VOICE TRANSFORMATION ALGORITHMS WITH REAL TIME DSP RAPID PROTOTYPING TOOLS

Graziano Bertini, Federico Fontana, Diego Gonzalez, Lorenzo Grassi, Massimo Magrini

Fondazione Scuola San Giorgio - CNR, Isola di San Giorgio Maggiore 30124, Venezia
phone: +39 41 5207757 fax: +39 41 5208135, email: diego.gonzalez@cini.ve.cnr.it
web: www.cini.ve.cnr.it

ABSTRACT

The main goal of the work here described is the DSP implementation of innovative algorithms for real-time voice transformation. This work represents part of the procedure (developed in the framework of the RACINE-S European Project) conceived for reconstructing voice and dialogue in audio tracks of old and highly damaged film movies .
Besides the implementation of a set of methods, as LPC/ VC (Linear Prediction Coding/ Voice Conversion), innovative methods of improving of the quality of synthesized voice have been considered. This whole set of operations represents a particular implementation of the so-called "Virtual Dubbing" procedure. The basic steps of the complete project include: designing a method for high-quality voice transformation, a suitable algorithm in Matlab/Simulink and, finally, translating it into Digital Signal Processor target code by means of a rapid prototyping approach.
The original code was developed in Matlab and so we used the Mathworks MATLAB's Real Time Workshop (RTW) DSP platform for rapid prototyping.

1. INTRODUCTION

Many different techniques have been developed for enhancing the quality of old and/or very deteriorated photographic images, films, audio and musical recordings.
Nevertheless there are still very active research areas involved with the development of special applications for solving non-standard problems, as in our studied case. The RACINE-S Project [1] involves both image (not described in this document) and synchronized audio restoration. In our case we are managing highly deteriorated or even missing parts of the audio score, as happens very often in very old movies. For this reason we are talking in this case about "reconstruction" rather than "restoration".
The Project involves different research/university units and industrial partners, each working on a different aspect: image/audio reconstruction and audio rendering improvement using intensimetric acoustic ambience reconstruction. In this document we will describe only the part of the work related to the voice reconstruction of the audio sequences. This work is developed under the responsibility of the Musical and Architectural Acoustics Lab. of FSSG-CNR of Venice(I).

2. PROJECT OVERVIEW

The starting point for virtual dubbing consists of digitized audio data arising from archive movie's sequences which are highly damaged or even partially missing. In this last case information are taken from segments positioned before and after the missing pieces of film. This audio data, which are supposed to be available as standard audio file, contains the relevant information about the target voice which needs to be reconstructed.

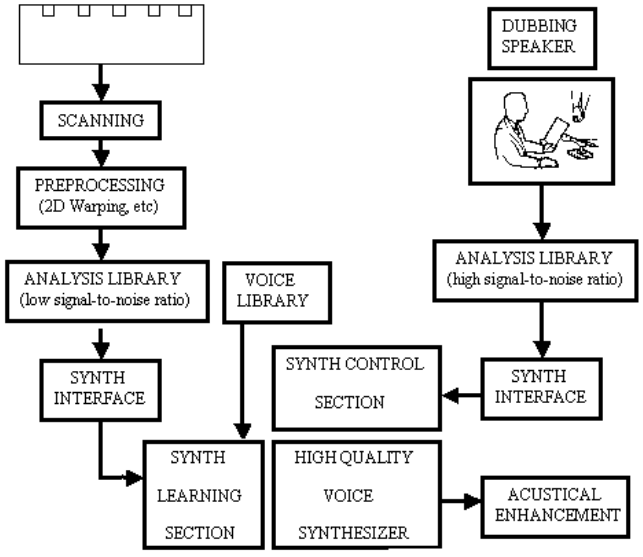

Fig. 1 Virtual dubbing block diagram

In a traditional dubbing studio a professional dubbing speaker dubs the sentences spelled by the target voice, if available, or simply reads the content of the script, if the corresponding sequence is missing. The main idea at this point is to provide the voice of the professional dubbing speaker with the features of the target voice (fig.1). This voice conversion process, described later, can be done in real time at the dubbing process or off-line at an audio post-production level. We tried to experiment the real time implementation using a dedicated DSP platform. In this document we will describe how this idea has been implemented, using a rapid prototyping approach.
For this task Mathwork's Real Time Workshop (RTW) has been used. Using RTW the previously simulated Simulink

models have been automatically translated into DSP source code, and then compiled and loaded on the DSP board.
For an easier use of the DSP starter kit board, a special enclosing hardware with additional analog circuitry has been built. This embedding hardware system provides input/output interfacing (analog and digital), power supply, signal conditioning (low noise preamplifier) and a user panel with state monitoring and reset buttons. The enclosing case also shields from electromagnetic interferences, improving the system reliability. A more detailed description can be found on [2].

3. LPC-VC VOICE ANALYSIS AND RE-SYNTHESIS

As said before our project requires the conversion of a source speaker's voice into another, as if it were pronounced by a different (target) speaker.
Today's available voice conversion algorithms mainly rely on residual-excited LPC synthesis [3-4-5] or on STFT-based synthesis [6-7].

In both cases the core of the algorithm is the definition of a map F(x) which transforms a feature vector x onto a new vector y. In the former case the feature vector is typically a set of LPC coefficients or, due to their better interpolation properties, a set of line spectral pairs coefficients (LSP) [3]. An example of such a voice conversion algorithm is available in the FESTIVAL TTS system within the OGIresLPC synthesis plug-in. In the latter case the feature vector can be some sort of spectrum magnitude representation, such as the FFT magnitude or the mel-cepstral coefficients [6].

A simple definition of the map F can rely on some parametric non-linear function approximation tools, e.g., neural networks or radial basis function networks (RBFN) [7]. Another common approach to the definition of the map structure is to use a probabilistic, locally linear function. A widely used technique makes use of Gaussian Mixture Models (GMM), which are capable to embed in the mapping function an acoustic model of the source speaker based on a Gaussian mixture [3,6].

The design of the conversion function requires a number of minimal steps. First, a database of sentences uttered by different speakers must be generated. Then, the feature vectors are computed by a frame-based LPC or sinusoidal analysis. Before the training step a dynamic time warping procedure (DTW) is required to time-align the feature vectors derived from the first speaker with the ones accounting for the second speaker. This guarantees that a input-output training is worked out, in which a phoneme of the first speaker corresponds to the same phoneme of the second speaker. At this stage, a conversion function based on a RBFN can be trained directly by identifying the parameters given by the input-output training pairs.

3.1 Voice conversion model

In our project we use a gaussian mixture voice conversion model in which two training sets of LPC coefficients extracted by the time window series and containing the source and the target (voice) signal, respectively, are used to form the conversion function (fig.2). This function is then used to map source into target LPC coefficient sets.

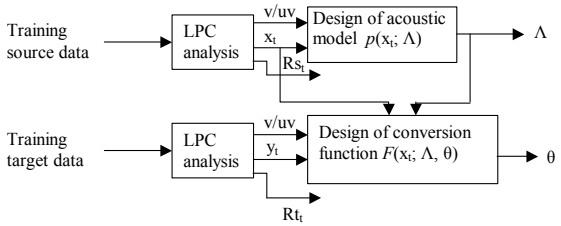

Fig.2 Voice conversion engine

.

Our LPC-VC model synthesizes the target voice by filtering the LPC residual of the source voice using target LPC coefficients obtained by converting the correspondent source LPC parameters by means of F. A careful design of the LPC-VC model would require to segregate the voice signal in "voiced" and "unvoiced" segments, each contained in a separate time window. At this stage of the project a standard segmentation of the signal into small segments having equal time length is sufficient to test the LPC voice conversion.

When used to model the speech process, the components of the GMM represent different phonetic events. Let us suppose that a sequence of P-dimensional column vectors {[x\vec]t}, t=1,...,T, which represents the time-varying spectral envelope of a source signal, has been fitted by a GMM. Moreover, let assume that a sequence of P-dimensional column vectors {[y\vec]t}, t=1,...,T, having the same length of the source signal, is the target of the conversion. We define a spectral conversion function as a map F to transform each vector in the input sequence into the vector which occupies the same position in the output sequence, thus preserving the time information of the input and output data. A common way to define a conversion function (1) is to use a parametric form for the spectral conversion function [6]:

$$\Im(\vec{x}_t) = \sum_{i=1}^{M} p(\lambda_i / \vec{x}_t)\left[\vec{\theta}_i + \Gamma_i \Sigma_i^{-1}(\vec{x}_t - \vec{\mu}_i)\right] \quad (1)$$

This conversion equation is equivalent to the solution of the following set of equations:

$$\vec{y}_t = \sum_{i=1}^{M} p(\lambda_i / \vec{x}_t)\left[\vec{\theta}_i + \Gamma_i \Sigma_i^{-1}(\vec{x}_t - \vec{\mu}_i)\right] \quad (2)$$

If we omit here the term $\Gamma_i\Sigma_i^{-1}([x\vec]_t - [(\mu)\vec]_i)$ in (2) we can use the following reduced form of the conversion function:

$$\Im(\vec{x}_t) = \sum_{i=1}^{M} p(\lambda_i / \vec{x}_t)\left[\vec{\theta}_i\right] \quad (3)$$

4. ALGORITHM IMPLEMENTATION

Following the Rapid prototyping approach we implemented our algorithms using Simulink, which provides a graphical user interface (GUI) for building models as block diagrams, using click-and-drag mouse operations.
Once a model has been defined, it is possible to simulate it. Using scopes and other display blocks, the simulation results can be viewed while the simulation is running. The simulation results can be put in the MATLAB workspace for post processing and visualization.

4.1 Simulink Implementation

The voice signal is first divided in frames having a strong overlapping factor in order to prevent from the birth of major artefacts occurring during the transition from one frame to another. For our specific purpose, here we have chosen a frame size of 1024 samples and a hop-size of 64 samples, along with a Blackman windowing that provides smooth transitions along adjacent frames. These figures seem to be widespread enough to provide sufficiently accurate results in the majority of the test cases. Then, each frame is LPC-encoded. An LPC order equal to 10 is considered general-purpose enough in most literature. For LPC analysis a classic extraction algorithm is employed, that performs an autocorrelation of the input signal followed by a Levinson-Durbin LPC extraction algorithm.
As previously said, the design and the training of the VC model are left off line, with a number of Gaussian Mixture Models (GMM's) set to 10. The design phase first needs the target voice LPC coefficients, that are treated the same way as before. These coefficients, transformed into line spectral pairs, are then used for the GMM modelling procedure, obtaining a set of transformation parameters (M, V, W, TH0), that will be used for the re-synthesis process.
The core steps of the re-synthesis are:

- *lpcar2ls* converts LPC coefficients into line spectral pairs. This requires a roots computation and a deconvolution.
- *lpcN_fun* and *lpcpi_fun* compute parameters for calculating the set of line spectral pairs (lsp) encoding the transformed voice according to formula (3).
- *lpcls2ar* converts the line spectral pairs back to transformed LPC coefficients.

The reported algorithms represent an important improvement over usual LPC techniques because through the voice transformation algorithms a detailed description of the spectral properties of the target and source signals is taken into account [8].

4.2 Real-time implementation

Because the complete implementation of the method involves a high algorithmic complexity, for practical applications is necessary to test the possibility to implement it in real-time.
As these basic algorithms are being developed at a research level, such a test need to be performed in a flexible platform which allows the implementation of the non-optimized algorithms with a reasonable effort. For this reason has been choice an implementation on a DSP hardware platform of Texas Instruments mounting a floating point, 32 bits processor: the TMS320C6711. This DSP is based on the VLIW (Very large Instruction word) technology which allows, using its optimized compiler, fast parallel computing.
For a fast evaluation of the TMS320C6711 processor a Developer Started Kit (DSK) is available from Texas Instruments This board have be connected to a standard PC running its development environment, Code Composer Studio.
On board there are 16 MBytes of SDRAM installed, plus 128 Kb of external Flash. Along other features, the most relevant one, for us, is the presence of an audio codec (AD/DA converter), the TLC320AD535. Even if it's quality is rather poor (16 bit, 8 kHz sampling rate), it is sufficient to test DSP applications in the speech audio band. Anyway the board can mount additional daughter boards, as better performance A/D converters: for example the PCM3003 daughter board, which allows 48 kHz sampling rate (16 bit stereo).
Mathworks' Real-Time Workshop builds applications from Simulink diagrams for prototyping, testing, and deploying real-time systems on a variety of target computing platforms, including Texas Instruments C6000 class DSP processors.
Once the Simulink model has been simulated it is possible to do some settings in model configurations in order to instruct Matlab to start the DSP code building process. The resulting translation will transform the Simulink model into a Code Composer Studio C language project and then, controlling the DSP compiler, into a DSP binary code, which can be finally downloaded on the DSK board. Before the translation, the model has been tuned using Mathwork's Model Advisor, a software tool that comprehensively analyzes the Simulink model to help the programmer to appropriately configure Simulink and Real-Time Workshop.

5. PRELIMINARY RESULTS AND CONCLUSIONS

The complete model has been simplified in a reduced model in order to develop in parallel the DSP application and to test the system. This reduced model lacks the GMM based conversion part and requires, as input, a source and a target voice that have to be synchronized. For the test run we have used a short pre-recorded sequence of vowels that is upload in the DSP hardware together with the code of the model. The output of the model is connected with the DAC line out.

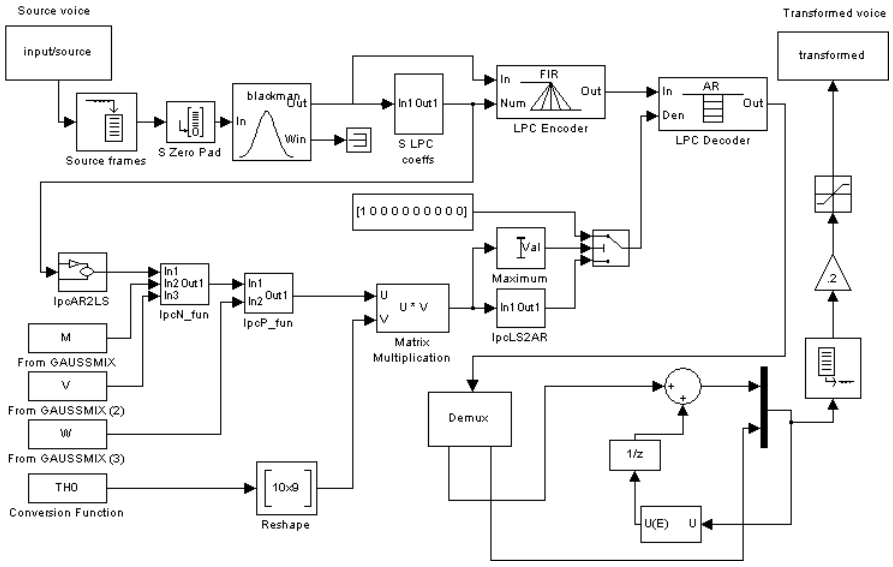

Fig.3 Complete Voice Conversion Algorithm

The reduced model has been successfully converted and loaded into the DSK boards. The signal at the DSK DAC output exactly reflects the simulation output.

The main goal of the project described has been the development of instruments and methods for reconstructing audio sequences starting from old, highly damaged, movie films. A solution based on LPC/VC algorithms plus GMM transformation has been proposed. The chosen solution provides successful approach to the physical modelling of the vocal tract, and physical meaningfulness and directivity of the control parameters. The quality of the resynthesized voice can be further improved using non-linear techniques [10]. This possibility is being investigated through interfacing of an efficient non-linear pitch detection model with the core of LPC voice conversion algorithms. The dynamical model allows for an efficient emulation of the perceptive function with a reduced number of parameters.

These algorithms have been firstly tested by Mathwork's Simulink environment. High quality voice reconstruction tests has been made with good results. According to one of the project's goal, we tested the possibility of a real-time implementation using a modern DSP platform provided by Rapid Prototyping tools. Considering the high complexity of the whole algorithm, together with some limits of the HW/SW development platform used, the real time implementation has been carried out only in a simplified form.

An interesting result of becoming familiar with rapid prototyping approach has been achieved: we found the used work flow very powerful and attractive for further investigations and improvements.

This work has been partially financed by the RACINE-S Project, contract number: IST 2001 -37117.

6. REFERENCES

[1] RACINE-S EU Project IST 2001 -37117.

[2] Bertini G., Di Giovannantonio L., Gonzales D., Grassi L., Magrini M., Dedola M. "*Sistema di sviluppo con DSP floating-point per il rapid prototyping*". Nota interna ISTI-CNR, B4-14 Dec. 2004

[3] A. Kain and M. W. Macon, "*Spectral voice conversion for text-to-speech synthesis*" Proceedings of International Conference on Acoustics, Speech, and Signal Processing, vol. 1, pp. 285-288, 1998.

[4] A. Kain and M. Macon, "*Personalizing a speech synthesizer by voice adaptation*" in Proc. of the 3rd ESCA/COCOSDA Int. Workshop in Speech Synthesis, pp. 225-230, 1998.

[5] A. Kain and M. W. Macon, "*Design and evaluation of a voice conversion algorithm based on spectral envelope mapping and residual prediction*" Proceedings of International Conference on Acoustics, Speech, and Signal Processing, 2001.

[6] Y. Stylianou, O. Cappé, and E. Moulines, "*Continuous probabilistic transform for voice conversion*" IEEE Transactions on Speech and Audio Processing, vol. 6, no. 2, pp. 131-142, March 1998.

[7] N. Ivahashi and Y. Sagisaka, "*Speech spectrum conversion based on speaker interpolation and multi-functional representation with weighting by radial basis function networks*" Speech Commun., vol. 16, pp. 139-151, 1995.

[8] D.Bonsi, C.Drioli, F.Fontana, D.L.Gonzalez and D. Stanzial,Deliverable 3.2.1 RACINE-S, "*Algorithms for dynamic control of sound synthesis and implementation of acoustic ambience extraction from tracks*" (2003).

[9] J. Cartwright, D.L. Gonzalez and O. Piro, "*Nonlinear dynamics of the perceived pitch of complex sounds*" Physical Review Letters, vol. 82, n. 26, pp 5389-5392 (1999).

[10] J.Cartwright, D.L. Gonzalez and O. Piro, "*Pitch perception: A dynamical-systems perspective*" PNAS, vol. 98, n.9, pp. 4855-4859 (2001).