Sistema de Reconocimiento de Dígitos Conectados con Módulo de Adaptación al Micrófono para PDAs

Yago Pereiro Estevan, Ana Isabel García Moral, Ascensión Gallardo Antolín, Oscar Motilla Rodríguez, Fernando Díaz de María
Departamento de Teoría de la Señal y Comunicaciones
EPS, Universidad Carlos III de Madrid, Avda. de la Universidad, 30, 28911 Leganés (Madrid)
Telf: 916248805, Fax: 916248805
Correo Electrónico: ypereiro@tsc.uc3m.es

Resumen

En esta comunicación se presenta el diseño e implementación de una interfaz vocal que permite el reconocimiento de dígitos conectados y que ha sido desarrollada en aritmética de punto fijo para su instalación en una PDA. El reconocedor incluye además un módulo detector de actividad vocal y un módulo de adaptación. Esta herramienta se ha desarrollado continuando el trabajo del grupo en el ámbito del reconocimiento de habla en dispositivos de baja capacidad, campo en el que ya se realizó una herramienta de reconocimiento de palabras aisladas [1], siendo este prototipo inicial en el que nos hemos basado para el desarrollo de la nueva aplicación.

1. Introducción.

Actualmente, los dispositivos portables (en particular, teléfonos y PDAs) forman parte de nuestra vida diaria y tienen suficiente capacidad computacional y de memoria para soportar aplicaciones complejas. Este hecho junto con la conectividad que poseen con una amplia gama de servicios a través de redes de comunicación inalámbricas ha hecho que aparezcan multitud de aplicaciones. Sin embargo el pequeño tamaño de estos dispositivos hace complejo el uso de muchas de estas aplicaciones a través del interfaz clásico (teclado).

Las interfaces vocales son una solución para combinar de un modo eficiente la posibilidad de aprovechar estas aplicaciones y el pequeño tamaño de estos dispositivos. Hace algunos años la implementación de algoritmos de reconocimiento automático de habla (RAH) en este tipo de dispositivos era inconcebible debido a la alta demanda computacional de esta tecnología. Sin embargo hoy en día es posible, para tamaños de vocabulario y tareas limitados; en [2, 3, 4, 5, 6, 7, 8] se pueden ver buenos ejemplos.

Para mejorar las capacidades de los RAH para sistemas de baja capacidad son necesarias ciertas modificaciones. En particular, los sistemas de reconocimiento deben ser optimizados para minimizar el uso de recursos computacionales y de memoria.

En esta comunicación describimos el diseño de una interfaz de voz para reconocimiento de dígitos conectados embebida en una PDA. Continuando el trabajo comenzado con el reconocedor de palabras aisladas [1] se ha realizado el paso al reconocimiento de dígitos conectados, paso que representa un salto cualitativo respecto al diseño anterior y constituye una etapa previa a la implementación del reconocedor de habla continua.

A continuación realizaremos, en las secciones 2, 3 y 4, una explicación del sistema de RAH, así como de las modificaciones introducidas respecto al prototipo de dígitos aislados presentado en [1]. En la sección 5 presentamos las consideraciones que debemos tener en cuenta para el diseño del reconocedor en punto fijo. Finalmente, en la sección 6 presentamos los resultados obtenidos y en la sección 7 las conclusiones y líneas futuras de trabajo.

2. Sistema de reconocimiento automático del habla.

El sistema consta de dos módulos principales, el módulo de parametrización y el módulo de reconocimiento, tal y como puede observarse en la Fig. 1.

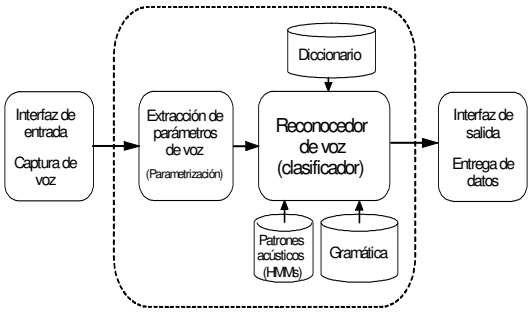

Figura 1. Esquema de bloques del sistema de reconocimiento automático del habla (RAH).

3. Parametrizador.

El módulo de parametrización o *"front-end"* está basado en el estándar desarrollado por ETSI [9] y se puede ver en la Fig. 2. El parametrizador extrae los 12 coeficientes MFCCs y la log-energía cada 10 ms, usando una ventana *Hamming* de 25 ms. Además, extrae 12 coeficientes delta-MFCC y la delta-log-energía, de modo que el vector de características final tiene 26 componentes.

En este bloque se han realizado dos mejoras respecto al prototipo inicial [1]. Por una parte, se ha optimizado el bloque de detección de principio y fin de voz para mejorar las prestaciones del sistema; y por otra, se ha realizado una transformación de Análisis de Componentes Principales (PCA, *Principal Component Analysis*) con el objetivo de reducir la dimensionalidad de los vectores acústicos y por tanto los requerimientos computacionales y de memoria.

3.1. Detector de principio y fin de voz.

El sistema de detección de actividad vocal implementado basa la decisión en la medida de la energía logarítmica de la señal de voz. Las etapas fundamentales del algoritmo son:

- Estimación del nivel de ruido de fondo lo cual posibilita la adaptación del umbral de decisión en tiempo real para obtener una respuesta que se acomode para distintos escenarios de funcionamiento.
- Supresión de ruidos cortos de alta energía exigiendo que las trazas de voz detectadas tenga una duración mínima.
- Etapa de reducción de ruido que mejora el rendimiento tanto del detector de voz como del reconocedor.

La finalidad principal del detector de voz es discriminar entre las tramas que deben ser evaluadas por un reconocedor de voz y las que no. Siguiendo esta premisa, se han añadido estas características al detector:

- La duración mínima de la traza de voz viene determinada por la duración mínima de una palabra.
- Se garantiza la no supresión del inicio y final de traza de voz que puede ser de baja energía mediante el añadido de unas tramas extras al principio y el final de las marcas de inicio y fin de voz obtenida por el detector.

La evaluación del algoritmo se ha llevado acabo utilizando una base de datos compuesta por 1000 ficheros de voz de distintos interlocutores libres de ruido en los que fueron marcados manualmente el inicio y final de la traza de voz. Posteriormente, han sido contaminados con diferentes tipos de ruido (blanco, F16) y con diferentes relaciones de señal a ruido (entre 3 y 12 dB) para simular diferentes escenarios de funcionamiento.

Los resultados obtenidos muestran la eficacia del algoritmo de detección al obtener unas tasas de acierto de voz y de silencio por encima del 90%, incluso en los escenarios más adversos.

3.2. Reducción de dimensionalidad.

Para reducir la dimensionalidad de los vectores de características, se aplicó una transformación utilizando PCA.

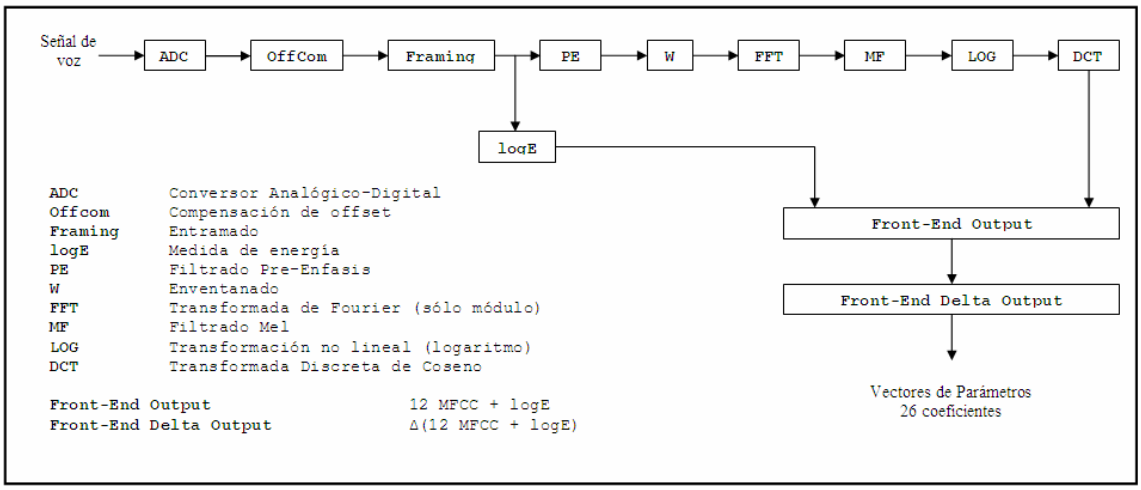

Figura 2. Diagrama de bloques del módulo de parametrización estandarizado por la ETSI [9].

PCA es una técnica ampliamente utilizada para la extracción de características y la reducción de dimensionalidad. Se basa en la premisa de que la mayoría de la información que nos permite distinguir entre las distintas clases consideradas está contenida en las direcciones de máxima varianza.

Supongamos que $X = [\, x_1\ x_2\ \ldots\ x_M \,]$ es la matriz *MxN* de *M* vectores acústicos, que suponemos centrada en el origen por simplicidad, y *N* dimensiones por cada vector. La matriz de covarianza puede ser aproximada por:

$$C = \frac{1}{M} X^T X$$

Por lo tanto la solución que buscamos será de la forma:

$$CU = C\Lambda$$

Donde **U** es la matriz de autovectores y Λ la matriz diagonal de autovalores.
Eliminando de la matriz de transformación **U** las filas correspondientes a los autovalores próximos a cero, podemos reducir la dimensionalidad *N* del espacio acústico inicial sin pérdida de información.

4. Módulo de clasificación.

Hemos tomado como referencia el sistema inicial desarrollado en [1], donde el reconocimiento de palabras aisladas se implementó utilizando el algoritmo *Token Passing* [10]. En dicho algoritmo, cada palabra del vocabulario se representa con un "modelo de palabra" construido a partir de modelos de fonema que forman una red de estados finitos, en la que cada par de estados conectados i y j tiene asociado un coste de transición p_{ij}, y cada estado j tiene asociada una función de coste local $d_j(t)$. Así, la extensión a palabras conectadas es trivial, pasando los modelos individuales de palabra a estar conectados en un modelo compuesto en bucle.

En el reconocimiento de dígitos conectados, utilizamos el mismo algoritmo de *Token Passing* en una versión extendida, tal y como se representa en la Fig. 3. Ahora los *tokens* incluyen además un 'identificador de camino' que es simplemente un puntero a la información de límite de palabra.

Además, se han introducido una serie de modificaciones respecto al sistema inicial, independientes de la aplicación de dígitos aislados, con el fin de mejorar las prestaciones y que describimos a continuación.

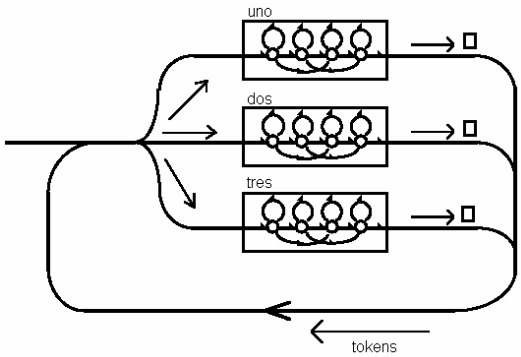

Figura 3. Diagrama de bloques de reconocimiento de dígitos conectados con el algoritmo Token Passing.

4.1. Modificación de los modelos acústicos.

Se ha realizado una modificación de los modelos acústicos con el fin de mejorar su fiabilidad. En estos modelos basados en HMMs (*Hidden Harkov Models)* hemos pasado de utilizar mezclas de una gaussiana por estado a trabajar con mezclas de 3 gaussianas por estado. Esta mejora incrementa en gran medida el rendimiento del reconocedor como veremos en la sección de experimentos.

4.2. Módulo de adaptación.

Debido a la necesidad del sistema de RAH de presentar un buen funcionamiento en diferentes situaciones (micrófonos, hablantes, ruidos, etc), y a que no es posible disponer de una base de datos específica para cada escenario posible, se ha añadido un módulo de adaptación al entorno. Su función es la de adaptar, a partir de unos pocos datos, los modelos de fonema generados independientemente del dispositivo sobre el que se va a utilizar, a las características propias del dispositivo en cuestión.

Este módulo evita tener que generar nuevos modelos cada vez que la aplicación se utiliza en un dispositivo diferente o incluso un entorno distinto, para lo cual se requerirían una gran cantidad de datos de voz grabados en el nuevo entorno y un software adicional de generación de modelos cuya implementación de forma eficiente no es posible hoy en día en una PDA.

En particular, hemos aplicado el algoritmo MLLR (*Maximum Likelihood Linear Regression*) [11] implementado con la herramienta de *HTK* [12] *HEAdapt*. Se han estudiado dos alternativas diferentes: la adaptación global y la adaptación utilizando clases de regresión. En el último caso, el árbol de regresión fue obtenido utilizando *HTK*.

5. Consideraciones de la implementación en punto fijo.

Los dispositivos portables, tienen, como ya hemos comentado antes, unas capacidades computacionales, de memoria y de consumo de potencia limitadas. La mayoría de estos dispositivos poseen una CPU que trabaja con una aritmética de punto fijo de 16 bits, por lo tanto la implementación del reconocimiento en estos dispositivos debe hacerse en aritmética de punto fijo. De otro modo la aplicación no sería eficiente y no podría realizarse en tiempo real. Por otra parte, hay que tener en cuenta que, aunque los algoritmos de punto fijo reducen el coste y el consumo de potencia, introducen dificultades como la cuantización, los errores de truncamiento, el control del desbordamiento, etc.

La representación en punto fijo de números enteros implica situar el punto que define las partes decimal y entera en algún punto del número, de modo que un número en punto fijo con Q_n significa que existen n bits a la derecha del punto.

La implementación en punto fijo del reconocedor necesita dos etapas de optimización: algorítmica y arquitectónica. La primera consiste en reemplazar funciones como el logaritmo natural o la raíz cuadrada por algoritmos de aproximación en punto fijo [13, 14, 15]. En la segunda se han considerado criterios como la representación en enteros de 16 bits, el uso de tablas precalculadas, el control de desbordamiento y la búsqueda de la mejor resolución Q_n en todos los pasos.

6. Resultados experimentales.

6.1. Base de Datos.

Para los experimentos que presentamos a continuación se han utilizado dos bases de datos diferentes. La primera de ellas es la llamada BDPDA, base de datos desarrollada por el Departamento de Teoría de la Señal y Comunicaciones (EPS- Universidad Carlos III de Madrid, España). Esta base de datos consta de 6300 locuciones de palabras aisladas grabadas por 50 hablantes en castellano, con hasta tres sesiones de cada hablante en un entorno de oficina sin ruido. Las locuciones han sido grabadas con una frecuencia de muestreo de 8 KHz. Debido a que la base de datos es demasiado pequeña para obtener resultados fiables se ha realizado una validación cruzada dividiéndola en 10 bloques y utilizando en cada simulación 9 de ellos para entrenar y el restante para el test.

La segunda base de datos utilizada ha sido la base de datos comercial en castellano *SpeechDat*, distribuida por ELDA [15] y grabada con una frecuencia de muestreo de 8 KHz en un entorno de teléfono. Para los experimentos de palabras aisladas utilizamos un subconjunto de dicha base de datos compuesto por 5053 palabras aisladas pronunciadas por 700 locutores, con un diccionario de, aproximadamente, unas 2600 palabras. Mientras que para los experimentos de habla continua (dígitos conectados) se empleó la base de datos completa: aproximadamente 71000 locuciones para entrenamiento pronunciadas por 3496 locutores diferentes y 2122 locuciones para test pronunciadas por 315 locutores diferentes.

6.2. Experimentos con PCA.

Se han realizado una serie de simulaciones con la BDPDA para determinar el comportamiento del sistema de reconocimiento al aplicar diferentes reducciones de dimensionalidad en los vectores acústicos. Estos experimentos se han realizado sobre el reconocedor de palabras aisladas.

En la Fig. 4 se puede observar que los vectores de características iniciales de 26 componentes pueden ser reducidos hasta a 15 componentes transformados por PCA sin una degradación significativa en la tasa de reconocimiento del sistema.

6.3. Implementación en punto fijo vs implementación en punto flotante.

Se han llevado a cabo una serie de experimentos para comparar el comportamiento de la implementación en punto fijo frente a la implementación en punto flotante. Estos experimentos se han realizado con la base de datos BDPDA con diferentes resoluciones (Q_7, Q_9 y Q_{11}) para las implementaciones en punto fijo y modelos acústicos de 1 y 3 gaussianas por estado.

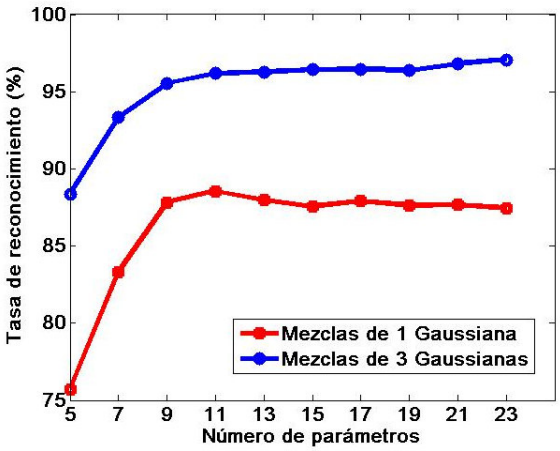

Figura 4. Representación de la tasa de reconocimiento aplicando diferentes reducciones de dimensionalidad con la transformación PCA.

En la Tabla 1 se pueden observar los resultados obtenidos con las diferentes implementaciones. Como se puede observar se consigue mantener las prestaciones sin ninguna pérdida al realizar el paso a punto fijo.

6.4. Experimentos de adaptación.

Estos experimentos se han llevado a cabo para estudiar el comportamiento de la adaptación del sistema al entorno en función de la cantidad de datos disponibles para la adaptación. El modelo base ha sido realizado con el subconjunto de la base de datos *SpeechDat* mencionado antes, mientras que para los cojnuntos de test y adaptación se ha utilizado la BDPDA.

La Fig. 5 muestra los resultados de reconocimiento conseguidos en las siguientes condiciones:

- "Sin adaptación": los modelos acústicos fueron entrenados usando sólo la base de datos *SpeechDat.*
- "Entrenamiento con la BDPDA": los modelos acústicos han sido entrenados utilizando diferentes subconjuntos del corpus de entrenamiento de la BDPDA, como se indica en el eje horizontal de la Fig. 5.
- "Adaptación Global": los modelos base se adaptan con el algoritmo MLLR usando para la adaptación diferentes subconjuntos del corpus de entrenamiento de la BDPDA, como se indica en el eje horizontal de la Fig. 5 y una sola transformación para todo el conjunto de unidades acústicas.
- "Adaptación con clases de regresión": igual que el caso anterior pero variando el método de adaptación, utilizando el MLLR con clases de regresión.

	Punto Flotante	Punto Fijo		
		Q7	Q9	Q11
1 mezcla	*88.85%*	*85.80%*	*85.80%*	*86.18%*
3 mezclas	*96.81%*	*96.85%*	*96.85%*	*96.71%*

Tabla 1. Tasa de reconocimiento de palabra para el reconocedor de palabras aisladas en punto flotante y punto fijo con diferentes niveles de precisión.

Como se puede ver en la Fig. 5 hay una gran diferencia entre los resultados con modelos de *SpeechDat* y BDPDA, obteniéndose unas prestaciones muy pobres del reconocedor en el caso de utilizar la *SpeechDat* sin adaptar. Si observamos la diferencia entre los resultados del entrenamiento con la BDPDA y los dos métodos de adaptación utilizados podemos sacar dos conclusiones: cuando tenemos una cantidad de datos suficiente del entorno en el que trabajamos, el entrenamiento con dichos datos es mejor que la adaptación con los mismos. Sin embargo, cuando la cantidad de datos disponibles decrece las prestaciones del método de adaptación se mantienen mientras que las del entrenamiento empeoran. Podemos observar también que la adaptación con clases de regresión mejora a la global pero tienden a converger cuando hay pocos datos. Esto se debe a que cuando la cantidad de datos disponible es muy pequeña, el árbol de regresión no tiene información suficiente para hacer múltiples transformaciones y el algoritmo se convierte en el mismo que el global.

6.5. Experimentos de dígitos conectados.

Una vez implementado el reconocedor de palabras aisladas se procedió al desarrollo de la versión de dígitos conectados.

En primer lugar se realizaron experimentos utilizando la base de datos completa *SpeechDat* (ver apartado 6.1) y la herramienta *HTK* [12] para obtener los modelos de lenguaje, y al mismo tiempo obtener unos resultados de referencia en punto flotante con los que comparar nuestro reconocedor de dígitos conectados.

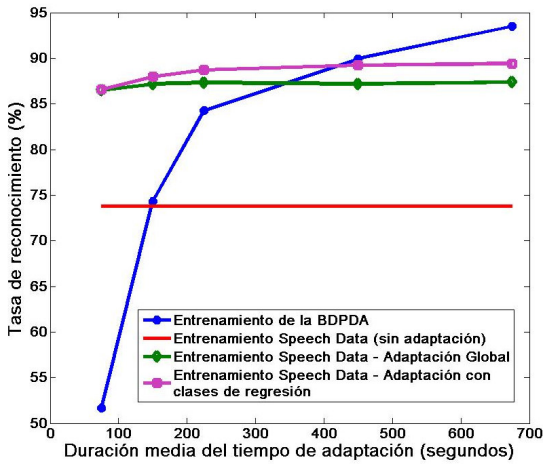

Figura 5. Representación de los resultados de la adaptación al entorno para diferentes cantidades de datos de adaptación con modelos acústicos de 3 gaussianas por estado.

Se obtuvieron modelos de con distinto número de mezclas por estado (1, 2, 4, 8, 16 y 32), pero sólo se utilizaron en las pruebas finales, es decir, con el reconocedor propio, los modelos de 4 mezclas por estado. La elección de este número de gaussianas por estado es una solución de compromiso, ya que los resultados en cuanto a tasa de reconocimiento son bastante buenos, a la vez que se evita la demora que provocaría en el proceso de reconocimiento el utilizar un número de mezclas muy elevado.

Al igual que en el prototipo inicial, se realizó primero una versión en punto flotante, comparando sus prestaciones con las obtenidas al realizar la misma tarea de reconocimiento utilizando la herramienta HTK. Así, conseguimos una tasa de reconocimiento de palabra (WAR: Word Accuracy Rate) de 86.20%, siendo este valor muy similar al obtenido con el reconocedor de referencia. Posteriormente, se pasó a la implementación en punto fijo del reconocedor de dígitos conectados, estando actualmente esta versión en pruebas.

7. Conclusiones y líneas futuras.

En esta comunicación se ha presentado un reconocedor de dígitos conectados para PDAs, como continuación del trabajo previo realizado en nuestro grupo sobre el desarrollo de un reconocedor de palabras aisladas embebido en dispositivos portables. Respecto a este prototipo de partida hemos incorporado mejoras en todas las etapas del sistema de reconocimiento.

En el módulo de parametrización se ha introducido un detector de principio y fin de voz con el objeto de mejorar la robustez frente al ruido y reducir el número de tramas a procesar por el reconocedor. También, se ha realizado una reducción de la dimensionalidad de los vectores acústicos mediante la aplicación de una transformación PCA que nos ha permitido reducir los requerimientos de memoria y de recursos computacionales del sistema.

Con respecto al módulo de reconocimiento, se han mejorado los modelos acústicos incrementando el número de mezclas por estado, lo que ha producido una disminución absoluta de la tasa de error de reconocimiento del sistema de, aproximadamente, un 7%. Asimismo, hemos añadido un módulo de adaptación al entorno acústico que favorece la portabilidad del sistema a otros dispositivos manteniendo unas prestaciones aceptables.

Finalmente, se ha extendido el algoritmo de reconocimiento de palabras aisladas al caso de dígitos conectados, obteniendo unos resultados prometedores.

Es importante destacar que el desarrollo de una herramienta de reconocimiento de dígitos conectados supone un salto cualitativo respecto al reconocimiento de palabras aisladas, y constituye un paso previo al reconocimiento de habla continua que es el futuro inmediato de este trabajo.

Referencias

[1] Keren Kesselman, Ana Isabel García Moral, Álvaro Pruneda Puertas, Ascensión Gallardo Antolín, Luis del Ser Toral y Fernando Díaz de María. "*Diseño de una interfaz vocal para la consulta de cotizaciones de valores del IBEX-35 desde una PDA 3ía GPRS*". Telecom. I+D 2004. Madrid, España.

[2] Goldstein, H., "*They Might Be Giants*", IEEE Spectrum, pp. 43-48, September, 2002.

[3] Gong Y. and Kao, Y.-H., "*Implementing a high accuracy speaker-independent continuous speech recognizer on a fixed-point DSP*", Proceedings of ICASSP, vol. 6, pp. 3686–3689, 2000.

[4] Deligne, S., Dharanipragada, S., Gopinath, R., Maison, B., Olsen, P. and Printz, H., "*A robust high accuracy speech recognition system for mobile applications*", IEEE Trans. on Speech and Audio Processing, Vol. 10, No. 8, pp. 551-561, Nov. 2002.

[5] Lam, Y.M. , Mak, M.W. and Leong, P.H.W., "*Fixed-point implementation of speech recognition systems*", Proc. of Int. Signal Processing Conf., Texas, USA, 2003.

[6] Lévy, C., Linarès, G:, Nocera P., Bonastre, J.-F., "*Reducing computational and memory cost for cellular phone embedded speech recognition System*", in Proceedings of ICASSP, vol. V, pp. 309-312, 2004.

[7] Varga, I., Aalburg, S., Andrassy, B., Astrov, S., Bauer, J.G. , Beaugeant, C., Geibler, C. and Höge, H., "*ASR in mobiles phones – an industrial approach*", IEEE Trans. on Speech and Audio Processing, Vol. 10, No. 8, pp. 562-569, Nov. 2002.

[8] Wang, D., Zhang, L., Liu, J. and Liu, R., "*Embedded speech recognition system on 8-bit MCU core*", ", in Proceedings of ICASSP, vol. V, pp. 301-304, 2004.

[9] ETSI Standard: ETSI ES 201 108 v1.1.2, "*Speech processing, transmission, and quality aspects (STQ); Distributed speech recognition; Front-end feature extraction algorithm; Compression algorithms*", 2000.

[10] S. J. Young, N. H. Russell, J. H. S. Thornton, "*Token passing: a simple conceptual model for connected speech recognition systems*". Cambridge University Engineering Department. 1989.

[11] Legetter, C. J. and Woodland, P. C., "*Maximum likelihood linear regression for speaker adaptation of continuous density Hidden Markov Models*", Computer Speech and Language, vol. 9, pp. 171-185, 1995.

[12] Young, S. et al, *"HTK-Hidden Markov Models toolkit (ver. 2.1)"*, Cambridge University, 1995.

[13] Crenshaw, J. W., "*MATH toolkit for REAL-TIME programming*", CMP Books, Lawrence, Kansas 2000.

[14] Motorola Application Note, "*Complex fixed-point Fast Fourier Transform optimization for AltiVec*".

[15] ELDA: "http://www.elda.fr"