

# Survey of the Speech Recognition Techniques for Mobile Devices

Dmitry Zaykovskiy

Department of Information Technology  
University of Ulm  
Ulm, Germany

dmitry.zaykovskiy@uni-ulm.de

## Abstract

This paper presents an overview of different approaches for providing automatic speech recognition (ASR) technology to mobile users. Three principal system architectures in terms of employing wireless communication link are analyzed: *Embedded Speech Recognition Systems*, *Network Speech Recognition (NSR)* and *Distributed Speech Recognition (DSR)*. Overview of the solutions which became standards by now as well as some critical analysis of the latest developments in the field of the speech recognition in mobile environments is given. Open issues, pros and cons of the different methodologies and techniques are highlighted. Special emphasis is made on the constraints and limitations the ASR applications encounter under the different architectures.

## 1. Introduction

The past decade has witnessed an unprecedented developing of the telecommunication industry. Market researches report that there are more than 1.6 billion mobile phone subscribers worldwide as of 2005 and this amount is expected to grow up to 2 billion by the end of 2006. The today's mobile technologies have far overcome person-to-person communication. The 2.5G networks supporting packet switched data exchange, such as GPRS with realistic bit rates of 30–80 kbit/s, have become an everyday matter. Whilst the networks of the 3rd generation, like UMTS or CDMA2000, are already operated in 72 countries, have 32 millions users (middle 2005) and continue to spread. These networks provide an effective data transfer rate up to 384 kbit/s.

At the same time the wireless local area networks (WLANs) based on the IEEE 802.11 specifications also known as Wi-Fi spots became widely available. This is a technology, which enables a person with a wireless-enabled computer or personal digital assistant (PDA) to communicate being within the coverage of an access point. With rates up to 11 Mbit/s Wi-Fi makes possible such applications as Voice over IP or video conferencing.

Alongside with expansion of the network technologies, the client devices have been developing at the same speed. Nokia has forecasted that by the end of 2006 one sixth of all cellular phones will be UMTS supporting devices. Also PDAs are getting more and more popular. For example, according to Gartner's study, the overall market for PDAs has grown by 20.7% in the third quarter of 2005, compared to 2004.

Of course such a perfect infrastructure gave rise for the development of many new data services for the handheld devices. However, the user interface, which has definitely improved over the last years, still limits the usability of the mobile devices. The

main interface problem of handheld gadgets is their miniature size. Typing on such tiny keyboards or pointing with stylus is very uncomfortable and error prone. Another issue is that PDA are often used when person is really "on the move". Operating in such conditions is either impeded or even prohibited, e.g. in the case of car driving.

The natural way to solve this problem consists in using speech recognition technology. Speech input requires neither visual nor physical contact with devices. It can serve as an alternative interface to the regular one or be a complement modality speeding up the input process and increasing its fidelity and convenience.

In the last decade a substantial effort has been invested in the automatic speech recognition techniques. As a result fast, robust and effective speech recognition systems have been developed [1–3]. The modern state-of-the-art ASR systems provide the performance quality (usually assessed by the recognition word error rate (WER) and by the ratio of the processing time to the utterance duration), which affords the comfortable use of ASR in real applications, c.f. Table 1.

However, the direct reproduction of the algorithms suitable for the desktop applications is either not possible or mostly leads to unacceptable low performance on the mobile devices. Due to the highly variable acoustic environment in the mobile domain and very limited resources available on the handheld terminals the implementation of such systems on the mobile device necessitates special arrangements [4]. In this paper we address the system optimization techniques, which enable the speech recognition technology on the portable computing devices.

The remainder of the paper is organized as follows. The process of the speech recognition from the perspective of the handheld device is presented in section 2. Sections 3, 4 and 5 draw in detail three principal system architectures: *client-based*, *server-based* and the *client-server* ASR. Section 6 summarizes and closes the discussion.

Table 1: Recognition rates of the state-of-the-art desktop ASR systems [2]

Task	Words	WER %	Real Time (RT)	
			1-CPU	2-CPU's
Connected Digits	11	0.55	0.07	0.05
Resource Manag.	1'000	2.74	0.50	0.40
Wall Street Journal	5'000	7.17	1.22	0.96

## 2. Architectures of ASR Systems for Mobile Devices

In this section we briefly describe the functional blocks of the current state-of-the-art speech recognition engines with their impact on the design of the ASR systems for mobile applications.

### 2.1. Basics of Automatic Speech Recognition

The goal of the ASR system is to find the most probable sequence of words  $W = (w_1, w_2, \dots)$  belonging to a fixed vocabulary given some set of acoustic observations  $\mathbf{O} = (\mathbf{o}_1, \mathbf{o}_2, \dots, \mathbf{o}_T)$ . Following the Bayesian approach applied to ASR [5] the best estimation for the word sequence can be given by

$$W^* = \arg \max_W P(W|\mathbf{O}) = \arg \max_W \frac{P(\mathbf{O}|W)P(W)}{P(\mathbf{O})}. \quad (1)$$

In order to generate an output the speech recognizer has basically to perform the following operations:

- extract acoustic observations (*features*) out of the spoken utterance;
- estimate  $P(W)$  - the probability of individual word sequence to happen, regardless acoustic observations;
- estimate  $P(\mathbf{O}|W)$  - the likelihood that the particular set of features originates from a certain sequence of words;
- find word sequence that delivers the maximum of (1).

The term  $P(W)$  is determined by the *language model*. It can be either rule based or of statistical nature. In the later case the probability of the word sequence is approximated through the occurrence frequencies of individual words (often depending on the previous one or two words) in some predefined database. The main shortcoming of the statistical language models from the mobile ASR viewpoint is the number of parameters to be stored, which may be as gross as hundreds of megabytes for very large vocabulary (LV) tasks.

The likelihoods  $P(\mathbf{O}|W)$  are estimated on most state-of-the-art recognizers using HMM based *acoustic models*. Here every word  $w_j$  is composed of a set of acoustic units like phonemes, triphones or syllables, i.e.  $w_j = (u_1 \cup u_2 \cup \dots)$ . And every unit  $u_k$  is modeled by a chain of states  $s_j$  with associated emission probability density functions  $p(\mathbf{x}|s_j)$ . These densities are usually given by a mixture of diagonal covariance Gaussians, i.e.  $p(\mathbf{x}|s_j) = \sum_{k=1}^M b_{mj} N(\mathbf{x}, \boldsymbol{\mu}_{mj}, \boldsymbol{\Sigma}_{mj})$ . The computation of the final likelihood  $P(\mathbf{O}|W)$  is performed by combining the state emission likelihoods  $p(\mathbf{o}_t|s_j)$  and state transition probabilities. The parameters of acoustic models such as state transition probabilities, means  $\boldsymbol{\mu}_{mj}$ , variances  $\boldsymbol{\Sigma}_{mj}$  and weights  $b_{mj}$  of Gaussian mixtures are estimated on the training stage and also have to be stored. The total number of Gaussians to be used depends on the design of the recognizer. However, even for a digit recognition task ending up with about one thousand 39-dimensional mixtures is a common situation, which also embarrasses a compact implementation of the ASR in a mobile device.

Finally, armed with both  $p(\mathbf{o}_t|s_j)$  and  $P(W)$ , we need an effective algorithm to explore all HMM states of all words over all word combinations. Usually modified versions of the Viterbi algorithm are employed to determine the best word sequence in the relevant lexical tree.

### 2.2. The Mobile ASR Dilemma

The implementation of effective mobile ASR systems is challenged by many border conditions. In contrast to the generic ASR, the mobile recognition system therefore has to encounter the following aspects: limited available storage volume (language model and acoustic models to be shorten, which leads to performance degradation), tiny cache of 8–32KB and small and "slow" RAM memory from 1MB up to 32MB (many signal processing algorithms are not allowed), low processor clock-frequency (enforces to use the suboptimal algorithms), no hardware-made floating point arithmetic, no access to the operating system for mobile phones (no low level code optimization possible), cheap microphones (often far away from the mouth - affects the performance substantially), highly challenging acoustic environment (PDA can be used everywhere: in the car, on the street, in large halls and small rooms; this introduces additive and convolutional distortions of the speech signal), no real PDA recorded speech corpora are currently available, high energy consumption during algorithms execution, and so forth. Finally, improvements which could be done in one functional block contradict with other parts of the system.

### 2.3. System Configurations for Mobile Speech Recognition

ASR systems as described in section 2.1 structurally can be decomposed into two parts: the acoustic *front-end*, where the process of the feature extraction takes place and the *back-end*, performing Viterbi search based on the acoustic and language models.

Since most of the portable devices use a communication link, we can classify all the mobile ASR systems upon the location of the front-end and back-end. This allows us to distinguish three principal system structures:

- **client-based** architecture or embedded ASR, where both front-end and back-end are implemented on the terminal;
- **server-based** or network speech recognition (NSR), where speech is transmitted over the communication channel and the recognition is performed on the remote server;
- **client-server** ASR or distributed speech recognition (DSR), where the features are calculated on the terminal, whilst the classification is done on the server side.

Each approach has its individual shortcomings, which influence the overall performance. Therefore, the appropriate implementation depends on the application and the terminal properties. Small recognition tasks are generally recommended to reside on terminals, while the large vocabulary recognition systems take advantage of the server capacities. In following we analyze the problems associated with particular architecture in detail and examine the remedies against undesired effects.

## 3. Embedded Speech Recognition Systems

In the case of client-based or embedded ASR the entire process of speech recognition is performed on the terminal device (see Fig. 1). Embedded ASR is often the architecture of choice for PDAs. First, these client devices are somewhat more powerful compared to the mobile phones. Second, they are driven under well established operating system, like Windows Mobile 5.0, allowing easier software extension at different system levels. Third, PDAs have well known processor architectures, e.g. Intel XScale, and there are some libraries and development kits optimized for a particular platform [6]. Finally, PDAs do not

always have a wireless communication link available, so the remote speech recognition is rather unwelcome on PDA.

The main advantage of the terminal based architecture relies in the fact that no communication between the server and the client is needed. Thus, the ASR system is always ready for use and does not rely on the quality of the data transmission. The most important issue for embedded ASR systems is the very limited system resources on the mobile device.

For the embedded ASR design two implementation aspects need to be considered. These are the **memory** usage of the underlying algorithms and the execution **speed** [7]. To achieve reliable performance of embedded speech recognition system the modifications improving both criteria should be introduced in every functional block of ASR system.

### 3.1. Front-end

The task of the acoustic front-end is to extract characteristic features out of the spoken utterance. Usually it takes in a frame of the speech signal every 10 msec and performs certain spectral analysis. The regular front-end includes among others, the following algorithmic blocks: Fast Fourier Transformation (FFT), calculation of logarithm (LOG), the Discrete Cosine Transformation (DCT) and sometimes Linear Discriminant Analysis (LDA).

In [8] Köhler et al. give an example of the ASR implemented on iPAQ H5550 Pocket PC with XScale 400 MHz processor. For 1 sec of speech the front-end requires 0.713 sec of processing time, where 0,622 sec is contributed by FFT, LOG, DCT and LDA.

DCT and LDA are normally performed by matrix multiplication, which requires  $O(n^3)$  floating point multiplications and additions. Because there is no floating point unit on PDAs, the operations with floating point numbers are software emulated. The alternative is to scale up float numbers by a factor  $S$ , drop out the fractional part, perform time efficient operation on integers and scale down the result. The weak point here is that scaling itself requires  $O(n^2)$  floating multiplications and that beforehand chosen  $S$  can generally lead to integer overflows.

Additional speed improvement can be gained by using a polynomial approximation of the logarithm  $\log_a f = \log_a(2^x m) \approx \log_a 2((m-1)(5-m)/3 + x)$  and by employing the processor optimized library for the FFT computation [6]. The combination of above techniques cuts five times the RT ratio of the entire front end from 0.713 down to 0.140 with moderate WER increase from 19.7% up to 20.2% on 2500 words task, (see Table 2 for details).

Table 2: Performance of the baseline and speed optimized algorithmic units [8]

Functional unit	Time		WER	
	Baseline	Opt.	Baseline	Opt.
DCT	42ms	5ms	19.7%	19.7%
LDA	134ms	10ms	19.7%	19.7%
LOG	67ms	9ms	19.7%	19.8%
FFT	379ms	25ms	19.7%	21.6%

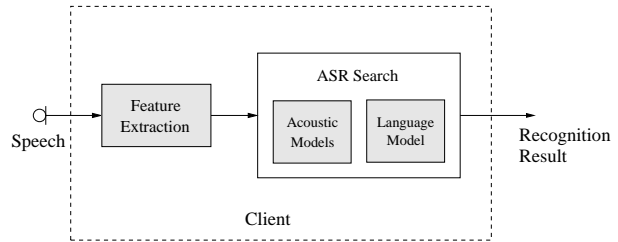


Figure 1: Client based ASR system - *Embedded Speech Recognition*

### 3.2. ASR Search

We have to notice that the feature extraction even if not optimized, takes just about 2% of all processing time in case of the medium vocabulary and even less in large vocabulary recognition tasks. The main computational burden relies in the ASR search, which is governed by two operations: the computation of Gaussians in the emission likelihoods  $p(o_t|s_j)$  for a given frame and the token propagation, i.e. the maintenance of the information about the survivors (best paths) during the search through the lexical tree.

#### 3.2.1. Likelihood Computation Methods

The calculation of the single state emission likelihood  $p(\mathbf{x}|s_j) = \sum_{m=1}^M b_{mj} N(\mathbf{x}, \boldsymbol{\mu}_{mj}, \boldsymbol{\Sigma}_{mj})$  with plain number of mixtures  $M=6$  and feature vector length  $D=39$  requires 480 floating multiplications, 234 additions and 6 exponentials. Therefore, the likelihoods are computed in the log-domain and are approximated by the impact of the nearest-neighbor:

$$\log p(\mathbf{x}|s_j) \approx \arg \max_{m=1, \dots, M} \left\{ b'_{mj} - \frac{1}{2} \sum_{d=1}^D \frac{(\mathbf{x}[d] - \boldsymbol{\mu}_{mj}[d])^2}{\boldsymbol{\Sigma}_{mj}^2[d, d]} \right\} \quad (2)$$

The use of this approximation shortens the number of required operations by 30% in [9], since scaled weights  $b'_{mj}$  can be calculated beforehand and some mixtures are early dropped out during the maximization.

Further improvements may be gained if we consider in (2) not all  $M$  mixture components but only some of them. The relevant components are chosen either via a projection search algorithm (PSA) or by  $k$ -d tree method [10]. In the  $k$ -d tree method the mixture components virtually divide the feature space into partitions. Structuring of the components into the tree allows to quickly determine which partition the vector  $\mathbf{x}$  belongs to. Only prototypes laying on the partition's border will be used in (2). In PSA only components located inside the  $\epsilon$ -neighborhood around the given feature vector are considered. The cost-effective Hamming approximation of  $l_1$ -norm can be used to search for the nearest prototypes. In [10] the  $k$ -d tree and PSA technics reduced CPU-time required for likelihoods computations respectively down to 50,7% and 39,9% compared to the baseline system without increase in WER.

In [11] Vasilache et al. present a phoneme based ASR, where HMM parameters  $\mu$  and  $\sigma$  are represented by one out of  $2^5$  (for mean) and one out of  $2^3$  (for variance) quantizer levels. If in addition the input features will be quantized with similar number of levels  $2^5$ , then the likelihoods can take only one out of  $2^5 \cdot 2^3 \cdot 2^5 = 8192$  fixed values, which can be precomputed and saved. With this substantial complexity reduction the perfor-

mance of the speed optimized ASR system was not worse than that of the baseline system.

3.2.2. Decoding Techniques

Once the likelihoods  $p(x|s_j)$  being calculated the best alignment of the HMM state sequence  $s_j$  to the sequence of feature vectors has to be found using the Viterbi algorithm. For this the trellis - network of admissible word sequences expanded to the level of the phonetic units  $u_k$  has to be constructed.

In the embedded ASRs this network is usually represented as a phonetic tree. Unlike the network architecture the tree representation requires no back-tracing: the leaf of the tree uniquely determines the entire path - word sequence. This results in faster search and also in saving the most demanded on PDAs dynamically allocated RAM.

The search cost also can be reduced by applying the two-pass search strategy. Assuming some simplifications the "fast-match" finds n-best paths, which are later rescored by the "detailed match" process. Despite on the small latency the two-pass algorithm is very efficient for large lists tasks, where the grammar can be effectively represented as a tree. In [12] on the task containing list of 1475 words the two-pass recognition took on 206 MHz PDA only 0.56 real-time, which is more than 8 time faster compared to the ASR with a single pass search.

4. Network Speech Recognition

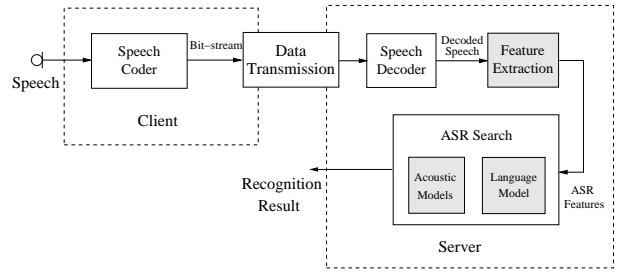
Practically all complications caused by the resource limitations of the mobile devices can be avoided shifting both ASR front-end and back-end from the terminal to the remote server. Such a server-based ASR architecture is referred in the literature as a network speech recognition (see Fig. 2a).

Unlike the embedded ASR, the NSR architecture can augment not only PDAs but also "thin" terminals, e.g. cellular phones, with a very large vocabulary ASR. Another advantage of NSR relies in the fact that it can provide access to the recognizers based on the different grammars or even different languages. Besides, the content of the ASR vocabulary often may be confidential, thus prohibiting its local installation. Finally, the NSR allows a seem-less to the end-user upgrade and modification of the recognition engine.

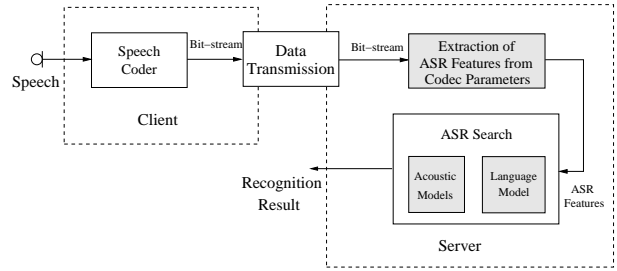
Characteristic drawback of the NSR architecture is the performance degradation of the recognizer caused by using low bit-rate codecs, which becomes more severe in presence of data transmission errors and background noise.

To a certain extent the distortion introduced by source coding can be diluted if the recognizer is trained on the respectively corrupted speech. However, the tandeming of the different source coding schemes in addition to the different channel noise levels spans a too large number of possible acoustic models. Better performance can be obtained if the recognition is performed based on the features derived from the parametric representation of the encoded speech without the actual speech reconstruction (see Fig. 2b). There are several successful implementations of such system intended for different codecs: ETSI GSM 06.10 [13], FS1015 and FS1016 [14] and ITU-T G.723.1 [15].

Another important issue related to NSR design is an arrangement of the server side. In contrast to generic recognition systems, the NSR back-end should be able to serve effectively hundreds of clients simultaneously. In [16] Rose et al. suggest an event-driven, input-output non-blocking server framework, where the dispatcher, routing all the systems events, buffers the



(a) ASR features are extracted from the transcoded speech



(b) ASR features are derived from the speech codec parameters

Figure 2: Server based ASR system - Network Speech Recognition

clients queries on the decoder proxy server, which redirects the requests to the one of free ASR decoder processes. Such NSR server framework composed of a single 1GHz proxy server and eight 1GHz decoder servers each running four decoder processes could serve up to 128 concurrent clients. [17] presents an alternative architecture, where the entire ASR system has been decomposed into 11 functional blocks. The components interconnected via DARPA Galaxy Hub can be accessed independently allowing a more efficient parallel use of the ASR system.

5. Distributed Speech Recognition

Distributed speech recognition represents the client-server architecture, where one part of ASR system, viz. primary feature extraction, resides on the client, whilst the computation of temporal derivatives and the ASR search are performed on the remote server (see Fig. 3).

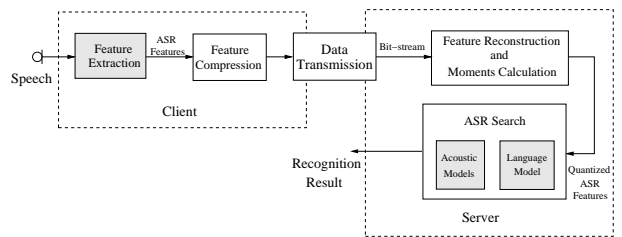


Figure 3: Client-Server based ASR system - Distributed Speech Recognition

Even though both DSR and NSR make use of the server-based back-end, there are substantial differences in these two schemes favoring DSR. First of all the speech codecs unlike the feature extraction algorithms are optimized to deliver the best perceptual quality and not for providing the lowest WER. Second, ASR does not need the high quality speech, but rather some set of characteristic parameters. Thus, it requires lower data rates - 4.8 kbit/s is a common rate for the features transmission. Third, since feature extraction is performed place on the client side, the higher sampling rates covering full bandwidth of the speech signal are possible. Finally, because in DSR we are not constrained to the error-mitigation algorithm of the speech codec, better error-handling methods in terms of WER can be developed.

The studies within the distributed recognition framework target three aspects indicative for DSR:

- the development of noise robust and computationally effective **feature extraction** algorithms;
- the investigation of procedures for feature **vectors quantization**, permitting compression of the features without losses in recognition quality;
- the elaboration of **error mitigation** methods.

The composite answer on all these question was given by the STQ-Aurora DSR Working Group established within the European Telecommunications Standards Institute (ETSI). The result of the four-year cooperative work of Aurora Group members, the world leading ASR companies, has become the ETSI standard ES 202 050 operating at the 4.8 kbit/s data rate and specifying the advanced front-end (AFE) feature extraction, feature compression and back-end error-mitigation algorithms [18]. In 2004 this standard was enriched to the extended advanced front-end (xAFE), allowing for the cost of additional 0.8 kbit/s reconstruction of the intelligible speech signal out of features stream.

The AFE includes the noise reduction, made by two stages of Wiener filtering, the calculation of 13 cepstral coefficients and a log energy, blind equalization and voice activity detection. The feature compression is performed by using the split vector quantization (SVQ). The error detection is based on the CRC and minimum consistency tests. A simple repetition of previously correctly received frame in place of erroneous one is applied in case if an error is detected.

Table 3 shows the performance of DSR with AFE and NSR with modern 4.45 kbit/s, 12.2 kbit/s and 12.66 kbit/s AMR codecs, where the packet-switched data transmission is performed over the channels with 1% and 3% block error rates. The average relative degradation of NSR compared to DSR exceeds 50%.

The simple error mitigation strategy of the AFE standard performs well under channels with a high carrier to interference ratio (C/I), but leads to the performance degradation under noisier conditions. Significant improvements can be obtained using the soft source-channel decoding approach introduced by Fingscheidt and Vary [20]. This method exploits the reliability (soft values) of the received bits and the a priori information about residual redundancy in feature vectors to calculate the a posteriori probabilities and thus the MMSE estimate of the transmitted vector. In [21] the authors show that if the bits reliability is not known it can be estimated using the minimum consistency tests in intraframe and interframe directions. This leads to the reduction in WER from 10% down to 2.5% with circuit-switched transmission over the channel with 2.5 dB C/I ratio for digit

Table 3: Comparison of the DSR with ETSI AFE and NSR with AMR codecs [19]

Sampl. rate	NSR codecs	BLER %	WER,%		Rel. Degr.
			DSR	NSR	
8 kHz	AMR-4.75	1	2.39	5.67	57.9%
		3	2.38	6.51	63.4%
	AMR-12.2	1	2.39	4.73	49.5%
		3	2.38	6.33	62.4%
16 kHz	AMR-12.66	1	1.84	2.74	32.9%
		3	1.84	3.44	46.5%

recognition task. Further improvement can be gained if soft decoding is applied also to the feature temporal derivatives [22].

An alternative solution to the vector quantization problem was suggested in [23]. The method based on the source modeling with Gaussian mixtures operates on bit-rates below 1 kbit/s with less than 1% relative WER degradation.

## 6. Conclusions

In this contribution we have analyzed three possible ASR architectures (embedded ASR, NSR and DSR) for providing the speech recognition technology to the portable end devices. The shortcomings associated with the particular design and possible solutions have been considered.

In our opinion the continuously increasing power of the mobile devices will give rise to the expansion of the embedded ASR systems. We suppose that in the near future the medium recognition tasks having 1000-2000 words, which represents a good coverage of the certain application domain, will be successfully running on the terminal devices, like PDAs or in-car embedded systems.

In the light of the incremental affordability of high data-rate networks the remote speech recognition systems will also remain of interest for performing the very large vocabulary recognition and for accessing to corporate ASR system with confidential contents. Because of the superior performance of DSR in presence of the transmission errors and surrounding noise, with xAFE being selected by 3GPP for Speech Enabled Services, and with real time transmission protocol for AFE features standardized by IETF, we believe that NSR will be totally supplanted by the DSR architecture.

## 7. References

- [1] B. Pellom and K. Hacioglu, "Sonic: The university of Colorado continuous speech recognition system," University of Colorado, Tech. Rep. TR-CSLR-2001-01, March 2001.
- [2] W. Walker, P. Lamere, P. Kwok, B. Raj, R. Singh, E. Gouvea, P. Wolf, and J. Woelfel, "Sphinx-4: A flexible open source framework for speech recognition," Sun Microsystems Laboratories, Tech. Rep. TR-2004-139, November 2004.
- [3] J. Odell, D. Ollason, P. Woodland, S. Young, and J. Jansen, *The HTK Book for HTK V2.0*. Cambridge University Press, Cambridge, UK, 1995.
- [4] R. C. Rose and S. Partharathy, "A tutorial on ASR for wireless mobile devices," in *ICSLP*, 2002.

- [5] L. Rabiner and B.-H. Juang, *Fundamentals of speech recognition*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1993.
- [6] Intel, "Intel performance libraries," <http://www.intel.com/cd/software/products/asmo-na/eng/perflib/index.htm>, 2006.
- [7] M. Novak, "Towards large vocabulary ASR on embedded platforms," in *Proc. Interspeech 2004 ICSLP*, 2004.
- [8] T. W. Köhler, C. Fügen, S. Stüker, and A. Waibel, "Rapid porting of ASR-systems to mobile devices," in *Proc. of the 9th European Conference on Speech Communication and Technology*, 2005, pp. 233–236.
- [9] A. Hagen, B. Pellom, and D. A. Connors, "Analysis and design of architecture systems for speech recognition on modern handheld-computing devices," in *Proc. of the 11th International Symposium on Hardware/Software Code-sign*, October 2003.
- [10] S. Ortmanns, T. Firzlaß, and H. Ney, "Fast likelihood computation methods for continuous mixture densities in large vocabulary speech recognition," in *Proc. Eurospeech'97*, Rhodes, Greece, September 1997, pp. 139–142.
- [11] M. Vasilache, J. Iso-Sipilä, and O. Viikki, "On a practical design of a low complexity speech recognition engine," in *Proc. ICASSP*, vol. 5, 2004, pp. 113–116.
- [12] M. Novak, R. Hampl, P. Krbec, V. Bergl, and J. Sedivy, "Two-pass search strategy for large list recognition on embedded speech recognition platforms," in *Proc. ICASSP*, vol. 1, 2003, pp. 200–203.
- [13] J. M. Huerta, "Speech recognition in mobile environments," Ph.D. dissertation, Carnegie Mellon University, April 2000.
- [14] B. Raj, J. Migdal, and R. Singh, "Distributed speech recognition with codec parameters," in *Proc. ASRU'2001*, December 2001.
- [15] C. Peláez-Moreno, A. Gallardo-Antolín, and F. Díaz-de-María, "Recognizing voice over IP: A robust front-end for speech recognition on the world wide web," *IEEE Trans. on Multimedia*, vol. 3, no. 2, 2001.
- [16] R. Rose, I. Arizmendi, and S. Parthasarathy, "An efficient framework for robust mobile speech recognition services," in *Proc. ICASSP*, vol. 1, 2003, pp. 316–319.
- [17] K. Hacioglu and B. Pellom, "A distributed architecture for robust automatic speech recognition," in *Proc. ICASSP*, vol. 1, 2003, pp. 328–331.
- [18] *Distributed Speech Recognition; Advanced Front-end Feature Extraction Algorithm; Compression Algorithm*, ETSI Standard ES 202 050, October 2002.
- [19] *Recognition performance evaluations of codecs for Speech Enabled Services (SES)*, 3GPP TR 26.943, December 2004.
- [20] T. Fingscheidt and P. Vary, "Softbit speech decoding: A new approach to error concealment," *EEE Trans. on Speech and Audio Processing*, vol. 9, no. 3, pp. 240–251, March 2001.
- [21] V. Ion and R. Haeb-Umbach, "A unified probabilistic approach to error concealment for distributed speech recognition," in *Proc. Interspeech 2005 ICSLP*, 2005.
- [22] A. James and B. Milner, "Soft Decoding of Temporal Derivatives for Robust Distributed Speech Recognition in Packet Loss," in *Proc. ICASSP*, vol. 1, 2005, pp. 345–348.
- [23] K. K. Paliwal and S. So, "Scalable distributed speech recognition using multi-frame GMM-based block quantization," in *Proc. Interspeech 2004 ICSLP*, 2004.