Advances in Very Low Bit Rate Speech Coding using Recognition and Synthesis Techniques * **

Geneviève Baudoin[5], François Capman[2], Jan Černocký[3], Fadi El Chami[6,5], Maurice Charbit[4], Gérard Chollet[4], and Dijana Petrovska-Delacrétaz[1]

[1] University of Fribourg, Dpt. of Informatics, dijana.petrovski@unifr.ch
[2] Thales Communications, francois.capman@fr.thalesgroup.com
[3] VUT Brno, Faculty of Information Technology, cernocky@fit.vutbr.cz
[4] ENST Paris, Dpt. Signal et Images, {charbit,chollet}@tsi.enst.fr
[5] ESIEE Paris, Dpt. Signal et Télécommunications, baudoing@esiee.fr
[6] Université Libanaise Tripoli, chamifadi@hotmail.com

Abstract. ALISP (Automatic Language Independent Speech Processing) units are an alternative concept to using phoneme-derived units in speech processing. This article describes advances in very low bit rate coding using ALISP units. Results of speaker-independent experiments are reported and speaker clustering using vector quantization is proposed. The improvements of speech re-synthesis using Harmonic Noise Model and dynamic selection of units are discussed.

1 Introduction

In order to achieve bit rates lower than 600 bps in speech coding, it is necessary to use recognition and synthesis techniques. By transmitting only the indexes of the recognized unit, the transmission bit rate is drastically reduced. The coder and the decoder share a dictionary of speech segments. At the decoder side speech synthesis is used in order to reconstruct the output speech, from the sequence of the transmitted symbols. The quality of the reconstructed speech data is also dependent on the choice of the synthesis method.

Current coders working at bit rates lower than 600 bps are based on segmental units related to phones (the phones being the physical realization of the corresponding phonemes). An alternative approach using automatically derived speech units based on ALISP tools was developed at ENST, ESIEE and VUT Brno [1]. These speech units are derived from a statistical analysis of a speech corpus, requiring neither phonetics nor orthographic transcriptions of the speech data. However the experiments conducted so far have only involved the speaker dependent case. In this paper we extend this technique to the speaker independent case (section 3) and present results with VQ based speaker clustering (sec. 4). We also suggest improvements using HNM synthesis and dynamic selection of synthesis units (Sections 5 and 4).

* The research has been supported by French RNRT project SYMPATEX and by Grant Agency of Czech Republic under project No. 102/02/0124.

** The authors appear in alphabetical order.

2 Principles of VLBR speech coding using ALISP units

To obtain speech units from a large speech corpus, first an *initialization* is performed. Spectrally stable zones in speech are found by temporal decomposition [2] and clustered to classes using vector quantization. This leads to initial phoneme-like transcription of the data. In the second phase, *model training*, HMMs are trained for all units. Iterations of Viterbi recognition of the training database and model re-estimations were found to be beneficial for the quality of units. A set of units, their models and transcriptions of the training data are the result of this step. The units are denoted *coding units*.

For the *synthesis* in the decoder, another type of units called *synthesis units* can be defined (see Section 6). Finally, the decoder must dispose of a certain number of *representatives* of each synthesis unit. When dealing with speech examples coming from multiple speakers, we are preserving the information about the identity of those speakers (in such a way we can select the same number of representatives per speaker). The coder must send the index of best-matching representative and information on the prosody: timing and pitch and energy contours. The decoder receives the information on coding units and derives the information on synthesis units, then it retrieves the representative from its memory. The synthesis modifies the prosody of the representative and produces output speech.

This approach was tested on American English [1], French [10] and Czech [11]. Intelligible speech was obtained for the three languages – low speech quality was attributed mainly to rudimentary LPC synthesis rather than the units themselves. The bit rate obtained is in the range of 100–200 bits/s for units encoding (without prosody information).

3 Speaker independent coding

In this section we address the issue of extending a speaker dependent very low bit-rate coder to a speaker independent situation based on automatically derived speech units with ALISP.

For the experiments we used the BREF database [6], a large vocabulary read-speech corpus for French. The BREF database is sampled at 16 kHz. The texts were selected from 5 million words of the French newspaper “Le monde”. In total 11,000 texts were chosen, in order to maximize the number of distinct triphones. Separate text materials were selected for training and test corpora. 120 speakers have been recorded, each providing between 5,000 and 10,000 words (approximately 40-70 min of speech), from different French dialects. Different subsets of the database were used for different experiments.

As a first step a gender dependent, speaker independent coder is experimented. For the speaker independent experiments, we have taken 33 male speakers to train the ALISP recognizer. Testing was done with another set of 3 male speakers. For a baseline comparison, we generated the equivalent speaker dependent experiments. Their speech data was divided into a training set for the speaker dependent ALISP recognizer and a common set for the test coding sentences.

The speech parameterization was done by classical Linear Predictive Coding (LPC) cepstral analysis. The Linear Prediction Cepstral Coefficients (LPCC) are calculated every 10 ms, on a 20 ms window. The temporal decomposition was set up to produce 16 events per second on the average. A codebook with 64 centroids is trained on the vectors from the gravity centers of the interpolation functions, while the segmentation was performed using cumulated distances on the entire segments. With the speech segments clustered in each class, we trained a corresponding HMM model with three states through 5 successive re-estimation steps. The 8 longest segments per model were chosen from the training corpus to build the set of the synthesis units, denoted as synthesis representatives present in the dictionary. The original pitch and energy contours, as well as the optimal DTW time-warps between the original segment and the coded one were used. The index of the best matching DTW representative is also transmitted. The unit rate is evaluated assuming uniform encoding of the indexes. The encoding of the representatives increases the rate by 3 additional bits per unit.

A conventional LPC synthesizer was used in the decoder. This synthesis method in known to be responsible to a lot of artifacts and unnatural sounds of the output speech. For a test segment, a comparison of the wide-band spectrograms of an original and synthesized speech, shows that the synthesis by itself introduces a lot of degradation. The same test segment was used to evaluate the transition from the speaker dependent to the speaker independent case. The corresponding spectrograms are shown in Fig. 1.

The resulting average rates for the spectral information are 140 bps for the speaker dependent case and 133 bps for the speaker independent case. Through informal listening we can conclude that the coded speech in the speaker independent mode is still intelligible. Not surprisingly, the speech quality was found to be worse in the speaker independent experiments.

4 VQ-based speaker clustering and adaptation

Several distinct approaches are possible for handling the speaker-independent mode. One could think of training the VLBR system using a sufficient amount of representative speakers, making no distinction between the different speakers as described in the previous section. But it could advantageously be combined with a pre-clustering of reference speakers, in order to select the closest speaker or the closest subset of speakers for HMM refinements and/or adaptation of synthesis units. In order to investigate further this idea, we have defined a VQ-based inter-speaker distance using the unsupervised hierarchical VQ algorithm [4]. The basic assumption is that training speech material from the processed speaker is available during a short training phase for running the VQ adaptation process. The inter-speaker distance is defined as the cumulated distance between centroids of the non-aligned code-books, using the correspondence resulting from the aligned code-books obtained through the adaptation process. This distance

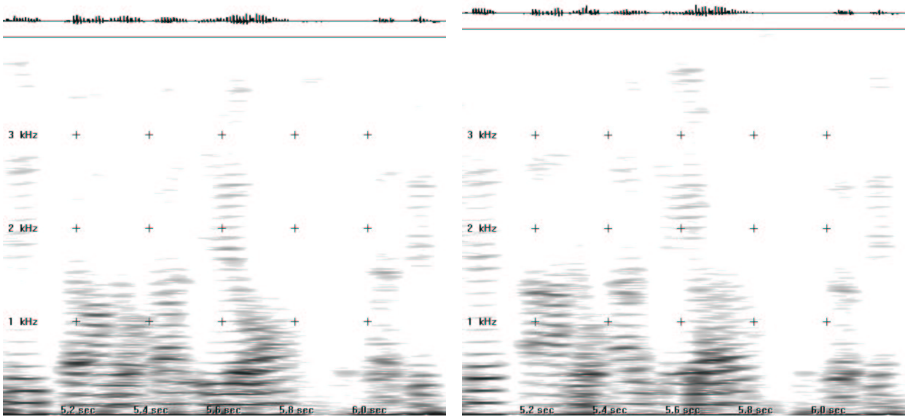

Fig. 1. Speaker dependent and speaker independent coding.

is used in the off-line pre-training phase for clustering the reference speakers, and during the on-line training phase for selecting the closest cluster to the user. From the distance matrix, sub-classes are extracted using a simplified split-based clustering method.

The proposed concept has been validated on the BREF corpus using phonetically balanced sentences. The resulting inter-speaker distance matrix illustrated in Fig. 2, was derived using 16 LPC-based cepstrum features, and 64 classes. Intra-speakers distances are located on the diagonal, and should be minimal for each speaker. Illustration of the clustering process is given for the largest class, (left panel of 3), a typical class (middle panel) and an isolated speaker (right panel) in terms of relative distance to the other speakers. One could note the similar positioning of speakers belonging to the same cluster. This distance is expected to be robust to channel variation, and moderate background noise, since it is based on an adaptation process, which should take into account part of the corresponding mismatch. This will be validated in future experiments using noisy data and distorted channel.

The obtained results in terms of speaker clustering using a small amount of data are encouraging. In our future works, we will study a speaker-independent VLBR structure derived from this concept, by adding HMM adaptation at the encoder, and voice conversion techniques at the decoder.

5 Harmonic-Noise Model synthesis

On contrary to source-filter based approaches, HNM (Harmonic Noise Model) [7, 9] increases the quality of speech processing by representing the speech signal by $x_t = \sum_{k=1}^{P} a_k \cos(2\pi f_k t + \phi_k) + n_t$, where the sum of cosinusoids is the harmonic part and n_t stands for the noise part. The noise part represents irregularities, as for example those produced by glottal disturbances. For speech signal, harmonic

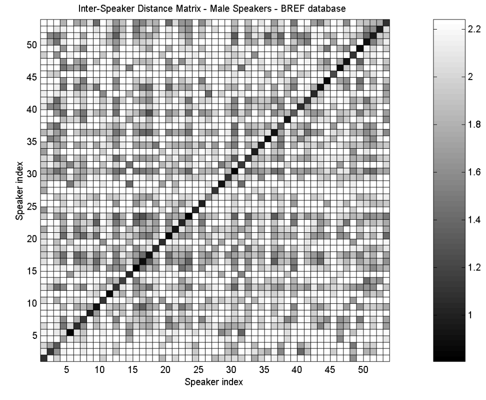

Fig. 2. Inter-speaker distance matrix for Male Speakers (54) from BREF corpus

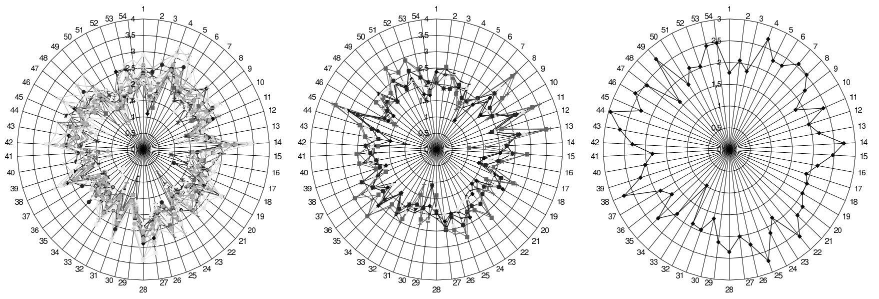

Fig. 3. Left panel: Relative distance of speakers from the largest cluster. Middle panel: Relative distance of speakers from a typical cluster (indexes 6, 14, 21, 29, 31, 43). Right panel: Relative distance of speakers from an isolated speaker (index 33).

components are mainly located below 4000 Hz while noise spectral part is located above 3000Hz. For unvoiced sound we keep the classical model, i.e. a white noise exciting an all-pole filter. For HNM, it is easy to modify speech timing. High quality speech synthesis is obtained by constraining phase continuity between successive windows.

Principal advantage of HNM is that gives a flexible method to modify voice characteristics as pitch or timbre. For this purpose, the most important features is the magnitude of the vocal tract transfer function, also called *spectral envelope.* Many works have been devoted to its estimation: The *LPC* approach, that considers that the source excitation is a white noise, performs poorly particularly for high-pitched voiced sounds. Some approaches use *interpolation techniques* between known harmonic values, but the problem has many degrees of freedom and the performed envelope may present non desirable high picks. We use an approach proposed by Galas and Rodet in [5] based on a *regularization technique* *with penalization.* The method is quite general and may be used with various parametrization as for example cepstral or pole envelope representations.

At first, pitch extraction is performed with sufficient accuracy to efficiently separate deterministic and noisy parts of HNM. Then we estimate spectral density η_k of noise with LPC approach. Because it is difficult to perform accurate estimation of harmonic components, particularly for high order harmonic, we restrict research of frequency harmonic to the band $0-4$ kHz. The estimation procedure proposed (described in detail by Eqs. (19)-(20) p. 473 in [3]) is based on iterative computation of the penalized least-square estimator of cepstrum coefficients, with a penalization (adjusted by a smoothing parameter λ) which controls the trade-off between observed data and regularizing function. In practice we observe that good spectral envelope estimation is obtained with very few iterations, typically less than 5.

In the context of speech synthesis, spectral envelope has been used as side information to modify speech segment of the dictionary. For example, spectral envelope yields the amplitudes of harmonic components for a given value of pitch and then, using the definition equation of HNM, we may construct the modified signal.

6 Synthesis with dynamic selection of units

The decoder is a speech synthesizer that operates using the information received from the coder that includes at least: ALISP coding units labels and prosody parameters. The speech synthesis is based on concatenation of synthesis units. Two types of synthesis units can be used:

Long synthesis speech units with spectrally stable extremities making concatenation easy. These long units can be constructed by aggregation of short ALISP coding units with re-segmentation in spectrally stable parts of the extremity units. The synthesizer is similar to a diphone one [8].

Short Synthesis speech units with dynamic selection of units, which is close to corpus based text speech synthesis and provided clearly better results than the first one. Here, for each ALISP class, a large number of representatives is extracted from the training corpus. These synthesis representatives are determined in order to fulfill criteria of good representation of a given segment to be coded and criteria of good concatenation of successive segments. A possible solution consists in defining representation and concatenation distance (D_R and D_C) and in choosing the representative to minimize a criterion of the form $aD_I + bD_C$. But, this solution requires to adjust the parameters a and b leading to heavy experimental trials. Therefore a different method was applied that does not requires the explicit calculation of a concatenation distance.

In the developed technique, after the training of ALISP coding units, each ALISP class is partitioned in sub-classes. Let's N_A be the number of ALISP coding classes (in practice, we used $N_A = 64$) and let's call H_j with $j \in [0, N_A-1]$ the ALISP coding classes. Each H_j class contains many segments of the training corpus that were recognized as generated by the H_j HMM model. Each H_j class is partitioned in N_A sub-classes called $H_i H_j$ containing all the speech segments of class H_j that were preceded by a segment belonging to the class H_i in the training corpus. It is possible to keep as synthesis representatives all the segments of the training corpus organized in classes and sub-classes as described above or to limit the size of each sub-class to a maximal value K. If the training corpus is not large enough some of the sub-classes may be empty.

In the coding phase, the coder after recognition of ALISP units, determines for each coding unit a representative for the synthesis. The coder transmits the indexes of ALISP coding class and of synthesis representative. In the decoding phase, the synthesizer concatenates the synthesis units corresponding to the chosen representative in each ALISP class.

During coding, if a segment is recognized as belonging to class H_j and is preceded by a segment in class H_i, the representative is searched in the subclass $H_i H_j$ of class H_j. The selection of the best representative in the sub-class is done on the distance D_C of good representation of the segment. The D_C distance is based on a spectral comparison by DTW between the segment to code and the potential synthesis representatives. The distance D_C can also include a distance on prosody parameters. The index of ALISP class is transmitted on 6 bits and the index of the representative on $\log_2(K)$ bits or $\log_2(Nmax)$ bits where N_{max} is the maximum number of segments in a sub-class. It is not necessary to transmit the index of the sub-class, since the decoder has the same information as the coder concerning the preceding unit. This approach gives very good results, but it requires a large memory size at the decoder for the codebook of synthesis representatives. If no limitation is done on the number of segments in a subclass, the complete training corpus must be present in the decoder. If the size is limited to $K = 16$ segments in each subclass (small quality degradation compared to no limitation), a maximum of 16*64*64 segments must be present in the decoder. If we suppose that the average segment length is 60 ms, this represents 1 hour of speech. When the number of representative segments is not limited the coder does an exhaustive search in the training corpus, but this is done efficiently (because of pre-classification by preceding segments the calculation is divided by 64).

Some of the subclasses $H_i H_j$ may be non represented in the training corpus, we developed an algorithm of substitution of the missing classes, using the fact that the ALISP classes have a numbering order that corresponds to an average spectral distance order between classes. Therefore, when a class $H_i H_j$ is missing, the algorithm searches if a class $H_{(i-1)} H_j$ or $H_{(i+1)} H_j$ exists. If not, it iterates the operation and when some class is found, it replaces the missing one.

7 Conclusions and Perspectives

This paper demonstrates that speech coding, at transmission rate lower than 400 bps, can be achieved with little degradation. In order to realize this, a speech memory is necessary at the coder and decoder sides. This memory should be identical on both sides. Only the indexes of speech segments (and some prosodic information) is transmitted between the coder and the decoder. The drawback of our proposal is the size of the memory required on both sides and the delay introduced by the maximal duration of the segments in memory (of the order of 200 msec). There are many applications which could tolerate both a large memory (let say 200 Mbytes) and the delay. Among such applications are the multimedia mobile terminal of the future (including the electronic book), the secured mobile phone, the compression of conferences (including distance education),... More work is necessary on voice transformation so that only typical voices will be kept in memory. This is an interesting topic to characterize a voice based on limited data and use this characterization to transform another voice. Applications in speaker recognition are being tested. The speech memory could also be labeled phonetically. In this manner our speech coder will be able to perform acoustic-phonetic decoding of speech, a major step toward speech recognition. In summary, we believe that speech coding by indexing is a useful step in most areas of Automatic Speech Processing.

References

1. G. Baudoin, J. Černocký, P. Gournay, and G. Chollet. Codage de la parole à bas et très bas débits. *Anales des Télécommunications*, 55(9–10):462–482, 2000.
2. F. Bimbot, G. Chollet, P. Deleglise, and C. Montacié. Temporal decomposition and acoustic-phonetic decoding of speech. In *Proc. IEEE ICASSP 88*, pages 445–448, New York, 1988.
3. M. Campedel-Oudot, O. Cappé, and E. Moulines. Spectral envelope estimation using a penalized likelihood criterion. IEEE, Trans. on Speech and Audio Proc.:469–481, July 2001.
4. S. Furui. Unsupervised speaker adaptation method based on hierarchical spectral clustering. In *Proc. ICASSP'89*, pages 286–289, 1989.
5. T. Galas and X. Rodet. Generalized functional approximation for sourcefilter system modeling. Proc. Eurospeech, (Genova):1085–1088, 1991.
6. L. F. Lamel, J. L. Gauvin, and M. Eskanazi. BREF: a large vocabulary spoken corpus for French. In *Proc. EUROSPEECH 1991*, Genova, Italy, 1991.
7. J. Laroche, E. Moulines, and Y. Stylianou. Speech modification based on harmonic + noise model. Proc. EUOROSPEECH. Madrid.:451–454, Sept. 1993.
8. P. Motlíček. Concepts of the dissertation. Technical report, Brno University of Technology, Inst. of Radioelectronics, April 2001.
9. Y. Stylianou, J. Laroche, and E. Moulines. High quality speech modification based on harmonic + noise model. Proc. IEEE, ICASSP. Minneapolis., Apr. 1995.
10. J. Černocký, G. Baudoin, and G. Chollet. Segmental vocoder - going beyond the phonetic approach. In *Proc. IEEE ICASSP 98*, pages 605–608, Seattle, WA, May 1998. http://www.fee.vutbr.cz/~cernocky/Icassp98.html.
11. J. Černocký, I. Kopeček, G. Baudoin, and G. Chollet. Very low bit rate speech coding: comparison of data-driven units with syllable segments. In V. Matoušek, P. Mautner, J. Ocelíková, and P. Sojka, editors, *Proc. of Workshop on Text Speech and Dialogue (TSD'99)*, number 1692 in Lecture notes in computer science, pages 262–267, Mariánské Lázně, Czech Republic, September 1999. Springer Verlag.