SEGMENTAL VOCODER – GOING BEYOND THE PHONETIC APPROACH

Jan Černocký [1,2], *Geneviève Baudoin* [2] *and Gérard Chollet* [3]

[1]Technical University of Brno, Institute of Radioelectronics, Brno, Czech Republic, cernocky@urel.fee.vutbr.cz
[2]ESIEE, Département Signal et Télécommunications, Noisy-le-Grand, France, {cernockj,baudoing}@esiee.fr
[3]ENST, Département Signal, Paris, France, chollet@sig.enst.fr

ABSTRACT

In our paper, the problem of very low bit rate segmental speech coding is addressed. The basic units are found automatically in the training database using temporal decomposition, vector quantization and multigrams. They are modelled by HMMs. The coding is based on recognition and synthesis. In single speaker tests, we obtained intelligible and naturally sounding speech at mean rate of 211.2 b/s. In the end, future extensions of our scheme (diphone-like synthesis and speaker adaptation) as well as possible use of automatically derived units in recognition are discussed.

1. INTRODUCTION

Among low bit rate speech coders [11], the very low rate region is populated mostly by *segmental* or *phonetic* vocoders. Only those schemes, based on recognition and synthesis, are able to use efficiently the limited number of bits and overcome the problems of standard frame-by-frame coding. One of main problems of segmental methods is the choice of basic units. Typically, *phonemes* are used [12, 7], which induces the need of a phonetically transcribed training database.

In our approach, those units are found *automatically*, so that only sufficient amount of raw training data is necessary. Therefore, this scheme is easily applicable in languages lacking standard speech databases and forms an important part of ALISP (Automatic Language Independent Speech Processing) tools [10]. With such a set of units, we have built a coder, consisting of recognizer (acoustically labelling the speech) and additional information encoder. In the decoder, a synthesis takes place, in our case concatenating examples from the training corpus, to obtain output speech.

The paper is organized as follows: section 2 presents a global view of our coder. Section 3 describes the search of basic units. Section 4 gives details on modelling and segmentation and section 5 discusses the additional information transmission and synthesis. Each of sections 3,4 and 5 is completed by description of experimental setup and partial results. The following section 6 comments the final results in terms of quality and bit rate. In section 7 we discuss future extensions of our scheme and possible application of this concept to speech recognition, and section 8 contains the conclusion.

This work is supported by the French Government scholarship No. 94/4516 and by the grant No. VS97060 of the Ministry of Education, Youth and Sports of the Czech Republic.

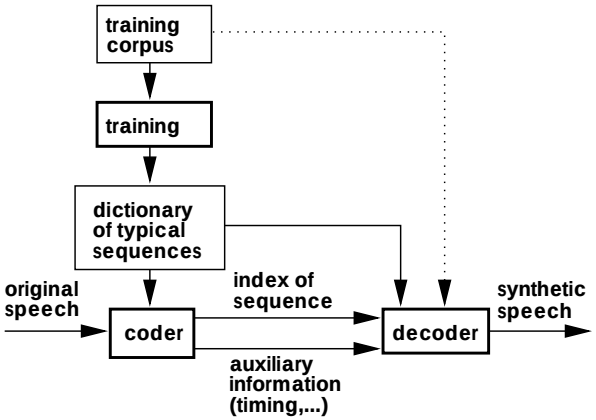


Figure 1: Scheme of the segmental coder.

2. SEGMENTAL CODING

The coding scheme is given on Fig. 1. The algorithm can be divided into five parts:

1. **Non-supervised search of characteristic segments.** On contrary to other phonetic vocoders [12, 7], the set of basic units is defined automatically. The temporal decomposition (TD), vector quantization (VQ) and multigram segmentation (MG) were used to find the initial set. It was further refined by hidden Markov model (HMM) training, and relabelling.
2. **Clustering and modelling of segments.** Standard left-right HMMs were used to model the sequences.
3. **Segment recognition.** The segmentation and segment recognition can be done using techniques known from continuous speech recognition. The index of recognized sequence is transmitted from the coder to the decoder. A simple "unigram" language model (LM) weighting the emission probabilities of HMMs was tested.
4. **Segment reconstruction.** To obtain synthetic speech on the decoder side, additional information must be transmitted. In this etap, a simple synthesis using examples from the training corpus was applied. Only the choice of example and energy correction is transmitted.
5. **Adaptation.** The resulting set of typical segments is strongly dependent on the training database. Several approaches can be considered to overcome the inter-speaker variability

(normalization of voices to a generic one, voice modification). The adaptation module has not yet been created and tested experimentally.

3. SEARCH OF TYPICAL SEGMENTS

First, the speech signal is divided into active and passive parts using a voice activity detector (VAD); only active parts are taken into account. Each part is parametrized on frame basis by a set of spectral coefficients, forming a $P \times n$ matrix Y, where P is the size of parameter vector and n the number of frames. This matrix is separated into limited amount of spectral *events*, each consisting of a *target* and an *interpolation function (IF)* using TD [1, 3] – the spectral parameters are approximated by a product of two matrices: $Y = G\Phi$, where G is a $P \times M$ target matrix and Φ is a $M \times n$ matrix of interpolation functions, concentrated in time. The number M of events is $M < N$. The method used for this decomposition is a short-time SVD with adaptive windowing, post-processing of interpolation functions (smoothing, decorrelation) and iterative refinement of G and Φ [3].

Then, the parameter vectors situated in gravity centers of IFs are quantized using simple VQ with low-size codebook in order to obtain a string of symbols. This string is used to determine a set of characteristic variable length symbol patterns called *multigrams*. The MG segmentation and dictionary training are based on decision oriented likelihood maximization:

$$L(W) = \max_{\{B\}} \prod_{k} p(S_k) \qquad (1)$$

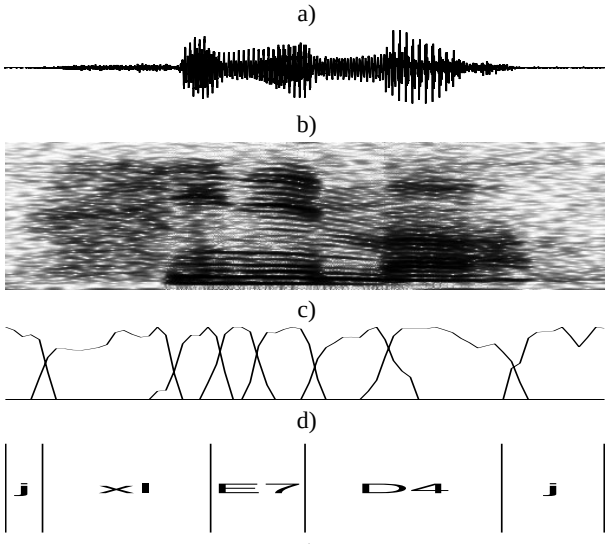
where $p(S_k)$ are the probabilities of symbol sequences and $\{B\}$ is the set of all possible segmentations. The process consists of initialization of dictionary using all occurences of $1-$ to $m-$symbol sequences, and of iterations of segmentation (Eq. 1) and probabilities reestimation. Although a MG-based method, where the symbols are replaced by unquantized vectors, has already been defined [2], the original method [4] was used, with two modifications:

- **forced segmentation** on the borders of active parts. Those are determined by VAD, and the resulting training string is created by their concatenation, so no multigram should cross their borders.
- **minimum occurence number** for multigram dictionary entries. This modification was done with respect to the following HMM training, where a minimum number of examples is needed.

The outputs of this procedure are a set of variable length characteristic sequences of quantized spectral events with associated probabilities, and a phonetic-like labelling of the training corpus.

3.1. Experimental setup, results

A single speaker data from the Swiss French telephone DB *Polyvar* recorded at IDIAP [8] were used. The set of 218 calls was divided into training ($\frac{4}{5}$) and test ($\frac{1}{5}$) sets; only the training one was used for the search. The signal was parametrized using 10 LPCC coefficients in frames of 20 ms, with 10 ms overlapping. The LPCC mean was substracted for each call. In the same time, the pitch (using FFT-cepstrum on 400 ms frames) and log-energy were computed. The voice activity was detected using one absolute and one relative energy thresholds, and the raw decisions were smoothed using a 11-tap OR-filter ("all around must be silence to consider a frame silent"). As result, 5.2 hours of active speech containing 15813 active parts and 1.8×10^6 frames were obtained. The TD was done using the `td95` package (see Acknowledgements). The parameter controlling the number of spectral targets was empirically set to have approximately the same number of events per second as the phonetic rate (15 events/sec). The average length of one interpolation function including overlapping is 91.8 ms. The total number of events in the training corpus is 271078. An example of TD can be seen on Figure 2c.



e)

0	450000	j
450000	2500000	xI
2500000	3650000	E7
3650000	6050000	D4
6050000	7300000	j

Figure 2: Example for the French word "cinema". a) signal, b) spectrogram, c) TD interpolation functions, d) MG segmentation, e) HTK-like transcription.

The LPCC vectors situated in centers of IFs were quantized using VQ with 64 code-vectors. For the codebook training, an LBG algorithm with $2 \rightarrow 4 \rightarrow \ldots \rightarrow 64$ splitting was used. For simplicity, the code-vectors were marked by symbols [A...Z, a...z, 0...9, @ and $]. The MG dictionary training and segmentation was performed on the symbol string resulting from VQ quantization. The maximal length of sequence was m=5 and 10 iterations of the segmentation–reestimation loop were performed. The threshold for minimum number of occurences was set to min. 20 representants of one sequence in the training string. The resulting dictionary contains 64 1-grams, 1514 2-grams and 88 3-grams. No 4- and 5-grams were found with the requested minimal number. The total number of sequences is therefore N=1666. The average length of one sequence in terms of spectral events is 1.638, which corresponds to 112.7 ms. An example of multigram segmentation converted to HTK-like transcription can be seen on Figure 2d and 2e.

4. MODELLING AND RECOGNITION

For each sequence from the MG dictionary, an HMM is trained using the labelled training corpus. The reasons for this modelling method are two: first, HMMs are the standard framework for speech recognition with developed theory and tools, second, the iterations of HMM segmentation and training can overcome the errors of initial DT+VQ+MG labelling and improve the acoustic coherence of units. We call the models trained using original segmentation the "1st generation", those obtained using preceeding HMM segmentation "next generations". The encoding of input speech into sequences is done using standard Viterbi recognizer maximizing the product of HMM emission probabilities and language model (LM) probabilities. A simple "unigram" LM was derived either from the MG probabilities (for 1st generation models) or from evaluation of HMM re-labelling (for next generation HMMs). So, the likelihood to maximize is:

$$L = \prod p(M_i)^{\gamma} p(O|M_i) \qquad (2)$$

where O are the observations, $p(M_i)$ is the a-priori probability of model M_i and γ is the LM scale factor. This factor can be compared to Lagrange multiplier weighting the probability of codevectors in Entropy Constrained VQ or in Variable to Variable Rate VQ [5]. It has a significant influence on HMM dictionary size (in the refinement etap) and on resulting bit rate and speech quality (during the coding).

4.1. Experimental setup, rate evaluation, results

Simple left-right HMMs without skipping of states were chosen. The number of emitting states was determined by the number of TD events in the modelled sequence. For an $i-$gram, a prototype HMM with $2i + 1$ states was created. In this case, one can say, that each stable part and each transition of original TD events is modelled by one state. As parameters, 10 LPCC (1st stream) and 10 ΔLPCC (2nd stream) parameters were used, together with log-energy and Δlog-energy (3rd stream). In each stream, the output probability distribution is given by a single Gaussian. To form the total output probability, the stream ones are multiplied with equal weight. For HMM training and recognition, the HTK toolkit was used. First, the model parameters were context-free initialized (`HInit` and `HRest` tools), then 5 iterations of context dependend reestimation were run (`HERest`). It should be noted, that in this etap, the HMM training relies still on the original TD+VQ+MG transcription of signals. These 1st generation models were then used for new labelling of training corpus, with three different LM factors γ=0.0,5.0,10.0. The tool `HVite` was modified to allow using of a-priori model probabilities instead of standard bigram LM. The new segmentation was used to retrain the models (2nd generation). As the threshold of minimal number of examples was applied also in the refinement step, the numbers of HMMs in the final set depend on γ (see Table 1). The training and test corpora were encoded using those models. To evaluate the rate needed for encoding of sequence indices, either $-\log_2 p(M_i)$ or $\log_2 N$ bits can be considered to code i-th sequence. In Table 1, this is denoted by R_e (entropy coding) and R_u (uniform coding).

5. ADDITIONAL PARAMETERS, SYNTHESIS

The most important information is the time alignment between the original and synthetic segment. It can be either a simple constant, or more sophisticated information (DTW path, etc.). However, it should be noted, that the number of bits for such information is very limited. The set of other additional parameters needed to be transmitted depends on the choice of synthesis method. For example, for overlap-and-add based synthesis, the transmission of energy and pitch is necessary. It is very likely, that also this information can be processed on segmental level: a codebook of general or model-dependent pitch and energy contours should be easy to find.

		train. set			test set		
γ	N	R_e	R_u	R_s	R_e	R_u	R_s
0.0	1514	113	117	11.07	116	120	11.40
5.0	1201	68	74	7.19	69	74	7.29
10.0	894	51	58	5.95	50	58	5.95

Table 1: Resulting rates [b/s] for sequence indices coding. R_s is the avg. number of sequences per second.

5.1. Experimental setup

Unfortunately, the above mentioned synthesis was not yet tested experimentally and only a simple method based on concatenation of acoustic examples could be realized. For each model, 8 examples (signal, LPCC, energy) were found in the training corpus. When matching with input speech segment, labelled by certain HMM, first 4 candidate examples with best duration match were chosen. Among them, the "winner" has minimal DTW distance from the original in the parameter space. The choice of example (1 out of 8) can be encoded by 3 bits per sequence. Also, to avoid energy jumps, having negative influence on the fluency of synthetic speech, a constant forcing the mean energy to match with the original was transmitted (5 bits). So, the total bit rate is the sum of sequence bit rate plus 8 bits per sequence.

6. RESULTS

Listening tests were performed with speech encoded in above mentioned experiences (LM factor γ=0.0, 5.0, 10.0). For all three factors, the synthetic speech sounds naturally, without typical artefacts, known from frame based methods, but for γ =5.0 and 10.0, the intelligibility is bad. Therefore, we must use null LM (γ =0.0), for which the resulting bit rate (uniform coding) is 117+11.07×8 =205.6 b/s for training and 120+11.40×8=211.2 b/s for test corpus. The best speech quality was found when listening to synthetic digits and command words. For longer phrases, the intelligibility was sometimes worse, but in many cases, the comprehension was not excellent even when listening to the original, due to speaker's not very clear pronunciation.

A comparison of spectra of original and synthetic word can be seen on Fig. 3, audio examples related to this article can be downloaded as WAV or AU files from:

`www.fee.vutbr.cz/~cernocky/Icassp98.html`

From the point of view of computational load, it must be noted that especially the recognition portion of coding is very complex and time consuming – in this etap, all models are independent, each with 3 to 7 states. Different pruning and tying schemes are currently being investigated in order to limit the number of parameters.

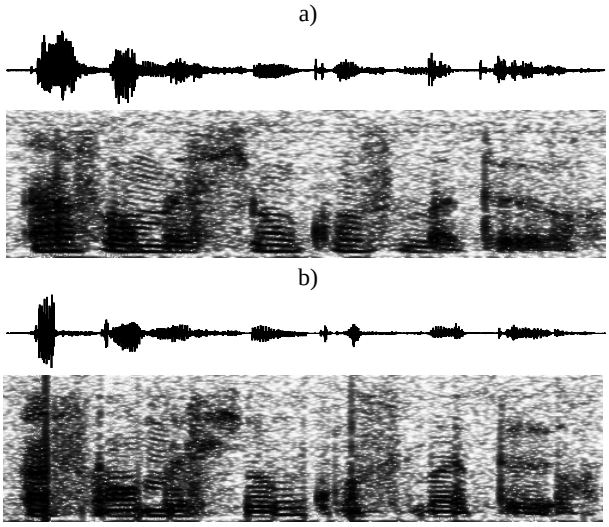


Figure 3: French words “information consommateur”: a) original, b) synthetic version with γ=0.0.

7. DISCUSSION, EXTENSIONS

One of main factors, influencing the quality of coding, is the database used for the training. The *Polyvar* is recorded over telephone, quite noisy and the speaking style of the speaker we used is not excellent. Future experiences will be performed with a different DB.

Presented coder is a laboratory prototype and is still subject of further research and improvements. First of all, the primitive concatenation based synthesis must be replaced by a more sophisticated method. Segmental synthesis has reached high degree of maturity and a PSOLA or MBROLA based approach should improve significantly the quality of output speech. Another issue is the speaker adaptation; a speaker normalization in coder and voice adaptation in decoder must take place. Useful spectral transforms are cited in [6] and [7], but for non-transcribed data, the classical approach (class dependent transforms) can not be used directly.

Better evaluation of synthetic speech quality, namely subjective tests with a group of listeners remain to be done. Also, the behavior of coder encoding different language from that (those) of training corpus, and possible adaptation of basic units set when confronted with a new language should be investigated.

In our opinion, automatically derived segments are good candidate for speech recognition, perhaps more coherent and reliable than widely used description units (context dependent phones). However, a link must be established between ortographic transcription and those units. A possible solution using joint multigrams was suggested in [9].

8. CONCLUSION

We have described a very low bit rate speech coder with automatically derived basic units. Our scheme does not need transcribed speech for training, only sufficient amount of acoustic data. With a simple synthesis, intelligible and naturally sounding speech was obtained at 211.2 b/s for single speaker. Among open issues, the limitation of complexity, more sophisticated synthesis, speaker adaptation and behavior in multilingual environment are of greatest importance. In our opinion, speaker and language independent coding with mean rate of hundreds b/s is possible using this scheme.

9. ACKNOWLEDGEMENTS

We would like to thank Frédéric Bimbot for the permision to use his temporal decomposition `td95` package. Thanks also to Andrei Constantinescu and Guillaume Gravier for HTK support. Most of the computations were done on SGI Power Challenge in Center of Information Services of TU Brno, we are grateful to system administrators for stable and reliable environment for our work.

10. REFERENCES

[1] B. S. Atal. Efficient coding of LPC parameters by temporal decomposition. In *Proc. IEEE ICASSP 83*, pages 81–84, 1983.

[2] G. Baudoin, J. Černocký, and G. Chollet. Quantization of spectral sequences using variable length spectral segments for speech coding at very low bit rate. In *Proc. EUROSPEECH 97*, pages 1295–1298, Rhodes, Greece, September 1997.

[3] F. Bimbot. An evaluation of temporal decomposition. Technical report, Acoustic research departement AT&T Bell Labs, 1990.

[4] F. Bimbot, R. Pieraccini, E. Levin, and B. Atal. Variable length sequence modelling: Multigrams. *IEEE Signal Processing Letters*, 2(6):111–113, June 1995.

[5] P. A. Chou and T. Lookabaugh. Variable dimension vector quantization of linear predictive coefficients of speech. In *Proc. IEEE ICASSP 94*, pages I–505–508, Adelaide, June 1994.

[6] K. Choukri. *Quelques approches pour l'adaptation aux locuteurs en reconnaissance automatique de la parole*. PhD thesis, École nationale supérieure des télécommunications (ENST), Paris, November 1987.

[7] C.M.Ribeiro and I.M.Trancoso. Phonetic vocoding with speaker adaptation. In *Proc. EUROSPEECH 97*, pages 1291–1294, Rhodes, Greece, 1997.

[8] A. Constantinescu and G. Chollet. Swiss PolyPhone and PolyVar: building databases for speech recognition and speaker verification. In *Speech and image understanding, Proc. of 3rd Slovenian-German and 2nd SDRV Workshop*, pages 27–36, Ljubljana, Slovenia, 1996.

[9] S. Deligne, F. Yvon, and F. Bimbot. Variable-length sequence matching for phonetic transcription using joint multigrams. In *Proc. EUROSPEECH 95*, pages 2243–2246, Madrid, Spain, 1995.

[10] G. Chollet et al. *NATO ASI: Computational models of speech pattern processing*, chapter Towards ALISP: a proposal for automatic language independent speech processing. Springer Verlag, in preparation.

[11] C. Jaskie and B. Fette. A survey of low bit rate vocoders. *DSP & Multimedia technology*, pages 26–40, April 1994.

[12] J. Picone and G. R. Doddington. A phonetic vocoder. In *Proc. IEEE ICASSP 89*, pages 580–583, Glasgow, 1989.