# On Portability of Automatic Speech Recognition: A Study Case

*Juan L. García Zapata*[*], *Juan C. Díaz Martín*[**], *José M. Rodríguez García*[**], *Pedro Gómez Vilda* [***]

[*]Departamento de Matemáticas, [**]Departamento de Informática

Universidad de Extremadura. Spain

{jgzapata, juancarl, jmrodri}@unex.es

[***]Departamento de Arquitectura y Tecnología de Sistemas Informáticos,

Universidad de Politécnica de Madrid. Spain

pedro@pino.datsi.fi.upm.es

**Abstract**

The current methodologies of Automatic Speech Recognition are not tied to specific platforms. This can help to the development of voice-driving applications using speech-processing libraries. In this line, our research group has rebuilt the code of IVORY, a methodology of isolated word speech recognition, as functions without dependencies, then able of concurrent execution. As a by-product, the resulting code is as portable as to suit in a generic system with minimal audio processing abilities. This paper shows this portability by mean of the development of a speech recognition application over a DSP board, coupling the ASR functions and an audio application, with lesser modifications.

## 1. Motivations: Portability and Modularity.

The digital signal processing algorithms, of high computing requirements, have traditionally implemented carefully suited to the available resources. Now, the growing demand of speech processing applications needs of an implementation of algorithms easy to reuse and fitting on several platforms on increasing power. Besides the portability, another technique that can help in algorithm integration is the modularity. A modular setup of the speech processing tasks allows building further procedures by rearrange or replacement of blocks.

With these goals in mind, the IVORY methodology of Isolated Word Speech Recognition in Noisy Environments can be described from a coarse point of view by this block diagram: [1] [2]

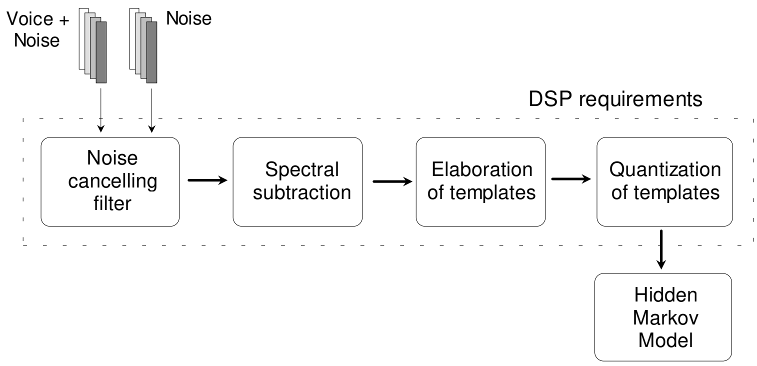


**Figure 1.** Pipeline distribution of IVORY tasks

*This work was founded by CICYT and Junta de Extremadura under the TIC99-0609 (DIARCA) and CICYTEX IPRR 98A039 grants.*

The voice processing algorithms have been divided in those main tasks, mutually independent and without internal state. The purpose of this distribution, in a pipeline parallelism, is the concurrent execution in a multiprocessor system[3]. In the paralelization of the methodology, several adjusts were taken to reach the disacoupling between stages, but the recognition efficiency are not affected.

With the tasks described in Figure 1 implemented in a dependence-free way, it can be possible to reuse such functions with little or no changes. For example, by this method, our group has developed ASR applications over PC (mono-threaded and then multithreaded), over linux cluster, with multi-user service [5] and in a DSP board.

In those heterogeneous platforms, ranging from clusters to DSP, the framework (the application that manages sound devices, user sessions, etc) is very different from one to other, but the recognition functions (the algorithmic) are essentially the same. In the next section, we summarize the porting of the algorithms to a DSP board.

## 2. Inside the building blocks

In this section is described the functions that implemented the IVORY methodology of isolated words speech recognition, with this aim of portability and ease algorithm integration. The target is TMS320C6711 DSP Starter Kit (DSK), a Texas Instruments C6711-based DSP board hardware module, parallel-port connected to a PC host[7]. In the development of the DSK recognition application, the code AUDIO acts as framework, and the recognition generic functions are inserted on it. This example code is distributed with Code Composer Studio (CCS), the programming environment of the DSK board [8].

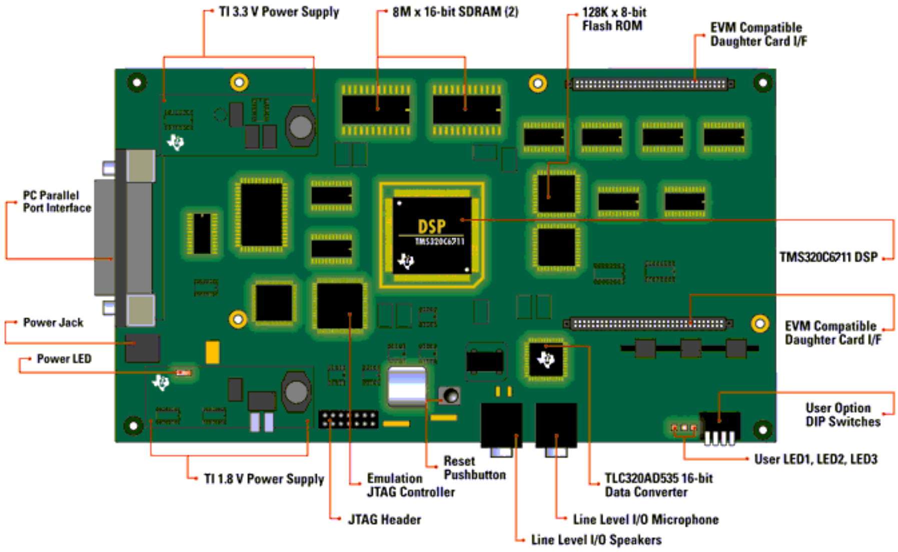


**Figure 2.** TMS320C6711 board

In an early stage of processing, the speech signal is segmented in 256-length windows. The buffers of AUDIO application are changed to fit this length requirement. The noise cancellation stage is an important characteristic of IVORY, but as the input channel of the board is mono, this stage cannot be ported to this system. The spectral extraction function applies a preemphasis filter and a windowed discrete transformation [1].

Then, according IVORY, it is built the template from each window. This template is the concatenation of a cepstral transformation and a delta-cepstrum, that is, a promediated difference of cepstrums corresponding to other near windows. This induces a temporal dependency and internal states. To allow the concurrent execution, that dependency should be managed in the main function audio, and not in the template elaboration function [9]. The concurrent code is not necessary in this application, but is needed in multi-threaded developments.

Each template is then codified with a vector quantization method, finding the nearest centroid in a codebook. The construction of this codebook for our speech database is described in the next section. This function is based on the computation of Euclidean distance, and with the above functions, shows high DSP requirements.

An energy criterion is used in the end point detection. At this point, a parser function compares the attained centroids sequence with the statistics models (Hidden Markov models) of the vocabulary words [10]. These statistics models are estimated from a database collected by the IVORY project, with 26 male and 26 female speakers, and several realizations of 30 words in Spanish language. The tools to build the models are described in the next section.

Despite the portability of the algorithms, it is required several adjusts to fit it in a system as DSK, with non-homogeneous memory model. Hand tuned options to CCS linker provides fast memory allocations to certain tables and code, in order to speed increasing; but no algorithm modification is needed.

## 3. Building the codebook and models

The essential speech recognition algorithms are implemented in relatively portable functions, but it is necessary to build the hidden Markov model of each vocabulary word. These statistical models embodies additional information about specific system features, as microphone adjust and quality [10]. They must be built with a functional prototype of the application, but still unable on recognition because of the model lack.

The source to build such models is the realization database. The templates obtained from this database are a huge set of multidimensional vectors. With a recursive technique of clustering, the distribution regions of these vectors are coded in 256 centroids. The vector quantization function described above operated with these centroids. The Markov models are based on this centroid-coded representation, and then depend on the codebook.

These tasks (to find the centroids from the templates and to find the models) can be made with generic tools (like HTK), but there are at our disposal specific ones, suited to the IVORY methodology, developed at DATSI. The codebook and models rebuilding tools are used in compile time, and they do not need reengineering to system integration.

In the DSK environment, to find the templates by the direct method of processing the database words and storing its templates on a file, is not feasible. Each word to process must be played by the sound card and reach the board via line level I/O microphone. Besides, the DSK board cannot write on PC host hard disk without interrupt the real-time execution of the algorithms. To contend with this obstacle, RTDX is used.

RTDX [7] is a Real-Time Data eXchange capability of the CCS IDE. The template obtained is stored in a board buffer, and transmitted to the host in the background, without conflict with the calculation of the next template. To use RTDX it is necessary to implement a COM (Microsoft's Component Object Model) client to be executed in PC host, which receives the data and stores it.
With the COM client built, for each realization in the database three tasks are launched: the playing of the word by the PC sound system, the recognition prototype over DSK, and the COM client. To implement this launching for the 1560 realizations available, a GEL (General Extension Language) file was setting. The GEL is a batch style language to control several switches of the CCS.

After the execution of the GEL file and the processing of the achieved templates by ISDE tool, the codebook and models must be linked to the application. The recognition system is then plainly functional.

## 4. Further applications of portability

An application of this algorithm modularity is the development of a speech-processing server. In the context of the DIARCA project, the computing engine is a distributed system of DSP, which serves several remote sessions of real time noise cancellation and voice processing[4].

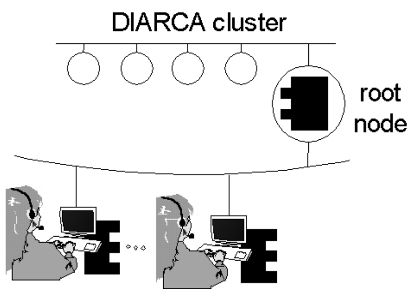


**Figure 3.** DIARCA DSP cluster as a distributed service of voice processing

## 5. Work in progress

Hardware providers, to shorten the development cycle of DSP-engineered applications, have promoted algorithm-encapsulated methods of programming. In this line, Texas Instruments, in its eXpressDSP technology initiative, has a subprogram of TMS320 DSP Algorithm Interoperability Standard (XDAIS)[6]. It is a set of programming rules and interface specifications intended to ease the system development.

Despite the portability of the algorithms described, they are not in eXpressDSP compliant format. It must be submitted to the interface specification and others rules of the standard.

The rebuilding of the codebook and models is system depending. It can be possible to obtain models reusable on other systems and/or word set, with standardization and normalization of cluster analysis algorithms[11].

## 6. References


[1] Gómez, P., Álvarez,A, Martínez, R., Pérez-Castellanos, V., Rodellar, V., Nieto, V., "A DSP-based Modular Architecture for Noise Cancellation and Speech Recognition", Proc. of the 1998 IEEE Int. Symp. on Circuits and Systems, ISCAS'98, Monterey, CA, USA, pp. V.178-181.

[2] http://dis.sema.es/projects/ivory

[3] Díaz Martín, J.C., Rodríguez García, J.M., García Zapata J.L., Gómez Vilda, P., "Robust Voice Recognition as a Distributed Service", Proc. of 8th IEEE Int. Conf. on Emerging Technologies and Factory Automation, EFTA 2001, Antibes, France, V2 pp 571-575

[4] Diaz Martin, JC. Garcia Zapata, JL. Rodriguez Garcia, JM. Alvarez Salgado, JF. Espada Bueno, P. Gómez Vilda, P.," DIARCA: A Component Approach to Voice Recognition", Proc. of 7th European Conference on Speech Communication and Technology, Eurospeech 2001, Aalborg, Denmark, 2001 V4 pp 2393-2396

[5] Diaz Martin, JC. Garcia Zapata, JL. Rodriguez Garcia, JM. Alvarez Salgado, JF. Espada Bueno, P. Gómez Vilda, P.,"Scalable Distributed Voice Recognition via Cluster Computing", Actas del Simposio de Informática y Telecomunicaciones SIT2001, A Coruña, España, 2001 pp. 163-171

[6] http://www.ti.com/sc/expressdsp

[7] http://focus.ti.com/docs/tool/toolfolder.jhtml?PartNumber=TMDS320006711

[8] Dirksen, S., "An audio example using DSP/BIOS", Application Report SPRA598, Texas Instruments Inc.,1999

[9] Lester, B, "the Art of Parallel Programming", Prentice-Hall, New Jersey, 1993

[10] Rabiner, L., Juang, B.H., "Fundamentals of Speech Recognition", Prentice-Hall, New Jersey, 1999

[11] Milligan, G. "An Examination of the Effect of Six types of Error Perturbation on Fifteen clustering Algorithms", Psycometrica 45 , September 1980, pp 325-342