CONCEPTS OF THE DISSERTATION

Jan Černocký

June 1996

Contents

1 Introduction 2

2 Organization of Czech-French PhD studies 2

3 Classical methods of speech processing 2

3.1 Signal approachs of speech coding 3

3.2 Modelling approach of speech coding 3

3.3 Parameters of the prediction filter 4

4 Concepts of segmental speech processing 4

4.1 Matrix quantization 4

4.2 Multiframe coding 5

4.3 Coding with phonetical segmentation 5

5 Multigrams 5

5.1 Theoretical basis 5

5.2 Application to speech spectrum representation 6

5.3 Evaluation 6

5.4 Implementation of the segmentation 7

5.5 Experimental results 8

5.6 Discussion 8

6 Modified multigrams 9

6.1 Notion of distance 9

6.2 Segmentation and dictionary training 9

6.3 Experimental results 10

6.4 Conclusion for the multigram methods 11

7 Remarks on spectral analysis for the segmental processing 12

8 Human perception and digital speech processing 12

8.1 Frequential and temporal masking 13

8.2 Auditory models 13

9 Conclusion 13

1 Introduction

Since my diploma project I elaborated in 1993 [14] I has been oriented to the digital speech processing. My PhD studies are focused to this field, mainly to the domain of segmental speech processing. This paper presents the basic ideas of what I already have done and what I would like to do until the end of these studies in 1998.

This paper is divided into 9 sections. My PhD is marked by the collaboration with the Département Signal of ESIEE in France. Section 2 describes the organization of this "international" PhD. Section 3 briefly deals with standard methods of speech processing, especially with the source-filter modelling. In Section 4 the segmental approachs of speech processing are introduced. Two following sections describe the work done in multigram speech spectrum representation. Section 5 deals with the "classical" method while Section 6 describes the modifications. This part of the paper is based on the proposal of article for IAPR Workshop held in Ljubljana in April 1996 [15]. Next two Sections (7 and 8) mark the way of continuation of my PhD studies – the former attempts to criticize the standard fixed-frame approach of speech spectral analysis, the later introduces the perceptive methods. Section 9 contains the conclusion.

2 Organization of Czech-French PhD studies

The starting point of my "international" PhD was the diploma project "Sub band speech coder 16 kbit/s" [14] which I had the chance to elaborate at ESIEE under the direction of Mme Geneviève Baudoin and in collaboration with the company SECMAT N.T. During the 1st year of my PhD in Brno I worked hard on my French and I applied for the French Government PhD scholarship . I succeeded and obtained a scholarship for the DEA diploma and 3 years of Czech-French PhD studies (programme CO-TUTELLE). Table 1 describes this system in detail.

1993-94	FEI VUT Brno beginning of PhD	
1994-95	Université d'Orsay DEA (Diplôme d'Etudes Approfondies) obligatory for each applicant for the PhD in France	DEA
1995-98	Czech-French PhD • 6-month periods in France ESIEE, Dpt. Signal Directors: Geneviève Baudoin, Gérard Chollet. • 6-month periods in the Czech Republic FEI VUT, Inst. of Radioelectronics Director: Vladimír Šebesta	PhD

Table 1: Organization of the PhD.

The final presentation should take place in Orsay in 1998 with a mixed Czech-French jury. There exist an agreement between the Univerity of Orsay and VUT Brno that the French "Docteur" title will be recognized as the Czech "Dr.".

3 Classical methods of speech processing

The speech processing can be roughly divided into three principal domains:

- *synthesis* which is attempting to create an utterance never before pronounced by a human speaker.
- *recognition* where we are trying to decide "what was said". The output can be a string of characters (dictation systems) or a command for a voice-driven device.
- *coding* defined as "an efficient representation of speech for the transmission and storage purposes".

As the main efforts of this dissertation are being done in the coding (or "representation") field, we will develop it in the following paragraphs. We insist that this is only a minimal form of the introduction to speech processing and recommend [11, 8] to an interested reader.

3.1 Signal approachs of speech coding

Digitalized speech signal can be considered as a member of the large family of digital digital signals and processed by standard coding methods:

- Pulse Code Modulation (PCM) represents the speech samples by numbers from a limited set, either with linear or logarithmic scale. Logarithmic PCM (with A- or μ-law) at 64 kbit/s is the standard for telephone quality speech coding.
- Differential Pulse Code Modulation (DPCM) where a sample is predicted from past samples using the prediction filter. Only the prediction error is coded and transmitted/stored.
- Adaptive Differential Pulse Code Modulation (ADPCM) equivalent to the previous one except for the adaptive prediction filter.

These methods are applicable to speech as well as to other audio signals (music, natural noises, etc.).

3.2 Modelling approach of speech coding

In this case, we take into account that the speech is produced by humans, in a specific manner. The input to the articulatory tract (vocal chords vibrations, noise, or a mixture of both) is modelized by the source $U(z)$ and the vocal tract by a filter $H(z)$. For simplicity reasons, the vocal tract is often considered as an all-pole system and modelized by an IIR filter:

$$H(z) = \frac{1}{A(z)} \tag{1}$$

where the polynomial $A(z)$ of P-th order is defined by

$$A(z) = 1 + a_1 z^{-1} + a_2 z^{-2} + \ldots + a_P z^{-P} \tag{2}$$

It can be shown that its coefficients, under the assumption that $U(z)$ has the character of white noise, can be estimated using the Linear Prediction (LP) method. If we denote the speech signal y_i, we can predict the n-th sample using P previous samples:

$$\tilde{y}_n = -\sum_{i=1}^{P} a_i y_{n-i} \tag{3}$$

When minimizing the energy of prediction error $e_n = y_n - \tilde{y}_n$, we obtain following relation for the optimal set of coefficients a:

$$\mathcal{E}(y_n y_{n-i}) + \sum_{j=1}^{P} a_j \mathcal{E}(y_n y_{n-i}) = 0 \quad \text{for} \quad i = 1 \ldots P \tag{4}$$

If we assume the signal to be stationary, we can replace the expectations by the autocorrelation coefficients $r_1 \ldots r_P$. The signal $E(z)$ obtained from $Y(z)$ by filtering

$$E(z) = Y(z)A(z) \tag{5}$$

is the estimation of the original excitation $U(z)$.

In the reality, the speech signal is not stationary but we suppose it to be within *frames* of 10 to 30 ms. In the coding, for each of those frames the filter parameters are calculated using temporal estimates of autocorrelation coefficients, the excitation is estimated using Eq. 5 and both filter parameters and excitation are further processed.

3.3 Parameters of the prediction filter

Although the excitation processing is an important part of all low-rate speech coders (represented nowadays mainly by the CELP algorithm, which uses vector quantization for the excitation representation), our work was focused to the parameters of the filter $H(z)$.

When observing the speech spectrum, we can divide it into two parts: the excitation is responsible for the fine structure (harmonics of the fundamental frequency or noise-like character), the vocal tract forms the spectral envelope. In modelling, this envelope is represented by the response of the IIR filter in the frequency domain. For simplicity reasons, we will call this response $H(f)$ the *spectrum* in the rest of this paper.

As mentiones above, the denomitor of $H(z)$ has the form of a polynomial (Eq. 2). The filter can be either parametrized by the LP coefficients a_i or by a set of derived values (PARCOR, LAR, LSF, etc.). We used LPC-cepstral (LPCC) coefficients, mainly for their advantages in the the calculation of spectral distance between the quantized and unquantized version of filter $H(z)$ (see Subsection 5.3 for details). The LPCC coefficients are easily derived from the LPC ones using:

$$c_n = -a_n - \sum_{k=1}^{n-1} \frac{k}{n} c_k a_{n-k} \quad \text{for} \quad n = 1 \dots \infty \tag{6}$$

4 Concepts of segmental speech processing

In both speech coding and recognition, there is a need of representing the speech signal by as little information as possible. In coding we are limited by the transmission channel capacity. In recognition, it is the generalization of speech events which we expect from the speech signal description. The standard systems work on a segmental basis - the signal is divided into fixed-length frames, which are processed independently, without taking into account the inter-frame dependencies. However, these dependencies exist and when observing a speech spectrogram, we can see on one hand stationary parts (eg. in long vowels), on the other hand many rapid spectral transitions.

In the literature, approachs taking profit of the segmental nature of speech are described and can be roughly divided into three domains: *Matrix Quantization*, *Multiframe Coding* and *Coding with Phonetical Segmentation*. These three approachs will be described more in detail in following subsections, but it is necessary to mention, that they are not exclusive. In a real algorithm we can easily meet all three of them. As we worked mainly with the spectral envelope, the description will be focused to spectrum coding and neglect the excitation aspects.

Two main questions of segmental coding schemes must be mentioned:

- *fixed* vs. *variable* length of frames. The variable length frames modelize better the speech events, which do not have constant duration. Contrarily, the fixed bit rate is mostly demanded in the coding and the variable length segment schemes must contain a buffering facility.
- *mathematical* vs. *phonetical* segmentation methods. The spectral vector flow may be either segmented purely by statistical methods (as it is the case of our experimental work) or by using some part or phonetical knowledge.

4.1 Matrix quantization

We define a spectral vector as a set of coefficients characterizing the filter $H(z)$. We write a column vector

$$\boldsymbol{x}_k = (x_{k,1} x_{k,2} \dots x_{k,P})^T \tag{7}$$

containing P coefficients $x_{k,i}$ (these may be LPC, PARCOR or any other set). In Matrix Quantization (MQ), one or more of these vectors form a spectral matrix

$$\boldsymbol{X}_k = (\boldsymbol{x}_1 \boldsymbol{x}_2 \dots \boldsymbol{x}_m) \tag{8}$$

with dimensions $P \times m$. This matrix is considered as a big vector and processed by the Vector Quantization. This method has an important complexity, not only for the creation of codebook of code-matrices, but also for a simple quantification. Methods described in the literature mostly employ stochastic codebooks to bypass the necessity of codebook training. Generally, the code-matrices are of fixed length and a time alignment (linear interpolation, DTW) is performed to match the matrices at the input with the code ones. Examples of this approach can be found in [13, 4].

4.2 Multiframe coding

With the matrix quantization in mind, we pass to Multifarme Coding (MFC). In this case, the spectral vectors are also grouped to matrices, but only some of the vectors are chosen and coded, the rest is discarded and obtained by the interpolation in the decoder. By this "diluting" of matrix $\boldsymbol{X}$, we obtain less complex methods. The choice of vectors to be transmitted can be done in two ways:

- *open loop* - the vectors to be transmitted are chosen by detecting the "important points" in the spectrum sequence. These can be either local maxima of temporal derivation, or places, where the spectral vectors are too far from a prealably calculated interpolation line.
- *closed loop* - all possible ways of "diluting" the spectral matrix are tested and the optimal one, minimizing the average spectral distance, is chosen.

The temporal information is important in this case - it is necessary either to transmit the information about the the repartition of valid frames among the discarded ones, or to add a time marker to each transmitted frame. As an example of this approach we can cite [10].

4.3 Coding with phonetical segmentation

In this kind of schemes, the frames with similar phonetical character are searched and grouped to segments. The frame classification can be simple (only consonants/vowels), or more sophisticated (definition of mixed segments, onset areas, etc.). The coding is modified according to the character of segment: the length of frames may be changed, the bit proportion for filter/excitation is modified according to the needs of segment, often we encounter variable numbers of spectral coefficients, even different types of coefficients (for example LAR and LSF) in one coding scheme. In [17], which may serve as an example of phonetical segmentation, the term *phonetical integrity* is introduced - the phonetical qualities of speech should be taken into account more than classical "signal" factors.

5 Multigrams

In our work we were looking for a method allowing us in the same time to segment a string of spectral vectors and to represent the sequences in an efficient way. We found the multigram method, originally developed and tested for written texts analysis. As this method needs a string of discrete symbols on its input, we used Vector Quantization of the spectral vectors to this conversion.

5.1 Theoretical basis

The article of Bimbot et al. [1] and following articles [6, 2] were the basis for the first part of our work with multigrams. We will limit us to a brief description and refer the reader to above mentioned articles. On the input of a segmentation we have a string of symbols $W = w_1 w_2 \dots w_N$. For a segmentation B into sequences $S_1, S_2, \dots S_q$ of length 1 to n we introduce the likelihood

$$L(B, W) = \prod_{k=1}^{q} p(S_k) \tag{9}$$

where q is the number of sequences in this segmentation, S_k is k-th sequence and $p(S_k)$ its probability. The optimal segmentation maximizes this likelihood, therefore

$$L(W) = \max_{\{B\}} \prod_{k} p(S_k) \tag{10}$$

where $\{B\}$ is the set of all possible segmentations. The number of all these segmentations is large and we use a Viterbi-based algorithm to find the optimal one. We define the likelihood of a partial string $w_1 \dots w_{k+1}$ as

$$L(w_1 \dots w_{k+1}) = \max_{1 \leq i \leq n} p([w_{k-i+2} \dots w_{k+1}]) \times L(w_1 \dots w_{k-i+1}) \tag{11}$$

and by evaluating it for $1 \leq k \leq N-1$, we obtain the optimal segmentation in sense of Eq. 10. The practical aspects of search of the optimal segmentation based on Eq. 11 will be discussed in Subsection 5.4.

To be able to segment, it is necessary to dispose of probabilities of sequences, but these are not known a-priori. That is why we build a *dictionary* of such sequences using a training string and an iterative procedure. We initialise the probability of each sequence T, that we can find in the string to

$$p_{init}(T) = \frac{c_{init}(T)}{C_{init}} \tag{12}$$

where $c_{init}(T)$ is the number of occurences of T in W and $C_{init} = N.n$ (number of all possible sequences of lengths 1 to n). Then we reestimate the probabilities in a loop:

- *Segmentation*, where we segment the training string using Eq. 10.
- *Reestimation of probabilities* on the basis of segmented string:

$$p(S) = \frac{c(S)}{C} \tag{13}$$

c(S) denotes here the number of occurences of S in the *segmented* string and C the total number of sequences after the segmentation. The probability may also be reestimated with a prunning factor a, which helps to eliminate the rare sequences from the dictionary:

$$p'(S) = \frac{c(S)}{C}\left(1 - a\sqrt{\frac{C - c(S)}{C.c(S)}}\right) \tag{14}$$

Article [6] gives another possibility of probability reestimation using ML-EM method.

5.2 Application to speech spectrum representation

We applied the multigram segmentation and dictionary training to vector quantized speech spectra. We disposed of a telephone database of one speaker, with 8000 Hz sampling frequency and 16 bit quantization. After suppression of silences and high-pass filtering by $1 - 0.95z^{-1}$, we created 20 ms frames with 10 ms overlapping and we calculated 10 LPC coefficients $a_1 \dots a_P$ (for $P = 10$). These were converted to LPC-cepstrum coefficients $c_1 \dots c_P$ which formed the vectors used for vector quantization and modified multigrams (see Subsection 6.3). We performed a VQ-codebook training (by a simple LBG algorithm) for codebook lengths $L = 2, 4, 8, 16, 32, 64, 128, 256, 512$. We quantized the spectral vectors by these codebooks and we obtained 9 training and 9 test strings of symbols. The length of the former ones was 213270, the length of the later ones 122903.

5.3 Evaluation

The results of VQ were evaluated in terms of *average spectral distortion SD* defined as the mean over all frames of the logarithmic spectral distance

$$D = \sqrt{\int_{-1/2}^{1/2} \left[10 \log S(f) - 10 \log \hat{S}(f)\right]^2 df} \quad \text{in dB} \tag{15}$$

where $S(f) = 1/\|A(f)\|^2$ is the power LPC-spectrum and $\hat{S}(f)$ the power spectrum with quantized coefficients. Using the Parceval's relation it can be calculated using LPC-cepstral coefficients (see Eq. 6 for the definition):

$$D = \mu\sqrt{2\sum_{i=1}^{\infty}(c_i - \hat{c}_i)^2} \tag{16}$$

where c_i and $\hat{c}_i$ are the unquantized and quantized cepstral coefficients respectively and $\mu = \ln(10)/10$ is a constant enabling the conversion to decibels. The infinity in the sum can be replaced by a reasonably chosen number, for ex. for 128, the result is very precise. To limit the computational load, we used only P (the order of prediction filter) as the upper limit of the sum in Eq. 16 in all evaluations. This simplifiacation introduces an error of up to 12% for SD, but for comparisons, the precision is sufficient.

For the evaluation of “classical” multigrams, it is not necessary to re-evaluate the spectral distortion – it is given by the VQ used. To measure the efficiency of the representation, we compare the entropy of VQ codebook with the entropy per symbole of the multigram dictionary. We define the former as:

$$H(V) = -\sum_{i=1}^{L} p(\boldsymbol{y}_i) \log_2 p(\boldsymbol{y}_i) \tag{17}$$

where $\boldsymbol{y}_i$ are the code-vectors of VQ-codebook and the later as

$$H'(M) = -\frac{\sum_{i=1}^{Z} p(M_i) \log_2 p(M_i)}{\sum_{i=1}^{Z} l(M_i) p(M_i)} \tag{18}$$

where Z is the total number of multigrams in the dictionary, $p(M_i)$ stands for the probability of the multigram M_i and $l(M_i)$ for its length. We note, that the term in the denominator is the average length of multigrams.

As these quantities are based only on the training string, it is necessary to validate them on a test one. We are evaluating the average *rate* for both the VQ and the multigrams. We compare this rate to $H(V)$ and $H'(M)$. The average rate for the VQ is defined by

$$R(V) = -\frac{\sum_{i=1}^{L} c_{test}(\boldsymbol{y}_i) \log_2 p(\boldsymbol{y}_i)}{N_{test}} \tag{19}$$

where $c_{test}(\boldsymbol{y}_i)$ is the number of vectors of the test string represented by code vector $\boldsymbol{y}_i$, and N_{test} is the length of the test string. For the multigrams, we obtain a similar formula:

$$H'(M) = -\frac{\sum_{i=1}^{Z} c_{test}(M_i) \log_2 p(M_i)}{N_{test}} \tag{20}$$

where in this case, $c_{test}(M_i)$ is the number of sequences represented by multigram M_i.

Another important criterion is the size of resulting multigram dictionary, which informs us how easily the appropriate entropy-code can be constructed.

5.4 Implementation of the segmentation

As we see in Eq. 11, during the segmentation we always decide, what part should be “cut” at the end of the partial string $w_1 \dots w_{k+1}$. We must memorize this information for all k, and we can obtain the optimal segmentation by a recursive search only after having finished with all symbols (for $k = N - 1$). This is not practical for long strings and useless for ex. for speech coding where it is impossible to wait until the end of signal, then segment and only then begin the transmission. That is why we use a buffer of limited length LB and we memorize a history of symbols $W_{hist} = w_{k-LB+2} \dots w_k$ and a history of optimal lengths $l_{hist}^{opt} = l_{k-LB+2}^{opt} \dots l_k^{opt}$. For $k + 1$, we perform following steps:

1. increment the length of buffer $LB = LB + 1$.
2. write the symbol w_{k+1} to the end of history of symbols. The history becomes $W_{hist} = w_{k-LB+2} \dots w_{k+1}$.
3. determine the optimal length “to cut” (using Eq. 11) and write it to the end of history of lengths: $l_{hist}^{opt} = l_{k-LB+2} \dots l_{k+1}$.
4. search the common point CP of n segmentations beginning by $l_{k-n+2}^{opt} \dots l_{k+1}^{opt}$ by backward search in the history of optimal lengths.
5. if CP was not found and LB reached the limit LB_{max}, force the common point $CP = k + 1$.
6. process the multigrams for the sub-string $w_{k-LB+2} \dots w_{CP}$ (process means updating of numbers of occurences for the training phase or writing of multigram length and index for the “representation” phase). Shorten the buffer to $LB = k + 1 - CB$ and go to Step 1.

If we find a CP in a natural way (we do not force it), it does mean that the segmentation from the beginning of buffer (w_{k-LB+2}) up to this point (w_{CP}) will not change for any $i > k$. The forcing of CP is necessary mainly when $l^{opt} = n$ for many successive symbols (in this case all segmentations are unique and they do not “meet”).

5.5 Experimental results

The results of the vector quantization with the codebooks of $2, 4, \ldots 512$ vectors are given in Table 2. We see that the entropy of VQ codebook is validated by the rate obtained on the test string – we do not observe a significant difference between $H(V)$ and $R(V)$.

L	$H(V)$ [bit]	$R(V)$ [bit]	SD [dB]
2	0.999	1.000	4.332
4	1.950	1.930	3.770
8	2.914	2.888	3.199
16	3.881	3.844	2.893
32	4.863	4.830	2.640
64	5.869	5.848	2.413
128	6.858	6.840	2.231
256	7.865	7.853	2.055
512	8.865	8.852	1.907

Table 2: Results of Vector Quantization: codebook size, entropy of the codebook, rate obtained on the test string and spectral distortion.

We performed several test of the multigram segmentation with the maximal lengths of multigrams $n =$ 2, 4, 6, 8, 10 and with the penalization factors $a =$ 0, 0.5, 1.0, 1.5, 2.0, 2.5, 3.0. For each combination, we performed a multigram dictionary training with 10 iterations of the process "segmentation" ⇒ "reestimation of probabilities". We evaluated the results in terms of entropy per symbole of the resulting multigram dictionary, we validated these numbers on the test string (calcul of $R(M)$) and we recorded the multigram dictionary size Z.

The optimal results for two VQ codebooks $L = 16$ and $L = 128$ are given in Table 3. Note, that there are two kinds of optimal results:

- (1) where one takes into account only the decrease of multigram dictionary entropy. In this case, we could conclude that for $L = 16$, the bit economy is $H(V) - H'(M) = 3.881 - 1.407 = 2.474$ bits or 63% of the original bit rate. This optimistic result looses its brightness if we look at the dictionary size and at the rate on the test string: 4.051 is more than 4 bits necessary to represent 16 symbols!
- (2) where the optimum is found as the minimal rate on the test string. For $L = 16$ we obtain worse, but more realistic results: the difference $H(V) - H'(M) = 3.881 - 2.264 = 1.617$ bit so that the economy due to the using of multigrams is 41%.

When increasing the number of codebook vectors, we obtain worse results. The difference $H(V) - H'(M)$ (with the (2) criterion) is only $6.858 - 5.414 = 1.444$ bit, or 21% for the VQ codebook $L = 128$.

L	16(1)	16(2)	128(1)	128(2)
n_{opt}	10	6	10	2
a_{opt}	0.5	1.0	0.0	0.0
Z	16225	2593	21368	3451
$H'(M)$	1.407	2.264	1.453	5.414
$R(M)$	4.051	2.328	12.737	5.465

Table 3: Optimal results for the multigram method: optimal length of multigrams, optimal penalization factor, size of resulting multigram dictionary, entropy per symbol of the multigram dictionary and rate obtained on the test string. (1) - optimum for the training string, (2) - optimum for the test string.

5.6 Discussion

Although this method offers good results for written text analysis (see [1, 6, 2]), we conclude, that a direct application to vector quantized speech spectra is impossible. A significant decrease of entropy is obtained only for small codebooks, which are, on the other hand, not enough precise for the spectrum representation. For greater codebooks, the main disadvantage is an enormous multigram dictionary, whose size is comparable to the length of the training string. This is a proof of a strong overlearning, verified on the test strings.

We see the reason in a great variability of characteristic symbol sequences, which increases with L: for greater codebooks we can say that almost each longer sequence is unique.

6 Modified multigrams

6.1 Notion of distance

In the written texts processing, it is correct to demand the exact representation of a sequence by a multigram. Hence, for ex. the word "BREAK" can never be represented by "BREAD" or by "BREAAAK". When working with spectral vectors, it is possible to relax this constraint – we can represent a sequence of vectors by a slightly different one, if they are similar enough from the perceptual point of view.

The introduction of a *distance* measure is a natural way to overcome the above mentioned problems. Having a distance notion, we will no more demand a strict equivalence between a sequence and a multigram but we will measure their similarity by their distance. Of course, in case of spectral vectors, we will be no more able to work just with the corresponding indices of code-vectors (symbols for "classical" multigrams), but we will return to unquantized vectors.

We have defined the distance of two sequences of vectors in a very simple manner as the average Euclidean distance of corresponding vectors. The distance of two sequences $\boldsymbol{X}$ and $\boldsymbol{Y}$ having the same length l is

$$D(\boldsymbol{X}, \boldsymbol{Y}) = \frac{1}{l} \sum_{i=1}^{l} d(\boldsymbol{x}_i, \boldsymbol{y}_i) \tag{21}$$

where $d(\boldsymbol{x}, \boldsymbol{y})$ is the Euclidean distance of two vectors. It is of course possible to use more sophisticated distance definitions (Itakura-Saito for ex.), or to use various weightings in the sum (for ex. depending on energy of corresponding frames). Another possibility is a time alignment of multigrams, which may take advantage of timing variations of characteristic speech parts (various lengths of the same vowel etc.).

6.2 Segmentation and dictionary training

In this approach we use a dictionary that consists no more of symbol sequences but of vector sequences (in analogy to VQ we will call them code-multigrams). In the same manner, the input string does not consist of symbols but of unquantized vectors. Similarly as in the "classical" multigram case, we are searching a segmentation into variable length (1 to n) sequences and a procedure for creation of dictionary of typical sequences.

The optimal segmentation of a string $\boldsymbol{X} = \boldsymbol{x}_1 \dots \boldsymbol{x}_N$ into sequences $\boldsymbol{U}_1 \dots \boldsymbol{U}_q$ of length 1 to n is given by maximization of the quantity (formerly called *modified likelihood*, but after the protest of Fréderic Bimbot renamed to "*a quantity*")

$$L'(B, \boldsymbol{X}) = \prod_{k=1}^{q} p'(\boldsymbol{M}_k) \tag{22}$$

over the set of all possible segmentations $\{B\}$

$$L'(\boldsymbol{X}) = \max_{\{B\}} \prod_{k} p'(\boldsymbol{M}_k) \tag{23}$$

The main difference from the previous likelihood definition (Eqs. 9 and 10) is the use of the *penalized probability* p'. It is no more a probability of a sequence but a modified probability of the code-multigram $\boldsymbol{M}_k$ representing the sequence $\boldsymbol{U}_k$. The multigram $\boldsymbol{M}_k$ is chosen among all multigrams of the length $l(\boldsymbol{U}_k)$ to minimise the distance $D(\boldsymbol{M}_k, \boldsymbol{U}_k)$.

The penalized probability is computed from the multigram probability (found in the dictionary) by

$$p'(\boldsymbol{M}_k) = Q\left[D(\boldsymbol{U}_k, \boldsymbol{M}_k)\right] p(\boldsymbol{M}) \tag{24}$$

where $p(\boldsymbol{M}_k)$ is the probability of multigram-code $\boldsymbol{M}_k$ and $D(\boldsymbol{U}_k, \boldsymbol{M}_k)$ is the distance between this multigram and the sequence $\boldsymbol{U}_k$.

The function $Q[.]$ must penalize the probability of multigram in function of its distance from the represented sequence. When the sequence is equal to the multigram, the function $Q = 1$, on the other hand, for a sequence "far" from the multigram, its probability must be severely penalized. We used a simple partially linear function defined by:

$$Q[D] = \begin{cases} 1 - \dfrac{D}{D_{max}} & \text{for} \quad D \leq D_{max} \\ \\ 0 & \text{for} \quad D > D_{max} \end{cases} \tag{25}$$

where D_{max} is a constant giving the maximal distance for which p' may be nonzero.

The segmentation algorithm is similar to Eq. 11 but instead of a likelihood of partial string and a probability, the modified likelihood of partial string and the penalized probability are used

$$L'(\boldsymbol{x}_1 \ldots \boldsymbol{x}_{k+1}) = \max_{1 \leq i \leq n} p'(\boldsymbol{M}_i^{opt}) \times L'(\boldsymbol{x}_1 \ldots \boldsymbol{x}_{k-i+1}) \tag{26}$$

where $\boldsymbol{M}_i^{opt}$ is the optimal multigram chosen among all multigrams of length i to minimize the distance $D([\boldsymbol{x}_{k-i+2} \ldots \boldsymbol{x}_{k+1}], \boldsymbol{M}_i)$ (see Eq. 21).

Similarly as for "classical" multigrams, we need a dictionary of code-multigrams with corresponding probabilities. We use an iterative algorithm too, but it differs slightly from that described in subsection 5.1. After an initialization (we will return to this step later), one iteration is composed of 3 steps:

- *Segmentation* which is found by maximizing the quantity $L'(\boldsymbol{X})$ (see Eq. 23).
- *Reestimation of probabilities* given for a multigram-code $\boldsymbol{M}$ by

  $$p(\boldsymbol{M}) = \frac{c(\boldsymbol{M})}{C} \tag{27}$$

  where $c(\boldsymbol{M})$ is the number of sequences represented by multigram $\boldsymbol{M}$ and C is the total number of sequences in the optimal segmentation. Using of reestimation formula with a pruning factor is also possible.
- *Recalculation of code-multigrams* - new multigram $\boldsymbol{M}$ is calculated as a centroid of all sequences represented by $\boldsymbol{M}$ in the optimal segmentation.

The initialization of such a dictionary is a crucial problem. Not only we have to choose the numbers of multigrams of different lengths and their initial values, but we must also initialize their probabilities. We have decided to initialize the dictionary using "classical" multigram dictionaries. We take a certain number of most probable multigrams, together with their probabilities, we search the corresponding code-vectors in the VQ-codebook, and we put them to the initial modified multigram-dictionary. The second problem is the choice of initial numbers of multigrams. During the iteration, these numbers can decrease (a probability appears to be 0 in the reestimation step, Eq. 27) but never increase, we did not include a cluster-splitting option in our algorithm. Two possibilities exist for the determination of number of multigrams. In the experimental part, we used fixed numbers: L (the size of VQ codebook) for the length 1 and $2L$ for the lengths 2 to n. Another possibility is to set the proportion of probabilities of "classical" multigrams we want to put in the modified multigram dictionary. For example, if we have n_i multigrams of length i in the "classical" dictionary, we choose the number of modified multigrams m_i to match the inequality

$$\frac{\sum_{j=1}^{m_i} p(M_{i,j})}{\sum_{j=1}^{n_i} p(M_{i,j})} \leq PR \tag{28}$$

where PR is the desired proportion.

6.3 Experimental results

We used the same strings of cepstral vectors as in the previous experiences (not quantified by VQ). We trained the modified multigram dictionary for the maximal length $n = 5$, for various dictionary

Mg. length	1	2	$\cdots$	n-1	n
Init. number	L	$2L$	$2L$	$2L$	$2L$

Table 4: Initial numbers of multigrams for modified multigram dictionary.

initializations and for different maximal distances D_{max}. The initialization of the modified multigram dictionary was done using pre-calculated classical multigrams and corresponding VQ codebooks. The initial numbres of multigrams for a codebook of L codevectors are given in Table 4.

The results are evaluated in terms of entropy per symbol (see Eq. 18), which is validated by evaluating the rate on the test string (see Eq. 20). The resulting dictionary size is no more an important criterion, as it can not exceed $L + 2(n-1)L$ multigrams (the total initial number). It is necessary to take into account the spectral distortion - in case of classical multigrams it is given by the distortion of VQ, for modified multigrams it necessitates to be evaluated.

Table 5 gives the results for the initialization by the VQ codebook $L = 128$, the initial numbers of multigrams were 128, 256, 256, 256, 256. The complete results for all initializations and maximal distances can be found on Figure 1, enclosed at the end of this document.

D_{max}	SD [dB]	1	2	3	4	5	$H'(\boldsymbol{M})$ [bit]	$R(\boldsymbol{M})$ [bit]
0.1	2.209	128	0	0	0	0	6.867	6.842
0.2	2.203	128	165	70	5	2	6.888	6.865
0.3	2.170	128	255	244	195	185	5.966	6.018
0.4	2.245	128	241	249	252	252	3.602	3.595
0.5	2.472	127	191	217	234	255	2.068	2.060
0.6	2.666	71	88	131	151	256	1.539	1.520
0.7	2.750	16	13	33	67	256	1.429	1.403
0.8	2.768	2	0	11	35	256	1.414	1.381
0.9	2.778	0	0	2	30	255	1.398	*
1.0	2.784	0	0	2	31	256	1.391	1.525

Table 5: Results of modified multigram method with $L = 128$ initialization: maximal distance, spectral distortion, numbers of multigrams of length 1...5 in the resulting dictionary, entropy of the dictionary, rate on the test string.

We see, that for all initializations we obtain interesting results – for certain D_{max}, the modified entropy significantly decreased and the spectral distance not much worsened than that of corresponding VQ used for the dictionary initialization. In addition, the dictionary size is limited and can be controlled by the initial choice of numbers of code-multigrams. However, the procedure is not completely without problems:

1. the choice of maximal distance D_{max} is critical for the resulting ratio spectral distortion/entropy. A good solution would be some kind of dynamical adjusting of this variable during the segmentation procedure. On the other hand, D_{max} can be used to change the above mentionned ratio which is not without interest in variable rate speech coders.

2. the computational load for the initializations with many code-multigrams is significant. We have to take into account, that for each symbol, we have to evaluate the distance to each multigram in the dictionary. Therefore we can not use as large and precise dictionary as we would have liked when defining this method.

6.4 Conclusion for the multigram methods

As we have already mentionned, the results when applying the "classical" multigrams to speech spectra sequences are not satisfactory. The method is not able to find a generalization of such sequences and for greater codebooks, it tends to symbol-by-symbol segmentation.

The modified method with distance notion gives significantly better results, summarized in previous section. With a limited dictionary size, this method is able to represent efficiently the spectral vectors with significantly less information and approximately the same distortion as a VQ. However, we have also mentionned the problems caused by the choice of initial dictionary and of parameters of the segmentation, and by the increased complexity of this algorithm. When we compare modified multigrams with VVVQ (see [4] for details), we find similarities between these two approaches although the ways were different: Chou obtained it by coupling the Matrix Quantization with Entropy Constrained VQ, in our case it is a multigram method combined with distance measures.

In the area of speech coding, the use of this method is in efficient coding of LPC-spectra for low rate coders. In order to reach a “transparent” spectrum quality, we can use this method on partial vectors (eg. of 4,3,3 cepstral coefficients) and use dictionaries of reasonable sizes to represent them. We have tested the method only on spectra but it should be implementable to other speech parameters (pitch or excitation vectors, gain) too.

As for recognition, the use of longer and less precise multigrams could lead to a set of robust units applicable in search of the correspondence of acoustical events and phoneme description. It is probable, that in this case a use of additional time alignment will have to be used. The work in this field is currently being done at ENST in Paris (especially by Sabine Deligne).

7 Remarks on spectral analysis for the segmental processing

This section as well as the following one designate the topics for the continuation of the PhD thesis. At the moment, the author does not have a consistent knowledge of the problems discussed and asks better informed readers for critical remarks and corrections.

The previous sections shew the interesting potential of the segmental speech processing. However, there is one major contradiction in these methods: first we perform the spectral analysis on *fixed* length frames, then we try to group them together to create *variable* length sements. Therefore, the basic temporal unit is still the fixed length frame, while we know that the speech events do not at all posses this synchronization. This problem is reflected also in the domain of speech recognition – the signal is usually analysed on the basis of fixed length segments and then a less or more sophisticated method of time alignment is employed (Dynamic Time Warping, Hidden Markov Models, etc.). The problems of paying little attention to the temporal aspects of speech were discussed for ex. by Hervé Bourlard in [3].

It seems to us that a spectral analysis with variable length frames could be a great contribution for the segmental methods. A candidate for this anaysis is *Wavelet Transform* (WT), defined for the continuous time by:

$$X_{WT}(a,b) = |a|^{-1/2} \int_{-\infty}^{\infty} x(t)\psi\left(\frac{t-b}{a}\right) dt \tag{29}$$

where the function $\psi(t)$ is called *mother wavelet* and b and a are called shifting and scaling factor respectively. On the contrary to widely used Fourier transform with window of the same length for all frequencies, the wavelet transform adjusts the size of the window according to the frequency and presents better localization in the time-frequency plane. The information about WT can be found for ex. in [16] or in the Daubechies book [5].

There are however important questions concerning the using of WT in segmental speech processing:

- wavelet construction and computational aspects of Discrete WT. For the speech coding, this transform must be reversible, it is necessary to reconstruct the signal in the decoder.
- matching of wavelets with the source-filter modelling of speech. For the speech processing, this scheme presents evidently an advantage in comparison to other types of signal. The basic WT definition does not exploit it.
- searching of speech segments for this type of transform. Having lost the frame notion, it will be no more possible to use multigram-based methods. This implicates a need of other segmentation scheme.

8 Human perception and digital speech processing

The aspects of human perception in the devices for automatic speech processing have been discussed since the beginning of existence of digital speech processing (eg. by Leipp [9] in 1977). While essential efforts were done in the understanding of speech production and in application of this work to digital speech processing (sophisticated articulatory tract models, synthesis, ...) significantly less was done in the human perception investigation.

8.1 Frequential and temporal masking

The psychoacoustic tests (eg. [18]) shew that the human ear does not detect arbitrarily short events both in the frequency and in the time domain. It is known that when a strong sound appears at certain frequency, we can plot a "masking curve" and that the weak components below this curve are not audible. More sophisticated masking curve can be calculated for the entire frequency band of speech – this approach is used in the MPEG-AUDIO coder and newly it was applicated also to low rate CELP based coding [12].

The temporal masking (as far as we know, very rarely, if ever, used in coding or recognition) is similar to above mentioned effect: if a strong sound appears at time t, the events appearing in the interval $t + \Delta$ will not be perceived (Δ is a variable depending on the masking sound, on individual abilities, etc.).

8.2 Auditory models

Physically, the human ear is well described [9, 18]. On the other other hand, we almost completely ignore the "high level" processing in the human brain. However, also the "low level" modelization apports interesting possibilities to the digital speech processing. Ghitza [7] modelizes the cochlea (a part of inner ear) by a set of filters and gives the models of "firing" of auditory cells to the cochlear nerve. The Ensemble Interval Histogram (EIH) is evaluated by counting the distances of successive firings (*spikes*). It is shown that while at low frequencies the firing is phase locked with the excitation sound, in high frequencies it is the instantaneous rate of firing which allows the detection of brief temporal events. Ghitza also presents the differences of temporal windows on which the ear performs the integration: for low frequences it tends to be long, for high frequences the windows are shorter. This corresponds with the Wavelets (Section 7) and justifies their using in the speech processing.

9 Conclusion

It is obvious, that the segmental approachs can improve the performances of many currently used speech processing schemes. The multigram method is the example of these changes – while using a principally simple mathematical method, we obtain more efficient spectrum representation in comparison to widely used Vector Quantization. The segmental approach is familiar with the human "processing" of speech: it is very unlikely, that we divide the perceived speech into fixed length frames processed independently... The current development in speech processing shows the importance of methods based on the human perception. However, it is applicable to standard "frame" techniques only with the greatest difficulties. It is not exaggerated to say that the essential efforts in the next decades will be done in three fields cited in this paper:

- segmental processing
- new methods of speech analysis
- perceptive approach

It must be mentioned with great insistance that these three points are not exclusive and that the most interesting and probably most efficient will be their combination.

As for the organizational aspects of this "international" PhD., I must say that despite some problems (administrative, adaptation to new environment, lack of family life, ...) there can not be better experience for a young man preparing for the research/education career. In France, and especially in the Parisien region, one has to choose among the number of interesting conferences, seminars, presentations, etc. This fruitful collaboration will certainly not end with the final PhD. presentation, I hope it will be kept and continue to be useful for both sides.

References

[1] F. Bimbot, R. Pieraccini, E. Levin, and B. Atal. Modèles de séquences à horizon variable: Multigrams. In *Proc. XX-èmes Journées d'Etude sur la Parole*, pages 467–472, Trégastel, France, June 1994.

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

[3] H. Bourlard. Nouveaux paradigmes pour la reconnaissance robuste de la parole. In *Proc. XXI-èmes Journées d'Etude sur la Parole*, pages 263–272, Avignon, France, June 1996.

[4] 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.

[5] I. Daubechies. *Ten Lectures on Wavelets.* Society for industrial and applied mathematics, Philadelphia, Pennsylvania, 1992.

[6] S. Deligne and F. Bimbot. Language modelling by variable length sequences: Theoretical formulation and evaluation of multigrams. In *Proc. IEEE ICASSP 95*, pages 169–172, Detroit, USA, 1995.

[7] O. Ghitza. Auditory models and human performance in tasks related to speech coding and speech recognition. *IEEE Trans. Speech and Audio Processing*, 2(1, part II):115–132, January 1994.

[8] N. S. Jayant and P. Noll. *Digital coding of waveforms.* Prentice Hall, New Jersey, 1984.

[9] E. Leipp. *La machine à écouter, essai de psycho-acoustique.* Masson, Paris, 1977.

[10] J. M. Lopez-Soler and N. Farvardin. A combined quantization-interpolation scheme for very low bit rate coding of speech LSP parameters. In *Proc. IEEE ICASSP 93*, pages II–21–24, Minneapolis, 1993.

[11] L. R. Rabiner and L. W. Schaeffer. *Digital processing of speech signals.* Prentice Hall, 1978.

[12] D. Sen and W. H. Holmes. Perceptual enhancement of CELP speech coders. In *Proc. IEEE ICASSP 94*, pages II–105–108, Adelaide, 1994.

[13] Y. Shiraki and M. Honda. LPC speech coding based on variable length segment quantization. *IEEE Trans. Acoust., Speech, Signal Processing*, 36(9):1437–1444, September 1988.

[14] J. Černocký. *Sub band speech coder, Diploma project.* ESIEE Paris, July 1995.

[15] J. Černocký, G. Baudoin, and G. Chollet. Efficient method of speech spectrum description using multigrams. In *International association of pattern recognition (IAPR) Workshop*, University of Ljubljan, Slovenia, April 1996. presented, but proceedings not yet published.

[16] V. Veselý. Wavelety a časově frekvenční analýza dat. In *Proc. ANALÝZA DAT 95/II*, Bohdaneč u Pardubic, 1995.

[17] S. Wang and A. Gersho. Phonetic segmentation for low rate speech coding. In Atal, Cuperman, and Gersho, editors, *Advances in speech coding*, pages 225–234. Kluwer Academic Publishers, 1991.

[18] xxx. *La parole et son traitement automatique*, chapter Perception auditive et perception de parole, pages 147–214. Masson, Paris, Calliope (CNET, ENST) edition, 1989.