Available online at www.sciencedirect.com

SCIENCE DIRECT®

Speech Communication 48 (2006) 888–902

SPEECH COMMUNICATION

www.elsevier.com/locate/specom

Efficient scalable encoding for distributed speech recognition ☆

Naveen Srinivasamurthy [a,1], Antonio Ortega [b], Shrikanth Narayanan [b,*]

[a] Standards Engineering, Qualcomm Incorporated, 5775 Morehouse Drive, San Diego, CA 92121, United States

[b] Department of Electrical Engineering-Systems, Signal and Image Processing Institute, Integrated Media Systems Center, University of Southern California, Los Angeles, CA 90089-2564, United States

Received 9 March 2005; received in revised form 28 August 2005; accepted 8 November 2005

Abstract

The problem of encoding speech features in the context of a distributed speech recognition system is addressed. Specifically, speech features are compressed using *scalable* encoding techniques to provide a multi-resolution bitstream. The use of this scalable encoding procedure is investigated in conjunction with a multi-pass distributed speech recognition (DSR) system. The multi-pass DSR system aims at progressive refinement in terms of recognition performance, (i.e., as additional bits are transmitted the recognition can be refined to improve the performance) and is shown to provide both bandwidth and complexity (latency) reductions. The proposed encoding schemes are well suited for implementation on light-weight mobile devices where varying ambient conditions and limited computational capabilities pose a severe constraint in achieving good recognition performance. The multi-pass DSR system is capable of adapting to varying network and system constraints by operating at an appropriate trade-off point between transmission rate, recognition performance and complexity to provide desired quality of service (QoS) to the user. The system was tested using two case studies. In the first, a distributed two-stage names recognition task, the scalable encoder operating at a bitrate of 4.6 kb/s achieved the same performance as that achieved using uncompressed features. In the second study, a two stage multi-pass continuous speech recognition task using HUB-4 data, the scalable encoder at a bitrate of 5.7 kb/s achieved the same performance as that achieved with uncompressed features. Reducing the bitrate to 4800 b/s resulted in a 1% relative increase in WER.

Keywords: Distributed speech recognition; Scalable encoding; Multi-pass recognition; Joint coding-classification

1. Introduction

With recent advances in wireless technology, there has been a wide proliferation of mobile devices with wireless connectivity, including mobile phones, personal digital assistants (PDAs) and tablet computers. Mobile users make use of a combination of buttons and a pointing device on the mobile devices to enter data into these devices. The restriction of having small displays, along with the potential need

☆ This research has been funded in part by the Integrated Media Systems Center, a National Science Foundation Engineering Research Center, Cooperative Agreement No. EEC-9529152, an NSF CAREER award (to Narayanan), and a DARPA Grant No. N66001-02-C-6023. Part of this work was previously published in (Srinivasamurthy et al., 2001a).

[*] Corresponding author. Tel.: +1 213 740 6432; fax: +1 213 740 4651.

E-mail addresses: naveens@qualcomm.com (N. Srinivasamurthy), ortega@sipi.usc.edu (A. Ortega), shri@sipi.usc.edu (S. Narayanan).

[1] Tel.: +1 858 651 5375; fax: +1 858 651 4880.

0167-6393/$ - see front matter © 2005 Elsevier B.V. All rights reserved.
doi:10.1016/j.specom.2005.11.003

for hands-free operation, makes interaction with these devices cumbersome and difficult even for simple tasks like directory lookup. Speech input provides an obvious and promising solution for improved data entry/retrieval flexibility. There has also been a recent trend in interactive business and information applications such as call centers to move from a touch-tone based solution to a voice-driven approach. Similarly, in automotive telematics, speech input provides hands free access for users. These applications provide additional motivation for providing speech recognition capability to mobile environments.

The scope and performance of speech recognition is primarily dictated by the acoustic and language models used; higher requirements along these dimensions typically imply increased complexity of the models. This, however, also increases the computation and memory requirements of the system. Current desktop/laptop computers usually have sufficient computation and memory resources to support large vocabulary continuous speech recognition (LVCSR) tasks requiring complex acoustic and language models (Woodland et al., 2001). Personal mobile devices, on the other hand, tend to have relatively limited computation, memory and storage capabilities, which have to be shared among multiple tasks, so that only a limited amount of resources can be devoted to speech applications. Since mobile devices are equipped with wireless connectivity, an alternative solution to overcome the complexity constraints is to adopt a client–server architecture wherein the speech utterance is acquired at the mobile device (client) and transmitted to a remote recognition engine (server). Thus the mobile device makes use of network resources to provide speech-driven services to the user. A further motivation for the use of a client–server based architecture stems from the fact that several speech-driven applications such as call centers and voice portals are inherently network based and domain specific knowledge (often dynamically changing) is usually only available at a centralized location. Additionally, in application scenarios where the ambient environment is highly variable, such as those involving mobile devices, there may be much to be gained in terms of running more complex schemes at a remote server for improved recognition performance (Gales and Young, 1996).

A typical speech recognizer consists of two distinct components, a feature extractor and a pattern recognizer that makes use of acoustic models, language models and other domain specific knowledge. The feature extractor computes relevant features (e.g., mel frequency cepstral coefficients (MFCCs), perceptual linear prediction (PLP) coefficients, linear predicted cepstral coefficients (LPCCs), etc.) from the input speech which are used by the pattern recognizer for recognition of input speech. When a client–server architecture is adopted for speech recognition, it is essential for the client to compress the speech before transmission to the server in order to conserve bandwidth and power. One such client–server system, shown in Fig. 1, is distributed speech recognition (DSR) (Digalakis et al., 1999), where the client contains the feature extractor (the computation requirements of feature extraction are almost the same as those of a vocoder based speech encoder). Only the extracted features are encoded and transmitted to the server with the speech recognizer which operates on the decoded feature data.

Encoding features instead of employing a traditional speech encoder (vocoder) at the client is desirable: while vocoders are optimized to maximize perceptual quality, DSR encoders can be designed to achieve the best possible recognition performance for a given bitrate. The effect of various speech coding techniques on speech recognition, including GSM (Digalakis et al., 1999; Srinivasamurthy et al., 2000; Kiss, 2000; Lilly and Paliwal, 1996; Srinivasamurthy et al., 2001b), G.723.1, G.727, G.728, G.729 (Turunen and Vlaj, 2001), ADPCM

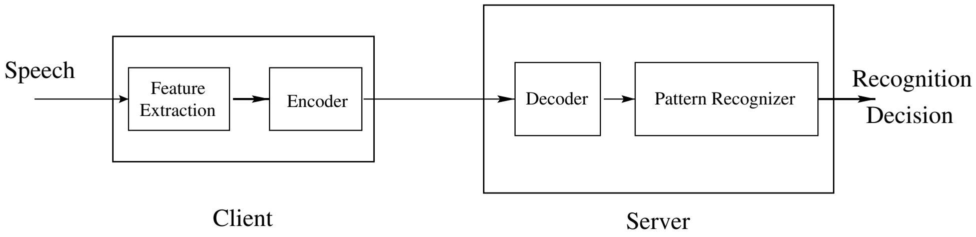

Fig. 1. A distributed speech recognizer. The client extracts the features, encodes and transmits it to the server which uses decoded features for recognition.

(Lilly and Paliwal, 1996) and MELP (Srinivasamurthy et al., 2000; Srinivasamurthy et al., 2001b), has been previously evaluated by a number of researchers. In all cases, it was shown that speech coding significantly degrades speech recognition performance.

Methods to improve the recognition performance when speech compressed by vocoders is used for recognition have also been proposed (Huerta, 2000; Kim and Cox, 2000). Here, features are extracted directly from the compressed bitstream. While this helps improve recognition performance, the rate-recognition performance achieved by vocoders is still inferior to those achievable by typical DSR encoders. The main drawback of DSR, however, is the lack of methods for reconstructing the speech of adequate quality from the received features at the server. Toward addressing this drawback, methods wherein the client augments the transmitted features with pitch and voicing information have been proposed (Chazan et al., 2000; Ramabadran et al., 2001).

A number of techniques have also been proposed for DSR encoding. These include scalar quantization (Digalakis et al., 1999; Zhu and Alwan, 2001; Kiss and Kapanen, 1999), vector quantization (VQ) (Ramaswamy and Gopalakrishnan, 1998) and product VQ (Digalakis et al., 1999; Bernard and Alwan, 2001; Speech processing, 2000) to quantize the features; single-step prediction (Ramaswamy and Gopalakrishnan, 1998) to exploit memory between successive feature vectors; transformation by DCT (Zhu and Alwan, 2001; Kiss and Kapanen, 1999) to exploit inter and intra-frame correlation and lossless coding (Zhu and Alwan, 2001) to losslessly encode the quantization indices. However, none of the previously proposed encoders are scalable. In this paper we describe a *scalable* DSR encoder, i.e., a system where the encoder provides a base layer which can be augmented by one or more enhancement layers to achieve higher fidelity data representation. A scalable DSR encoder enables the client to progressively refine the information at the server using multiple enhancement layers. In the proposed scalable DSR encoder, each feature is *independently* quantized after single-step prediction using a uniform scalar quantizer (USQ). The quantization index is losslessly coded using a combination of Huffman coding and run length coding. Separate prediction loops are maintained for the base and each of the enhancement layers. The proposed scalable encoding technique offers several advantages: (i) it provides multiple layers, where the enhancement layer data refines the base layer data to provide higher fidelity representation, (ii) the rate of each layer can be easily modified, and (iii) it provides flexibility for using unequal error protection schemes (Weerackody et al., 2002) to ensure adequate protection for the bitstream. This is especially useful in mobile channels where the channel conditions are constantly changing.

The other contribution of this paper is to show the potential of utilizing the progressive refinement of data provided by the client for progressive processing. Specifically, we show that the proposed scalable encoder can be combined efficiently with a multi-pass recognition scheme at the server in order to set up a *multi-pass DSR system*. This system can be adapted to achieve reduced average bitrate and or reduced recognition latency, thus providing increased flexibility in the system design. The multi-pass DSR system can be made to operate at various suitable trade-off points between transmission rate, recognition performance and complexity, depending on (i) the QoS (recognition performance, latency) the client wants to provide the user (ii) the channel conditions, and (iii) the computational bottlenecks in the different parts of the overall system.

The rest of the paper is organized as follows. The proposed scalable DSR encoder is described in Section 2. The operation of the multi-pass DSR system under different constraints is explained in Section 3. Details of the multi-pass recognition schemes considered in this paper are given in Section 4 and the actual experiments are described in Section 5. Results are provided in Section 6. Finally, conclusions and discussion are given in Section 7.

2. Scalable encoding

The objective of the DSR client is to transmit feature data at the minimal bitrate required to achieve the desired recognition performance. This optimal bitrate, however, depends on several factors including the recognition task, ambient noise, channel conditions and the speaker characteristics. The final objective, i.e., recognition performance and confidence, is known only at the server and, hence, the client in general will have to transmit data at a higher than the optimal bitrate to accommodate the different signal conditions and achieve the desired recognition performance.

The goal of this paper is to design scalable encoding of features in the DSR context. Previously proposed DSR encoders are *not scalable*, i.e., they do not provide a multi-resolution or embedded bitstream. The idea here is to enable an embedded bitstream such that the decoder can reconstruct the features prior to recognition using only the base layer (at bitrate R_1) to achieve coarse recognition performance, or alternatively can decode the whole bitstream, including both base layer and enhancement layer (total bitrate $R_2 > R_1$) in order to achieve improved recognition performance. The flexibility offered by scalable encoding can be exploited in a number of ways. For example, the server has the option of only requesting data for portions of the speech which it considers will help in improving the recognition performance, i.e., the server could request enhancement data only for low confidence portions of the speech. An example application benefiting from this approach is a distributed dictation system, where initially only the base layer is transmitted to the server. After completion of dictation the server transmits to the client the recognized text which is shown to the user. The user can then identify the misrecognized words. The enhancement data corresponding to these misrecognized words is transmitted to the server. The server now generates a list of possible replacement words for each misrecognized word. These are now provided as choices for the misrecognized words.

Ideally, given base layer rate R_1 and enhancement layer rate $R_2 - R_1$, we would like the full resolution performance (when R_2 bits are decoded) to be the same as if we had used a single resolution (only one layer) encoder at rate R_2. Encoders having this property are said to be "successively refinable". This property has been found to be achievable in some particular cases, e.g., for some memoryless *i.i.d.* sources under standard signal distortion metrics, such as mean square error (MSE) (Equitz and Cover, 1991). In our DSR case, the speech features from adjacent frames are correlated, i.e., the source is not memoryless. Moreover, we are considering here a non-traditional metric to measure encoding performance at a given rate (i.e., recognition performance rather than MSE). Thus, in general, we do not expect scalable DSR methods to be successively refinable, even with respect to standard MSE metrics. The DSR encoder proposed in this paper is scalable, enabling refinement of low fidelity data with enhancement bits to achieve higher recognition fidelity. The scalable DSR encoder is built by a novel combination of several non-scalable predictive (DPCM) encoders. Before presenting the scalable encoder, the component predictive encoder (i.e., prediction with scalar quantization) is explained.

Typically, features for ASR such as MFCCs are computed from speech utterances that have been segmented using overlapping Hamming windows. Due to this overlap and the underlying correlation in the speech (because of the slow movement of articulators), it is reasonable to expect that MFCC vectors from adjacent frames will exhibit high correlation. To achieve good compression efficiency this correlation has been exploited using linear prediction (Ramaswamy and Gopalakrishnan, 1998; Srinivasamurthy et al., 2000), where a given MFCC in a frame was predicted from the corresponding MFCC in the previous frame.[2] The prediction error $e_i = u_i - \alpha\hat{u}_{i-1}$ was quantized using uniform scalar quantization (USQ), where u_i is the current sample and $\hat{u}_{i-1}$ is the reconstruction of the previous sample generated by the coarse prediction loop. We chose USQ instead of a more complex quantizer because it offers the advantages of low complexity encoding, simple design (little training is required), good quantization performance[3] and additionally, given a desired rate, techniques are available to determine its optimal step size (Gray and Neuhoff, 1998). Furthermore, the importance of the MFCCs can be naturally incorporated in USQ quantization. It is well known that lower MFCCs are more important than higher MFCCs for speech recognition. Hence the USQ step size is set to be a multiple of the standard deviation of the MFCCs (this was motivated by the fact that the standard deviation of the lower MFCCs is smaller than that of higher MFCCs). This scheme implicitly implies a bit-allocation with lower MFCCs being allocated a higher portion of the total bitrate when compared to the higher MFCCs.[4] The USQ indices were losslessly encoded with a Huffman entropy coder. To achieve lower

[2] Single-step prediction seems a reasonable choice given that most of the time overlap occurs only between adjacent frames, and indeed our experiments showed that the gain in applying multi-stage prediction was limited.

[3] When combined with entropy coding, high rate USQ can theoretically achieve the same distortion as the best possible quantizer with only a 0.25 bits/sample penalty in rate (Gray and Neuhoff, 1998).

[4] Refer to our work in (Srinivasamurthy et al., 2003; Srinivasamurthy et al., 2004) for an alternative technique for bit allocation across MFCCs. These bit allocation techniques can be applied both to the scalable technique we propose here (Srinivasamurthy et al., 2003) and to the standard Aurora codec (Srinivasamurthy et al., 2004).

bitrate than that achievable by Huffman coding, a bitmap was transmitted to the decoder to indicate the position of the non-zero coefficients in every frame, and the non-zero coefficients were losslessly coded by the Huffman coder. A binary arithmetic coder could be used to encode this bitmap efficiently. However since the encoding complexity of arithmetic coding is high, run length coding was used. This encoder (coarse DPCM quantizer) is shown in the bottom part of Fig. 2 and constitutes the base layer of the proposed scalable encoder.

A straightforward method for generating the enhancement (refinement) layer would be to encode the residue, $u_i - \hat{u}_i$, after coarse DPCM quantization, where $\hat{u}_i$ is the reconstruction produced by the coarse DPCM quantizer for the current sample u_i. However, our initial experiments showed that encoding these residues was not very efficient. Instead, our system comprises two separate DPCM encoders, which perform quantization independently (with a coarse and fine quantizer, respectively) and where the redundancy between the two resulting sequences of quantization indices is reduced via entropy coding. More specifically, we use techniques that (i) eliminate impossible values using consistency criteria (Singh and Ortega, 1999; Rose and Regunathan, 2001) (see Section 2.1) and (ii) exploit statistical dependencies between the coarse and fine quantization indices using context-dependent entropy coding (see Section 2.2).

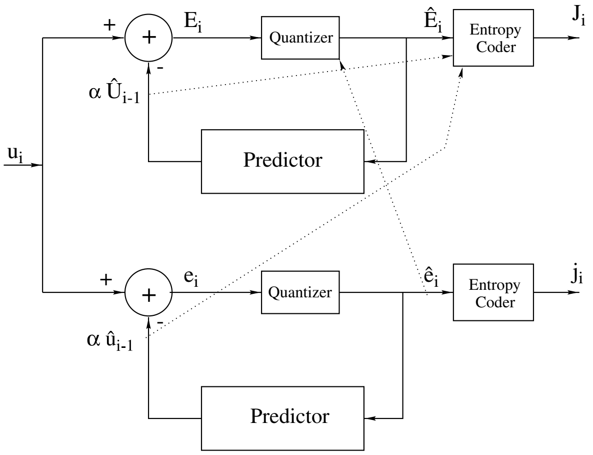

Fig. 2. The proposed scalable DPCM scheme, where two independent DPCM prediction loops are used. The quantized prediction error of the coarse DPCM quantizer, $\hat{e}_i$, is used by the fine DPCM quantizer to find the valid bins. The entropy coder uses both fine and coarse predictions, $\hat{u}_{i-1}$ and $\hat{U}_{i-1}$, to determine the context.

We now describe the design of the scalable predictive encoding scheme. It should be noted that this design technique is applicable not just to MFCCs but also to other sources with memory.

2.1. Consistency criteria for independent DPCM quantizers

Let u_i be an input sample (in our case this would be one of the MFCCs, all components of the MFCC vector are coded independently), and let e_i and E_i be the prediction errors of the coarse and fine DPCM quantizers, respectively (see Fig. 2). Then

$$\begin{aligned} &e_i = u_i - \alpha\hat{u}_{i-1}, \quad E_i = u_i - \alpha\hat{U}_{i-1} \\ &\quad \Rightarrow E_i = e_i + \alpha(\hat{u}_{i-1} - \hat{U}_{i-1}) \end{aligned} \tag{1}$$

where $\hat{u}_{i-1}$ and $\hat{U}_{i-1}$ are the reconstructed samples of the coarse and fine DPCM quantizers, respectively, and α is the prediction coefficient. Define $z_i \triangleq (\hat{u}_{i-1} - \hat{U}_{i-1})$. Given that $e_i \in [a_k, b_k]$ (i.e., e_i is quantized to the bin which spans the interval $[a_k, b_k]$), it is easy to see that E_i is constrained to belong to an interval I_c defined as follows:

$$E_i \in I_c = [a_k + \alpha z_i, b_k + \alpha z_i]. \tag{2}$$

Let B_f be the set of all quantization bins of the fine DPCM quantizer. Define $B_{f_i} \triangleq B_f \cap I_c \neq \phi$. It is apparent that only the bins in B_{f_i} are valid choices for the fine DPCM prediction error. This is illustrated in Fig. 3 where the interval I_c intersects 3 bins from B_f (highlighted). If Δ_c and Δ_f are the step sizes used in the coarse and fine DPCM quantizer, then the number of valid bins $|B_{f_i}|$ is at most $\lceil \frac{\Delta_c}{\Delta_f} \rceil + 1$. If $|B_{f_i}| \ll |B_f|$ then significant savings in bitrate can be achieved by encoding only those bins of the fine quantizer that are valid choices given the coarse quantizer.

2.2. Context-dependent entropy coding

Let j_i and J_i, be the quantization indices of the coarse and fine DPCM quantizers, respectively, at time i. Context information from previous coarse and fine reproductions can be used to reduce the entropy of the current fine DPCM prediction error. $\hat{U}_{i-1}$ and $\hat{u}_i$ have already been used to find the valid bins B_{f_i}. In addition, we can use the previous reconstructed value of the coarse DPCM quantizer $\hat{u}_{i-1}$ to select different a priori models for the probabilities of the fine quantization indices J_i. While $\hat{u}_{i-1}$ by itself does not provide explicit information, the

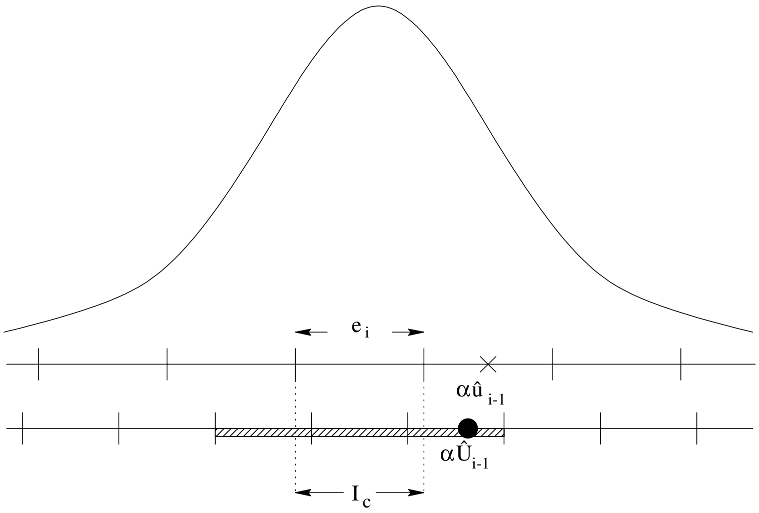

Fig. 3. Overlapping shifted quantizers. Using information from the coarse reproduction, the number of potential bins for the fine DPCM quantizer is reduced. Only those bins from B_f that overlap with the interval I_c are valid. The valid bins B_{f_i} are highlighted. With the consistency criterion the fine encoder only needs to signal to the decoder which among the valid fine quantization bins is the correct one, thus resulting in significant rate savings as compared to sending the fine DPCM bin independently of the coarse information. The predictor values $\alpha\hat{u}_{i-1}$ and $\alpha\hat{U}_{i-1}$ for the coarse and fine prediction loops are shown in the figure. Due to quantization these values will typically be different.

difference $|\hat{u}_{i-1} - \hat{U}_{i-1}|$ is useful. Consider the case when $|\hat{u}_{i-1} - \hat{U}_{i-1}|$ is small. Then if the coarse prediction error is quantized to zero ($j_i = 0$), it is highly likely that the fine prediction error will also be quantized to zero ($J_i = 0$) and, conversely if $j_i \neq 0$, it is highly likely that $J_i \neq 0$. On the other hand, when $|\hat{u}_{i-1} - \hat{U}_{i-1}|$ is large, then it is highly likely that $J_i \neq 0$. This information can be exploited by using context dependent entropy coding (Weinberger and Seroussi, 2000; Chrysafis and Ortega, 1997). We define two different contexts for J_i

(C1) $|\hat{u}_{i-1} - \hat{U}_{i-1}| \leqslant T_q$ and
$j_i = 0 \Rightarrow p(J_i = 0) \gg p(J_i \neq 0)$,
(C2) $|\hat{u}_{i-1} - \hat{U}_{i-1}| > T_q$ or
$j_i \neq 0 \Rightarrow p(J_i \neq 0) \gg p(J_i = 0)$.

This information can be exploited by using a bitmap. The more probable event in each context is always indicated in the bitmap by the same symbol, “0”: i.e., in context (C1) we transmit a “0” if $J_i = 0$ and a “1” otherwise and in context (C2) we transmit a “0” if $J_i \neq 0$ and a “1” otherwise. This will ensure that $p(0) \gg p(1)$ in the bitmap, which as before, we will encode using run length coding. Additionally, different Huffman encoders were used in the two different contexts.

Using the information from the coarse DPCM quantizer to find the valid bins, and the context information from $\hat{u}_{i-1}$ and $\hat{U}_{i-1}$, we were able to reduce the bitrate for the enhancement layer compared to encoding the enhancement layer independently. Specifically, the consistency criteria enabled about 26% reduction in bitrate and the context based entropy coding resulted in an additional 9.5% reduction. In the next section, we will describe the use of scalable feature encoding in a DSR context.

2.3. Packetization

In order to minimize the user latency (especially for continuous speech recognition), speech utterances are accumulated for short durations (e.g., 1 or 2 s). MFCCs are calculated and compressed for these accumulated speech segments; then they are packetized and transmitted. Transmission over error-prone transport channels will typically result in some packets being lost. To mitigate the effect of these losses, frame concealment techniques, such as insertion, interpolation and regeneration (Milner and Semnani, 2000; Mayorga et al., 2002) and unequal forward error correction techniques (Weerackody et al., 2002; Riskin et al., 2001) can be used. Now it is required that each packet be independently decodable. Hence, the prediction should be limited to only intra-packet MFCCs (i.e., the first MFCC vector of each packet is not predicted from the last MFCC vector of the previous packet). This will obviously result in some loss in compression efficiency. However, assuming each packet corresponds to 2 s of speech data, this implies the prediction loop is broken (to insert an independently coded frame) once every 80th MFCC. This will lead to a very small loss in compression efficiency.

3. Multi-pass DSR

The statistical, data-driven approach to ASR using HMMs and *N*-gram language models has been widely adopted given its adaptability to varying signal conditions and application domains. Depending on the scope of the application and the nature of the domain, often requiring the use of fairly complex acoustic and language models, various strategies are adopted to achieve a good operating balance between recognition performance and computational requirements. Note that in the DSR context an additional computational bottleneck at the servers may arise when providing simultaneous service to many clients. Multi-pass schemes have been effectively used to trade-off complexity and accuracy in ASR; we can extend this idea so that bitrate selection is also part of the trade-offs to consider in a multi-pass DSR system.

For example, consider a low-complexity initial recognizer (“coarse” processing) that operates on the coarsely quantized data and provides as an output a word lattice, an *N*-best hypothesis or a progressive search lattice (Murveit et al., 1993), which can then be used to reduce the search space of a more complex recognition scheme in a later phase. Possible initial recognition schemes include a reduced complexity HMM, where the HMM complexity is reduced by using smaller acoustic and/or language models and/or a narrower pruning beam. In a multi-pass recognition system, a more complex recognition strategy can be used to rescore the word lattice or the *N*-best hypothesis or the progressive search lattice to provide a more accurate decision, *but only if* the recognition confidence after the initial recognition pass was sufficiently low. This reduces the average complexity, as compared to using a single stage recognizer, since (i) the search space for the complex recognizer can be significantly reduced by the low complexity initial recognizer, and (ii) the complex recognizer is only used when the initial recognition pass did not provide an answer with a sufficiently high confidence level.

Let us consider now the additional potential advantages of the multi-pass DSR system depending on whether the key constraint in the application is to maintain *response latency* or to operate within *limited bandwidth* conditions.

3.1. Response latency

In a multi-pass system, the initial recognizer, which only attempts to narrow down the search space, can provide good performance with coarsely quantized data as an input, while the second, more complex, recognizer is more sensitive to the quality of the input data. Thus, when using a non-scalable encoder, the selected quantization quality has to be “good enough” for the second recognizer. Therefore, the first operates on an input that has better quality than strictly necessary, and thus *the bitrate provided to this recognizer is higher than needed.* Typically, in a networked environment each packet consists of several feature frames of the speech utterance. Additionally, the recognition operations are *pipelined*, hence the final recognition stage depends on the results of the initial stage. This can potentially result in higher response latency, because the recognizer can get the data from the network layer only after the packet has been received in its entirety.

In contrast, a scalable encoder can reduce the latency by exploiting this pipelining. The data required by the different stages are transmitted *simultaneously*, but in different layers, i.e., the base layer contains the data required by the initial recognizer and the enhancement layer contains the additional data required by the final recognizer. Since the initial recognizer is simple it can be assumed to be much faster than a more sophisticated scheme. As the base and enhancement layer data are being received, based on the intermediate results of the initial recognizer, the search space of the final recognizer can be constrained. Since the enhancement data has also been made available, the final recognizer can proceed with its recognition. However, the bitrate required by the base layer will be significantly lower than that required for transmitting the entire data.[5] When the main bottleneck in the system is a very slow link used for communication

[5] For the same recognition performance the base layer bitrate was 60% of the bitrate required for the entire data (see Section 6.1).

between the client and server this reduction in bitrate will enable the initial recognizer, and consequently the final recognizer, to complete faster, as compared to the case of a non-scalable encoder. However, the client and server bandwidth requirements will be increased as both layers always have to be transmitted together to the server.

3.2. Client and server bandwidth

In bandwidth constrained situations the layers are transmitted *sequentially*. Initially only the base layer is transmitted to the server. After the first recognition stage, if required (depending on the recognition confidence), the enhancement layer is requested from the client. By this procedure, both client and server bandwidths requirements can be kept low. However the absolute delay can be high (for cases when the recognition confidence after the first stage is low or when the round trip delay is large).

4. Example multi-pass DSR systems

We consider two different recognition experiments to illustrate the performance of our proposed scalable encoding in the context of DSR, namely, (i) a *large* spoken names recognition task where a free phone loop HMM recognizer is used as the first stage and a lattice based HMM recognizer is used in the second stage, and (ii) a continuous speech recognizer (CSR) for the HUB-4 broadcast news task, where a bigram language model recognizer was used as the initial state and a lattice based trigram recognizer as the second state. In both DSR systems, the client employed a scalable encoder to compress the MFCC features.

4.1. Spoken names recognition

The spoken names recognition task (Behet et al., 2001; Gao et al., 2001) is used, among others, in network based applications such as directory assistance and caller identification. In these applications the list of names tends to be quite large, in the order of hundreds of thousands. Variability in pronunciation further increases the perplexity. The traditional approach to name recognition has been to use a finite state grammar (FSG), where all the names (with all possible pronunciation variants) are alternate paths for recognition. For a spoken name utterance the recognizer evaluates all possible paths and selects the name corresponding to the most likely path. As the names list grows it is evident that the computational complexity increases. To ensure reduced complexity, extensive pruning will be required which will result in recognition performance degradation. Attempts at weighting certain names higher by using some prior statistics is possible but limits the task scope.

An alternative approach with reduced computational complexity but acceptable recognition performance is to adopt a two stage recognizer with dictionary lookup (Sethy et al., 2002). Fig. 4 illustrates this approach. Such a two stage approach has also been used for spelled name retrieval (Junqua, 1997), information retrieval (Coletti and Federico, 1999) and complexity reduction of a phone based continuous speech recognition system (Abe et al., 1999). For the spoken names recognition task, the first stage is a low complexity bigram phone loop which is used to identify the *N*-best phone sequence corresponding to the input utterance. The next step involves a string match, where each of the *N*-best phone sequences is compared to the entries in a dictionary. The utterances corresponding to phone sequences which have a distance less than a given threshold (the threshold is usually chosen as a function of the number of phones in the recognized phone sequence) from the recognized *N*-best phone sequence are selected to generate a lattice. The final stage involves rescoring this generated lattice using more complex acoustic (triphone) models.

The accuracy obtained by using a two stage recognizer for the spoken names task is comparable to the conventional single stage FSG based approach as shown in (Sethy et al., 2002) but results in significant savings in complexity, since the lattice only consists of a subset of the entire names list. The dictionary used for lookup is a names dictionary which consists of all possible names along with their pronunciations. In our experiments we used the Levenshtein (or edit) distance during dictionary lookup to compute the string match distance between phone sequences.[6] The Levenshtein distance between phone sequences p_1 and p_2, $LD(p_1,p_2)$, is the minimum cost associated in transforming p_1 into p_2 by deletions, insertions and

[6] More sophisticated distances, for e.g., including the phone confusion matrix scores to weight the Levenshtein distance can also be incorporated.

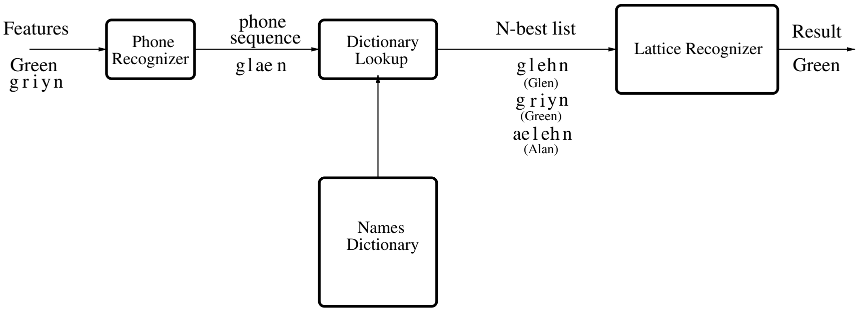

Fig. 4. A two stage approach using dictionary lookup for the names recognition task. When a Levenshtein distance threshold of 2 is used, although the recognized phone sequence is closer to Glen than Green, the dictionary lookup will ensure that Green is in the final lattice used for rescoring.

substitutions. This two stage names recognition procedure is summarized below.

Algorithm 1 (*Multi-pass Spoken Names Recognition*).

```
Step 1:  Identify the N-best phone sequences $p_r^n$ for the name utterance using a bigram phone loop, $n = 0, 1, \ldots, N - 1$.
Step 2:  Find $T_n$ corresponding to $p_r^n$ from Table 1, $n = 0, 1, \ldots, N - 1$.
Step 3a: Initialize $i = 0$.
Step 3b: For name $i$ in the names dictionary find the corresponding phone sequence $p_i$.
Step 3c: If $LD(p_r^n, p_i) < T_n$, for any $n = 0, 1, \ldots, N - 1$, add name $i$ to the names lattice.
Step 3d: If there are more names in the dictionary set $i = i + 1$ and go to Step 3b else go to Step 4.
Step 4:  Rescore the names lattice using context-dependent models to get the final result.
```

This two stage names recognition procedure can be efficiently combined with the proposed scalable encoder by using the base layer with the bigram phone recognizer and the enhancement layer with the lattice rescoring stage to reduce the overall latency.

Table 1
The threshold used during dictionary lookup is a function of the recognized phone sequence length

Length of phone sequence	Threshold T_n
Less than 4	3
4 or 5	4
Greater than 5	5

Shorter phone sequences are assigned a lower threshold and vice-versa. Thresholds were chosen by experiments on the training data.

4.2. A two-stage continuous speech recognition

As a second example, we consider a more standard continuous speech recognition task using the HUB4 data. Many practical speech recognition applications are impeded from using advanced speech recognition techniques due to their intensive computation/memory complexities. Simpler speech recognition technologies can instead be used. However, they result in recognition performance degradations. Similar to the two stage recognizer for the spoken names task, a multi-pass recognition scheme has been proposed (Murveit et al., 1993) which enables improved speed/accuracy tradeoff by using progressive search techniques. These techniques use an "early-pass" reduced complexity speech recognizer to reduce the search space of a "later-pass" more accurate but complex speech recognizer. This procedure can be repeated iteratively, with each stage result used to constrain the search space for the next stage.

The early-stage recognizer builds a word lattice[7] containing several most likely word sequences using its low complexity models. The latter-stage uses this word lattice to constrain its search space and find the most likely word sequence hypothesis using its more complex models. Maintaining a reasonable sized word lattice at the early stage typically ensures that the "true" most likely word sequence is

[7] We have used the early-pass recognizer to build word lattices. However, our proposed system can also use progressive search lattices (Murveit et al., 1993).

included in the word lattice, while only unlikely word sequences are discarded. Thus the word lattice, while constraining the search pass of the later-pass recognizer, does not significantly degrade its recognition performance. Hence, this multi-pass procedure ensures that *simultaneously* good recognition performance and reduced decoding time are achieved, making them ideal for use in practical speech recognition applications. Fig. 5 illustrates a multi-pass recognition system consisting of two stages.

The multi-pass recognition procedure when a scalable encoder is used at the client is summarized below.

Algorithm 2 (*Multi-pass Distributed Continuous Speech Recognition*).

```
Step 1: Use the base layer data with the early-pass
        low complexity speech recognizer.
Step 2: Recognize the utterance and dump the word
        lattice corresponding to it.
Step 3: Use the enhancement layer data and rescore
        the word lattice with the high complexity
        latter-pass speech recognizer.
        (a) Update the acoustic probabilities using
            the enhancement layer data.
        (b) Update the language probabilities using
            the more complex LMs.
        (c) Find the most likely word sequence.
```

If more than 2 stages are used in the multi-pass recognizer, the second stage also provides a (more refined) word lattice. The third stage uses this word lattice along with an additional enhancement layer and so on. It should be noted that the number of recognition stages required depends on the performance requirements for a given application, and the confidence in the recognition results of a given stage such as for example determined by confidence scores. As a consequence, it is possible that just a single stage recognition with base layer data may be adequate in some scenarios.

5. Experimental setup

An HMM-based recognizer (HTK) was used to test the scalable DSR encoder developed in Section 2. In all our experiments 12 MFCCs and the zeroth cepstral coefficient were extracted at the client using an overlapping Hamming window 25 ms long, with adjacent windows separated by 10 ms. The Δ and $\Delta\Delta$ coefficients were derived at the server from the decoded MFCCs.

5.1. Spoken names recognizer

Two sets of acoustic models (American English) were considered: for context-independent (CI) models, 3 stage left to right HMMs with 8 Gaussian mixtures per state were trained, and for context-dependent (CD) models 3 stage left to right HMMs with 4 Gaussian mixtures per state were trained. The number of CI and CD models were 46 and 6600, respectively. The speech corpus used for testing was the OGI NAMES corpus (CSLU OGI). This is a collection of name utterances spoken by different speakers over the telephone. The spoken names data used represent only first or last names (and not the full name). The speech utterances having been collected under realistic acoustic conditions exhibit significant environmental variations and almost every name is spoken by a different person. For

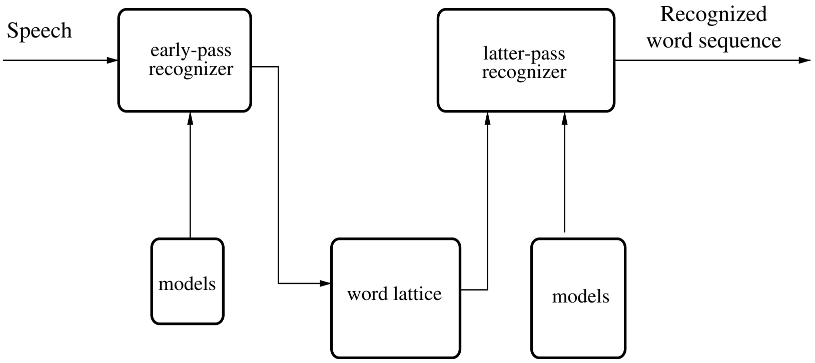

Fig. 5. A two stage multi-pass continuous speech recognizer. The low complexity early-pass recognizer is used to constrain the search space of a latter-pass more complex recognizer by generating a word lattice.

our experiments we used 4619 names of which 3498 were unique (i.e., 1121 names had multiple pronunciations). Spoken names (6356) were used for training and 3000 different spoken names were used for testing. The names dictionary we used had 100,000 name entries. The top two phone sequences from the phone recognizer were used by the dictionary lookup to generate the lattice of names for rescoring.

5.2. Continuous speech recognizer for the HUB-4 broadcast news task

The acoustic models were 1128 context-dependent models trained as 3 stage left to right HMMs with 8 Gaussian mixtures per state. The speech corpus used for training was the 1994 HUB-4 speech corpus. Speech (2200 s) from the 1995 HUB-5 speech corpus were used for testing.[8] Both the train and test speech utterances contain significant variability (background music, different recording conditions and speakers), making this a difficult recognition task. The vocabulary size was 1300 words. The language models were the bigram and trigram broadcast news LMs provided by CMU. The number of bigrams and trigrams were 427×10^3 and 1.8×10^6 respectively. The early-pass recognizer employed a bigram LM and used the base layer data. The word lattices were generated as HTK SLF files by retaining the 20 best word sequences and using 5 tokens at every state (Young et al., 2000). The word lattice was used as the network for recognition by the latter-pass recognizer. The acoustic probabilities were updated using the enhancement layer data and the language probabilities were updated by rescoring the word lattice with a trigram LM.

6. Results for the multi-pass DSR system

6.1. Spoken names recognition

When the proposed scalable DSR encoder is used at the client, a base layer and an enhancement layer are transmitted to the server for every name utterance. The bigram CI phone loop uses the base layer to generate the *N*-best phone sequence. This is used by the dictionary lookup to build the list of names for the lattice (FSG) recognizer. The lattice recognizer rescores the names list using the enhancement layer data to get the final recognized name result. Note that the phone recognizer and the dictionary lookup need not wait for the enhancement layer data to be received. The average number of names in the lattice when compressed data was used was approximately 1140, which is the same when uncompressed data was used, i.e., compression did not increase the lattice size. However, note that this lattice size 1140 is significantly smaller than the number of names in the dictionary (100,000).

The recognition results obtained with the above procedure for the names task are shown in Fig. 6. Transparent recognition performance (i.e., the recognition performance was the same as that achieved with uncompressed data) with CD models was achieved when the base layer rate was 2580 b/s and the enhancement layer was 2000 b/s. To achieve transparent recognition with our proposed encoder operating with a single layer (i.e., in a non-scalable mode) the required bitrate is 4040 b/s (in this case this single layer of encoded data at 4040 b/s is used by both recognition stages). Let the total user latency be defined as the sum of transmission time and recognition time. The use of scalable coding enables us to lower the user latency. Assume each name utterance is put into a single packet (non-scalable codec) or two packets (scalable codec with two layers). Table 2 shows the transmission times of the packets for different utterance lengths when using an 8800 b/s transmission link (8800 b/s is the maximum bitrate used by a speech codec on the fundamental channel (FCH) in a cdma2000© mobile network). It can be seen that the transmission time required for the base layer of the scalable codec is lower than the time required for the non-scalable codec. Hence, the first stage recognizer can be started earlier in the scalable case. This ensures that the second stage (lattice recognizer) can also complete faster in systems employing a scalable coder while achieving the same recognition performance. To illustrate the reduction in user latency, consider the transmission times shown in Table 2. Specifically consider the case when we are using a multi-pass scheme for both scalable and non-scalable encoding cases, and the length of the name utterance is 1 s. For systems employing a non-scalable encoder, the time required to transmit the data from the client to server is 0.5 s. Hence, the first stage recognizer can only be started 0.5 s after the user finishes speaking. If we assume, without loss of generality, the speech recognizer works in real time (i.e.,

[8] Utterances with significant speech and music overlap were eliminated from the test set. A few examples of the test utterances are available at http://biron.usc.edu/~snaveen/speech_examples.

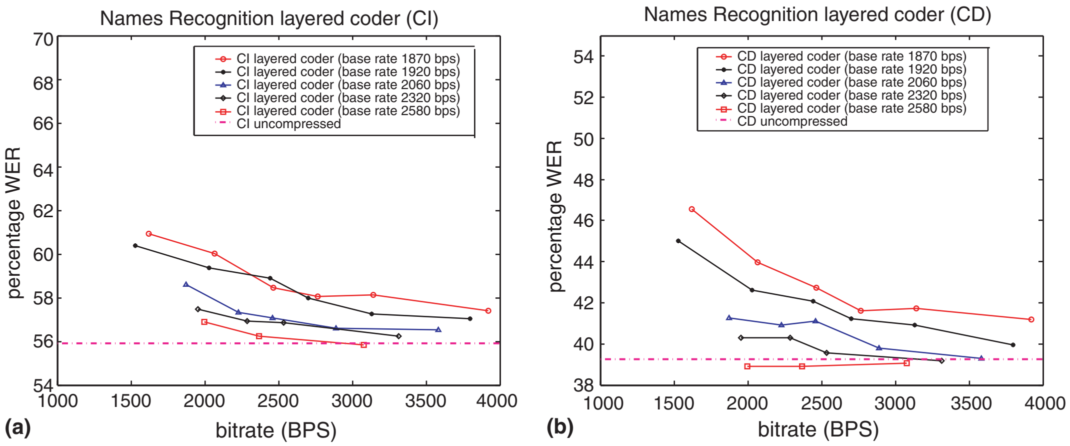

Fig. 6. Results for the names recognition task when the proposed scalable DSR encoder is used at the client. The results are shown for different base layer rates. When the base layer rate is 2580 b/s and the enhancement layer rate is 2000 b/s the recognition result for CD models is the same as that obtained with the proposed encoder operating in a non-scalable mode at a bitrate of 4040 b/s. However since the base layer rate is lower than the full rate of the non-scalable encoder, the recognition latency is reduced. (a) CI models and (b) CD models.

Table 2
Transmission times for scalable and non-scalable codec on an 8800 b/s transmission link

Length of name utterance (s)	0.5	1	2
Non-scalable codec	0.27	0.50	0.95
Scalable codec (base layer)	0.18	0.33	0.62
Percentage reduction in user latency	10.8	11.1	11.2

The recognizer is assumed to operate in real-time. Hence time required for recognition is equal to the length of the name utterance. The user latency is the sum of the transmission and recognition time. Reduction in user latency for the proposed scalable encoder is shown in the last row. Around 11% user latency reduction is possible. Each packet is assumed to have 40 bytes of RTP/UDP/IP headers. If RoHC (Robust Header Compression) were used the header size could be reduced to 5 bytes.

it takes 1 s to recognize an utterance of length 1 s), then the recognizer completes recognizing the utterance 1 s after reception of the data. Hence, the total delay as experienced by the user is 1.5 s. However, when we use the scalable codec, the base layer is transmitted in 0.33 s. Now the first stage recognizer can be started 0.33 s after the user finishes speaking. As the first stage recognizer is working the server can receive the enhancement layer. Again assuming the same real time recognition, the total delay for the user is now 1.33 s. Hence, by using a scalable codec we could reduce the user latency by 0.17 s. For different lengths of the name utterance it was observed that on average around 11% reduction in user latency can be achieved. The successive transmissions in the scalable case, can be pipelined, if needed.

Table 3 shows the degradation in speech recognition performance caused due to compression when compared to using uncompressed MFCCs, i.e., it shows the increase in WER when recognition was performed using compressed MFCCs when compared to uncompressed MFCCs. Results are shown for the ETSI standard DSR encoder *Aurora* and the proposed scalable encoder. For the proposed encoder we show results when two layers are used and also when only the base layer is used. The rates for the proposed encoder when two layers are used is the combined (i.e., base layer plus enhancement layer) rate. From the table we can see that the increase in WER for CD models when *Aurora* is used is 2.04%. For the proposed encoder, the increase in WER is 2.64% and 0.0% when both layers are used and only base layer used, respectively. Hence, we observe that the proposed encoder

Table 3
Relative percentage increase in WER for the proposed encoder and *Aurora* in the spoken names task

Encoding technique	CI	CD	Rate (b/s)
Aurora	1.54	2.04	4400
Proposed scalable	2.84	2.64	4270
Encoder (both layers)	0.63	0.00	4950
Proposed scalable	1.65	1.48	3260
Encoder (only base layer)	0.57	0.00	4040

Observe that the rate for the proposed scalable encoder is the total rate (base + enhancement). Generating multiple layers results in compression inefficiency. In spite of this, the rate-recognition performance of the proposed scalable encoder is comparable to the non-scalable *Aurora* encoder.

has better rate-recognition performance when compared to *Aurora*. We also observe that although scalability results in compression inefficiency, the proposed *scalable* encoder has comparable rate-recognition performance to the *non-scalable Aurora* encoder.[9]

6.2. Continuous speech recognition

When the proposed scalable DSR encoder is used at the client, the bigram recognizer uses the base layer to generate a word lattice. This word lattice is rescored by a trigram recognizer which in addition uses the enhancement layer to update the acoustic probabilities. As in the spoken names task, the initial bigram recognizer does not have to wait for the enhancement layer data to be received.

The recognition performance achieved for the two stage multi-pass recognizer for several different base layer rates is shown in Fig. 7. Observe that with a base layer rate of 2470 b/s and an enhancement layer rate of 3230 b/s, the recognition performance achieved is the same as that achieved with uncompressed data. To achieve transparent recognition with our proposed encoder operating with a single layer (i.e., in a non-scalable mode) the required bitrate is 5240 b/s. Hence we observer that, while using a scalable encoder provides more flexibility in the overall system design it only results in an 8.7% increase in bitrate.

The trade-off between bitrate and recognition performance is also clear from the figure. For the base layer rate of 2470 b/s, reducing the bitrate from 3230 b/s to 1880 b/s (42% reduction) results in only a 2.7% relative increase in WER. In Figs. 6 and 7, the recognition performance with compressed data is sometimes better than with uncompressed data. The difference in recognition performance for these points from the uncompressed recognition performance is not statistically significant.

An interesting observation from the results is that the recognition performance plateaus at different WERs for different base layer rates (with the plateau being higher for lower base layer rates). This indicates that a certain minimum data fidelity is required at the initial recognition stage, below which improving the data fidelity only at the later stages does not enable the system to achieve the same recognition performance as that achieved by uncompressed data, no matter how high the enhancement layer bitrate is made. This provides a guideline for selection of bitrates for the base and enhancement layers. Given that a particular recognition performance is required for the task, this implicitly decides the minimal bitrate that can be used for the base layer. For example in the CSR task if the desired recognition performance of the overall system has to be as good as that achievable with uncompressed features then, from Fig. 7, we observe that the base layer rate has to be greater than 2470 bps. However, this also implies that given the constraint that a particular rate has to be used for the base layer (this can be due to application/channel constraints) then there is no necessity to use a (high) rate for the enhancement layer which lies beyond the knee of the recognition performance curve corresponding to that particular base layer rate. For example, at a base layer rate of 2250 b/s there is no gain in increasing the enhancement layer rate beyond 1940 b/s.

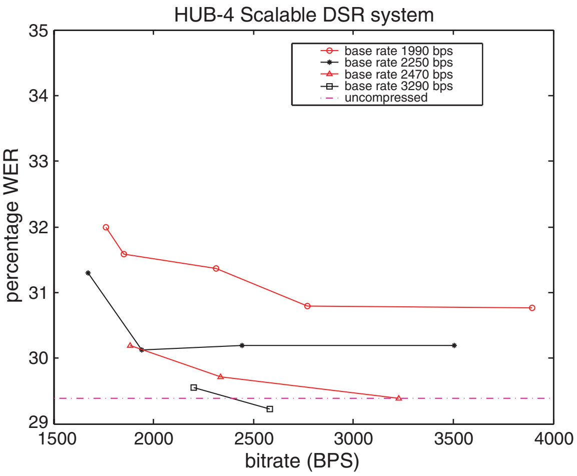

Fig. 7. Recognition results of the multi-pass DSR for the HUB-4 broadcast news task. Observe that with a base layer rate of 2470 b/s and an enhancement layer of 3230 b/s we achieve the same recognition performance as with uncompressed data. Also the recognition performance versus bitrate trade-off is clear from the results.

Fig. 8 shows both the distortion due to compression and the percentage WER for different bitrates. It is clear that as the bitrate is increased the distor-

[9] *Aurora* encodes both energy and C0. But, only C0 has been used in the recognition experiments. If the energy component was eliminated the bitrate could be reduced. For e.g., if we assume 6 bits are required to encode C0, then the bitrate required by *Aurora* would be 4200 b/s. When only the base layer of the proposed encoder was used at both stages, the recognition performance at 4040 b/s was better than that achieved by the *Aurora* encoder.

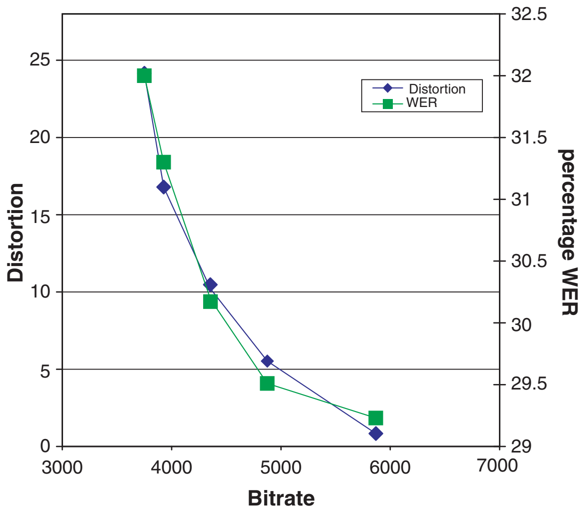

Fig. 8. Comparison of trade-off between distortion and percentage WER versus bitrate. Distortion scale is shown on the left *y*-axis and percentage WER scale is shown on the right *y*-axis. The correlation between distortion and recognition performance is clearly illustrated.

tion is reduced. Similarly, the percentage WER also decreases with increase in bitrate. The two curves have a similar tradeoff against bitrate. This shows that there is a strong correlation between distortion and percentage WER. This relationship can be used to predict the impact compression has on recognition performance.

Another interesting observation from Figs. 6 and 7 is that the bitrate required to ensure that speech recognition performance is not degraded due to compression is about 4600 b/s for the spoken names task and 5700 b/s for the CSR task. Additionally, it has been shown that the approximate minimum bitrate for transparent operation for an isolated digits task was 1100 b/s (Srinivasamurthy et al., 2001a) and for a connected digits task was 2000 b/s (Srinivasamurthy et al., 2001b). This illustrates that the minimum bitrate for transparent speech recognition is strongly task dependent. In general, more complex speech recognition tasks require higher bitrate. It will be interesting to analytically quantify the minimum bitrate requirement for different speech recognition tasks. We leave this as potential future work.

7. Conclusions

In this paper we addressed the problem of speech encoding for a DSR system. We showed that using speech encoders optimized for recognition rather than perceptual distortion provide better rate-recognition performance. To handle the practical scenario of large client density which can severely overload a DSR server, a multi-pass DSR system which provides trade-offs between bitrate, complexity and recognition performance is desirable. It was shown that there is more flexibility in the operation of the DSR system when a scalable coder is used at the client. To provide a multi-resolution compressed stream it is more convenient to work with feature encoders, as the client now has greater flexibility in directly controlling the input to the HMMs. We showed that the proposed multi-pass recognizer combined with a scalable feature encoder can provide flexibility in adapting the DSR system to the changing bandwidth requirements and server load while achieving the best recognition performance possible. It also has the additional advantage of reducing the recognition latency. We illustrated the multi-pass DSR approach on two varied complex tasks: a large spoken names task and a continuous speech recognition task.

By addressing the DSR system optimization at the encoder, at the recognizer and at the system level, we can ensure that the system operation can be made highly robust to user, device, channel and network conditions. The proposed schemes gives us more freedom to fine tune the system, enabling the DSR system to adapt and provide the desired QoS to the user while constantly adjusting to the network and server conditions.

Acknowledgements

The authors thank Abhinav Sethy for discussions regarding the names recognition task.

References

Abe, Y., Itsui, H., Maruta, Y., Nkajima, K., 1999. A two-stage speech recognition method with an error correction model. In: Eurospeech'99, Budapest, September.

Behet, F., de Mori, R., Subsol, G., 2001. Very large vocabulary proper name recognition for directory assistance. In: IEEE Internat. Conf. on Automatic Speech Recognition and Understanding, pp. 222–225.

Bernard, A., Alwan, A., 2001. Source and channel coding for remote speech recognition over error-prone channel. In: ICASSP 2001, Vol. 4.

Chazan, D., Cohen, R.H.G., Zibulski, M., 2000. Speech reconstruction from mel-frequency cepstral coefficients and pitch frequency. In: IEEE ICASSP 2000.

Chrysafis, C., Ortega, A., 1997. Efficient context-based entropy coding for lossy wavelet image compression. In: DCC, Data Compression Conference, Snowbird, Utah, March.

Coletti, P., Federico, M., 1999. A two-stage speech recognition method for information retrieval applications. In: Eurospeech'99, Budapest, September.

"Names 1.1." The CSLU OGI Names corpus, http://cslu.cse.ogi.edu/corpora/names.

Digalakis, V.V., Neumeyer, L.G., Perakakis, M., 1999. Quantization of cepstral parameters for speech recognition over the world wide web. IEEE J. Select. Areas Comm. 17 (January), 82–90.

Equitz, W., Cover, T., 1991. Successive refinement of information. IEEE Trans. Inform. Theory 37 (March), 269–275.

Gales, M., Young, S., 1996. Robust continuous speech recognition using parallel model combination. IEEE Trans. Acoust. Speech Signal Process. (September), 352–359.

Gao, Y., Ramabhadran, B., Chen, J., Erdogan, H., Picheny, M., 2001. Innovative approaches for large vocabulary name recognition. In: ICASSP 2001, Vol. 1, pp. 53–56.

Gray, R.M., Neuhoff, D.L., 1998. Quantization. IEEE transactions on information theory 44 (October), 2325–2383.

Huerta, J.M., 2000. Speech recognition in mobile environments. Ph.D. thesis, Department of Electrical and Computer Engineering, Carnegie Mellon University, April.

Junqua, J.-C., 1997. SmarTspelL™: A multipass recognition system for name retrieval over the telephone. In: IEEE Trans. on Speech and Audio Processing, March, Vol. 5, pp. 173–182.

Kim, H.K., Cox, R., 2000. Bitstream-based feature extraction for wireless speech recognition. In: ICASSP-2000, Vol. 3, pp. 1607–1610.

Kiss, I., 2000. A comparison of distributed and network speech recognition for mobile communication systems. In Internat. Conf. on Spoken Language Processing 2000 (ICSLP 2000).

Kiss, I., Kapanen, P., 1999. Robust feature vector compression algorithm for distributed speech recognition. In: Eurospeech 1999.

Lilly, B.T., Paliwal, K.K., 1996. Effect of speech coders on speech recognition performance. In: ICSLP 96, Philadelphia, PA, pp. 2344–2347.

Mayorga, P., Lamy, R., Besacier, L., 2002. Recovering of packet loss for distributed speech recognition. In: Eusipco 2002, Toulouse, France, September.

Milner, B., Semnani, S., 2000. Robust speech recognition over ip networks. In: ICASSP 2000, Istanbul, Turkey, June.

Murveit, H., Butzberger, J., Digalakis, V., Weintraub, M., 1993. Large vocabulary dictation using SRIs DECIPHER™ speech recognition system: Progressive search techniques. in ICASSP-93, April, Vol. II, pp. 319–322.

Ramabadran, T., Meunier, J., Jasiuk, M., Kushner, B., 2001. Enhancing distributed speech recognition with back-end speech reconstruction. In: Eurospeech 2001, Aalborg, Denmark, September.

Ramaswamy, G.N., Gopalakrishnan, P.S., 1998. Compression of acoustic features for speech recognition in network environments. In: IEEE ICASSP 1998, pp. 977–980.

Riskin, E., Boulis, C., Otterson, S., Ostendorf, M., 2001. Graceful degradation of speech recognition performance over lossy packet networks. In: Eurospeech 2001, Aalborg, Denmark, September.

Rose, K., Regunathan, S., 2001. Toward optimality in scalable predictive coding. IEEE Trans. Image Process. 10 (July), 965–976.

Sethy, A., Narayanan, S., Parthasarathy, S., 2002. Syllable-based recognition of spoken names. In: ISCA Pronunciation Modeling and Lexicon Adaptation Workshop.

Singh, R., Ortega, A., 1999. Erasure recovery in predictive coding environments using multiple description coding. In: IEEE Workshop on Multimedia Signal Processing.

Speech processing, transmission and quality aspects (STQ); distributed speech recognition; front-end feature extraction algorithm; compression algorithms. Tech. Rep. Standard ES 201 108, European Telecommunications Standards Institute (ETSI), April 11 2000.

Srinivasamurthy, N., Ortega, A., Zhu, Q., Alwan, A., 2000. Towards efficient and scalable speech compression schemes for robust speech recognition applications. In: ICME 2000, New York, NY, July.

Srinivasamurthy, N., Ortega, A., Narayanan, S., 2001a. Efficient scalable speech compression for scalable speech recognition, in Eurospeech 2001, Aalborg, Denmark, September 2001.

Srinivasamurthy, N., Narayanan, S., Ortega, A., 2001b. Use of model transformations for distributed speech recognition. In: ISCA ITR-Workshop. Adaptation Methods for Speech Recognition, Sophia-Antipolis, France, August 2001.

Srinivasamurthy, N., Ortega, A., Narayanan, S., 2003. Towards optimal encoding for classification with applications to distributed speech recognition. In: Proc. Eurospeech 2003, Geneva, Switzerland, September.

Srinivasamurthy, N., Ortega, A., Narayanan, S., 2004. Enhanced standard compliant distributed speech recognition (aurora encoder) using rate allocation. In: Proc. of ICASSP 2004, Montreal, Canada, May.

Turunen, J., Vlaj, D., 2001. A study of speech coding parameters in speech recognition. In: Eurospeech 2001, Aalborg, Denmark, September.

Weerackody, V., Reichl, W., Potamianos, A., 2002. An error-protected speech recognition system for wireless communications. IEEE Trans. Wireless Comm. 1 (April), 282–291.

Weinberger, G.S.M., Seroussi, G., 2000. The LOCO-I lossless image compression algorithm: principles and standardization into JPEG-LS. IEEE Trans. Image Process. 9 (August), 1309–1324.

Woodland, P., Hain, T., Evermann, G., Povey, D., 2001. CU-HTK march 2001 hub5 system. In: LVCSR Hub5 Workshop 2001, Linthicum Heights, May.

Young, S., Kershaw, D., Odell, J., Ollason, D., Valtchev, V., Woodland, P., 2000. The HTK Book (for htk version 3.0). Available from: <htk.eng.cam.ac.uk/prot-docs/HTKBook/htkbook.html>, July.

Zhu, Q., Alwan, A., 2001. An efficient and scalable 2D DCT-based feature coding scheme for remote speech recognition. In: ICASSP 2001, Vol. 1.