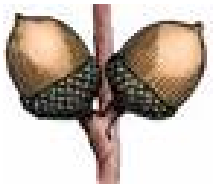

Project no. 034362

ACORNS

Acquisition of COmmunication and RecogNition Skills

Instrument: STREP
Thematic Priority: IST/FET

D1.1 Modules for Conventional Feature Set

Due date of deliverable: 2007-11-30
Actual submission date: 2007-11-25

Start date of project: 2006-12-01 Duration: 36 Months

Organisation name of lead contractor for this deliverable: KTH

Revision: 0.1

Project co-funded by the European Commission within the Sixth Framework Programme (2002-2006)		
Dissemination Level		
PU	Public	X
PP	Restricted to other programme participants (including the Commission Services)	
RE	Restricted to a group specified by the consortium (including the Commission Services)	
CO	Confidential, only for members of the consortium (including the Commission Services)	

VERSION DETAILS
Version: 0.3
Date: 21 December 2007
Status: Final

CONTRIBUTOR(S) to DELIVERABLE	
Partner	**Name**
SE-KTH	Chris Koniaris

DOCUMENT HISTORY			
Version	**Date**	**Responsible**	**Description**
0.0	30/09/07	Chris Koniaris	writing
0.1	20/11/07	Chris Koniaris	reformat and extend
0.1	25/11/07	Bastiaan Kleijn	editing
0.2	18/12/07	Chris Koniaris	editing comments of Hugo Van hamme
0.3	21/12/07	Chris Koniaris	editing comments of Louis ten Bosch

DELIVERABLE REVIEW			
Version	**Date**	**Reviewed by**	**Conclusion***
0.1	17/12/07	Hugo Van hamme	Approved
0.1	18/12/07	Louis ten Bosch	Approved

Table of Contents

1 Summary

This report describes an initial set of feature extraction algorithms delivered under deliverable D1.1. The first set of features consists of the Logarithmic Mel Spectrum and the Mel-Frequency Cepstral Coefficients (MFCCs).

2 Introduction

A first stage of almost any speech processing system consists of the extraction of features that are more convenient for processing than the sequence of speech samples itself. This report describes the initial set of features that are to be used in the ACORNS project. This first set of features consists of the Logarithmic Mel Spectrum, the Mel-Frequency Cepstral Coefficients (MFCCs) and the pitch period. These features will later be extended with features that are perceptually meaningful (as evaluated by a model of the human auditory system).

3 Logarithmic Mel Spectrum and Mel-Frequency Cepstral Coefficients

3.1 List of files and folders

The deliverable was sent in a zip format file containing the files and folders shown in Table 1:

Table 1: List of files and folders for the spectrum and MFCCs.

Name of the file or folder	Format type	Description
1. fe_demo.m	*Matlab file*	*Main function; a demo with 3 TIMIT wave files is provided*
2.mfcc_extract.m	*Matlab file*	*Computes the mel spectrum and the MFCCs*
3.filtbank.m	*Matlab file*	*A triangular filterbank in mel domain*
4.deltas.m	*Matlab file*	*Computes Delta Coefficients*
5.deltasdeltas.m	*Matlab file*	*Computes Delta-Delta Coefficients*
6.cmvn.m	*Matlab file*	*Applies a cepstral mean and variance normalization to the MFCCs*
7.readhtk.m	*Matlab file*	*Reads a HTK parameter file (from VOICEBOX toolkit)*
8.writehtk.m	*Matlab file*	*Writes data in HTK format (from VOICEBOX toolkit)*
9.readsph.m	*Matlab file*	*Reads a SPHERE/ TIMIT format wave file (from VOICEBOX toolkit)*
10.read_mfcc.pdf	*PDF file*	*A Readme file describing the code and its use*
11.GNU GENERAL PUBLIC LICENSE.pdf	*PDF file*	*Describes the terms for using VOICEBOX toolkit routines*
12.lists	*Folder*	*Contains list files for the 3 TIMIT wave files*
13.timit_demo	*Folder*	*Contains the 3 TIMIT wave files; Also the extracted features are to be saved there*

3.2 The extraction process of Spectrum and MFCCs

Our goal is to provide ACORNS with a set of standard features. Mel spectrum and mel-frequency cepstral coefficients are one of them. For each speech block we consider a high pass pre-emphasis filter of the form $x(n) = \breve{x}(n) - a\breve{x}(n-1)$, where $\breve{x}(n)$ is the original speech and $a = 0.97$ [1]. Then, a Hamming window is applied to the output of the pre-emphasis block

$$x'(n) = \left\{0.54 - 0.46\cos(\frac{2\pi(N-1)}{N-1})\right\}x(n), \quad n=1,..,N,$$

where N is the length of the window. A Discrete Fourier Transform (DFT) is applied to the windowed frame to compute the magnitude spectrum of the signal

$$X(k) = \sum_{n=0}^{N-1} x'(n)e^{-j2\pi kn/N}, \qquad k=1,\ldots,K,$$

where K is the length of the DFT. We then compute the DFT power spectrum which we multiply with the triangular mel weighted filterbank. The result is summed to give the logarithmic mel spectrum

$$S(m) = \ln\left[\sum_{k=0}^{K-1} |X(k)|^2 \; H_m(k)\right],$$

where $|X(k)|^2$ is the periodogram, $H_m(k)$ is the m^{th} triangular filter, and M is the number of the filters of the filterbank. In the end, we consider the Discrete Cosine Transform (DCT) of the logarithmic filterbank energies to get the uncorrelated mel-frequency cepstral coefficients (MFCCs) [2] as

$$c(q) = \sum_{m=0}^{M-1} S(m)\cos\left(q(m-\frac{1}{2})\frac{\pi}{M}\right), \qquad q=1,\ldots,Q,$$

where Q is the number of cepstral coefficients.

3.3 Description of the software

In this section we describe the implementation of the above process and discuss some practical matters that needed to be solved.

fe_demo.m

Purpose: The main function which takes as input the speech file, calls the front-end routines and saves the extracted features for future use.

Synopsis: fe_demo

Description: Initially, the first speech file (out of three demo files taken from TIMIT database) is considered, and the routine mfcc_extract is then called to process the file and extract the logarithmic mel spectrum and the MFC coefficients. The extracted features are then saved in a HTK format (using the writehtk function from VOICEBOX) and then the next speech file is considered. The routine ends when all speech files have been considered and the corresponded features have been saved.

mfcc_extract.m

Purpose: The function which takes as input the speech files and extracts the mel spectrum and the mel-frequency cepstral coefficients.

Synopsis: [c,e,melspec] = mfcc_extract(x,fs,ncep,win,shf,NFFT,nf,fb_step,nlf,deriv,norm)

Description: The input and the output parameters are shown in Table 2 and Table 3.

Table 2: Input parameters for MFCC extraction.

parameter	Description
X	*The speech input*
fs	*The sampling frequency (in Hz)*
ncep	*The number of cepstra*
win	*The window's length (in samples)*
shf	*The window's shift (in samples) [win/2]*
NFFT	*The length of the DFT*
nf	*The number of the filters in the filterbank*
fb_step	*The filterbank's step [bandwidth]*
nlf	*The number of low filters to throw away*
deriv	*If deriv=1, compute Delta coefficients only else if deriv=2, compute in addition, Delta-Delta coefficients*
norm	*If norm=1, compute cepstral mean normalization else if norm=2, compute cepstral mean and variance normalization. If norm=0, no normalization is performed*

Table 3: output parameters for MFCC extraction

parameter	Description
c	*The mel-frequency cepstral coefficients*
e	*The energy coefficient*
melspec	*The logarithmic mel spectrum*

The routine starts by calling the function filtbank to compute the triangular mel filterbank. Next, the speech signal is taken and a pre-emphasis filter is considered. The speech file is divided into blocks of 25ms using a Hamming window, overlapped every 12.5ms. Next, the DFT power spectrum of the windowed signal is computed, which then is pruned (due to symmetry, only the first half of DFT

is used for further processing), and emphasized. In our experiments we found that there were unexpected errors in case of zero input. To avoid such cases, we check if the power spectrum takes values less than a small threshold (we considered the number e^{-10}). If that is true, then we assign this small value to be the new value of it. The logarithmic mel spectrum is calculated by multiplying the power spectrum by each of the triangular Mel weighting filters and then summed the result. In parallel, the energy is computed. Both the mel spectrum and the energy are floored to avoid taking values less than -50. Finally, the DCT of the spectrum is considered, and 12 MFCCs are calculated, ignoring the 0^{th} coefficient. If necessary, the Delta and the Delta-Delta coefficients can be calculated to augment the feature set. Finally, there is an option of performing a cepstral mean normalization or a cepstral mean and variance normalization for $norm = 1$ or $norm = 2$, respectively.

filtbank.m

Purpose: The function which computes the triangular mel filters of the filterbank.

Synopsis: [H,lfreq,rfreq]=filtbank(fb_step,fs,nf,NFFT)

Description: The input and the output parameters are shown in Table 4 and Table 5

Table 4: input parameters filter bank.

parameter	Description
fb_step	*The filterbank's step [bandwidth]*
fs	*The sampling frequency (in Hz)*
nf	*The number of the filters of the filterbank*
NFFT	*The length of the DFT*

Table 5: Output parameters filter bank.

parameter	Description
H	*The triangular windows of the filterbank*
lfreq	*The left edge of the triangulars (in Hz)*
rfreq	*The right edge of the triangulars (in Hz)*

Initially, the central frequencies of the filters are computed. We consider a linear spacing for frequencies from 0 to 1 KHz with a constant bandwidth of 100 Hz, and then a logarithmic scaling with a factor of 1.1. The reason we chose this is because this filterbank simulates best how the human hearing system functions, resolving frequencies in a nonlinear manner [3]. Next, the amplitude of the triangular filterbank is computed as

$$H_m(k) = \begin{cases} 0 & k < f(m-1) \\ \dfrac{(k - f(m-1))}{(f(m) - f(m-1))} & f(m-1) \le k \le f(m) \\ \dfrac{(f(m-1) - k)}{(f(m+1) - f(m))} & f(m) \le k \le f(m+1) \\ 0 & k > f(m+1) \end{cases},$$

which satisfies $\sum_{m=1}^{M} H_m(k) = 1$ according to [4]. For a 16 KHz speech signal, we considered a number of 30 filters. The following Figuree shows the output of the filterbank.

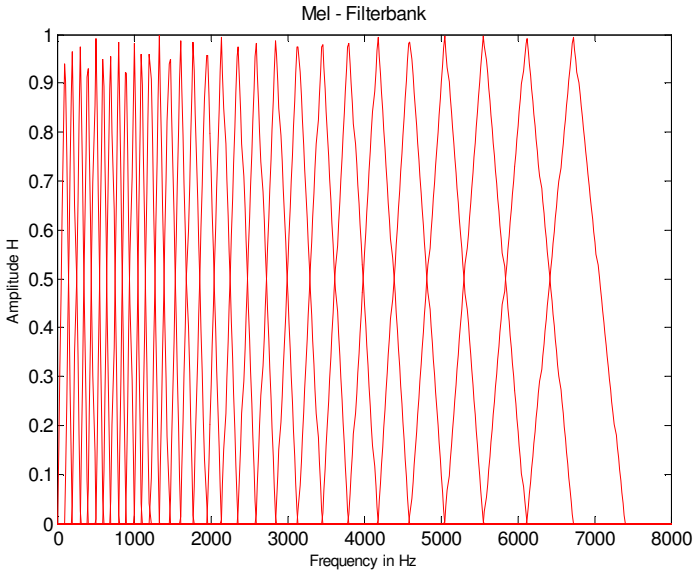

Figure 1: Mel filter bank.

In Figure 1, we see the linear spacing up to 1 KHz and then the logarithmic while the amplitude of the filters is constant as in [2].

deltas.m

Purpose: The function which computes the Delta coefficients.

Synopsis: d = deltas(c)

Description: This routine takes as input the mel-frequency cepstal coefficients, applies a 9-point window to computes the first time derivatives [5] as

$$d_t = \frac{\sum_{\theta=1}^{\Theta} \theta (c_{t+\theta} - c_{t-\theta})}{2 \sum_{\theta=1}^{\Theta} \theta^2},$$

and returns them to the mfcc_extract.

deltasdeltas.m
Purpose: The function which computes the Delta-Delta or acceleration coefficients.

Synopsis: dd = deltasdeltas(c)

Description: This routine, yields in computing the Delta coefficients as we presented in the previous paragraph and in addition in computing the acceleration coefficients with a 3-point window. The formula for computing the acceleration coefficients is the same as in previous case, replacing the cepstra with their Delta coefficients.

$$a_t = \frac{\sum_{\theta=1}^{\Theta} \theta(d_{t+\theta} - d_{t-\theta})}{2\sum_{\theta=1}^{\Theta} \theta^2},$$

cmvn.m
Purpose: The function which computes the cepstral mean and variance normalization.

Synopsis: c = cmvn(c,norm)

Description: In this function, cepstral mean and variance normalization is performed, a technique that is designed to handle convolutional distortions and to increase the robustness of the speech recognition system to unknown linear filtering operations [4]. For the given cepstral vector $c(q)$, we subtract its mean value $\bar{c}(q)$, resulting in the cepstral mean normalized vector $\hat{c}(q)$,

$$\hat{c}(q) = c(q) - \bar{c}(q),$$

in case $norm = 1$. When $norm = 2$ is used, then a cepstral mean and variance normalization is performed as

$$\hat{c}(q) = \frac{c(q) - \bar{c}(q)}{\bar{\sigma}(q)},$$

where $\bar{\sigma}(q)$, is the standard deviation of $\bar{c}(q)$.

readsph.m
Purpose: The function which reads a TIMIT/SPHERE format sound file

Synopsis: [y,fs,ffx] = readsph(filename,mode,nmax,nskip)

Description: We use this VOICEBOX toolkit routine to read the 3 TIMIT sound files we used in our demonstration. According to GNU GENERAL PUBLIC LICENSE documentation we can use routines from VOICEBOX if we have read and accepted the terms that are included there. We used these routines without modifying them, just for the purpose of demonstration.

readhtk.m
Purpose: The function which reads a HTK parameter file

Synopsis: [d,fp,dt,tc,t] = readhtk(filename)

Description: We use this VOICEBOX toolkit routine to read the extracted features in our demonstration. According to GNU GENERAL PUBLIC LICENSE documentation we can use routines from VOICEBOX if we have read and accepted the terms that are included there. We used these routines without modifying them, just for the purpose of demonstration.
writehtk.m
Purpose: The function which writes data in HTK format.

Synopsis: writehtk(filename,d,fp,tc)

Description: We use this VOICEBOX toolkit routine to save the extracted features in our demonstration. According to GNU GENERAL PUBLIC LICENSE documentation we can use routines from VOICEBOX if we have read and accepted the terms that are included there. We used these routines without modifying them, just for the purpose of demonstration.

4 Pitch Track Estimation

The second set of delivered features is the pitch.

4.1 List of files and folders

The pitch-period tracking algorithm was sent in a zip format file containing the files show in Table 6 Pitch tracking routines.

Table 6 Pitch tracking routines.

Name of the file or folder	Format type	Description
1. main.m	*Matlab file*	*Main function; a demo with a test file is provided*
2. process_data.m	*Matlab file*	*Process a speech signal and computes its pitch period*
3.lpanaly.m	*Matlab file*	*Concerns the LPC filter*
4.inplsf.m	*Matlab file*	*Interpolates LSFs*
5.lpcls2ar.m	*Matlab file*	*Converts line spectrum pair frequencies to AR polynomial (from*

Name of the file or folder	Format type	Description
		VOICEBOX toolkit)
6.lpcar2ls.m	*Matlab file*	*Converts AR polynomial to line spectrum pair frequencies (from VOICEBOX toolkit)*
7.lpweight.m	*Matlab file*	*Filters weighting*
8.lpanafil.m	*Matlab file*	*To get predicted residual and updated state*
9.lpsynfil.m	*Matlab file*	*To get speech from the residual and update state*
10.g729pitch.m	*Matlab file*	*Estimates pitch period according to [6]*
11.iirfilter.m	*Matlab file*	*An IIR filter*
12.spclab.m	*Matlab file*	*A speech signal presentation program*
13.fileread.m	*Matlab file*	*To read a file*
14.filewrit.m	*Matlab file*	*To write a file*
15.read_pitch.pdf	*PDF file*	*A Readme file describing the code and its use*
16.GNU GENERAL PUBLIC LICENSE.pdf	*PDF file*	*Describes the terms for using VOICEBOX toolkit routines*
17.testfile.wav	*Sound file*	*A .wav file for demonstration*

4.2 The pitch period track estimation process

The method we used to estimate the pitch period is based on [6]. We begin by estimating an autoregressive model of the speech. To achieve this, a linear predictive (LP) analysis is performed at a rate of 100 Hz, using the autocorrelation method. The LP coefficients are then transformed to the line spectral frequency (LSF) domain for interpolation to 200 Hz update rate. An adaptive perceptual weighting filtering based on the autoregressive model is then used to de-emphasize the speech. The de-emphasized speech signal will be used to perform the pitch period estimation.

A smoothed open-loop pitch for a speech frame is used. The algorithm considers three pitch-period candidates t_1, t_2 and t_3 from three search ranges for a speech frame. To prevent pitch doubling and halving, the pitch period is selected for each update using the following process:

- Initial candidate; look for the maximum pitch correlation and select the initial estimate

$$R'_{\max} = \max_{(i=1,2,3)} R'(t_i)$$
$$T_{op} = \arg\max_{(i=1,2,3)} R'(t_i)$$

- If $t_2 < T_{op}$

If $|T_{op} - t_2| < 10, \quad \delta = 0.7$ else $\delta = 0.9$

If $R'_{\max}\delta < R'(t_2)$, $R'_{\max} = R'(t_2)$ and $T_{op} = t_2$

- If $t_3 < T_{op}$

If $|T_{op} - t_3| < 5, \quad \delta = 0.7$ else $\delta = 0.9$

If $R'_{\max}\delta < R'(t_3)$, $T_{op} = t_3$

The final selection of the open-loop pitch for the current frame is T_{op}. Finally, a median filtering is applied to T_{op} to remove outliers from the pitch track.

4.3 Description of the software

In this section we describe the implementation of the above process.

main.m
Purpose: The function which calls most of the available routines to perform de-emphasis and pitch estimation.

Synopsis: a = main(testfile,16)

Description: Initially, the speech file is re-sampled from 16 to 8 KHz and then the routine process_data is called to de-emphasize and to estimate the pitch period.

process_data.m
Purpose: The main function which takes as input a speech file and calls the appropriate routines to de-emphasize and estimate the pitch.

Synopsis: [ptrack1,ddata] = process_data(idata,sf,mf)

Description: This routine is responsible to de-emphasize the speech signal and to perform a series of processes before estimating the pitch. The procedure goes through the following steps; the speech signal is pre-processed and then Linear Prediction (LP) analysis is performed. The LP coefficients are transformed to Line Spectral Frequency (LSF) for interpolation purposes. A perceptual weighting filtering is then applied to de-emphasize the speech signal before calling the pitch tracking function g729pitch which will estimate the pitch period. In the end, a median filtering is applied to remove outliers.

lpanaly.m
Purpose: Performs LPC analysis.

Synopsis: out = lpanaly(s, order, mode)

Description: Gets A parameter of LPC filter from a speech segment s. Used to transform LPCs to LSFs.

inplsf.m
Purpose: Used to interpolate.

Synopsis: lsf = inplsf(lsf_prev, lsf_curr, nsub)

Description: Interpolation of the LSFs in center point.

lpcls2ar.m
Purpose: Converts LSFs to AR polynomial AR = (LS) .

Synopsis: ar = lpcls2ar(ls)

Description: We use this VOICEBOX toolkit routine to convert interpolated LSFs to LPCs. According to GNU GENERAL PUBLIC LICENSE documentation we can use routines from VOICEBOX if we have read and accepted the terms that are included there. We used these routines without modifying them.

lpcar2ls.m
Purpose: Converts a polynomial AR polynomial to LSFs LS = (AR).

Synopsis: ls = lpcar2ls(ar)

Description: We use this VOICEBOX toolkit routine to convert LPCs to LSFs. According to GNU GENERAL PUBLIC LICENSE documentation we can use routines from VOICEBOX if we have read and accepted the terms that are included there. We used these routines without modifying them.

lpweight.m
Purpose: Applies a perceptual weighting filtering to de-emphasize the speech.

Synopsis: lpu = lpweight(lpu, gamma)

Description: Implementation of an adaptive perceptual weighting filtering based on the autoregressive model to de-emphasize the speech. The de-emphasized speech is used later to estimate the pitch.

lpanafil.m
Purpose: Used in the perceptual weighting.

Synopsis: [ress, sn] = lpanafil(s,lpcs,sn)

Description: Gets predicted residuals and update state.

lpsynfil.m
Purpose: Used in the perceptual weighting.

Synopsis: [sps,sn2] = lpsynfil(res,lpcs,sn2)

Description: Gets speech from excitation/residual and update state sn. There is a possibility that this will be done for several subframes.

iirfilter.m
Purpose: Used in the perceptual weighting.

Synopsis: out = iirfilter(res,mem,lpcs)

Description: Implementation of an IIR filter.

g729pitch.m
Purpose: Implementation of an open-loop pitch analysis based on the ITU-T G.729.1 Recommendation [6].

Synopsis: pperiod = g729pitch(pdata,indexmatrix,blockl,maxpp,sf)

Description: For each frame we obtain three pitch-period candidates t_1, t_2 and t_3 from three search ranges. If we denote as δ the threshold which will help us decide the best candidate for the current loop then, the best pitch candidate is selected after the following steps:

- Initial candidate; look for the maximum pitch correlation and select the initial estimate

$R'_{\max} = \max_{(i=1,2,3)} R'(t_i)$

$T_{op} = \arg\max_{(i=1,2,3)} R'(t_i)$

- If $t_2 < T_{op}$

If $|T_{op} - t_2| < 10, \quad \delta = 0.7$ else $\delta = 0.9$

If $R'_{\max}\delta < R'(t_2), \quad R'_{\max} = R'(t_2)$ and $T_{op} = t_2$

- If $t_3 < T_{op}$

If $|T_{op} - t_3| < 5, \quad \delta = 0.7$ else $\delta = 0.9$

If $R'_{\max}\delta < R'(t_3), \quad T_{op} = t_3$

The final selection of the open-loop pitch for the current frame is T_{op}. The above open-loop pitch analysis prevents from common errors such as pitch doubling and halving,

spclab.m
Purpose: A speech signal presentation program.

Synopsis: spclab(varargin)

Description: SPCLAB is a Matlab tool to analyze and listen to speech signals. It provides an intuitive interface, based on 3-button mouse clicks. The user can mark a segment with the left mouse button, zoom in or out with right button, and listen with the middle button. Various analysis tools, such as FFT, LPC analysis, cepstral, pitch analysis, autocorrelation, etc. are available. Finally, it can be used to compare and analyze several signals simultaneously, in synchronized plots. In our case, we use this tool to plot the original signal, the de-emphasized version of it, and the estimated pitch period track as it is shown in Figure 2: signal, de-emphasized signal and pitch track.

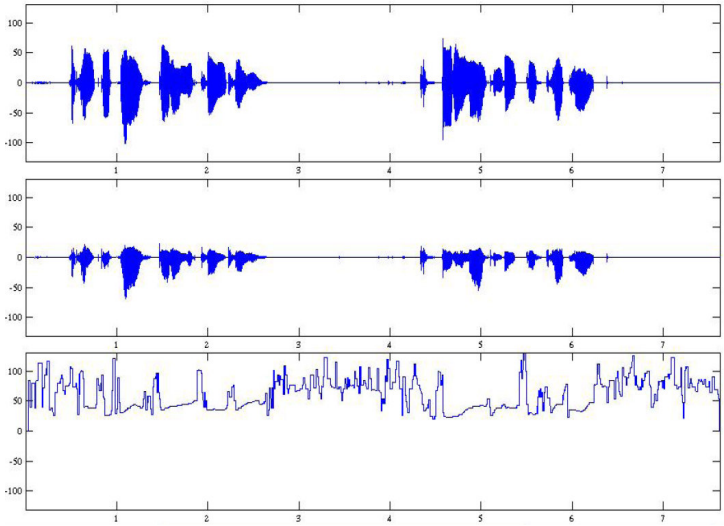

Figure 2: signal, de-emphasized signal and pitch track.

In the third graph, the vertical axis shows the pitch period while the horizontal shows time.

fileread.m
Purpose: To read a file.

Synopsis: [s,header] = fileread(FileName,dataform,len)

Description: Use this routine to read a file and its header.

filewrit.m
Purpose: To write data to a file.

Synopsis: filewrit(FileName,s,dataform,hdr)

Description: Use this routine to save data (i.e., pitch) in a certain data form for future use.

5 References

[1] ETSI ES 201 108 v1.1.2., "Speech Processing Transmission and Quality aspects; Distributed Speech Recognition; Front-end feature extraction algorithm; Compression Algorithms.", 2000-2004.

[2] S.B. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabicword recognition in continuously spoken sentences", *IEEE Transactions on Acoustics Speech and Signal Processing,* vol. 28, no. 4, pp. 357-366, Aug. 1980.

[3] S.S. Stevens, J. Volkman, and E.B. Newman, "A scale for the measurement of the psychological magnitude pitch", *The Journal of the Acoustic Society of America*, vol. 8, pp.185-190, Jan.1937.

[4] X. Huang, A. Acero, and H.W. Hon, "Spoken Language Processing: A Guide to Theory, Algorithm and System Development", *Prentice Hall*, 2001.

[5] S. Young et al., "The HTK Book (for HTK Version 3.2).", *Cambridge University, Engineering Department*, Dec. 2002.

[6] ITU-T Rec. G.729.1, "G.729 based embedded variable bit-rate coder: An 8-32 kbit/s scalable wideband coder bitstream interoperable with G.729", 2006.