Exploiting Temporal Correlation of Speech for Error Robust and Bandwidth Flexible Distributed Speech Recognition

Zheng-Hua Tan, *Senior Member, IEEE*, Paul Dalsgaard, *Senior Member, IEEE*, and Børge Lindberg, *Member, IEEE*

Abstract—In this paper, the temporal correlation of speech is exploited in front-end feature extraction, client-based error recovery, and server-based error concealment (EC) for distributed speech recognition. First, the paper investigates a half frame rate (HFR) front-end that uses double frame shifting at the client side. At the server side, each HFR feature vector is duplicated to construct a full frame rate (FFR) feature sequence. This HFR front-end gives comparable performance to the FFR front-end but contains only half the FFR features. Second, different arrangements of the other half of the FFR features creates a set of error recovery techniques encompassing multiple description coding and interleaving schemes where interleaving has the advantage of not introducing a delay when there are no transmission errors. Third, a subvector-based EC technique is presented where error detection and concealment is conducted at the subvector level as opposed to conventional techniques where an entire vector is replaced even though only a single bit error occurs. The subvector EC is further combined with weighted Viterbi decoding. Encouraging recognition results are observed for the proposed techniques. Lastly, to understand the effects of applying various EC techniques, this paper introduces three approaches consisting of speech feature, dynamic programming distance, and hidden Markov model state duration comparison.

Index Terms—Distributed speech recognition (DSR), error concealment (EC), error recovery, low bit-rate, split vector quantization (SVQ).

I. INTRODUCTION

AIMED at optimal performance of automatic speech recognition (ASR) over mobile communication networks, an important research topic within ASR has been to focus on the issue of distributed speech recognition (DSR) [1]. In the client–server architecture, a DSR system splits ASR processing into two parts, the client-based front-end feature extraction and the server-based back-end recognition, where data transmission between the two parts may take place via heterogeneous networks. However, the transmission of data across networks presents a number of challenges to ASR research, e.g., bandwidth limitations and transmission errors. As a consequence,

considerable efforts have been made ranging from front-end processing, source coding/decoding, channel coding/decoding, packetization to error concealment (EC) aimed at maintaining ASR performance in the distributed environments with adverse transmission channels [2].

Since the mel-frequency cepstral coefficient (MFCC) features are extensively used and have proved to be successful for ASR, MFCCs are used for most DSR front-ends. In general, the goal of source coding is to compress information aiming at a low bit-rate. One common class of source coding schemes for DSR applies split vector quantization (SVQ) [3] for the coding of ASR features in addition to the recently introduced transform coding such as the discrete cosine transform to pursue a very low bit-rate [4], [5]. Source coding can also be applied for achieving error resistance for example multiple description coding (MDC) and layer coding, which are also considered as a joint source and channel coding [2].

Channel coding techniques attempt to protect information from transmission errors. Linear block codes and a soft decision decoding are introduced in [6], and Reed–Solomon coding is applied in [7]. For packetization, a number of interleaving schemes is investigated in [8] to handle burst-like packet losses. All these client-based error recovery techniques are able to recover a large amount of transmission errors, however, at such cost as additional delay, increased bandwidth, and higher computational overhead [9].

In this paper, front-end processing, source and channel coding, and packetization are investigated aimed at low bit-rate and high error robustness at the same time minimizing additional cost. Specifically, a half frame rate (HFR) front-end with feature duplication is presented and extensively analyzed. This has the advantages of both low bit-rate and reduced computations which are opposed to source coding where additional computations are required. The effectiveness of the HFR front-end further motivates the introduction of a set of client-based error recovery techniques including MDC and interleaving.

In general, the deficiencies of client-based error recovery techniques may be avoided by applying EC techniques which exploit signal redundancy at the server side. Feature domain EC techniques attempt to generate substitutions for the erroneous/lost frames as close to the original as possible to improve recognition accuracy. The commonly used EC techniques include substitution [7], repetition [10], interpolation [11], [12], and splicing [13] where the erroneous frames are simply dropped. A partial splicing scheme [14] substitutes lost/erro-

neous frames partly by a repetition of neighboring frames and partly by a splicing. Under certain assumptions, the partial splicing scheme is equivalent to a weighted Viterbi decoding (WVD).

All the feature domain EC techniques previously referenced share the common characteristic of conducting EC at the vector level. A vector—equivalent to a frame—is regarded as the target unit. However, it is very likely that not all subvectors in an erroneous vector are erroneous. To exploit the existing error-free information embedded in each erroneous vector, this paper proposes a subvector-based EC technique where each subvector is considered as a supplementary element for error detection and mitigation. This is achieved by exploiting the temporal correlation present in the speech features to identify and replace inconsistent subvectors within erroneous vectors.

Different from feature domain techniques, model domain EC schemes introduce exponential weighting factors into the calculation of the observation probability by applying WVD such that contributions made by features or feature vectors with low reliability are decreased [6], [15], [16]. Weighting factors may be computed from the bit reliability information given by the network channel decoder which, however, is not often feasible [17]. The proposed subvector EC technique potentially retains or creates unreliable features but automatically generates a reliability measure for each feature during the subvector EC process. A WVD scheme is therefore introduced and combined with the subvector EC technique.

The paper is organized as follows. Section II describes the European Telecommunications Standards Institute (ETSI)-DSR standard and the motivation for this paper. Sections III and IV present the HFR front-end and the client-based error recovery techniques, respectively. The subvector-based EC technique and its combination with WVD are presented in Section V. Experimental evaluations and discussions are given in Section VI. Section VII provides a number of comparative studies. Conclusions are presented in Section VIII.

II. BACKGROUND AND MOTIVATION

Incorporating ASR technology into mobile networks is currently done in one of three scenarios [3], [6], [18]. In the first, ASR is conducted in the client only. Such fully client-embedded ASR has the advantage of not introducing extra distortion to the speech signal. However, the requirements to the client are high in terms of computing, memory, and power consumption. This has inspired the development of high-speed, low-resource ASR techniques for mobile devices [19]. In the second scenario, the client transmits the encoded speech to the server where speech is resynthesized, features are extracted and recognition is subsequently performed. In this scenario, a low bit-rate speech coder may cause significant degradations in recognition performance [20]. The feature set may, however, also be estimated directly from the coded speech bitstream without reconstructing the speech [13], [18], [21]. In the third scenario (the DSR setup), speech features suitable for recognition are calculated, quantized, and encoded in the client and transmitted to the server where they are decoded, submitted to a suitable EC technique, and handled by a recognizer. The DSR scenario provides a good tradeoff between bit-rate and recognition
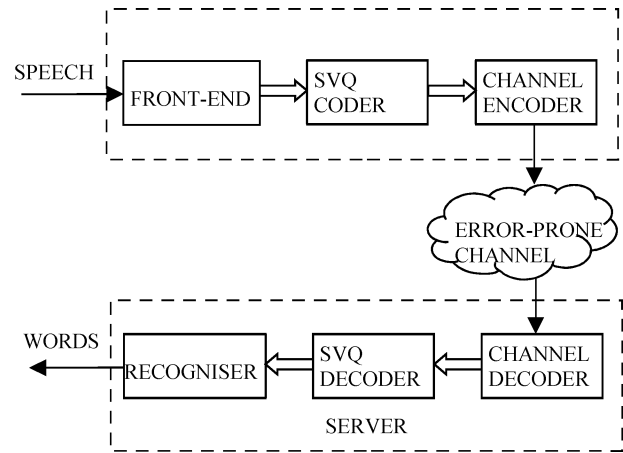

Fig. 1. Block diagram for DSR system.

accuracy [6]. A block diagram for a DSR system is illustrated in Fig. 1. The client includes the following three modules: front-end feature extraction, SVQ, and channel encoder, while the server back-end comprises the channel decoder, the SVQ decoder, and the recognizer.

A. ETSI-DSR Standard

The ETSI-DSR standard defines the feature extraction front-end processing together with an encoding scheme [10]. The front-end produces a 14-element vector consisting of log energy ($\mathrm{Log}E$) in addition to 13 MFCC coefficients ranging from c_0 to c_{12}—computed every 10 ms. Feature compression uses an SVQ algorithm that groups two features (either $\{c_i$ and $c_{i+1}, i = 1, 3, \ldots, 11\}$ or $\{c_0$ and $\mathrm{Log}E\}$) into a feature pair subvector resulting in seven subvectors in one vector. Each subvector is quantized using its own SVQ codebook. The size of each codebook is 64 (6 bits) for $\{c_i$ and $c_{i+1}\}$ and 256 (8 bits) for $\{c_0$ and $\mathrm{Log}E\}$, resulting in a total of 44 bits for each vector. Before transmission, two quantized frames (vectors) are grouped together creating a frame pair. A 4-bit cyclic redundancy check (CRC) is calculated for each frame pair and appended, resulting in 92 bits for each frame pair. Twelve frame pairs are combined to form an 1104-bit feature stream. By adding the overhead bits of a synchronization sequence and a header, each multiframe is represented by 1152 bits to represent 240 ms speech, corresponding to a bit-rate of 4800 b/s.

Over error-prone channels, the bitstream received at the server may have been contaminated by errors. Two methods are applied to determine if a frame pair is received with errors, namely a CRC checksum test and a vector consistency test. The vector consistency test determines whether or not the decoded features for each of the two consecutive feature vectors in a frame pair have a minimal continuity. A frame pair is labeled as erroneous when its CRC is detected as incorrect. The vector consistency test is applied to the frame pair received before the one failing the CRC test and to the frame pairs following. The preceding frame pair is also classified as erroneous if the vector consistency test then fails. Frame pairs following are classified as erroneous until one frame pair is received with a correct CRC and meets the vector consistency requirement.

TABLE I
ERROR RATES AND BERs OF FRAME PAIR, ONE-FRAME (VECTORS),
AND SUBVECTORS VERSUS GSM EPs

GSM EPs	C/I ratios (dB)	%BER	% Error rate			
			Frame Pairs (92 bits)	Vectors (48 bits)	Sub-vectors	
					$[c_i, c_{i+1}]$, i=1,3...11 (6 bits)	$[c_0, \log E]$ (8 bits)
EP1	10	0.0049	0.16	0.06	0.02	0.03
EP2	7	0.18	3.08	1.88	0.59	0.68
EP3	4	3.55	30.7	22.2	10.2	11.1

In the ETSI-DSR EC scheme, repetition is applied to replace erroneous vectors. It is shown in [22] that a poor channel, e.g., having 4-dB carrier to interference (C/I) ratio, still severely reduces the accuracy of speech recognition using the implementation of the ETSI-DSR standard.

B. Effect of Transmission Errors

One problem of employing the ETSI-DSR frame pair format is that two entire vectors in a frame pair will be in error and substituted even though only a single bit error occurs in the 92-bit frame pair. This has motivated the introduction of a one-frame-based error protection [23], in which each frame is protected by its own 4-bit CRC, generating a 48-bit one-frame. The one-frame scheme causes the overall probability of one frame in error to be lower at the cost of only a marginal increase in bit-rate, from 4800 to 5000 b/s.

This concerns the problem of the data block size (measured in bits) for error detection and concealment. Table I shows error rates and bit error rates (BER) calculated on the basis of the Global System for Mobile communication (GSM) error patterns (EP). The GSM EPs are often used as they are more realistic by representing a merge of both random and burst-like errors as opposed to artificially created test data.

Table I shows that error rates of subvectors are significantly lower than error rates of vectors. Consequently, it may be advantageous to exploit the existing error-free subvectors in erroneous vectors rather than simply neglecting them. Section V focuses on the detection, extraction, and exploitation of error-free subvectors on the basis of existing redundancy within speech features.

C. Temporal Redundancy in Speech Features

Temporal correlation and redundancy exist in the speech feature stream due to both the overlapping in feature extraction processing and the speech production process itself. The redundancy makes ASR resistant to random transmission errors but vulnerable to burst-like errors. In [8], recognition experiments show that the baseline ASR accuracy can be maintained at a frame loss rate of 50%, provided that the average burst length is short. Due to the high correlation between consecutive speech frames, the odd and even numbered frames carry almost the same information. In [24], it is experimentally shown that increasing frame shift from 10 ms up to 17.5 ms even gives higher recognition accuracy and that 22.5-ms frame shift still gives comparable accuracy to the 10-ms full frame rate (FFR).

This motivates the investigation of HFR front-end and a number of client-based error recovery techniques for FFR front-end by arranging the odd and even numbered frames in different ways as presented in the following two sections.

FFR	c_i^{t-2}	c_i^{t-1}	c_i^t	c_i^{t+1}	c_i^{t+2}
HFR-Duplication - Case 1	c_i^{t-2}	c_i^{t-2}	c_i^t	c_i^t	c_i^{t+2}
HFR-Duplication - Case 2	c_i^{t-3}	c_i^{t-1}	c_i^{t-1}	c_i^{t+1}	c_i^{t+1}

Fig. 2. Comparison of data used for calculating delta features by FFR and by HFR-Duplication.

III. HALF FRAME RATE FRONT-END

The commonly used front-end processing computes the speech features using a 25-ms frame length and a 10-ms frame shift, resulting in a 15-ms overlap between consecutive frames. In the HFR front-end a 20-ms frame shift is used resulting in a 5-ms overlap. At the server side and prior to recognition, each HFR feature vector is duplicated to reconstruct the FFR vector equivalent (called HFR-duplication), and no modifications are introduced to the back-end recognizer. This is similar to the HFR method briefly presented in [25] where, however, linear interpolation is used to generate the equivalent FFR vector. The work presented here applies a repetition because of the superior performance of the repetition scheme shown in Sections VI and VII. In addition to providing a low bit-rate feature stream for DSR, the HFR front-end has the advantage of only requiring half the computational cost in its feature extraction process. This may be a significant advantage for capacity limited devices in terms of computing power and battery life. This is opposed to source coding that also achieves low bit-rate but at the cost of introducing additional computations.

A. Delta and Delta–Delta Feature Analysis for HFR Front-End

The delta and delta-delta features are calculated at the server side on the basis of the reconstructed FFR vectors. Therefore, the effect of the HFR-based reconstruction of the static features on the delta and delta-delta features is analyzed in the following. In HTK [26] the delta features are calculated according to

$$d_i^t = \frac{(c_i^{t+1} - c_i^{t-1}) + 2(c_i^{t+2} - c_i^{t-2})}{10} \quad (1)$$

where d_i^t ($i = 0, 1, \ldots, 12$) is the ith delta feature in frame t. Formula (1) shows that two proceeding and two following static features are used in the calculation, as illustrated in the first row of Fig. 2.

When calculating delta features from HFR-Duplication static features, two cases occur as shown in the second and third row of Fig. 2. The corresponding delta features for Case 1 and Case 2 are, respectively, calculated as

$$d_i^t(1) = \frac{(c_i^t - c_i^{t-2}) + 2(c_i^{t+2} - c_i^{t-2})}{10} \quad (2)$$

and

$$d_i^t(2) = \frac{(c_i^{t+1} - c_i^{t-1}) + 2(c_i^{t+1} - c_i^{t-3})}{10}. \quad (3)$$

The difference between $d_i^t(1)$ and d_i^t is

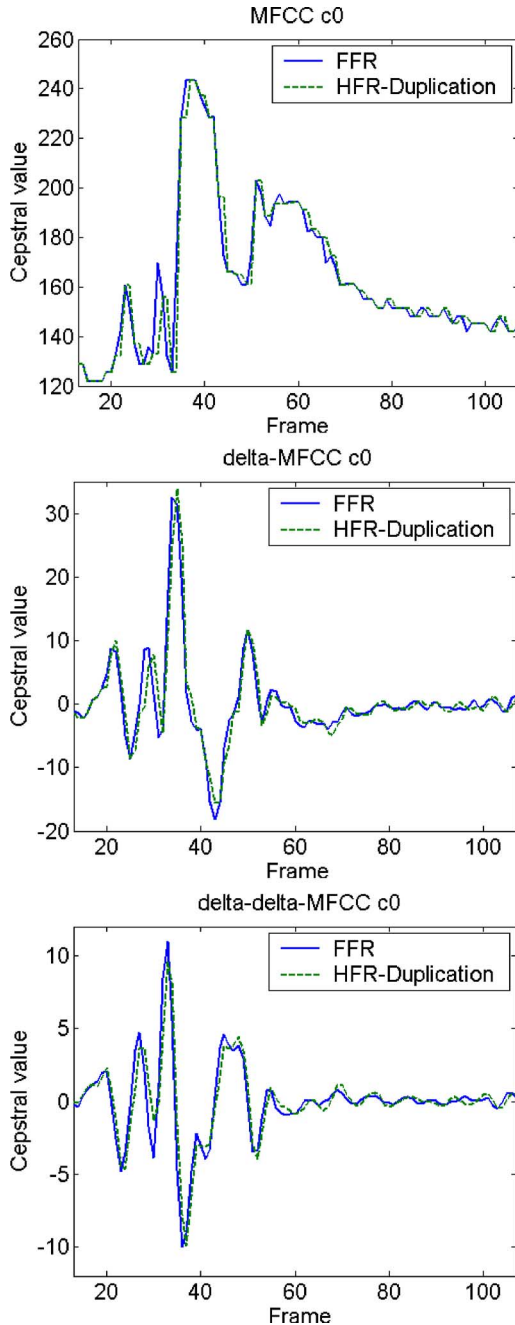$$d_i^t(1) - d_i^t = \frac{(c_i^{t-1} - c_i^{t-2}) - (c_i^{t+1} - c_i^t)}{10} \quad (4)$$

Fig. 3. Comparison of static, delta, and delta-delta features obtained from FFR and HFR-Duplication.

and the difference between $d_i^t(2)$ and d_i^t is

$$d_i^t(2) - d_i^t = \frac{(c_i^{t-2} - c_i^{t-3}) - (c_i^{t+2} - c_i^{t+1})}{5}. \tag{5}$$

Formulas (4) and (5) show that the differences are approximately equal to the second-order differentiation and thus have small values, indicating that the influence on the delta features from applying frame duplication is marginal. Similar relationships exist between the delta and delta-delta features.

Further, this marginal influence is evidenced by a visual comparison of FFR and HFR-Duplication features. Fig. 3 shows data taken from an utterance for the Danish word "et." The MFCC
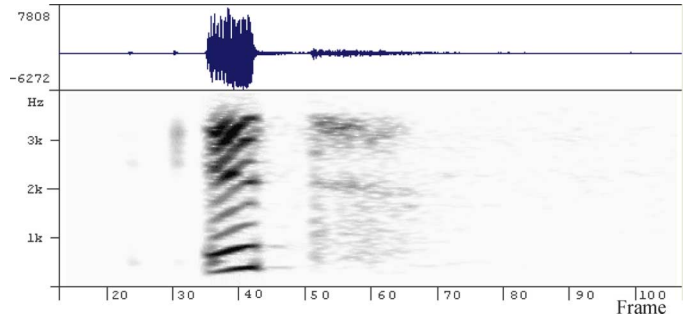

Fig. 4. Waveform and spectrogram of the Danish word "et."

TABLE II
PERCENT WER ACROSS THE FRONT-ENDS FOR DANISH DIGITS AND CITY NAMES USING FFR-BASED TRIPHONE MODELS WITHOUT QUANTIZATION

Front-end	WER (%) on test set	
	Danish digits	City names
FFR	0.21	20.71
HFR-Duplication	0.41	20.71
HFR-NoDuplication	3.32	38.75

coefficient c_0 is especially chosen due to its capacity of emphasizing transitions between vowels and consonants. The left graph shows the c_0 feature calculated from FFR data and corresponding data for the HFR-Duplication. The middle and the right graphs show similar analyses for the delta and delta-delta features.

Fig. 3 shows that all the HFR-Duplication features closely trace the corresponding FFR features even in the transient regions. Fig. 4 shows the waveform and spectrogram of the word "et" and that frames 32 and 48 are approximately labeled as a silence/vowel boundary and a vowel/consonant boundary, respectively.

B. HFR Front-End With and Without Feature Duplication

The performance of the HFR front-end is evaluated on the two databases the Danish SpeechDat (II) [27] and Aurora 2 [28]. The experiments in this subsection are all conducted without transmission errors. The speech features are standard MFCC with 13 static coefficients and their delta and delta-delta features, resulting in a total of 39 coefficients.

1) Danish SpeechDat (II): The SpeechDat (II) compatible database DA-FDB 4000 comprises speech from 4000 Danish speakers collected over the fixed telephone. A part of the database is used for training 32 Gaussian mixture triphone models based on the SpeechDat/COST 249 reference recognizer. Two subdatabases have been used as test data namely the Danish digits (vocabulary size = 11) and the city names (vocabulary size = 449).

The triphone models used in this experiment are all trained using the FFR features directly and without quantization. The features for the test data, however, are all calculated on the basis of quantized data. The results for the Danish digits and city names tasks for the HFR and FFR front-ends are shown in Table II. It is seen that the HFR front-end with duplication achieves results close to the FFR front-end for both tasks. However, using the HFR features without duplication gives substantially higher word error rate (WER).

TABLE III
PERCENT WER ACROSS THE FRONT-ENDS FOR TEST SET A USING
FFR-BASED 16-STATE MODELS WITHOUT QUANTIZATION

Front-end	WER (%) on test set				
	Clean1	Clean2	Clean3	Clean4	Average
FFR	1.14	1.00	0.92	0.77	0.95
HFR-Duplication	1.07	1.03	1.01	0.96	1.02
HFR-NoDuplication	29.87	28.08	28.15	29.44	28.88

TABLE IV
PERCENT WER ACROSS THE FRONT-ENDS FOR TEST SET A USING
MATCHED 16-STATE MODELS AFTER QUANTIZATION

Front-end	WER (%) on test set				
	Clean1	Clean2	Clean3	Clean4	Average
FFR	1.04	0.97	1.10	0.89	1.00
HFR-Duplication	1.10	0.97	1.10	0.89	1.02
HFR-NoDuplication	11.08	10.28	10.44	10.71	10.63

2) Aurora 2: The Aurora 2 database is the TI digit database artificially distorted by adding noise and using a simulated channel distortion. Whole word models are created for all digits using the HTK recognizer. Each of the whole word digit models has 16 HMM states with three Gaussian mixtures per state. The silence model has three HMM states with six Gaussian mixtures per state. A one state short pause model is tied to the second state of the silence model.

The word models used in the experiments are trained on clean speech data. The test data are the clean data from Test Set A. The models are trained on FFR features without quantization whereas the test data are quantized. Table III shows the WERs for Test Set A across three same front-ends as applied in Table II. Again, the HFR front-end with feature duplication demonstrates comparable WERs to the FFR although the models are trained using the FFR features, whereas HFR-NoDuplication gives significantly higher WERs.

Table IV shows results for experiments in which matched training and test models are used. Here, the features are the same for both training and test and quantization is applied in their calculation. The results show that the HFR front-end with feature duplication gives results close to the FFR front-end. The WER of HFR-NoDuplication is still substantially higher although both training and test features are matched, demonstrating that duplication of each HFR feature vector is critical even when using matched models. An explanation to the observed high WERs for HFR-NoDuplication is that the number of its feature vectors does not match the number of HMM states.

C. Duplication of Features Versus Number of HMM States

To verify the above explanation, a set of further experiments are conducted by using HMM models with eight states instead of 16 states for the Aurora 2 task. Table V shows the results for eight-state HMM models for the three front-ends.

By comparing the results in Tables IV and V, it is found that the performance change of FFR and HFR-NoDuplication goes to two completely different directions when the number of states is halved. A significant increase in the average WER (from 1.00% to 6.30%) is seen for FFR whereas the average WER for HFR-NoDuplication decreases substantially (from 10.63% to 1.40%) with results that are close to the average WER for

TABLE V
PERCENT WER ACROSS THE FRONT-ENDS FOR TEST SET A USING
MATCHED EIGHT-STATE MODELS AFTER QUANTIZATION

Front-end	WER (%) on test set				
	Clean1	Clean2	Clean3	Clean4	Average
FFR	6.39	6.35	5.99	6.48	6.30
HFR-Duplication	5.74	6.41	5.40	5.83	5.84
HFR-NoDuplication	1.23	1.75	1.64	0.99	1.40

FFR using 16-state models (1.00%). However, the performance of HFR-Duplication is still close to the FFR, indicating a good reconstruction of the equivalent FFR features. The results underline the correlation between the number of states of the back-end models and the frame rate used in the front-end.

IV. CLIENT-BASED ERROR RECOVERY TECHNIQUES

The HFR front-end offers a low bit-rate and a comparable performance to the FFR front-end when there are no transmission errors. This provides an effective alternative to the FFR front-end if the available bandwidth is restricted. Based on the HFR front-end concept, an adaptive multiframe rate scheme can furthermore be implemented in which the DSR system is enabled to adaptively switch between HFR and FFR front-ends based on the quality of the transmission channel [29]. Such switching between the HFR and FFR front-ends results in a bandwidth flexible DSR codec, and there is no requirement for switching back-end HMM models.

However, due to the fact that less redundant information is available, the HFR features are likely to be more sensitive to transmission errors. The error robustness of the HFR front-end can be improved by applying channel coding techniques (such as Reed–Solomon coding and linear block coding) that deliberately add redundancy into the speech source. Since the HFR front-end achieves a low bit-rate for the source, it allows more bits for channel coding. This is similar to the adaptive multirate (AMR) speech codec [30], where more bandwidth is available for channel coding when the speech source has a low bit-rate. As opposed to adding redundancy as normally done in typical channel coding techniques, the FFR front-end can be considered as a channel coding to the HFR front-end, using half of the FFR features as redundant information. Since the HFR feature frames are simply the odd numbered frames in the FFR front-end, the even numbered frames are redundant information that can be arranged in different ways in the process of source coding and packetization.

A. MDC

MDC is an error-resistant source coding technique. The technique encodes the signal source into substreams (descriptions) of equal importance in the sense that each description can independently reproduce the original signal into some basic quality [31]. The quality of the reconstructed signal incrementally increases when more descriptions are received. In the FFR front-end, the odd numbered vectors together with the even-numbered vectors create two descriptions, and each of them may be transmitted independently, resulting in an MDC coding scheme exploiting channel diversity. If one description is received without errors, in this two-description MDC coding, there is principally no need to wait for, or exploit, the other description.
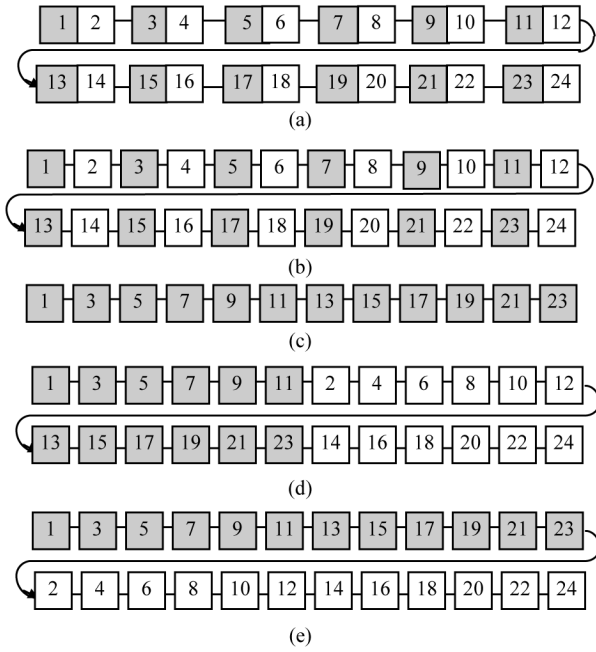
Fig. 5.   HFR and four different FFR coding schemes. (a) ETSI-DSR FFR-based frame pair scheme. (b) FFR-based one-frame scheme. (c) HFR scheme. (d) FFR-based Interleaving12 scheme. (e) FFR-based Interleaving24 scheme.

B. Interleaving

Interleaving techniques counteract the effect of burst errors normally at the cost of delay. Specifically, interleaving rearranges the ordering of a sequence of frames in order to disperse burst errors for efficient error recovery and concealment [32], [8]. At the server, the counterpart de-interleaving restores the sequence to its original order. A common way to implement interleaving is to divide symbol sequences into blocks corresponding to a two-dimensional array and to read symbols in by rows and out by columns.

This paper presents an interleaving scheme that manages the ordering of odd- and even-numbered frames. Specifically, a chosen number of odd-numbered frames may be concatenated and transmitted first and followed by their corresponding even-numbered frames. The characteristic difference between conventional interleaving and this special interleaving scheme is that the latter may offer less or no overall transmission delay dependent on whether the transmission has caused errors or not. In the case of no errors, the odd-numbered feature vectors can be used immediately to reconstruct the equivalent FFR feature vectors by duplication without causing any delay.

C. Multiframe Structures

Fig. 5(a) and (b) shows that the 24-frame multiframe structures for the FFR-based ETSI-DSR frame pair scheme [10] and FFR-based one-frame scheme [23], respectively. The HFR scheme, however, encompasses only twelve frames in each multiframe as shown in Fig. 5(c). Each frame is protected by a 4-bit CRC resulting in a 48-bit frame and 12 frames are joined and appended with overhead bits resulting in a 624-bit multiframe, concatenated into a 2600 b/s bit-rate (contrasting the 4800 b/s for the ETSI-DSR FFR-based scheme). The same bit-rate of 2600 b/s is obtained without observing degradation in

ASR performance, however, by quantizing FFR-based feature vectors in [3], [7]. In the decoding at the server side, the CRC is used as the only error detection method, and no vector consistency test is conducted for the HFR scheme. Fig. 5(d) and (e) illustrates two FFR-based interleaving schemes designated as "Interleaving12" and "Interleaving24" grouping a sequence of 12 vectors and a sequence of 24 vectors into one block, respectively, and for each block the odd-numbered features are concatenated and transmitted first and their corresponding even-numbered features transmitted later.

V. SUBVECTOR-BASED EC AND ITS COMBINATION WITH WVD

This section presents the detection, extraction, and exploitation of error-free subvectors. Since there is no CRC like coding applied (or error checking bits allocated) at the subvector level, error detection at this level can only make use of a subvector consistency test which relies on the temporal correlation existing in the speech features.

A. Subvector-Based EC

Given that t denotes the frame number and V^t the feature vector, V^t is formatted as

$$
\begin{aligned}
\boldsymbol{V}^t &= [c_1^t, c_2^t \ldots c_{12}^t, c_0^t, \log E^t]^T \\
&= [[c_1^t, c_2^t] \ldots [c_{11}^t, c_{12}^t], [c_0^t, \log E^t]]^T \\
&= [[\mathrm{S}_0^t]^T, [\mathrm{S}_1^t]^T \ldots [\mathrm{S}_6^t]^T]^T
\end{aligned}
\tag{6}
$$

where $\boldsymbol{S}_j^t$ ($j = 0, 1, \ldots, 6$) denotes the jth subvector in frame t. Two consecutive frames in a frame pair are represented by $[\boldsymbol{V}^t, \boldsymbol{V}^{t+1}]$. The consistency test is conducted within the frame pair so that each subvector $\boldsymbol{S}_j^t$ from $\boldsymbol{V}^t$ is compared with its corresponding subvector $\boldsymbol{S}_j^{t+1}$ from $\boldsymbol{V}^{t+1}$ to evaluate the consistency of the two subvectors. If any of the two decoded features in a feature pair (subvector) does not possess a minimal continuity, the subvector is classified as inconsistent. Specifically, subvectors $\boldsymbol{S}_j^t$ and $\boldsymbol{S}_j^{t+1}$ in a frame pair are classified as inconsistent if

$$
(d(S_j^{t+1}(0) - S_j^t(0)) > T_j(0)) \text{ OR } (d(S_j^{t+1}(1) - S_j^t(1)) > T_j(1))
\tag{7}
$$

where $d(x, y) = |x - y|$ and $S_j^t(0)$ and $S_j^{t+1}(0)$ and $S_j^t(1)$ and $S_j^{t+1}(1)$ are the first and second element, respectively, in the subvectors $\boldsymbol{S}_j^t$ and $\boldsymbol{S}_j^{t+1}$ as given in (6); otherwise, they are consistent. Thresholds $T_j(0)$ and $T_j(1)$ are constants based on measuring the statistics of error-free speech features.

Assuming there are $2N$ frames (N frame pairs) in error to be mitigated, using the notation A for the last error-free frame t_A and B for the first following error-free frame t_B, the ETSI-DSR buffered vectors are $[\boldsymbol{V}^A, \boldsymbol{V}^{A+1}, \boldsymbol{V}^{A+2}, \ldots, \boldsymbol{V}^{A+2N-1}, \boldsymbol{V}^{A+2N}, \boldsymbol{V}^B]$, as illustrated in Fig. 6 at the subvector level.

In the above buffering matrix, columns $\boldsymbol{V}^A$ and $\boldsymbol{V}^B$ are the error-free vectors with $2N$ erroneous vectors received in between. The $2N$ vectors $\boldsymbol{V}^{A+1} \ldots \boldsymbol{V}^{A+2N}$ are all identified as

$$\begin{array}{cccccccc} \boldsymbol{V}^A & \boldsymbol{V}^{A+1} & \boldsymbol{V}^{A+2} & . & \boldsymbol{V}^{A+2N-1} & \boldsymbol{V}^{A+2N} & \boldsymbol{V}^B \end{array}$$

$$\begin{bmatrix} S_0^A & S_0^{A+1} & S_0^{A+2} & . & S_0^{A+2N-1} & S_0^{A+2N} & S_0^B \\ S_1^A & S_1^{A+1} & S_1^{A+2} & . & S_1^{A+2N-1} & S_1^{A+2N} & S_1^B \\ S_2^A & S_2^{A+1} & S_2^{A+2} & . & S_2^{A+2N-1} & S_2^{A+2N} & S_2^B \\ S_3^A & S_3^{A+1} & S_3^{A+2} & . & S_3^{A+2N-1} & S_3^{A+2N} & S_3^B \\ S_4^A & S_4^{A+1} & S_4^{A+2} & . & S_4^{A+2N-1} & S_4^{A+2N} & S_4^B \\ S_5^A & S_5^{A+1} & S_5^{A+2} & . & S_5^{A+2N-1} & S_5^{A+2N} & S_5^B \\ S_6^A & S_6^{A+1} & S_6^{A+2} & . & S_6^{A+2N-1} & S_6^{A+2N} & S_6^B \end{bmatrix}$$

Fig. 6. ETSI-DSR buffering matrix.

$$\begin{array}{cccccccccc} \boldsymbol{C}^A & \boldsymbol{C}^{A+1} & \boldsymbol{C}^{A+2} & \boldsymbol{C}^{A+3} & \boldsymbol{C}^{A+4} & \boldsymbol{C}^{A+5} & \boldsymbol{C}^{A+6} & \boldsymbol{C}^{A+7} & \boldsymbol{C}^{A+8} & \boldsymbol{C}^B \end{array}$$

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 \end{bmatrix}$$

Fig. 7. Example of consistency matrix.

erroneous by the frame error detection methods. In the subvector-based EC, these erroneous vectors are further submitted to a subvector consistency test which generates a consistency matrix C of dimensions $7 \times (2N+2)$ with elements defined as shown by (8) at the bottom of the page.

The consistency matrix shown in Fig. 7 shows results from a subvector consistency test applied to data corrupted by the GSM EP3. On the basis of this consistency matrix, the subvector based EC is implemented in such a way that all inconsistent subvectors are replaced by their nearest neighboring consistent subvectors, whereas the consistent subvectors are kept unchanged [33]. To exemplify this, consider the first row of the consistency matrix in Fig. 7. This row contains four zeros located in columns C^{A+3}, C^{A+4}, C^{A+5}, and C^{A+6}, respectively, indicating that there are four inconsistent subvectors $\boldsymbol{S}_0$ in vectors $\boldsymbol{V}^{A+3}, \boldsymbol{V}^{A+4}, \boldsymbol{V}^{A+5}$, and $\boldsymbol{V}^{A+6}$ in the corresponding buffering matrix, and that each subvector needs to be substituted by its nearest neighboring subvector. Thereby, subvectors $\boldsymbol{S}_0$ in $\boldsymbol{V}^{A+3}$ and $\boldsymbol{V}^{A+4}$ will be replaced by $\boldsymbol{S}_0$ in $\boldsymbol{V}^{A+2}$, while subvectors $\boldsymbol{S}_0$ in $\boldsymbol{V}^{A+5}$ and $\boldsymbol{V}^{A+6}$ will be replaced by $\boldsymbol{S}_0$ in $\boldsymbol{V}^{A+7}$.

B. Combining Subvector EC and WVD

Subvector EC handles subvectors within erroneous vectors in two different ways such that all consistent subvectors are retained and inconsistent subvectors are substituted with their nearest neighboring consistent subvectors. This strategy exploits error-free information embedded in each erroneous vector, but it is observed that neither the retained consistent subvectors are necessarily correct (or reliable) nor do the nearest neighboring substitutions generate the same features as their original. Consequently, these potentially unreliable features should not be given the same weight as error-free (reliable) features in the ASR decoder. This subsection aims at calculating a reliability measure for each feature and exploiting the measure using the WVD technique.

1) Weighted Viterbi Decoding: The general vector-based WVD modifies the observation probability of each feature vector in the Viterbi decoding by using the reliability of each vector as an exponential weighting factor [34]. The WVD uses the following formula to update the likelihood score accordingly:

$$\delta_t(j) = \underset{i}{\mathrm{Max}} \left[\delta_{t-1}(i) a_{ij} \right] \left[b_j(\boldsymbol{V}^t) \right]^{\gamma(t)} \tag{9}$$

where $\delta_t(j)$ is the likelihood of the most likely state sequence at time t that ends in state j and has generated the observation (feature vectors) from $\boldsymbol{V}^1$ to $\boldsymbol{V}^t$, a_{ij} is the transition probability from state i to state j, $b_j(\boldsymbol{V}^t)$ is the probability of emitting observation $\boldsymbol{V}^t$ when state j is entered. The weighting factor $\gamma(t)$ is a normalized reliability coefficient—of value between 0 and 1—that adjusts the contribution of each vector to the overall likelihood score. The formula shows that choosing the value of $\gamma(t)$ close to one causes the output probability for the particular vector to contribute almost fully to the likelihood score and choosing a value of $\gamma(t)$ close to zero causes the output probability to be equal to one and identically contribute to all models, thereby neutralizing the vector contribution. The vector-based WVD is applied in [35] where a time varying weighting factor is used to handle the fact that the longer a burst is, the less effective the vector repetition technique is.

In combining WVD with the subvector EC, each feature is given its own weighting factor. Consider an observation vector $\boldsymbol{V}^t = [v^t(1), v^t(2), \dots, v^t(K)]^T$ where the component $v^t(k)$ is either one of the MFCC coefficients c_k^t, $k = 1, 2, \dots, 12$ or $\log E^t$ for $k = 13$, and c_0 is not included. The mapping between $v^t(k)$ and $\boldsymbol{S}_j^t$ is defined by (6), e.g., $v^t(1) = c_1^t = S_0^t(0)$ and $v^t(2) = c_2^t = S_0^t(1)$. In assuming a diagonal covariance matrix, the overall observation probability is the product of the probabilities of emitting each individual feature. A feature-based VWD thus computes the likelihood score as follows:

$$\delta_t(j) = \underset{i}{\mathrm{Max}} \left[\delta_{t-1}(i) a_{ij} \right] \prod_{k=1}^{K} \left[b_j(v^t(k)) \right]^{\gamma_k(t)} \tag{10}$$

where $b_j(v^t(k))$ is the observation probability of observing feature $v^t(k)$ when entering state j, and $\gamma_k(t)$ is the reliability measure for feature $v^t(k)$ as given below.

2) Reliability Measure: The reliability of each feature $v^t(k)$ is calculated during the subvector EC processing. When the two corresponding subvectors $\boldsymbol{S}_j^t$ and $\boldsymbol{S}_j^{t+1}$ in a frame pair pass the

$$c_{ij} = \begin{cases} 1, & j = 1 \text{ or } j = 2N+2, 1 \leqq i \leqq 7 \\ 0, & 2 \leq j \leq 2N+1, \mathrm{S}_{i-1}^{A+j-1} \quad \text{inconsistent from (7)}, 1 \leq i \leq 7 \\ 1, & 2 \leq j \leq 2N+1, \mathrm{S}_{i-1}^{A+j-1} \quad \text{consistent from (7)}, 1 \leq i \leq 7 \end{cases} \tag{8}$$
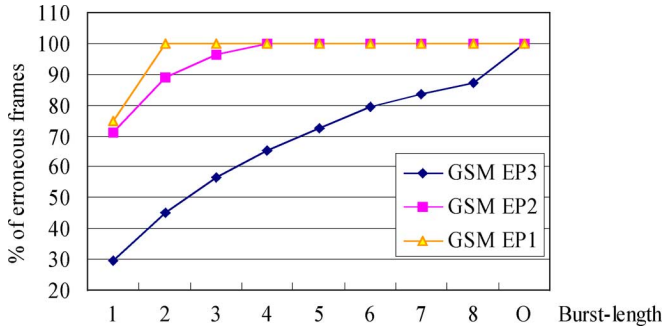
Fig. 8. Distribution functions of erroneous frames by burst length. "O" covers burst lengths larger than 8.

consistency test as given in (7), the reliability of each feature in the subvectors is calculated on the basis of the difference $d(v^t(k), v^{t+1}(k))$ between two corresponding features. For features that do not pass the consistency test, the reliability depends on both the reliability of the substituting feature and the temporal distance between the substituted feature and the substituting feature. Specifically, weightings are assigned according to the following formula:

$$\gamma_k(t) = \begin{cases} \alpha^{d(v^t(k), v^{t+1}(k))/T_k}, & S_j^t \text{ consistent from (7)} \\ \gamma_k(t+p) \cdot \beta^{|p|}, & v^t(k) \text{ substituted by } v^{t+p}(k) \end{cases} \tag{11}$$

where α and β are two adjustable parameters, p is the temporal distance between the two features, and T_k is the threshold for subvector consistency test as used in (7). For error-free vectors, the weighting factors are all equal to one.

VI. EXPERIMENTAL EVALUATION ON ERROR ROBUSTNESS AND DISCUSSION

This section evaluates and discusses the performance of the proposed techniques on error robustness. To enable comparison with a number of often used techniques, the same database and the same channel condition are applied in all experiments. The database applied is Aurora 2 database Test Set A as described in Section III-B. Acoustic models are trained by using FFR features without quantization.

A. Channel Condition and Baseline Techniques

Channel simulation relies on the three GSM EPs as analyzed in Section II-B since they are widely used—for example—by the ETSI-DSR Working Group [22]. To investigate the error distributions of the GSM EPs, they are segmented into frames each corresponding to 10 ms—the shift used in the FFR feature extraction process. Each frame is then classified as error free or erroneous depending on if there are any errors in the frame segment. The resulting distribution functions of erroneous frames as a function of burst length (that is, the number of consecutive erroneous frames) are shown in Fig. 8.

Fig. 8 shows that approximately 100%, 96%, and 56% of erroneous frames have a burst length of less than or equal to three (which are not considered as burst-like) for GSM EP1, EP2, and EP3, respectively. It is noted that EP1 and EP2 essentially

contain random errors only and that EP3 contains both random and burst-like errors. In channel simulations, both random and burst-like errors should be considered as they typically occur in real communication environments. Recognition experiments in [14] moreover demonstrate that the effect of EP1 and EP2 on ASR performance is insignificant. Therefore, the characteristics of the transmission channel in this evaluation are given by EP3.

The compared client-based techniques encompass one-frame scheme [23] and Reed–Solomon code. Different from [7], the Reed–Solomon code implemented in this experiment is RS(32, 16) with 8-bit symbols in which 16 information symbols are encoded into 32 coded symbols. This code has a capability of correcting eight symbol errors or 16 symbol erasures in the code word.

The repetition used by the ETSI-DSR standard [10] and the linear interpolation [11] are chosen as representatives for conventional server-based EC techniques. The performance of error-free transmission and the performance without using any error concealment (NoEC) are provided for comparison. On the experiments of NoEC, transmission errors remain in the speech features and are passed onto the ASR decoder. WVD is used in [6], [16], [35] and shows good performance. In this paper, only the vector-based WVD in [35] is implemented for comparison.

B. Experimental Results

The HFR front-end evaluation uses feature duplication. The client side error recovery techniques encompass MDC, Interleaving12 and Interleaving24. For testing the MDC, two description encodings are transmitted over two uncorrelated EP3 channels.

The tested EC techniques encompass the subvector EC and its combination with WVD. The threshold values given in the ETSI-DSR standard for vector consistency test as specified in Section II-A are directly used in the experiments for the subvector consistency test as introduced in Section V. The parameters of calculating the reliability in (11) for the combination of subvector EC and WVD are chosen to be $\alpha = 0.45$ and $\beta = 0.4$. The results are shown in Table VI and commented in the following.

First, the HFR front-end with feature duplication (HFR-Duplicat) significantly outperforms the ETSI-DSR FFR-based scheme. This is due to the fact that when compared to the 92-bit frame pair, the frame package size in HFR is 48 bits resulting in lower frame error rate for the same channel condition. When the one-frame FFR-based scheme, with a 48-bit frame package size is used, the performance is much better than both the HFR and the ETSI-DSR standard. The WVD (combined with ETSI-DSR repetition EC) demonstrates a better performance than ETSI-DSR indicating the effectiveness of the WVD and a slightly worse performance than the HFR-Duplication again due to the frame pair scheme. Linear interpolation is worse than repetition (ETSI-DSR) as also observed in [14]. The worst performance given by NoEC emphasizes the importance of applying EC schemes in general.

Second, the subvector EC gives substantially better results than both the ETSI-DSR and the WVD scheme. The subvector EC is even better than the one-frame scheme and the RS(32, 16) coding scheme verifying the effectiveness of the subvector

TABLE VI
PERCENT WER AND BIT-RATE (kb/s) FOR SOME SCHEMES FOR EP3 FOR TEST SET A. THE PROPOSED TECHNIQUES ARE HIGHLIGHTED BY BOLD FONT

Scheme	NoEC	Inter-polation	ETSI-DSR	WVD	**HFR-Duplicat**	RS	One-frame	**Sub-vector EC**	**Inter-leaving12**	**Sub-vector EC +WVD**	**Inter-leaving24**	**MDC**	Error free
WER	8.88	7.35	6.70	4.78	4.56	3.45	3.41	2.65	2.43	2.05	1.74	**1.04**	0.95

TABLE VII
PERCENT WER ACROSS DIFFERENT α SETTINGS WITH A FIXED $\beta = 0.4$ FOR THE GSM EP3 FOR AURORA 2 TEST SET A

α	0.35	0.40	0.43	**0.45**	0.47	0.50	0.55
WER	2.08	2.07	**2.05**	**2.05**	2.06	2.06	2.10

TABLE VIII
PERCENT WER ACROSS DIFFERENT β SETTINGS WITH A FIXED $\alpha = 0.45$ FOR THE GSM EP3 FOR AAURORA 2 TEST SET A

β	0.30	0.35	0.38	**0.40**	0.42	0.45	0.50
WER	2.08	2.08	**2.05**	**2.05**	2.07	2.09	2.11

TABLE IX
PERCENT WER ACROSS DIFFERENT THRESHOLD SETTINGS FOR SUBVECTOR EC AND ITS COMBINATION WITH WVD FOR THE GSM EP3 FOR AURORA 2 TEST SET A

λ	-1.0	0.1	0.6	0.8	**1.0**	1.2	2.0
Sub-vector	6.70	4.42	2.90	**2.60**	2.65	2.70	3.88
Sub-vector + WVD	4.73	2.85	2.09	**2.01**	2.05	2.18	2.81

EC technique. It should be noted that the RS(32, 16) scheme gives a performance close to the one-frame scheme but this scheme is demanding with regard to bandwidth and computations. The subvector EC technique has the advantages of neither introducing increased complexity nor resource requirement. A further performance gain is obtained by employing the combination of subvector EC and WVD.

Finally, the interleaving and MDC schemes perform remarkably well. One advantage of applying the special interleaving schemes is that delays are introduced only if there are transmission errors. MDC renders results close to that of error-free transmission.

C. Reliability Measure Parameters α and β

This subsection investigates the relationship between performance and the parameters α and β used in calculating the reliability in (11). Table VI shows that the choice of $\alpha = 0.45$ and $\beta = 0.4$ gives significantly better performance than the ETSI-DSR standard. The effects on WER of varying the values of the two parameters are presented in Tables VII and VIII. The results show that a setting of the two parameters α and β around their optimum values only has a minor influence on the resulting WER.

D. Consistency Test Thresholds

The effect of the settings of the threshold values [i.e., $T_j(0)$ and $T_j(1)$ in (7), which are the same as T_k in (11)] on performance is investigated in this subsection. It is noted that two sets of thresholds are applied in this paper. The first set is used in the vector consistency test as an additional test to the CRC checking according to the ETSI-DSR standard, and the second set in the subvector consistency test. In the experiments conducted above, the two sets of thresholds are given the same values as provided by the ETSI-DSR standard.

In the experiments in this subsection, the first set of thresholds is kept as given in the ETSI-DSR standard, whereas the second set of thresholds is varied across a range. This is done by multiplying the ETSI-DSR standard values with a scaling factor λ as given in Table IX, showing the WER across the λ

range. With the setting $\lambda = -1$, the subvector EC gives the same WER as the ETSI-DSR standard as shown in Table VI, and the performance of the combination of subvector EC and WVD is close to the WER obtained by applying vector-based WVD (4.78%). These results are in correspondence since negative threshold values result in all subvectors being replaced by its nearest neighboring error-free ones, i.e., equivalent to the ETSI-DSR repetition scheme. The results also show that for $\lambda = 0.1$, where only almost identical subvectors are classified as consistent, the subvector-based EC still gives 34.0% relative improvement as compared to the ETSI-DSR standard, and the combined method gives 39.7% relative improvement as compared to the vector-based WVD. A possible explanation for this improvement is that keeping the almost identical features unchanged may be better than using substitutions. For $\lambda = 2$, with only a small number of subvectors detected as inconsistent and, thus, the majority of erroneous subvectors may remain in the speech feature stream, improvements are still observed for both methods. This result together with the worst performance for NoEC, shown in Table VI, suggests that transmission errors harm the recognition performance the most as they are causing feature values largely different from their original. The lowest WER is achieved for $\lambda = 0.8$ showing 61.2% relative improvement compared to the ETSI-DSR standard for the subvector EC and 57.9% relative improvement compared to the vector-based WVD for the combination. The results indicate that varying the scaling factor λ around the default setting only has a minor influence on the resulting WER.

VII. COMPARATIVE STUDY

Comparative studies are conducted to explain the variation in WER as observed for the tested EC techniques encompassing repetition, interpolation, and subvector EC. This involves direct visual inspections of MFCC features, comparison of dynamic programming (DP) distances and comparison of hidden Markov model (HMM) state durations [36]. The database applied is the Danish digits database as described in Section III-B. For simplicity, the experiments in this section use randomly distributed errors with a BER value of 2%.

A. Visual Inspection of MFCC Features

The error-free MFCC features are visually compared with the features corrupted by transmission errors and reconstructed by
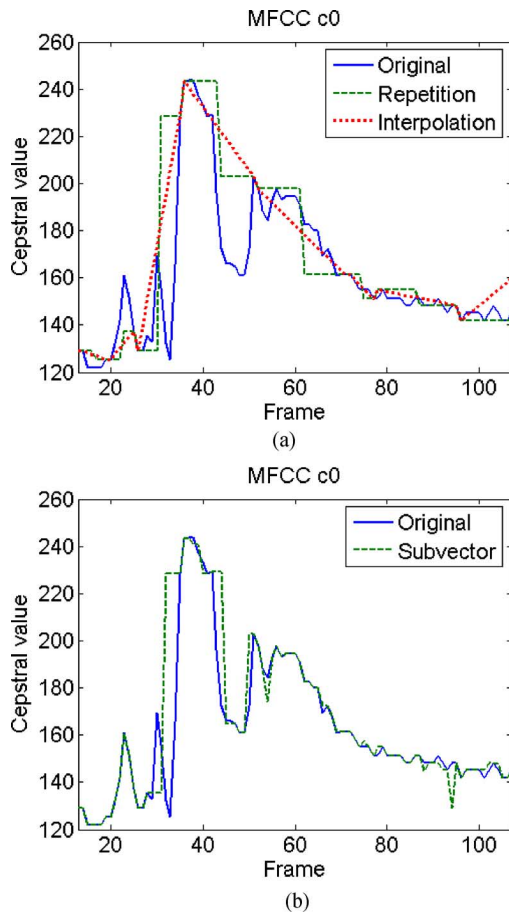
Fig. 9. Coefficient c_0. (a) MFCC, rMFCC, and iMFCC. (b) MFCC and sMFCC.

Fig. 10. Coefficient delta-c_0. (a) Delta-MFCC, delta-rMFCC, and delta-iMFCC. (b) Delta-MFCC and delta-sMFCC.

either repetition (rMFCC), interpolation (iMFCC), or subvector (sMFCC) concealment. The test utterance is the Danish word "et" also used in the experiments presented in Figs. 3 and 4. The MFCC coefficient c_0 is especially chosen to show transitions. The impacts of applying the three EC techniques on the static, the delta (velocity), and the delta-delta (acceleration) features are shown in Figs. 9–11, respectively.

Fig. 9(a) shows that the rMFCC feature tracks the error-free MFCC feature closer than the iMFCC feature indicating a better reconstruction. Fig. 9(b) shows that the sMFCC feature tracks the MFCC feature closest. Figs. 10(a) and 11(a) show that the delta-rMFCC and delta-delta-rMFCC features track the corresponding error-free features closer than the delta-iMFCC and delta-delta-iMFCC features. The explanation may be that the interpolation technique reconstructs each frame by interpolating along a straight line of iMFCC features as shown in Fig. 9(a). This results in segments of constant delta-iMFCC and zero-valued delta-delta-iMFCC segments which cause less available information for the Viterbi decoding. In applying the repetition technique, a fast change in feature value is observed in the middle of erroneous frames. Figs. 10(b) and 11(b) show that delta-sMFCC and delta-delta-sMFCC features track the corresponding error-free features closest. Figs. 9(a), 10(a), and 11(a) reveal that the MFCC and the rMFCC feature curves appear to display similar shapes even though there are some shifts along
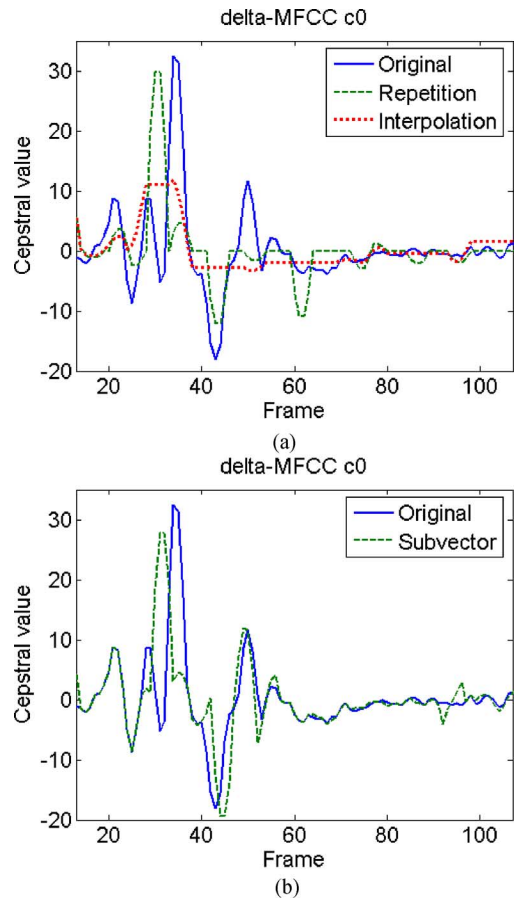
the time axis as compared to the iMFCC feature curves. However, the DP embedded in the Viterbi algorithm makes this shift relatively irrelevant, which is demonstrated in the discussion on DP distances in the next subsection.

In general, it seems that the rapid changes often appearing in MFCC coefficients do not justify the introduction of the linear interpolation scheme. Rapid changes often occur in segments spanning over phoneme boundaries (for example, in Fig. 4 around frame 48).

B. Comparison of DP Distances

It is expected that the interpolation technique must result in smaller Euclidean distance values between corresponding MFCC and iMFCC features than between MFCC and rMFFC features—when averaged over an entire utterance. The conclusions on average Euclidean distances are, however, not directly comparable to Viterbi-based matching which is in essence a DP approach. In the following experiment, Euclidean distances are analyzed in connection with time normalized DP distances between error-free features and features derived by repetition, interpolation, and subvector concealment by using the symmetric dynamic time warping (DTW) according to [37]. As an example, the Euclidean and DP distances between the error-free MFCC c_0 and the corresponding MFCC c_0 generated by the three EC techniques for the word "et" are shown in Fig. 12. The results show that the rMFCC feature has smaller DP distance

TABLE X
NUMBER OF UTTERENCES FOR WHICH iMFCC c_0 OR rMFCC c_0 OR sMFCC c_0 HAS THE SMALLEST
OR SECOND SMALLEST EUCLIDEAN OR DP DISTANCE (OUT OF 328 UTTERNACES)

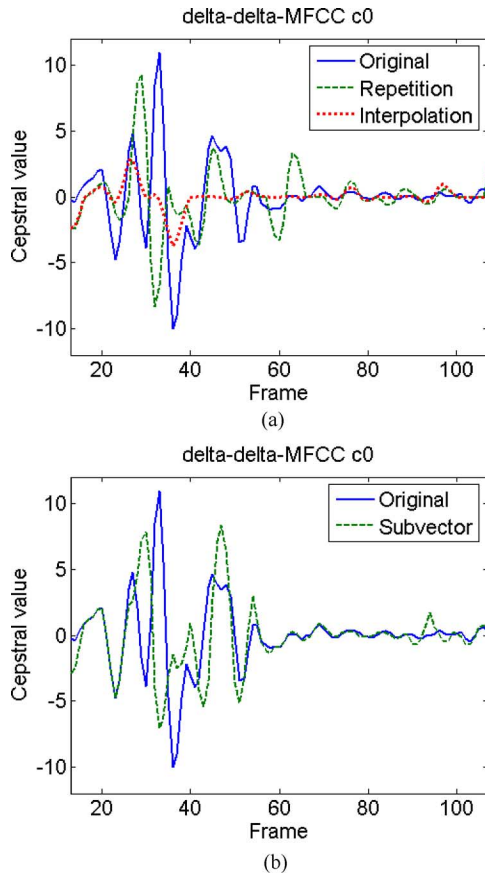| Feature	Number of utterances			
	Smallest Euclidean distance	Second Smallest Euclidean distance	Smallest DP distance	Second Smallest DP distance
iMFCC c_0	0	295	0	146
rMFCC c_0	0	33	0	182
sMFCC c_0	328	0	328	0

Fig. 11. Coefficient delta-delta-c_0. (a) Delta-delta-MFCC, delta-delta-rMFCC, and delta-delta-iMFCC. (b) Delta-delta-MFCC and delta- delta-sMFCC.
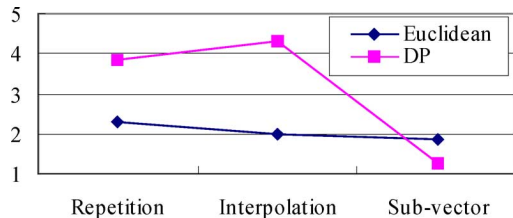

Fig. 12. Euclidean and DP distances between c_0 of MFCC and MFCC generated by three EC techniques for word "et."

to the original MFCC feature than the iMFCC feature though the opposite is observed for the Euclidean distance. Moreover, the results show that the distances between original MFCC and sMFCC are always the smallest.

The experiment is enlarged to encompass a large number of utterances. The experiment compares the Euclidean as well as

TABLE XI
AVERAGE STATE DURATIONS OVER ELEVEN TEST UTTERANCES FOR
ERROR-FREE MFCC, rMFCC, iMFCC, AND sMFCC

Feature	MFCC	rMFCC	iMFCC	sMFCC
State duration	5.253	4.023	3.736	5.345

DP distances of iMFCC c_0, rMFCC c_0, and sMFCC c_0 to the error-free MFCC c_0 for 328 test utterances. The numbers of utterances having the smallest or second smallest Euclidean or DP distance are counted and shown in Table X. The results show that sMFCC features always have the smallest distances to error-free features for both measures as compared to distances for iMFCC and rMFCC features. The results also show that for the majority of utterances, iMFCC c_0 has smaller Euclidean distance to the original MFCC c_0 than rMFCC c_0, whereas the number of utterances for which iMFCC c_0 has smaller DP distance is less than the number for rMFCC. This indicates that repetition performs better in terms of DP distance although worse in terms of Euclidean distance.

C. Comparison of HMM State Durations

To study the Viterbi decoding process, a set of experiments are conducted where the HMM state duration (counted as the number of frames) is tracked during decoding. The experiment applies strict left-to-right HMM models. The normalized duration is calculated as the number of speech frames of each utterance—nonspeech frames excluded—divided by the total number of states of the models that the utterance matches.

The average state durations over eleven test utterances (one for each digit including two variants for digit "one") for error-free MFCC, rMFCC, iMFCC and sMFCC are shown in Table XI. Two facts are observed from these data. First, interpolation gives the smallest average state duration indicating that features calculated by interpolation result in faster state transition, whereas features reconstructed by repetition result in longer average state occupancies. As seen in Section VII-A, interpolation generates artefact features and may therefore mislead the Viterbi search. Second, the average state duration for features calculated by subvector EC is very close to the one for error-free features, justifying that the subvector EC is better for reconstructing erroneous features.

VIII. CONCLUSION

This paper presented a set of techniques for DSR by exploiting the temporal correlation present in the speech features. First, an HFR front-end processing with feature duplication was described. The HFR front-end achieves low bit-rate with

reduced computational cost as opposed to source coding techniques. Moreover, it was demonstrated that the effect of static feature duplication on the delta and delta-delta features is marginal, that feature duplication is essential for obtaining comparable performance to the FFR front-end, and that the frame rate should match the number of HMM states. Second, this paper presented a number of client-based error recovery techniques including MDC and interleaving. An innovative aspect of these techniques is that half of the FFR features are considered as the source information and the other half as redundant information which can be arranged in different ways. The proposed interleaving techniques do not introduce any transmission delay when there are no transmission errors. The third contribution of this paper is proposing a subvector-based EC technique and its combination with WVD, adding only marginal extra complexity and resource requirements to the back-end.

The general results on error robustness are very encouraging. The HFR front-end with feature duplication, subvector EC combined with WVD, Interleaving24 and the MDC technique show 31.9%, 69.4%, 74.0%, and 84.5% improvement on error robustness over the ETSI-DSR standard, respectively. Except for the subvector EC, all techniques can be equally applied in circuit-switched and packet-switched networks.

Finally, this paper presented three comparison methods encompassing MFCC feature, DP distance, and HMM state duration comparison. These approaches provide useful insight into the behavior of a number of EC techniques.

ACKNOWLEDGMENT

REFERENCES

[1] D. Pearce, "Enabling new speech driven services for mobile devices: An overview of the ETSI standards activities for distributed speech recognition front-ends," in *Proc. Appl. Voice Input/Output Soc. Conf. Speech Applicat. Conf.*, May 2000.

[2] Z.-H. Tan, P. Dalsgaard, and B. Lindberg, "Automatic speech recognition over error-prone wireless networks," *Speech Commun.*, vol. 47, no. 1-2, pp. 220–242, Sep.- Oct. 2005.

[3] V. Digalakis, L. Neumeyer, and M. Perakakis, "Quantization of cepstral parameters for speech recognition over the World Wide Web," *IEEE J. Select. Areas Commun.*, vol. 17, no. 1, pp. 82–90, Jan. 1999.

[4] Q. Zhu and A. Alwan, "An efficient and scalable 2D DCT-based feature coding scheme for remote speech recognition," in *Proc. ICASSP*, May 2001, pp. 113–116.

[5] W.-H. Hsu and L.-S. Lee, "Efficient and robust distributed speech recognition (DSR) over wireless fading channels: 2D-DCT compression, iterative bit allocation, short BCH code and interleaving," in *Proc. ICASSP*, May 2004, pp. 69–72.

[6] A. Bernard and A. Alwan, "Low-bitrate distributed speech recognition for packet-based and wireless communication," *IEEE Trans. Speech Audio Process.*, vol. 10, no. 8, pp. 570–579, Nov. 2002.

[7] C. Boulis, M. Ostendorf, E. A. Riskin, and S. Otterson, "Gracefully degradation of speech recognition performance over packet-erasure networks," *IEEE Trans. Speech Audio Process.*, vol. 10, no. 8, pp. 580–590, Nov. 2002.

[8] A. B. James and B. P. Milner, "An analysis of interleavers for robust speech recognition in burst-like packet loss," in *Proc. ICASSP*, May 2004, pp. 853–856.

[9] C. Perkins, O. Hodson, and V. Hardman, "A survey of packet loss recovery techniques for streaming audio," *IEEE Network*, vol. 12, no. 5, pp. 40–48, Sep./Oct. 1998.

[10] *Distributed Speech Recognition; Extended Advanced Front-End Feature Extraction Algorithm; Compression Algorithm, Back-End Speech Reconstruction Algorithm*, ETSI Standard ES 202 212, Nov. 2003.

[11] B. Milner and S. Semnani, "Robust speech recognition over IP networks," in *Proc. ICASSP*, May 2000, pp. 1791–1794.

[12] Z. A. Bawab, I. Locher, J. Xue, and A. Alwan, "Speech recognition over bluetooth wireless channels," in *Proc. Eurospeech*, Sep. 2003, pp. 1233–1236.

[13] H. K. Kim and R. V. Cox, "A bitstream-based front-end for wireless speech recognition on IS-136 communications system," *IEEE Trans. Speech Audio Process.*, vol. 9, no. 4, pp. 558–568, Jul. 2001.

[14] Z.-H. Tan, P. Dalsgaard, and B. Lindberg, "Partial splicing packet loss concealment for distributed speech recognition," *Inst. Electron. Eng. Electron. Lett.*, vol. 39, no. 22, pp. 1619–1620, Oct. 2003.

[15] A. Potamianos and V. Weerackody, "Soft-feature decoding for speech recognition over wireless channels," in *Proc. ICASSP*, May 2001, pp. 269–272.

[16] V. Weerackody, W. Reichl, and A. Potamianos, "An error-protected speech recognition system for wireless communications," *IEEE Trans. Wireless Commun.*, vol. 1, no. 2, pp. 282–291, Apr. 2002.

[17] V. Ion and R. Haeb-Umbach, "A unified probabilistic approach to error concealment for distributed speech recognition," in *Proc. Interspeech*, Sep. 2005, pp. 2853–2856.

[18] H. K. Kim and R. V. Cox, "Bitstream-based feature extraction for wireless speech recognition," in *Proc. ICASSP*, May 2000, pp. 1607–1610.

[19] X. Li, J. Malkin, and J. A. Bilmes, "A high-speed, low-resource ASR back-end based on custom arithmetic," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 14, no. 5, pp. 1683–1693, Sep. 2006.

[20] P. Haavisto, "Audio-visual signal processing for mobile communications," in *Proc. Eur. Signal Process. Conf.*, Sep. 1998.

[21] C. Pelaez-Moreno, A. Gallardo-Antolin, and F. Diaz-de-Maria, "Recognizing voice over IP: A robust front-end for speech recognition on the World Wide Web," *IEEE Trans. Multimedia*, vol. 3, no. 2, pp. 209–218, Jun. 2001.

[22] "Recognition with WI007 compression and transmission over GSM channel," Ericsson, 2000, Aurora document no. AU/266/00.

[23] Z.-H. Tan, P. Dalsgaard, and B. Lindberg, "OOV-detection and channel error protection for distributed speech recognition over wireless networks," in *Proc. ICASSP*, Apr. 2003, pp. 336–339.

[24] J. Macias-Guarasa *et al.*, "Revisiting scenarios and methods for variable frame rate analysis in automatic speech recognition," in *Proc. Eurospeech*, Sep. 2003, pp. 1809–1812.

[25] A. Bernard and A. Alwan, "Joint channel decoding—Viterbi recognition for wireless applications," in *Proc. Eurospeech*, Sep. 2001, pp. 2703–2706.

[26] S. J. Young *et al.*, *HTK: Hidden Markov Model Toolkit V3.2.1, Reference Manual.* Cambridge, U.K.: Cambridge Univ. Speech Group, 2004.

[27] B. Lindberg *et al.*, "A noise robust multilingual reference recogniser based on SpeechDat(II)," in *Proc. ICSLP*, Oct. 2000.

[28] H. G. Hirsch and D. Pearce, "The Aurora experimental framework for the performance evaluation of speech recognition systems under noisy conditions," in *Proc. ISCA ITRW ASR00*, Sep. 2000, pp. 181–188.

[29] Z.-H. Tan, P. Dalsgaard, and B. Lindberg, "Adaptive multi-frame-rate scheme for distributed speech recognition based on a half frame-rate front-end," in *Proc. IEEE MMSP*, Nov. 2005, pp. 1–4.

[30] "AMR speech CODEC: General description (release 6) 2004, 3GPP TS 26.071 V6.0.0.

[31] V. K. Goyal, "Multiple description coding: Compression meets the network," *IEEE Signal Process. Mag.*, vol. 18, no. 5, pp. 74–93, Sep. 2001.

[32] J. L. Ramsey, "Realization of optimum interleavers," *IEEE Trans. Inf. Theory*, vol. IT-16, no. 3, pp. 338–345, May 1970.

[33] Z.-H. Tan, P. Dalsgaard, and B. Lindberg, "A subvector-based error concealment algorithm for speech recognition over mobile networks," in *Proc. ICASSP*, May 2004, pp. 57–60.

[34] N. B. Yoma, F. R. McInnes, and M. A. Jack, "Weighted Viterbi algorithm and state duration modelling for speech recognition in noise," in *Proc. ICASSP*, May 1998, pp. 709–712.

[35] A. Cardenal-Lopez, L. Docio-Fernandez, and C. Garcia-Mateo, "Soft decoding strategies for distributed speech recognition over IP networks," in *Proc. ICASSP*, May 2004, pp. 49–52.

[36] Z.-H. Tan, B. Lindberg, and P. Dalsgaard, "A comparative study of feature-domain error concealment techniques for distributed speech recognition," in *Proc. Robust2004*, Aug. 2004.

[37] H. Sakoe and S. Chiba, "Dynamic programming algorithm optimization for spoken word recognition," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-26, no. 1, pp. 43–49, Feb. 1978.

Zheng-Hua Tan (M'00–SM'06) received the B.S. and M.S. degrees in electrical engineering from Hunan University, Changsha, China, in 1990 and 1996, respectively, and the Ph.D. degree in electronic engineering from Shanghai Jiao Tong University, Shanghai, China, in 1999.

He is an Associate Professor in the Department of Electronic Systems, Aalborg University (AAU), Aalborg, Denmark, which he joined in May 2001. Prior to that, he was a Postdoctoral Fellow in the Department of Computer Science, Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Korea. He was also an Associate Professor in the Department of Electronic Engineering at Shanghai Jiao Tong University. His research interests include speech recognition over communication networks, robust speech recognition, acoustic modeling, machine learning, and computational intelligence. He has published extensively in these areas in refereed journals and conference proceedings.

Paul Dalsgaard (M'74–SM'89) was born in Denmark on May 18, 1937. He received the M.S. degree in electronic engineering from the Technical University of Denmark, Lyngby, in 1962.

He was a Lecturer at the Danish Engineering Academy, Copenhagen, Denmark, from 1963 to 1969, and in Aalborg from 1969 to 1974 at what time Aalborg University (AAU) was inaugurated. He was a one-year Visiting Researcher at McMaster University, Hamilton, ON, Canada, in August 1974, studying optimal design of electronic circuit systems on the basis of component tolerance and tuning capabilities. He was an Associate Professor at AAU from 1974 to 1993 when he was appointed Full Professor in Speech Technology and Multimedia. He initiated research on ASR in Denmark in 1979 and a large part of this research was done in collaboration with Danish and European companies on the basis of public funding. His research on ASR has, to a large extent, focused on the integration of cross-disciplinary information including acoustic phonetics and linguistics. Since 1993, the research in the speech group was carried out in the context of the general research that took place within the Centre for Personal Communication (CPK) and on the basis of public funding received from 1993 to 2003. Since 2001, his research interests have been focused on the Integration of ASR in wireless network systems and lately on methods that may lead to richer and more robust representation of speech features from signal preprocessing.

Børge Lindberg, (M'94) was born in Denmark, November, 1959. He received the M.Sc. degree in electrical engineering from Aalborg University (AAU), Aalborg, Denmark, in 1983.

In 1983, he became a Research Assistant at AAU, in 1986, he joined Jydsk Telefon, R&D Laboratory, Erhus, Denmark, and in 1992, he became a Research Assistant at AAU [since 1993 at the Center for PersonKomunikation (CPK)]. In 1995, he was a Visiting Researcher at the Defence Research Agency, Great Malvern, U.K., studying methods for predicting the performance of automatic speech recognition systems. Since 1996, he has been an Associate Professor at the Department of Electronic Systems, AAU, and since 2006, he has been the Head of this department. A large part of his automatic speech recognition research has been done in collaboration with Danish and European companies on the basis of public funding. From 2001 to 2006, he was the Chairman of the EU COST Action 278 on Spoken Language Interaction in Telecommunication. His current research interests include speech recognition with a focus on robustness and acoustic modeling techniques.