se desea reconocer sea igual del largo del comando almacenado.
Se calcula el número de muestras que deben ser agregadas y se determina la posición i e i+1, entre las cuales una nueva muestra será agregada. Esta se calculará promediando los valores de la posición i e i+1. Para insertar esta nueva muestra es necesario realizar un desplazamiento a la derecha una posición a partir de la posición i.
En la Figura 12, se muestra esta idea. Este proceso será iterativo mientras la señal que se desea reconocer no alcance la longitud de la señal de voz almacenada.

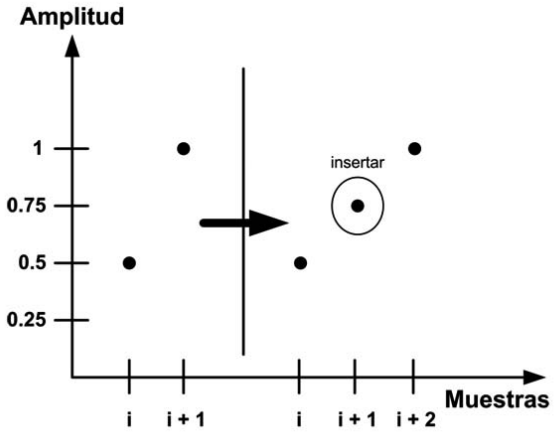


Figura 12: Inserción de Muestras

3.4 Codificar

El proceso de decimación e interpolación son la base para la correcta aplicación de la transformada wavelet Haar, fundamentalmente porque necesitamos que la longitud de ambas señales sea la misma y de esta forma obtener coeficientes que puedan ser comparables entre sí.

Se divide la señal una vez acondicionada en 4 lapsos iguales, cada uno es tratado por la transformada Haar entregando cada uno de ellos coeficiente de escala y wavelet. Como la señal tiene un lago máximo 4096 muestras, cada lapso que procesa la transformada Haar cuenta con 1024 muestras. El número de niveles de resolución se obtendrá de la ecuación (10).

Haciendo el número de iteraciones igual a N-2, tendremos los 8 coeficientes propuestos. Luego de esta etapa se graba en la base de datos los coeficientes encontrados.

3.5 Reconocer

En esta etapa se confrontan los coeficientes obtenidos del proceso de codificar, con los obtenidos de la base de datos para un comando determinado. Dada la imposibilidad de obtener los mismos valores se define un cierto intervalo de validez, para una comparación exitosa.

Se debe tener presente que existirá una iteración en el algoritmo donde la muestra capturada se compara con cada una de las almacenadas. En caso de recorrer todos los comandos y no tener éxito en la búsqueda, el algoritmo termina sin reconocer el comando.

El algoritmo fue implementado en Matlab para su simulación y prueba. Es relevante mostrar que se utilizó un arreglo de 4096 elementos para la obtención de los resultados. Los tiempos de ejecución estuvieron en el entorno de un segundo, en un computador personal HP/Compac 6510b, con procesador Intel Centrino y Windows XP, lo que está dentro de especificación.

4 Algoritmo

El algoritmo de Captura y Almacenamiento del Comando de Voz, es el siguiente:

Algorítmo: Captura y Almacenamiento del Comando

```
ReadSignal (Signal, Fs, Bits, Word)
LS = length (Signal)
Acondsignal (Signal, LS, posinit, posfinal)
LengthSignal        =posfinal – posinit + 1
LevelFloat          =log2( LengthSignal)
Levels              = floor(levelFloat)
EraseNumber         =LengthSignal-2levels
Samples             =floor((2levels-2levels+1) / 2)
If (EraseNumber <   = Samples)
Decimation(Signal, LS, Levels)
Else
Interpolation(Signal, LS, (Levels + 1))
End
H-WTCalculation(Signal,  Levels, SignalCoef)
SaveParametres(Word, SignalCoef, Coef, Levels)
```

La variable Signal contiene el vector del comando de voz, en el cual se acondicionan y se eliminan los silencios (AcondSignal), se calcula el número de muestras a eliminar (EraseNumber) con el cual se decide si se decima o interpola. Luego se calcula la Transformada Haar (CalculateHWT) y finalmente se graban los parámetros (SaveParameters).

El algoritmo de reconocimiento del comando de voz es el siguiente:

Algorítmo: Reconocimiento del Comando

```
ReadSignal(Signal, Fs, Bits, Word)
LS              = length(Signal)
Acondsignal(Signal, LS, posinitial, posfinal)
LengthSignal    = posfinal – posinit + 1
LevelFloat      = log2( LengthSignal)
Levels          = floor(levelFloat)
EraseNumber     = LengthSignal-2levels
GetCommands(Word, CommandCoef, Levels, Parms, eof)

Do while not (eof)
If Levels > CommandLevels
Decimation(Signal, LS, CommandLevels)
Else
Interpolation(Signal, LS, CommandLevels)
End
H-WTCalculation(Signal,  Levels, SignalCoef)
Comparison(CommandCoef, SignalCoef, Word, ErrorVec-
tor)
GetCommands(Word,CommandCoef,CommandLevels,
Parms, eof)
End
Recognition(ErrorVector, Successful)
```

En el algoritmo de reconocimientos se tiene la variable Signal, en la cual se acondicionan y eliminan los silencios (Acondsignal), se obtiene un comando de voz de la base de datos y se determina si se debe decimar o interpolar. Una vez realizado esto se calcula la transformada Haar (CalulateHWT). Luego se efectúa la comparación (Compare) en función de los coeficientes wavelets determinando un intervalo de error. Una vez terminado el recorrido de todos los comandos, se realiza el reconocimiento teniendo en la variable SucessFull, el comando reconocido: 'abrir', 'cerrar' o 'grabar', en caso contrario, el literal 'not-exist'.

Las pruebas que se realizaron con los comandos, fueron repetidos 50 veces cada uno, por un solo hablante, en formato WAV a 11025 [Hz].

5 Resultados

Para la medición de la eficiencia del sistema se utilizó el porcentaje de aciertos, expresada en la ecuación (12):

$$\text{Aciertos} = \frac{\text{Nro.Aciertos}}{\text{Nro.Casos}} \text{x100} \qquad (12)$$

Para el reconocimiento se utilizaron los 4 coeficientes wavelets correspondientes al nivel de resolución n-2, estos coeficientes evolucionan de manera diferente en cada uno de los cuartos y representan la razón de cambio en cada uno de ellos. En la tabla 1, se pueden ver los resultados para los tres comandos considerados.

Tabla 1: Resultados de Búsqueda, para un hablante.

Comando	Repeticiones	% Aciertos
Cerrar	50	70,83
Grabar	50	85,10
Abrir	50	61,70

Efectuando un análisis más detallado de los resultados y en particular sobre aquellos comandos que fueron rechazados, se observa que el efecto de la decimación o interpolación y la variabilidad en la energía, fueron parámetros importantes que afectaron los resultados. En efecto, en la tabla 2, se entregan resultados considerando aquellas repeticiones que están en una banda de +/-20% en la energía respecto del promedio y en +/- 20 % en el largo del comando respecto de la media.

Tabla 2: Resultados de Búsqueda, para un hablante, considerando la energía y el largo del comando.

Comando	Repeticiones	% Aciertos
Cerrar	10	80
Grabar	10	90
Abrir	10	80

Esto nos sugiere que para mejorar los resultados es necesario considerar una rutina que aplique una ganancia de corrección de la amplitud del comando que se desea reconocer respecto del comando almacenado.
Por otra parte, para determinar el impacto que tiene la cantidad de comandos considerados en los resultados obtenidos, se eliminó el comando "Abrir" del análisis de resultados.
En la tabla 3, se entregan los resultados para los dos comandos, "Cerrar" y "Grabar".

Tabla 3: Resultados de Búsqueda, para un hablante.

Comando	Repeticiones	% Aciertos
Cerrar	50	75,00
Grabar	50	89,36

6 Conclusiones

Este trabajo presenta un algoritmo simple para el reconocimiento de un conjunto acotado de comandos de voz, que hace uso limitado de memoria para la captura, almacenamiento y reconocimiento, permitiendo así viabilizar el uso de microprocesadores de recursos limitados.

La eliminación de silencios, en el paso de acondicionamiento, resultó ser significativo el la tasa de rechazo que llegó a ser de un 6% para los resultados de la tabla 1, esto fue producto del transitorio eléctrico de conexión que aparece en el micrófono después de activar la captura.

El análisis de las tablas 1 y 3, sugiere que se puede manejar el número de comandos y estudiar a priori aquellos comandos que produzcan bandas lo mas distanciadas posible para asegurar un mejor resultado.

La tabla 2, hace explicito el caso en el que la energía y el comando varían mas menos en un 20%, el número de aciertos es igual o superior al 80%, lo que a subes deja entrever que el volumen de voz debe ser alto y parejo en los ejecutantes del comando.

Los efectos de decimación, interpolación, variabilidad de la energía de la señal y efectos transitorios de conexión del micrófono, fueron los aspectos más importantes que afectaron en forma directa los resultados obtenidos.
La aplicación de la transformada Haar resultó eficaz al nivel del 75% con dos comandos en el desarrollo de este algoritmo, a pesar de su sencillez, posibilitando obtener los resultados de acuerdo con las características de diseño impuestas en este trabajo.

La simplicidad de los algoritmos planteados, probablemente no permiten una tasa de reconocimiento cercana al 95%, pero deja ver la viabilidad de implementarlos para el reconocimiento de voz.

Referencias

1. Barbon S., Capobianco G., Sasso L., Silva E., Lopes F., Scalassara P., Maciel C., Pereira J. C. and Chen S. (2009). Wavelet-based dynamic time warping, Journal of Computational and Applied Mathematics, Elsevier Science Publishers, vol. 227, pp. 271-287.

2. Burrus C., Gopinath R. and Guo H. (1998). Introduction to Wavelets and Wavelet Transforms. Prentice Hall. New Jersey.

3. Chui C. (1997). Wavelets: A Mathematical Tool for Signal Analysis. SIAM. Philadelphia.

4. Faundez P. and Fuentes A. (2000). Procesamiento Digital de Señales Acústicas Utilizando Wavelets. Acoustic Engineering Thesis, Instituto de Matemáticas, Universidad Austral de Chile.

5. Gorin A. and Mammone R.J. (1994). Introduction to the Special Issue on Neural Networks for Speech Processing. IEEE Transaction on Speech and Audio Processing, vol. 2, pp. 113-114.

6. Gupta M. and Gilbert A. (2001). Robust Speech Recognition Using Wavelet Coefficient Features. IEEE Workshop on Automatic Speech Recognition and Understanding, ASRU '01, pp. 445- 448.

7. Huang X., Acero A. and Hon H-W. (2001). Spoken Language Processing: A Guide to Theory, Algorithm and System Development. Prentice Hall PTR. New Jersey.

8. Jackson L. B. (1985). Digital Filters and Signal Processing. Kluwer Academic Publishers, University of Louisville, Department of Electrical and Computer Engineering. U.S.A.

9. Jian H., Han Z., Scucces P. and Robidoux S. and Sun Y. (2000). Voice-Activated Environmental Control System for Persons with Disabilities. Proceedings of the IEEE 26th Annual Northeast Bioengineering Conference,

10. Juang B. H. and Rabiner L. R. (1991). Hidden Markov Models for Speech Recognition. Technometrics, vol. 33, no 3, pp. 251-272.

11. Karasawa S. and Sakuraba H. (2007). Use of Haar wavelet transform based multiple template matching for analyses of speech voice. EATIS '07 Proceedings of the 2007 Euro American conference on Telematics and information systems.

12. Kaschel H., Watkins F. and San Juan E. (2005). Compresión de Voz Mediante Técnicas Digitales para el Procesamiento de Señales y Aplicación de Formatos de Compresión de Imágenes. Revista SCIELO-INGENIARE: Revista Chilena de Ingeniería, vol. 13, nº3, pp. 4-10.

13. Kirschning I. (1998). Automatic Speech Recognition with the Parallel Cascade Neural Network. PhD Thesis. Tokushima University Japan.

14. Kosko B. (1991). Neural Networks for Signal Processing. Prentice Hall, U.S.A.

15. Mendoza A., Archila L. and Ardila J. A. (2001). Caracterización del Intervalo QT en una Señal Electrocardiografica Usando la Transformada Wavelets. II Congreso latinoamericano de Ingeniería Biomédica, La Habana.

16. Nievergelt Y. (2001). Wavelet Made Easy. Department of Mathematics, Eastern Washington University. U.S.A.

17. Ocampo J., Mena E., Cabrera A. and Chierez, J. (2005). Análisis Matricial basado en la Transformada Wavelets para diagnostico cardiaco. VI Congreso de la Sociedad Cubana de Bioingeniería, La Habana.

18. Olmos-Castillo H. I. (2006). Uso de los wavelets para identificar el comportamiento de un proceso, Revista Mexicana de Ingeniería Química, vol. 5, pp. 183-187.

19. Phelps E., Pruehsner W. R. and Enderle J. D. (2000). Soni-Key Voice Controlled Door Look. Proceedings of the IEEE 26th Annual Northeast Bioengineering Conference, pp. 165-166.

20. Rabiner L. (1989). A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. Proceeding of the IEEE, vol. 77 (2), pp.257-286.

21. Rumelhart D. E., McClelland J. L. and the PDP research group. (1986). Parallel distributed processing: Explorations in the microstructure of cognition, vol. 1. MIT Press. U.S.A.

22. Soong F.K. and Huang E.-F. (1991). A tree-trellis based fast search for finding the N best sentence hypotheses in continuous speech recognition. International Conference on Acoustics, Speech, and Signal Processing, vol. 1, pp. 705 - 708.

23. Sydral A., Bennet R.and Greenspan S. (1995). Applied Speech Technology. CRC Press. U.S.A.

24. Thawonmas R. and Iizuka K. (2008). Haar Wavelets for Online-Game Player Classification with Dynamic Time Warping. Journal of Advanced Computational Intelligence and Intelligent Informatics, Special issue on Intelligence Techniques in Computer Games and Simulations, vol. 12, no 2, pp. 150-155.

25. Tsai D-M and Chiang C-H. (2002). Rotation-Invariant pattern matching using wavelet descomposition. Pattern Recongnition Letters, Elsevier Science Inc., vol. 23, pp 191-201.

26. Vaidyanathan P. (2008). The Theory of Linear Prediction. California Institute of Technology, Morgan & Claypool Publishers. U.S.A.

27. Vera P., Fernandez G., Chavez S. and Leiva A. (2006). Compresión de Imagen para Aplicaciones en telemedicina por transformada Wavelets. XII Congreso Internacional de Telecomunicaciones, Universidad Austral, Valdivia. Chile.

28. Waibel A. and Lee K. (1990). Readings in Speech Recognition. Morgan Kaufmann Publishers Inc., U.S.A.