BioVoiceDemo: Prototipo para Verificación de Locutor en la plataforma iPhone *

Jon A. Gómez and Leandro Graciá

Instituto Tecnológico de Informática,
Universidad Politécnica de Valencia, Spain
jon@dsic.upv.es,lgracia@iti.upv.es
http://www.iti.upv.es

Resumen Presentamos el desarrollo de un prototipo para verificación de locutor en entornos móviles, particularmente para el iPhone de Apple. El prototipo ofrece una interfaz de usuario sencilla con dos apartados, entrenamiento y test. El entrenamiento facilita la adquisición de voz para obtener muestras y entrenar los modelos acústicos del usuario a autenticar. El test solicita al usuario una secuencia de dígitos generada aleatoriamente.
La verificación del locutor se realiza mediante doble comprobación. Con los modelos acústicos se comprueba si se trata del usuario auténtico o de un impostor. Con la secuencia fonética de los dígitos solicitados y un grafo de fonemas que representa la pronunciación se valida si lo pronunciado corresponde a dicha secuencia.

Keywords: identity verification, voice biometrics, speech recognition

1. Introducción

Los métodos biométricos de identificación permiten reconocer a una persona basándose en características fisiológicas o de comportamiento. Se caracterizan por la necesidad de que la persona esté físicamente en el lugar de identificación. Y requieren más o menos colaboración del usuario según los métodos utilizados y las características a medir.

Al elegir las características biométricas para identificar personas en una aplicación de seguridad, conviene tener presente que las características fisiológicas no están sujetas a variaciones voluntarias como sí lo están las características de comportamiento. La voz es un ejemplo de característica biométrica de ambos tipos que resulta atractiva en biometría por varias razones: las muestras se toman de manera no molesta para el sujeto, y es particularmente útil para accesos remotos o realizar transacciones a través de Internet [1].

La desventaja que ofrece la voz es la variabilidad, ya sea ésta voluntaria o involuntaria. Como ejemplo de variabilidad involuntaria podemos destacar los cambios espectrales cuando el locutor fuerza la voz en ambientes ruidosos o si

* Trabajo subvencionado por el MICINN en el contrato TIN2008-06856-C05-02

está congestionado. Otros factores ajenos al locutor son el ruido ambiente y las distintas características de los sistemas de adquisición. En cuanto a variaciones voluntarias, el caso más preocupante es cuando un locutor intenta suplantar a otra persona simulando su voz. El objetivo principal de un sistema como el presentado aquí es justamente no autenticar impostores.

Los sistemas de reconocimiento biométrico de personas basados en la voz no requieren apenas esfuerzo para el usuario. Sin embargo, el éxito de estos sistemas está íntimamente ligado a la fatiga que puedan llegar a causar en el locutor, es decir, cuanta mayor colaboración se requiera por parte del usuario más fácilmente se verán alterados los parámetros biométricos, en nuestro caso las características acústicas y fonéticas. Un entorno ruidoso puede provocar que el usuario deba repetir varias veces una frase o una secuencia de dígitos, llegándose, en casos extremos, a que la alteración momentánea impida que el sistema verifique o identifique al locutor. En el caso de los dispositivos móviles este problema se agrava debido a los diferentes entornos en los que se requerirá la identificación o verificación mediante la voz.

Existen dos modos de funcionamiento para los sistemas de reconocimiento basados en parámetros biométricos: verificación e identificación. En el primer caso, el usuario se identifica mediante un método no biométrico, como un código numérico (PIN) o una tarjeta, y el sistema ha de verificar que la identidad proporcionada se corresponde con la realidad. En el segundo caso, se trata de averiguar la identidad del sujeto sin información adicional (no biométrica) buscando en una base de datos una representación de parámetros biométricos suficientemente aproximada. La identificación puede realizarse en dos modalidades: conjunto de locutores cerrado o abierto. Cuando el conjunto de locutores es cerrado se asume que el locutor que reclama una identidad siempre es uno de los existentes en la base de datos. En esta modalidad se dispone de un modelo previamente entrenado para todos los locutores posibles. Por el contrario, cuando se trabaja con un conjunto de locutores abierto no se puede asumir que el locutor reclamante de una identidad está en la base de datos, en este caso se deben manejar umbrales para decidir si se acepta una identidad de las posibles o se rechaza el intento de identificación. El ajuste de los umbrales debe encontrar un punto equilibrio en el que apenas se den falsos positivos y el ratio de falsos rechazos sea aceptable para los usuarios. La verificación de un único locutor es un caso particular de la modalidad en la que el conjunto de locutores es abierto, únicamente hay un locutor objetivo a verificar frente al resto de posibles locutores (impostores).

El prototipo que presentamos en este trabajo está desarrollado para la plataforma iPhone[1], se trata de un sistema de verificación donde la identidad del usuario se asume, el propietario del dispositivo. Es un sistema de verificación para un sólo locutor que debe resolver, lo mejor posible, como representar correctamente al resto de posibles locutores mediante un modelo universal. La autenticación de un usuario debe superar dos comprobaciones: verificación del locutor y validación de lo pronunciado. La verificación del locutor se lleva a cabo con independencia del texto solicitado en base a dos GMM, uno que representa

[1] iPhone es una marca registrada de Apple Inc.

el universo de locutores posibles o UBM (*Universal Background Model*) y otro adaptado al usuario a autenticar [1,2]. La validación de lo pronunciado consiste en calcular una medida de confianza de lo pronunciado con respecto a la secuencia fonética de la clave. Cuando dicha medida supera cierto umbral ajustado empíricamente se valida la muestra de voz, en caso contrario se rechaza. La clave es una secuencia de dígitos (PIN) generada aleatoriamente que se le presenta al usuario para que la pronuncie. La generación automática de la clave a solicitar no requiere que el usuario la memorice y minimiza el riesgo de que su voz grabada se reutilice para suplantarlo.

Existen en el *AppStore* aplicaciones que ofrecen solución al mismo problema para el iPhone, pero en su mayoría requieren el envío de la voz del usuario a un servidor, tanto de las adquisiciones para entrenamiento como la de las pronunciaciones a verificar. Este aspecto puede no gustar a muchos usuarios dado que su voz puede quedar registrada sin su consentimiento. Nuestro prototipo realiza todos los cálculos sobre el dispositivo físico sin necesidad de conexión a ningún servidor, tanto los cálculos de adaptación de modelos como los de verificación.

A continuación, la segunda sección describe el prototipo. La tercera sección presenta los resultados de los experimentos off-line utilizados para ajustar los umbrales de verificación del locutor y validación de lo pronunciado. La cuarta sección comenta los obstáculos más importantes encontrados en la implementación sobre la arquitectura del iPhone OS. Y finalmente se discuten las conclusiones en la última sección.

2. Descripción del sistema

El prototipo que presentamos en este trabajo consta de dos componentes independientes y bien diferenciados. El primer y principal componente es la librería o API de procesamiento de voz, que puede utilizarse en aplicaciones de cualquier índole cuya interacción con el usuario sea susceptible de mejorarse mediante la voz, especialmente aquellas en las que se requiera autenticar al usuario propietario para restringir el acceso a la propia aplicación o a datos personales almacenados en el iPhone. El segundo componente es la interfaz de usuario, escueta pero suficiente para demostrar el funcionamiento del sistema de verificación de locutor.

La Figura 1 ilustra el funcionamiento multihilo del subsistema de verificación del locutor. Las aplicaciones que hagan uso de la librería interactuarán con el subsistema mediante un reducido conjunto de funciones que forman la API (*Application Programming Interface*). Los distintos hilos de ejecución (*threads*) se comunican mediante colas FIFO controladas con semáforos, actuando cada hilo como productor y/o consumidor según el caso.

El hilo de ejecución nº 1 lo constituyen el subsistema *Core Audio* de Apple y el preprocesador. Conforme se adquiere la señal vocal ésta se preprocesa y los vectores acústicos obtenidos se van encolando. Este hilo de ejecución actúa como el productor de la primera cola.

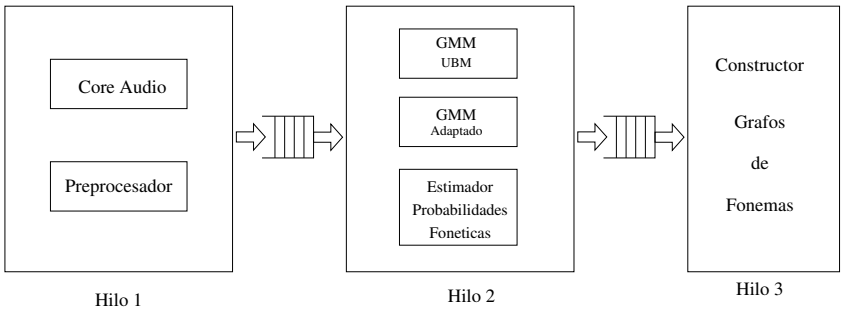


Figura 1: Módulos del subsistema que operan en modo *pipe-line*.

El hilo de ejecución nº 2 va consumiendo vectores acústicos, realiza los cálculos pertinentes para obtener vectores con probabilidades fonéticas, y añade éstos en la siguiente cola. Este hilo actúa como consumidor de la primera cola y productor de la segunda.

El hilo de ejecución nº 3 va consumiendo vectores con probabilidades fonéticas, explora la evolución de las probabilidades de las distintas unidades fonéticas, y progresivamente va construyendo un grafo de fonemas como representación de la pronunciación. Su salida es un grafo de fonemas que se envía al módulo que valida la pronunciación según la secuencia fonética solicitada.

2.1. Captura y preproceso de la señal vocal

Core Audio es un subsistema de control de dispositivos de audio que facilita y estandariza la reproducción y captura de señal de audio sobre la plataforma iPhone OS. Su puesta en marcha crea automáticamente un hilo de ejecución que se encarga de manejar el micrófono y va rellenando un conjunto de búfferes preparados antes de arrancar la captura. Cada vez que se llena un búffer se invoca a una función facilitada por el programador (*callback*) que extrae la señal del búffer. En nuestro caso esta función se encarga de preprocesar la señal y añadir los vectores acústicos en la primera cola.

El preprocesador aplica un filtro de preénfasis sobre la señal vocal en el dominio del tiempo y seguidamente la convierte en una secuencia de vectores acústicos. Extrae un vector cada 10 ms aplicando una ventana de Hamming de 20 ms. Cada vector acústico contiene 39 parámetros: la energía, los primeros 12 MFCC (*Mel Frequency Cepstral Coefficients*), más las primeras y segundas derivadas.

Adicionalmente, con el objeto de minimizar el ruido ambiente según la ubicación en la que se encuentre el usuario con su iPhone, se ha aplicado una técnica de eliminación de ruido con resultados similares a los obtenidos mediante filtros Wiener pero asequible en cuanto a consumo de CPU para un dispositivo móvil [3]. Esta técnica de eliminación de ruido se aplica en dos fases. La primera fase consiste en restar a cada canal del banco de filtros la componente de ruido

estimada cuando no se detecta voz, y en compensar su nuevo valor según un coeficiente calculado para cada canal en función del ratio entre la energía del canal y la componente de ruido. La segunda fase se aplica después de obtener los 39 parámetros acústicos, y consiste en el mapeo de cada parámetro por separado según dos distribuciones. Una distribución calculada sobre el conjunto de muestras de señal vocal utilizadas en el aprendizaje, que representan la distribución estándar de cada parámetro acústico, y otra distribución calculada sobre las muestras de voz capturadas por el sistema en su funcionamiento habitual, que se actualizan progresivamente.

2.2. Modelos acústico-fonéticos y verificación del locutor

El segundo hilo procesa cada vector acústico para obtener un vector con probabilidades fonéticas. Cada componente es la probabilidad *a posteriori* de una unidad fonética dado un vector acústico.

El cálculo de las probabilidades fonéticas se realiza en dos pasos. El primero obtiene las densidades de probabilidad de las clases acústicas a partir del GMM. Cada clase acústica se modela mediante una distribución normal o de Gauss. El segundo paso combina las densidades de probabilidad de las clases acústicas con probabilidades condicionales para obtener las densidades de probabilidad de cada unidad fonética, aplicando la regla de Bayes se obtienen las probabilidades *a posteriori* de las unidades fonéticas [4,5].

Para la tarea de verificación del locutor se dispone de dos GMM, el general estimado a partir de un corpus multilocutor que representa el UBM, y el adaptado al usuario a autenticar. Para el cálculo de las probabilidades fonéticas descrito en el párrafo anterior se utiliza el GMM del UBM. En paralelo se calculan las densidades de probabilidad en base al GMM adaptado al locutor. Entonces se dispone de dos verosimilitudes para una misma pronunciación, una con respecto al UBM y otra con respecto al locutor, calculadas en términos logarítmicos y normalizadas para no depender de la talla de la muestra de voz.

$$\log L(X|UBM) = \frac{1}{T}\sum_{t=1}^{T} \log p(x_t|UBM) = \frac{1}{T}\sum_{t=1}^{T} \log\Big(\sum_{a\in A_{UBM}} p(x_t|a)\Big)$$

donde $X = \{x_1, ..., x_T\}$ es la secuencia de vectores acústicos de una pronunciación, A_{UBM} es el conjunto de clases acústicas o gaussianas del GMM del UBM, y $p(x_t|a)$ es la densidad de probabilidad condicional de observar el vector acústico x_t con respecto a la clase acústica a, calculada como una distribución normal o de Gauss. $p(X|LOC)$ se calcula idénticamente con respecto al conjunto de clases acústicas del GMM adaptado al locutor A_{LOC}.

La decisión de si es el locutor auténtico o un impostor se toma comparando el *log Likelihood Ratio (LLR)* frente a un umbral ajustado empíricamente [1,2]:

$$\Lambda(X) = \log p(X|LOC) - \log p(X|UBM)$$

El LLR se puede normalizar con el propósito de hacer más robusto el sistema de verificación. Existen distintas aproximaciones para la normalización como son Z-norm, H-norm y T-norm, cuya diferencia radica en cómo son estimados los factores de desplazamiento (la media) y de escalado (la desviación típica) [2]. En nuestro prototipo queda pendiente probar qué normalización dará mejores resultados, actualmente no se utiliza normalización del LLR.

Adaptación del modelo al locutor. El GMM adaptado al usuario del iPhone se calcula a partir del GMM del UBM. Por cada clase acústica se calcula una nueva estimación de la media y la varianza como en el algoritmo EM, sin embargo, la media y la varianza originales se alterarán a partir de las nuevas estimaciones según un coeficiente que pondera los valores nuevos frente a los existentes. Las clases acústicas poco representadas en el conjunto de muestras del locutor apenas serán alteradas [1,6].

2.3. Grafos de fonemas y validación de lo pronunciado

Para la verificación del locutor el sistema solicita al usuario que pronuncie una secuencia de dígitos generada aleatoriamente cada vez. Entonces el sistema captura la muestra de voz hasta un máximo de 10 segundos. Además de obtener las verosimilitudes acústicas como se comenta en la subsección anterior, con la secuencia de vectores de probabilidades fonéticas se construye un grafo de fonemas [5]. La validación de lo pronunciado con respecto a la secuencia fonética del PIN solicitado se realiza buscando el mejor camino de dicha secuencia fonética por el grafo de fonemas. Para obtener el mejor camino se utiliza un algoritmo de alineamiento temporal no lineal o DTW (*Dynamic Time Warping*), de fácil implementación sobre los grafos de fonemas obtenidos.

Los nodos del grafo de fonemas representan instantes de tiempo, y los arcos representan unidades fonéticas detectadas entre cada dos nodos. Los arcos incluyen además la probabilidad de que la unidad fonética haya sido pronunciada en el periodo de tiempo que abarca el arco.

La construcción de estos grafos se realiza explorando la evolución de las probabilidades fonéticas (probabilidades *a posteriori* de cada unidad fonética, incluidos los silencios). Manejando dos umbrales, uno para detección y otro para ampliación, se van detectando segmentos temporales en los que se considera que una unidad fonética ha sido pronunciada con cierta probabilidad. Los nodos se crean en aquellos instantes de tiempo en los que comienza o finaliza un segmento de cualquier unidad fonética. A medida que se avanza en el tiempo se van colocando arcos entre los nodos, con la restricción de que los arcos únicamente unen nodos consecutivos tal y como se puede ver en la Figura 2. De esta manera, un segmento de una unidad fonética estará representado por una secuencia de arcos entre cada dos nodos consecutivos [5].

Con el propósito de simplificar la DTW para que no considere los casos de inserciones, borrados o sustituciones, los grafos son modificados para incluir un modelo de error consistente en añadir arcos para aquellas unidades que no fueron detectadas entre cada par de nodos. El grafo resultante es más denso que el

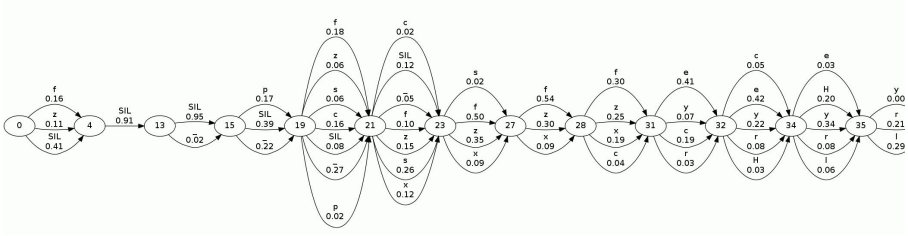


Figura 2: Trozo de un grafo de fonemas.

presentado en la Figura 2, pero su exploración mediante la DTW queda simplificada al únicamente contemplar las operaciones de coincidencia. Existen distintas aproximaciones para calcular la probabilidad fonética o medida de confianza de los arcos añadidos por el modelo de error. Por simplificar, en nuestra implementación se utiliza la propia probabilidad de cada unidad fonética ya disponible en los vectores con las probabilidades fonéticas.

Los grafos de fonemas son útiles para continuar el procesamiento del habla de cara al reconocimiento, pero en nuestro prototipo resultan útiles para que la DTW sea más liviana. Una DTW aplicada directamente sobre la secuencia de vectores con probabilidades fonéticas daría el mismo resultado, pero con mayor consumo de CPU debido a que el número de nodos del grafo es bastante inferior al número de vectores.

Fruto de buscar el mejor camino de la secuencia fonética por el grafo de fonemas se obtiene la siguiente medida:

$$score(X|UBM, SF) = \frac{1}{T}\sum_{t=1}^{T} \log Pr(u_{f,t}|x_t)$$

donde $SF = \{u_1, .., u_F\}$ es la secuencia de unidades fonéticas para validar X, $X = \{x_1, .., x_T\}$ es la secuencia de vectores acústicos, $u_{f,t}$ es la unidad fonética que la búsqueda del mejor camino ha determinado que se considere en el instante t, y $Pr(u|x_t)$ es la probabilidad *a posteriori* de que haya sido pronunciada la unidad fonética u al observarse el vector acústico x_t.

3. Preparación del UBM y experimentación off-line

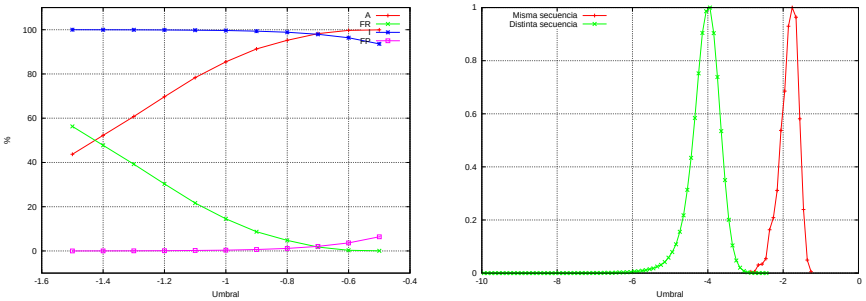
Para el entrenamiento de los modelos acústicos se ha utilizado el corpus fonético de la base de datos Albayzin [7]. Los modelos acústicos de nuestro sistema se componen de un GMM como representación de las clases acústicas, más una matriz de probabilidades condicionales que refleja el grado de relación de cada clase acústica con cada unidad fonética [4]. El GMM de los modelos acústicos representa el UBM, el GMM adaptado al locutor se obtiene a partir de este según se describe en la sección 2.2.

Utilizando los 204 locutores del subcorpus fonético de Albayzin se ha realizado una adaptación del GMM genérico a cada locutor por separado. De cada

locutor se han tomado la mitad de sus frases escogidas aleatoriamente para adaptar el GMM, y la otra mitad para obtener el LLR (ver sección 2.2). En la Figura 3a se presenta la evolución de varios ratios frente a un rango de valores del LLR. Estos ratios son: (A) el porcentaje de aciertos al validar al locutor correcto, (FR) el porcentaje de falsos rechazos, (I) el porcentaje de aciertos al rechazar una pronunciación de un locutor distinto, y (FP) el porcentaje de falsos positivos. El objetivo es conseguir un FP nulo con un FR lo más pequeño posible. Se observa que un umbral de decisión menor o igual a $-0{,}9$ cumple el objetivo. Tomar un valor más negativo asegura un FP prácticamente nulo a costa de aumentar el FR. Debemos tener presente que un FR alto aborrecerá al usuario por la cantidad de veces que deberá intentar ser autenticado.

Utilizando los mismos locutores y las mismas frases se ha diseñado un experimento similar para ajustar el umbral de validación. Para ello se ha estimado el valor de $score(X|UBM, SF)$ cuando el audio de la frase contiene la misma secuencia que se sabe fue pronunciada, y cuando el audio contiene una secuencia distinta. El resultado del experimento se muestra en la Figura 3b, donde puede observarse claramente la separación entre ambos histogramas según el valor de $score(X|UBM, SF)$. Fijar el umbral de validación mayor o igual a $-2{,}2$ garantiza que una grabación del locutor pronunciando un PIN distinto no será aceptada.

Aunque el prototipo funciona correctamente con el umbral de verificación a $-0{,}9$ y el de validación a $-2{,}2$, queda pendiente por realizar un estudio exhaustivo del comportamiento del sistema sobre el iPhone con un amplio conjunto de locutores.



(a) Verificación (b) Validación

Figura 3: Gráficas mostrando el comportamiento de distintas medidas para el ajuste de los umbrales de verificación y de validación.

4. Implementación sobre la plataforma del iPhone

El desarrollo de la librería sobre el iPhone ha estado plagada de obstáculos por las limitaciones de CPU y memoria del iPhone 3G. Aunque se trata de un

dispositivo móvil bastante potente, los requerimientos computacionales para el procesamiento del habla son muy altos. Las nuevas versiones ofrecen entornos más potentes pero el objetivo del prototipo es el iPhone 3G, pues todavía existen muchos dispositivos de esa generación en funcionamiento que representan un amplio mercado. Además, conseguir que funcione sobre el iPhone 3G asegura el funcionamiento sobre las siguientes versiones y el iPad.

4.1. Pérdida de precisión frente a consumo de CPU

Se ha reducido considerablemente el número de clases acústicas de los GMM para conseguir que el prototipo funcione en tiempo real sobre el iPhone 3G. El mismo sistema sobre un ordenador personal trabaja con GMM de unas 2000 gaussianas, mientras que para el iPhone se utiliza un GMM con menos de 500.

Todos los cálculos de coma flotante se realizan con reales de 32 bits o simple precisión. Utilizar doble precisión sobre el iPhone resultó directamente inviable.

Aunque internamente se trabaja con los logaritmos de las probabilidades, el calculo de la verosimilitud se debe realizar con el sumatorio de las densidades de probabilidad, también necesario para calcular las densidades de probabilidad a nivel fonético. Estos sumatorios de probabilidades cuyos valores están almacenados como sus logaritmos se han aproximado interpolando a partir de una tabla de valores precalculados. Calcular estos sumatorios utilizando las funciones exponencial y logaritmo también hace inviable el funcionamiento sobre el iPhone.

Las nuevas versiones del iPhone (3GS y 4) disponen de procesadores más potentes sobre los cuales se podrá encontrar un nuevo punto de equilibrio entre precisión y consumo de CPU.

4.2. Adaptación del modelo al locutor

Un detalle importante es el número mínimo de muestras del locutor necesarias para estimar un GMM adaptado que consiga unos resultados aceptables. A mayor cantidad de voz disponible mejores resultados se conseguirán, pero quizás el usuario de un dispositivo móvil no está dispuesto a grabar gran cantidad de muestras para entrenamiento. El prototipo exige un mínimo de 3 adquisiciones para realizar una primera adaptación al locutor. Como facilidad la aplicación admite que el usuario realice más adquisiciones cuando lo desee, guardándose únicamente los últimos 2 minutos de voz grabada. La adaptación al locutor, una vez disponible suficiente voz, se ejecuta a petición del usuario. Tras un periodo prudencial de uso, el usuario habrá grabado al menos 2 minutos de voz y los modelos adaptados obtendrán mejores resultados.

La adaptación de los modelos es un cálculo iterativo que consume mucha CPU. Todos los aspectos comentados en el punto anterior han influido significativamente para que la adaptación de los modelos se ejecute en un tiempo aceptable. En una versión definitiva puesta a la venta se aconsejará al usuario que realice la adaptación de los modelos cuando el iPhone esté cargándose. En caso contrario se reducirá significativamente la carga de la batería.

5. Conclusiones

El mayor obstáculo en el desarrollo del prototipo ha sido el desfase entre las necesidades de cálculo y la capacidad de cómputo del iPhone 3G, plataforma objetivo de nuestro desarrollo. El éxito sobre esta plataforma aseguraba el éxito en las versiones posteriores del dispositivo que obviamente son más potentes. El requerimiento inicial nos ha obligado a buscar un equilibrio entre consumo de CPU y precisión de los cálculos aproximados.

Hacer funcionar un sistema de reconocimiento del habla para un vocabulario grande sobre un dispositivo móvil no es todavía viable. Pero para tareas sobre dominios restringidos ya comienza a ser una realidad. Los primeros intentos de reconocedores sobre la plataforma iPhone capturaban la señal vocal y la enviaban a un servidor más potente para realizar el reconocimiento. Nuestra aproximación funciona sobre el dispositivo físico sin necesidad de apoyos externos a ningún nivel.

La verificación del locutor sobre el iPhone es ya una realidad aunque todavía nos quedan aspectos a mejorar. El reconocimiento del habla para una tarea con dominio restringido es un objetivo pendiente. En cualquier caso, el desarrollo de librerías para el procesamiento del habla sobre dispositivos móviles es una línea de trabajo con un futuro asegurado gracias al paulatino aumento de la capacidad de cómputo de estos dispositivos.

Referencias

1. Reynolds, D. A., Campbell, W. M.: Text-Independent Speaker Recognition, Springer Handbook of Speech Processing and Communication, Springer-Verlag GMBH, Heidelberg, Germany, 763–781 (2007).
2. Bimbot, F., Bonastre, J.F., Fredouille, C., Gravier, G., Magrin-Chagnolleau, I., Meignier, S., Merlin, T., Ortega-García, J., Petrovska-Delacrétaz, D., Reynolds, D.A.: A tutorial on Text-Independent Speaker Verification, EURASIP J. Appl. Signal Processing, Hindawi Publishing Corp., New York, NY, United States, 430–451 (2004).
3. Choi, E.: A Noise Robust Front-end for Speech Recognition Using Hough Transform and Cumulative Distribution Mapping, ICPR '06: Proceedings of the 18th International Conference on Pattern Recognition, IEEE Computer Society, Washington, DC, USA, 286–289 (2006).
4. Gómez, J.A., Castro, M.J.: Automatic Segmentation of Speech at the Phonetic Level, Structural, Syntactic, and Statistical Pattern Recognition, volume 2396 of *LNCS*, Springer-Verlag, 672–680 (2002).
5. Gómez, J.A., Calvo, M., Sanchis, E.: Localización de Palabras basada en Grafos de Fonemas, Procesamiento del Lenguaje Natural, nº 44, 59–66 (marzo 2010).
6. Reynolds, D.A., Quatieri, T.F., Dunn, R.: Speaker verification using adapted Gaussian mixture models, Digital Signal Processing **10**(1–3), 19–41 (2000).
7. Moreno, A., Poch, D., Bonafonte, A., Lleida, E., Llisterri, J., Mariño, J.B., Nadeu, C.: Albayzin Speech Database: Design of the Phonetic Corpus, Eurospeech, Berlín, Alemania, 653–656 (1993).