Diphone-like units without phonemes - option for Very Low Bit Rate Speech Coding

Petr Motlíček [1], Geneviève Baudoin [2] and Jan Černocký [1]

[1] Technical University of Brno, Institute of Radioelectronics
Brno, Purkyňova 118, Czech Republic
{*motlicek,cernocky*}*@urel.fee.vutbr.cz*
[2] ESIEE, Département Signal et Télécommunications
Noisy-le-Grand, CEDEX, France
baudoing@esiee.fr

Abstract

The aim of our effort is to reach higher quality of resulting speech coded by very low bit rate (VLBR) segmental coder. The basic units are found automatically in a training database using temporal decomposition and vector quantization. They are modeled by HMMs. Then two methods of re-segmentation were used in order to find new longer units. In the first approach borders are set to the centers of previous units. In the second, borders are fixed to the centers of middle HMM states of previous units. Number of frames in new units is conditioned to be bigger than a fixed constant. Hence, new units can consist of a several previous segments. Decreasing transition noise of resultant speech was obtained using these techniques.

1. Introduction

When we speak of very low bit rate coders, segmental or phonetic vocoders are meant [5]. Only those vocoders based on recognition and synthesis are able to efficiently limit bit rate. The coder and the decoder share the database of speech units (segments) that are considered to be representatives of any speech uttered by any speaker. Only the indices of representatives and some prosodic informations are transmitted by this coder. Hence, the bit rate of these types of coders can be less than 300 bps. The quality of this speech coding approach depends on a lot of factors. Among the most important is the quality of recognition of speech units. But speech analysis and synthesis are not less significant.

In our approach defining of speech units influences resulting quality of coder. Using phonetically labeled or transcribed speech database would discharge us these troubles. Unfortunately, current phone recognizers provide sufficient performances when the recognized speech is similar to the training database only. Hence, in our approach, speech units are supposed to be found automatically (by Automatic Language Independent Speech Processing (ALISP) tool) before training of recognizer. The fact that we do not need transcribed and labeled speech database can be great benefit of this method. The coder can be easily used in languages lacking standard speech databases.

When a set of speech units is obtained, it can be used for coding. Coder consists of recognizer acoustically labeling the speech and additional information encoder. In decoder, synthesis built on concatenating of examples from the training corpus is applied to obtain output speech.

An alternative technique built on using automatically derived speech units was developed at ENST, ESIEE and VUT-Brno [1]-[3]. However, the quality of resulting synthesized speech is not sufficient. This paper reports experiments based on re-segmentation of original units obtained by temporal decomposition. The aim of the re-segmentation is removing transition noise from the resulting speech. This noise is caused by concatenating of representatives in decoder. It is obvious that representatives will not match one each other as well as original coded speech. Hence, resulting decoded speech will always be influenced by this noise. However, its value can be largely decreased when this technique is applied.

The outline of the paper is the following: Section 2 describes the database used in our experiments. Section 3 presents the general principles of ALISP on which our work is based. Section 4 gives details on re-segmentation techniques and section 5 discusses the the problems with synthesis caused by re-segmentation of previous units. The following section 6 comments the final results in term of improving resulting speech applying the re-segmentation techniques.

2. Database

All our experiments are built on Boston University Radio Speech Corpus, database collected in 1995. The whole database contains data from 7 professional FM-radio speakers. Detailed description of this DB is given in its documentation [6]. For our purpose, data only of one female speaker were used. Clean and noisy data are included in DB. For training, as well as testing of our coder, only clean data were taken into account. At the beginning, the DB used in our experiments was split into training and testing parts. The both parts were classically pre-processed by LPC-cepstrum.

3. ALISP tools

Using of ALISP units for very low bit rate speech coding is in more details described in [1]-[3]. All our techniques of the re-segmentation are based on ALISP. Hence, only brief description is included.

3.1. Temporal decomposition

Temporal decomposition (TD) [4] is used for the initial segmentation of the speech into quasi-stationary parts. The matrix of spectral parameters is decomposed into a limited number of events. Each event is represented by a target vector and an in-

This work is supported by the grant No. VS97060 of the Ministry of Education, Youth and Sports of the Czech Republic.

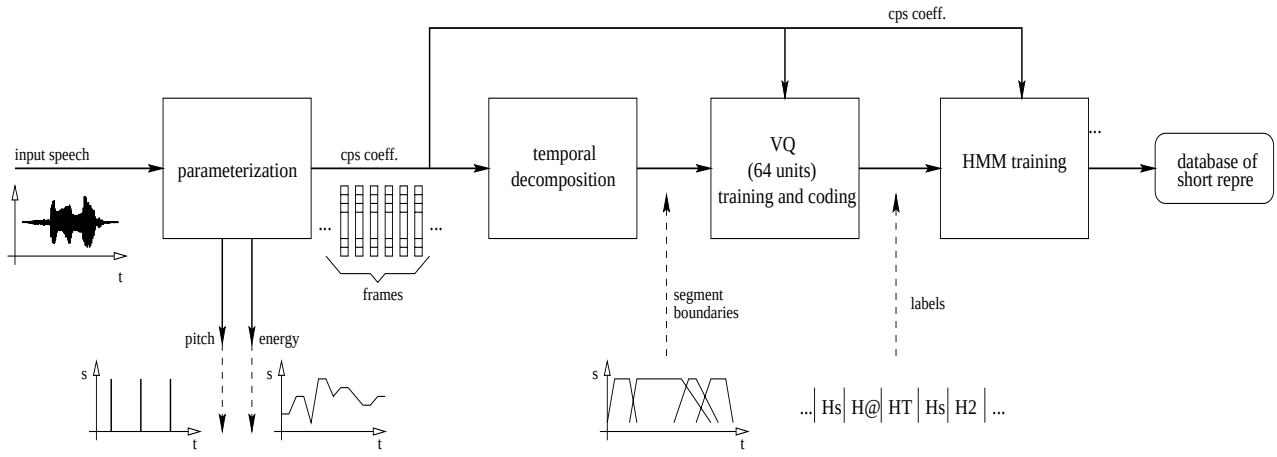

Figure 1: Scheme of training process.

terpolation function. The td95 package is used for the TD[1]. Only on training corpus, the temporal decomposition was applied. In addition, segments are generated in spectrally stable parts of speech. For each segment, the gravity center frame is computed.

3.2. VQ clustering

Vector Quantization (VQ) [7] is used for the clustering of the segments created by TD. It is obvious, that VQ is applied on training data, only. VQ consist of training, quantization and post-processing:

- The training of VQ codebook is built on the K-means algorithm. Only gravity center frames of the TD were used in its training. The length of VQ codebook is set to $L = 64$.
- In the quantization, each segment determined by TD is quantized by labels of the VQ codebook. On contrary to training, the quantization takes into account whole segments and uses cumulated distance of all the vectors from the segment.
- In order to work with HMMs, in post-processing the resulting labels from VQ are converted to symbolic form: *A, B, C, ...Y, Z, 0, 1, ...8, 9, a, b, c , ...y, z, @, $*. Each symbol has prefix *H*. These 64 symbols are used for description of unique labels.

3.3. HMMs

Hidden Markov Models (HMMs) are widely used in speech recognition because of their acoustic modeling capabilities. HMMs are related to original VQ symbols, so that their number is 64. The number of emitting states per model is fixed to 3. The models are initialized as left-right without state skipping. Not only one set of HMMs is generated. We have found that an iterative approach can improve the acoustical quality of units. Hence, several generations of models are created. It is obvious that in each iteration, not only the training data set, but also the test one was aligned with models. The test data set have not been segmented and quantized by TD and VQ, so that HMMs were used for detecting the units in unseen speech.
The whole training process described above is shown in Fig. 1.

[1] Thanks to Frédéric Bimbot (France) for permission to use it.

4. Re-segmentation

At this point, the re-segmentation of the original units recognized by HMMs is used. This technique is applied in order to decrease the influence of transition noise on the synthesized resulting speech. It is obvious that the transition noise will be always presented in our resulting decoded speech, because of the technique on which our approach is based. But applying appropriate segmentation, its influence can be notably decreased. Original TD segments, on which HMMs are afterwards trained, are created, so that they have contained stable parts of processed speech. Therefore, the boundaries of these segments are set to non-stable parts of speech that mostly contains small energy of signal. In decoder, where chosen appropriate representatives are concatenated to create resulting speech; these representatives are concatenated mostly in parts with small energy, as well. It is obvious that representatives do not come from original coded speech. They are chosen from training data set, so that they the best substituted coded units. Hence, transition noise appears in resulting speech.

The quality of signal can be described by ratio of signal to noise. In localities of segment boundaries, the ratio is small because of small energy of signal and quite big value of transition noise. But, if representatives in synthesis are concatenated in localities of stable parts of signal, where the energy of signal is mostly much higher, the transition noise would not come out so much in resulting speech and the ratio of signal to noise would be higher, as well. This is the general principle of our approach. One can say that instead of the re-segmentation of original units, new alternative segmentation could have been done at the beginning of our job. However the aim is not only creating units, so that their boundaries are set to the stable parts of speech signal. A new longer units that cover more non-stable parts of signal are required. It would be difficult to create them by TD and to train HMMs afterwards. Hence, the re-segmentation of original units is done after HMM recognition. The method is applied on recognized label sequence that comes from last HMM iteration. Obviously, it must be applied on training data set, as well as on the test one.

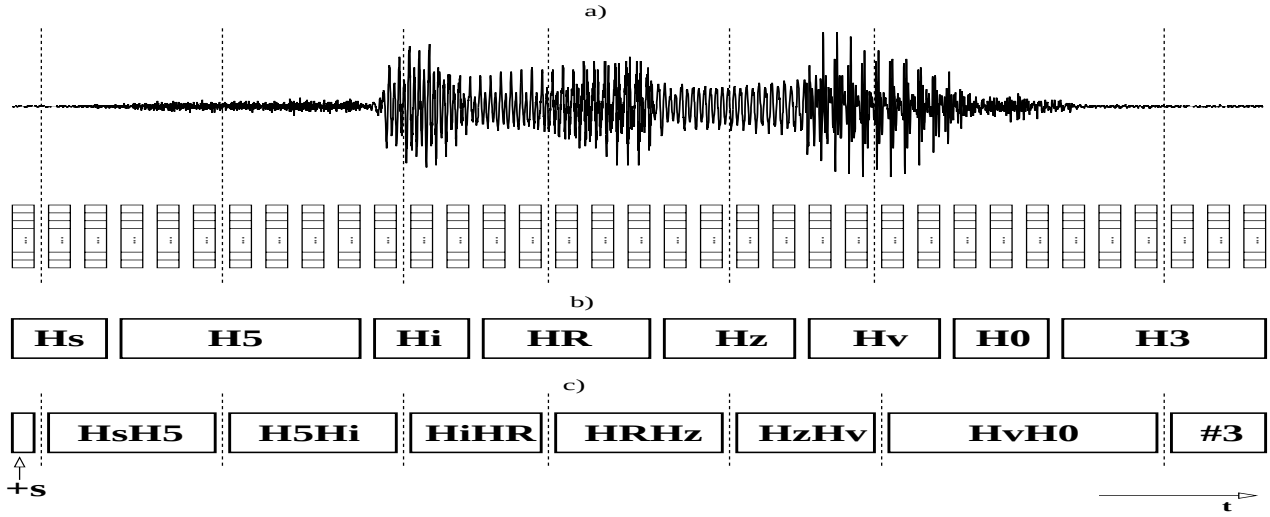

Figure 2: Example of re-segmentation according to middle frames of old units. Minimal length of new units is 4 frames. a) speech signal with its splitting into the frames, b) original segmentation recognized by HMMs, c) new re-segmentation.

One of the difficulties can be where to put the boundaries of new created segments in original ones. Hence, two experiments were done.
In first experiment, the new segment boundaries are set to the centers of previous segments. These centers were determined according to their middle frames. In the second experiment, the middle frames of middle HMM states in original segments interpret new segment boundaries. As mentioned before, new units are supposed to be longer than original ones. Hence, not each original segment will contain new segment boundary.

4.1. Re-segmentation according to middle frames of old units

In this approach, the boundaries of new units are put to the centers of old ones. Several experiments were done with this method that were different in minimal length of new units, as mentioned before. The minimal length represents the minimal number of frames in created new units. The algorithm of the re-segmentation is: First, the centers of old units are found. Then, we move from one center to another and remember the number of frames we went over. If number of frames between two neighboring centers is less than required, the second center is not declared as new segment boundary and we move to another old unit's center. This advance is still repeated unless we go over required minimal number of frames. Graphically the re-segmentation method can be seen in Fig. 2. It is obvious that the re-segmentation starts from the first center of first original unit, as it is seen in Fig. 2c. The "prefix" part of old unit is declared as an independent new unit. Its name will be start with character '+' in order not to be mistaken for real new unit. The same problem appears in the last processed old unit. Its name begins with '#'. The names of the whole new units consist of the names of old units that are covered by new one, as again seen in Fig. 2c.

4.2. Re-segmentation according to middle frames of middle states of HMMs

As mentioned before, in this approach, the new segment boundaries are represented by center frames of the middle HMM states of old units. The number of emitting states per one HMM is fixed to 3, as described in 3.3. Each state must contain one frame, at least. Hence, the minimal number of frames in an original unit is 3. If the number is higher, the frames are split into states according to score of recognition. It is obvious that the resulting segmentation based on this approach will be different to the first one.

5. Representatives and synthesis

In our experiments, parametric LPC synthesis was applied. To complete the coder we need to define define the synthesis units that will be used in the decoder to synthesize the resulting speech. Hence, for each unique dictionary unit, the three longest units from the training data set are kept, so that they are mostly downsampled when being converted to shorter segments. It is obvious that the attention is already payed to the training units after the re-segmentation.
When coding a previously unseen speech, first the coding units are detected using the HMM recognizer. Then, the stream of recognized units is re-segmented by one of the method described in 4. For each coding unit, the best synthesis unit (from 3 representatives) is chosen. The choice is done using minimum Dynamic Time Warping (DTW) distance between a representative and an input speech segment (coding unit). Important part of the synthesis is the prosody. The original prosody was used in our experiments.
When selecting the representatives to synthesize a previously unseen speech, we can easily find out that in coding speech, there are some coding units which do not have equivalent representative stored in DB of representatives (based on training data set). It is caused by the re-segmentation of original units. The theoretical number of unique units after re-segmentation is infinite. The explanation of this difficulty can be easily seen in Fig. 2. A new unit created by some of the re-segmentation

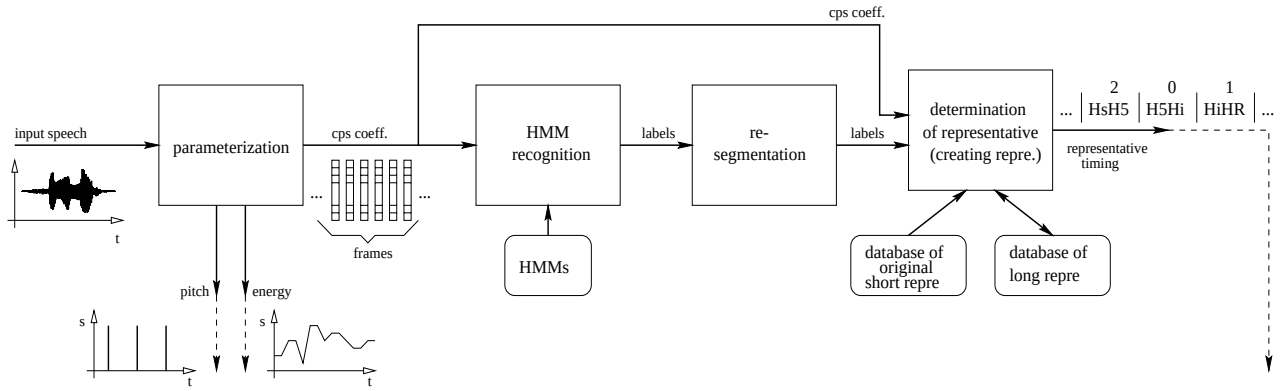

Figure 3: Scheme of coder used in our experiments. Only indices of coding units and numbers of chosen representatives are transmitted to the decoder.

...	...	...	...	...	...
6000000	7000000	H@	6000000	7000000	H@
7000000	7400000	Hj	7000000	7400000	Hj
7400000	8100000	H1	7400000	8100000	H2
8100000	8600000	H3	8100000	8500000	H3
8600000	8900000	Hn	8500000	8900000	Hn
...	...	...	...	...	...

Table 1: Example of two sequences of recognized labels from different HMM generations.

H**6**HnH3HP	↑	H4HnH3H**S**
H**5**HnH3HP	↑	H4HnH3H**R**
...	**H4HnH3HP**	...
H**3**HnH3HP	↓	H4HnH3H**O**
H**2**HnH3HP	↓	H4HnH3H**N**

Table 2: Seeking the existing representative for the unit H4HnH3HP by changing the first or last parts of units.

method can consist of two, three, or more original units, depending on the minimal required length of new units. Hence, a lot of re-segmented coding units can appear that have not been seen in training data set and for which we do not have any appropriate synthesis unit.
Therefore, two approaches were developed in order to obtain resulting speech.

5.1. Seeking the best synthesis unit from existing ones

Instead of non-existing synthesis unit, the best existing one will be used. Seeking this existing synthesis unit by DTW or another method, based on searching minimum distance between two segments, would result in very long search time. Hence, our seeking is based on a differences between the HMM generations. When comparing two original recognized units (before re-segmentation) from different HMM generations, we can easily notice that the sequences of recognized units are not the same.

They are quite similar, but some units are different in the recognized sequence on contrary to another one. An example can be seen in Tab. 1. Instead of unit *H1*, in second sequence, *H2* is recognized. According to this example, we try to replace non-existing synthesis unit by existing one. The replacing unit is being sought according to its name. We try to find the nearest name (of existing synthesis unit, of course) to the name of non-existing unit. The seeking process is created according to sequence of units' names, mentioned in 3.2.
The example from Tab. 1 is part of sequence of original units recognized by HMM. Number of these unique original units is 64, as mentioned in 3.3. It is obvious that in this case, we do not have any problems with non-existing units. The difficulty appears after the re-segmentation, as described in 5. In our experiments, the re-segmented units mostly consist of three or more original units. But the first and last parts are only the halves of the previous original unit. Only the middle ones are entire. Hence, applying the rule, explained above, on first or last part of the whole unit, we will not make as big mistake as applying the rule on the middle part. The explanation is given in Tab. 2. The non-existing unit *H4HnH3HP* is supposed to be replaced by *H5HnH3HP* or *H3HnH3HP*, ...
Unfortunately it can happen that any appropriate representative will be found using this method. In this case, the appropriate representative will be created.

5.2. Creating the representative for non-existing coding unit

The representative for non-existing coding unit can be created from original representatives. These representatives were created before re-segmentation from original recognized units. The technique is the same, as described in 5. The difference is that only one longest representative was chosen for each original unit (from training data set, of course). Then this representative was split into two halves, according to 4.1.
Creating new representative from original ones will be explained on the example:
When creating non-existing representative *H4HnH3HP*, the second half of original representative *H4*, the whole *Hn* and *H3* and the first half of original representative *HP* are used. This parts are concatenated into one complex which interprets representative for *H4HnH3HP* coding unit.
This technique is used only in case of failing of the previous one, described in 5.1, because of its high time complexity.

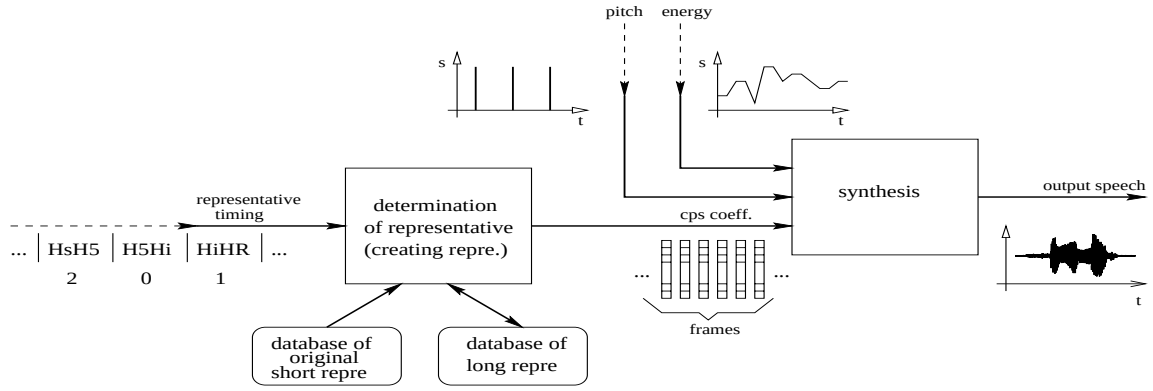

Figure 4: Scheme of decoder.

6. Results

The scheme of coder is given in Fig. 3. Only indices of coding re-segmented units with numbers of chosen representatives, as well as the DTW paths are transmitted to the decoder. The scheme of decoder can be seen in Fig. 4. Several experiments based on re-segmentation were done. They distinguish from each other in method used for re-segmentation, as described in 4.1 and 4.2. Furthermore, they are different in the length of created units, as mentioned in section 4.

6.1. Quality of resulting speech

It is obvious that the quality of resulting speech depends on experiment used for its coding. If new re-segmented units are short, the probability of not-existing representative for coding unit is small. Hence, an appropriate representative will be used almost every time. However, the re-segmentation is not applied on long parts of speech and the quality of resulting speech will not be higher than without re-segmentation.
In case of too long re-segmented units, a small number of transition appear in resulting units' sequence. However, using not large DB in our experiments, the probability of non-existing representative is much bigger. Hence, the most suitable representative has to be chosen from existing ones (or created from original representatives (more transition parts will appear there)). The quality of resulting speech is then lower, of course. Therefore, the optimal lengths of re-segmented units should be found according to the best resulting speech.

6.2. Bit rates

When applying the re-segmentation methods on original units, the number of created units in coding sentence is always less than without re-segmentation. In spite of this fact, the bit-rate does not necessarily decrease. The re-segmentation greatly increases the number of re-segmented unique units. Hence, more bits are needed when transmitting the indices of coding units. Therefore, resulting bit-rate depends on the lengths of new re-segmented units.

7. Conclusion

The purpose of applying the re-segmentation techniques was to reach higher quality of resulting speech coded by VLBR coders. This aim was achieved with all our experiments that were built on the re-segmentation. Some examples of resulting speech can be found on:
http://www.fee.vutbr.cz/∼motlicek/speech.html.
The speech coded only using original units (re-segmentation not used) and the resulting average bit rates of speech coming from all our experiments are given there, as well.
In our experiments, the prosody and timing (DTW) path have was not coded, at all. However in the future, the resolving of this task will be required in our coder. One of the approach is described in [8].
Unfortunately, the LPC synthesis used in our experiments is responsible for a lot of artifacts and unnatural sounds of resulting speech. Hence, a better synthesis producing clearer speech, without increasing the bit rate, will be used in coder. Harmonic Noise Model (HNM) is one of the candidate.

8. References

[1] J. Černocký. Speech Processing Using Automatically Derived Segmental Units, *PhD Thesis, ESIEE*, France, 1998.

[2] J. Černocký, G. Baudoin, and G. Chollet. Segmental Vocoder-going beyond the phonetic approach. *Proc. ICASSP Seattle* pp. 605-608, May 1998.

[3] 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*, pp. 1295-1298, Rhodes, Grece, September 1997.

[4] B. S. Atal. Efficient Coding of LPC Parameters by Temporal Decomposition. In *Proc. IEEE ICASSP 83*, pp. 81-84, 1983.

[5] J. Picone, and G. R. Doddington. A phonetic vocoder. In *Proc. IEEE ICASSP 89*, pp. 580-583, Glasgow, 1989.

[6] M. Ostendorf, P. J. Price, and S. Shattuck-Hufnagel. The Boston University radio news corpus. *Technical report*, Boston University, 1995.

[7] J. Makhoul, S. Roucos, and H. Gish. Vector quantization in speech coding. In *Proc. IEEE*, 73(11):1551-1589, November 1995.

[8] Y. P. Nakache, P. Gournay, G. Baudoin. Codage de la prosodie pour un codeur de prole a tres bas debit par indexation d'unites de taille variable, CORESA 2000.