2009 International Conference on Computational Science and Engineering

*i*Know Where You Are

Kalyan Subbu, Ning Xu and Ram Dantu
Department of Computer Science and Engineering
University of North Texas
3940 N. Elm St., Denton, TX 76207, USA
E-mail: {kp0124, nx0001, rdantu}@unt.edu

***Abstract*—“Smart” phones, such as G1 and iPhone, have made their way into people’s lives with more intelligence owing to the continually decreasing cost and increasing access to memory capability, processing power and network bandwidth. This work attempts to detect presence in different environments like office, conference, meeting and traveling outdoors, using audio sensors on the G1 phone. Specifically, a two step process involving classifying the background first and then detecting the number of people in an environment is followed. Using the Vector Quantization method, audio is classified into five predefined classes. A recognition rate ranging from 86% to 100% for individual classes and 91% overall recognition accuracy was obtained. For speaker change detection via Bayesian Information Criterion, about 79.4% of all 350 test audio clips was correctly categorized.**

I. Introduction

Mobile phones have been more and more integrated into people’s cultural and social practices with the advantage of surpassing the space limitation and facilitating instant and comprehensive access to a variety of information over the traditional phones and even computers. “Smart” phones, such as G1 and iPhone, have made their way into people’s lives with more intelligence owing to the continually decreasing cost and increasing access to memory capability, processing power and network bandwidth [1].

Audio data possesses distinguishing features which can be used to separate classes of audio like silence, pure speech, impure (noisy) speech, environmental and music. These features can be considered as elements of an n-dimensional vector. Environmental cues can be utilized to automatically present dynamic information to users relevant to their current context. Background sounds in places like the office, classrooms, streets and vehicles can be a rich source for modeling user context. Furthermore, the information obtained can be used to assist activity and location detection, based on the results of which the mobile devices can respond intellectually such as preventing disruption in the meetings, performing emergency dial, etc.

In this work, an attempt is made to infer speech situation by first classifying the background environmental sounds into predefined classes and then detecting the number of speakers based on analysis of the audio recording samples collected using G1 phones. The Vector Quantization (VQ) technique is incorporated for classification phase and the Bayesian Information Criterion (BIC) technique for speaker segmentation.

The remainder of this paper is organized as follows: Section II provides an overview of related work. Section III presents the data collection and preprocessing and the feature extraction methodology. Section IV discusses the classification algorithm and the technique for speaker change detection used in this work. Section V presents the experimental set up and the results obtained for both the classification and detection phase. Finally, Section VI concludes.

II. Related Work

Peltonen et al. [2] performed auditory scene recognition of 26 different acoustic environments of which 17 scenes were classified with an accuracy of 68.4% using a feature vector consisting of multiple features. The recognition accuracy proved to be approximately 18 times better than the random guess rate. Work done by Sawhney [3] in environmental noise classification showed that feature extraction mechanisms like power spectral density (PSD) and frequency bands generated from filterbank analysis and nearest neighbor classification technique proved to be robust. Ma et al. [4] used a 11-state HMM classifier and obtained accuracy of the individual scenes ranging from 75% to 100%. However, the noise classifier designed in this system is not capable of recognizing multiple and simultaneously occurring environmental noises. Clarkson [5] looked at obtaining environmental context through audio for applications and user interfaces like Nomadic computing [6]. They categorized their ASA system into sound scenes and sound objects. Results showed that their system detected 9/10 of the scene transitions.

There have been numerous existing speech analysis research and related techniques. For example, speaker segmentation, sometimes referred to as speaker change detection [7], is the task of splitting a recorded audio document into acoustically homogeneous segments, so that every segment ideally contains speech from only

978-0-7695-3823-5/09 $26.00 © 2009 IEEE
DOI 10.1109/CSE.2009.55

IEEE computer society

one speaker [8]. Speaker clustering aims at classifying speech segments based on speaker voice characteristics [9], which is to identify speech segments uttered by the same speaker and assign a unique label to them [10]. Acoustic change detection is a technique for detecting changes in speaker identity, environmental condition and channel condition [11]. The above techniques are useful in finding the number of speakers whose speeches are documented in the same audio file. It is assumed that no a priori knowledge about speakers, such as the number of speakers, reference data or model of speakers, is available. The detection process is conducted in an offline, non-real-time fashion where analysis is performed on full-length audio documents and not on device.

To our knowledge no work has been reported on detection of presence in different environments like office, conference, meeting using audio sensors on the phone. Moreover, the sensor measurements are constrained due to conditions like random orientation of microphone, limited processing power on the phone, isolating background, multiple speakers, speaker changes in a meeting and integrating data from multiple sensors. Also there has not been any research work done in glueing both classification of environmental sound and detecting number of speakers for determining social context in that particular environment. In this work, we adopted a two step approach exploiting background sound and multi party speech to achieve presence detection. Additionally, microphones together with other sensors like GPS and accelerometers can be used for developing applications that include avoiding interruptions and unwanted calls.

III. Methodology

A. Data Collection and Preprocessing

Audio samples were collected using the inbuilt microphone on the phone. An Android program was written to complete this task using a class implemented as part of the Android platform called *android.media.MediaRecorder* that captures audio [12].The steps are described below:

1) Create a new instance of *android.media. MediaRecorder*;
2) Create a file path where the data will be save;
3) Use *MediaRecorder.setAudioSource()* to set the audio source to *MediaRecorder.AudioSource. MIC*;
4) Use *MediaRecorder.setOutputFormat()* to set output file format. The popular format in mobile devices is 3GPP;
5) Use *MediaRecorder.setAudioEncoder()* to set the audio encoder. The default audio encoder/decoder provided in android is AMR-NB which samples at 8kHz with various data rate from 4.75 to 12.2 kbps;
6) Call *start()* to start audio capture, and call *stop()* to stop
7) When the recording task is done, *release()* needs to be called to release the resources allocated on the *MediaRecorder* instance.

After data collection, the analog signals were digitized by sampling them. The recordings were segmented into 30 msecs or 256 samples of duration audio files and each sample was labeled according to the environment where they were recorded for analysis. Then the original audio document was divided into frames and Hamming windowing technique was used on each individual frame so as to minimize the signal discontinuities at the beginning and end of each frame. After this, the windowed frame was converted from time domain to frequency domain using Fast Fourier Transform.

B. Feature Extraction

The purpose of this step is to extract useful discriminative information from the time-domain waveform which will result in a compact set of feature vectors. Mel Frequency Cepstral Coefficients (MFCCs) are the most commonly used acoustic features since they take the human perception sensitivity with respect to frequencies into consideration. The mel filter banks are a set of triangular band pass filters obtained by mapping the normal frequency values over the mel scale. Fig. 1 shows the MFCC features extracted from a particular environmental sound.

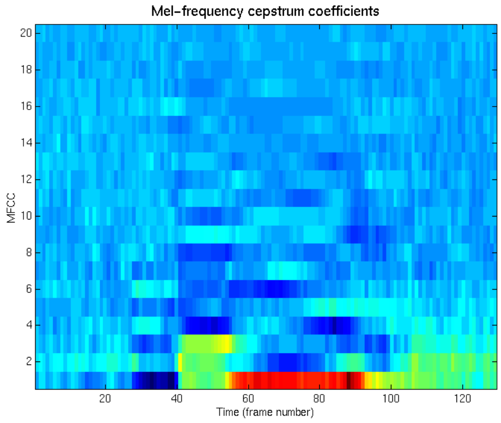

Figure 1. MFCCs Extracted from Audio Sample

IV. Classification and Detection

A. Classification Model

The VQ method [13] was used as a classification algorithm due to its ease of implementation and accuracy. We assume that $x = [x_1, x_2, x_3, \cdots]^T$ is an

N-dimensional vector whose components $\{x_k, 1\leq k\leq N\}$ are real-valued, continuous-amplitude random variables. The vector x is mapped onto another real-valued, discrete-amplitude, N dimensional vector y. The expression $y = q(x)$ where $q(\bullet)$ is the quantization operator indicates that y is the quantized value of x. Typically, y takes on one value from the finite set of values Y referred to as the reconstruction codebook, or simply the codebook and is given by $Y = \{y_i, 1\leq i\leq L\}$ where L is the size of the codebook. The vectors y_i are the set of code vectors given by $y_i = [y_{i1}, y_{i2}, \cdots]^T$. To design a codebook the N-dimensional space of the random vector x is partitioned into L regions or cells $\{C_i, 1 \leq i \leq L\}$ and a vector y_i is associated with each cell C_i . The quantizer assigns the code vector y_i if $x \in C_i$.

Based on the tutorial presented in [14], the well-known LBG vector quantization algorithm [15] for clustering a set of L training vectors into a set of M codebook vectors has been used for clustering the training samples and classifying the test samples with the trained ones. Algorithm (1) lists the pseudocode.

Algorithm 1 LBG Vector Quantization Algorithm

```
1:  Design a 1-vector codebook $y_i = \{y_{i1}, y_{i2}, \cdots\}$
2:  Split codebook according to the rule:
      • ${y_n}^+ = y_n(1+\epsilon)$
      • ${y_n}^- = y_n(1-\epsilon)$
3:  $m = 2 * m$
4:  Cluster vector using Nearest-Neighbor Search
5:  Find and update centroids
6:  Compute distortion $D$
7:  if $\frac{D^{'} - D}{D} < \epsilon$ then
8:      if $m < M$ then
9:          Goto Step 2
10:     else
11:         Stop
12:     end if
13: else
14:     $D^{'} = D$
15:     Goto Step 4
16: end if
```

B. Speaker Change Detection via Bayesian Information Criterion

The input audio stream can be modeled as a Gaussian process in the cepstral space. Thus, we used the technique introduced in [11], namely, a maximum likelihood approach to detect turns of a Gaussian process based on the Bayesian Information Criterion (BIC).

Denote $\mathbf{x} = \{x_1, x_2, \ldots, x_N\}$ as the d-dimensional cepstral vectors extracted from the audio samples within the sliding window; assuming $\mathbf{x}$ subjects to an independent multivariate Gaussian distribution, then the element of each dimension satisfies:

$$x_i \sim N(\mu_i, \sigma_i) \tag{1}$$

where μ_i is the mean vector and σ_i is the full covariance matrix.

Let's first examine a simplified problem: there is at most one speaker change point in the Gaussian process. Given the aforementioned assumption, the problem becomes a binary hypothesis testing of a change occurring at time t:

$$H_0 : x_1 \cdots x_N \sim N(\mu, \sigma) \tag{2}$$

versus

$$H_1 : x_1 \cdots x_t \sim N(\mu_1, \sigma_1); x_{t+1} \ldots x_N \sim N(\mu_2, \sigma_2); \tag{3}$$

where σ, σ_1 and σ_2 are the sample covariance matrices from all the data within the window, from $\{x_1, \cdots, x_t\}$ and from $\{x_{t+1}, \cdots, x_N\}$, respectively. Then, the maximum likelihood estimate of the changing point is

$$\hat{t} = argmax_t(Nlog|\sigma| - N_1 log|\sigma_1| - N_2 log|\sigma_2|). \tag{4}$$

where N, N_1, N_2 represent the number all the data within the window, from $\{x_1, \cdots, x_i\}$ and from $\{x_{i+1}, \cdots, x_N\}$, respectively. Based on BIC approach, we can score the two hypotheses ($i = 1, 2$) for each possible changing point t as below

$$\begin{aligned} BIC(H_i, t) &= Nlog|\sigma| - N_1 log|\sigma_1| - N_2 log|\sigma_2| \\ &- \frac{1}{2}\left(d + \frac{1}{2}d(d+1)\right) logN \end{aligned} \tag{5}$$

Thus, we decide there is a change if the maximum BIC value is greater than zero, which implies it is more likely a change occurs in the audio document, and the time stamp of the changing point is the t value satisfying the criterion just mentioned.

Based on the same idea, the algorithm to sequentially detect the changing points in the Gaussian process $\mathbf{x}$ is presented in Fig. 2. By expanding the window $[S_0, S_1]$, the maximum likelihood decision is made based on as all the data samples the program has seen. Fig. 3 shows the BIC values computed over a test sample of 30 seconds long, where the dash-dot line indicates the speaker change. In this case, we can infer that two speakers are involved in the speech document.

V. Results

For the Classification phase, different acoustic environments viz., bus, lab, office, lecture, home, urban driving in a car and walking in a street were chosen. Thirty recordings of each category were collected during different days and time of a day. Throughout the entire process of recording samples, the phone was placed in the user's trouser pocket. The duration of the recording lasted for about 10 minutes but only 10 secs of

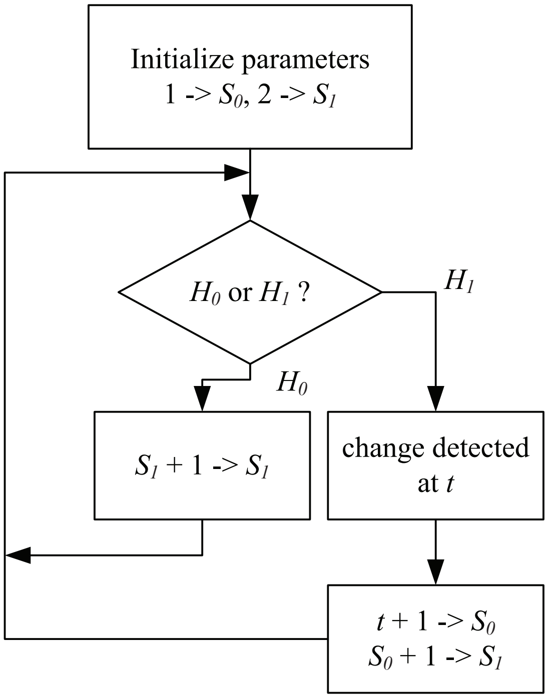

Figure 2. Detection of Multiple Speaker Change Points in the Audio Document

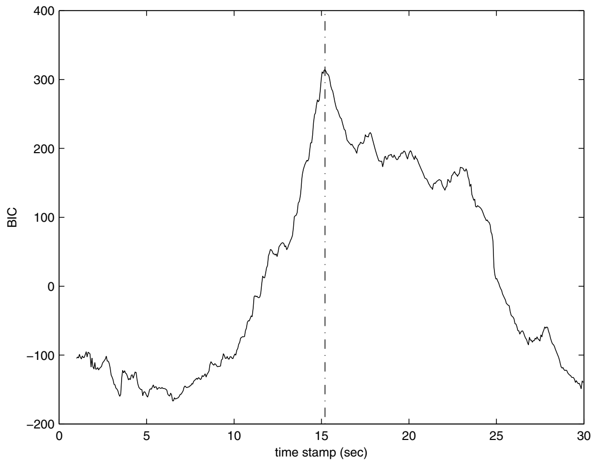

Figure 3. BIC Values of a 30-Second Audio Clip. (The greater the BIC value is, the more likely there is a speaker change at that time. The dash-dot line indicates detected speaker change.)

audio was used for analysis . The number of training and test samples were equally divided among the recordings obtained.

Then, using the VQ classification method, a group of known test samples from bus, lab, gym, class and car and unknown test samples from a street and a room with the television on were compared with the training samples. Fig. 4 compares the new samples to the centroids of four different classes and finally clusters or classifies them to their respective classes of audio.

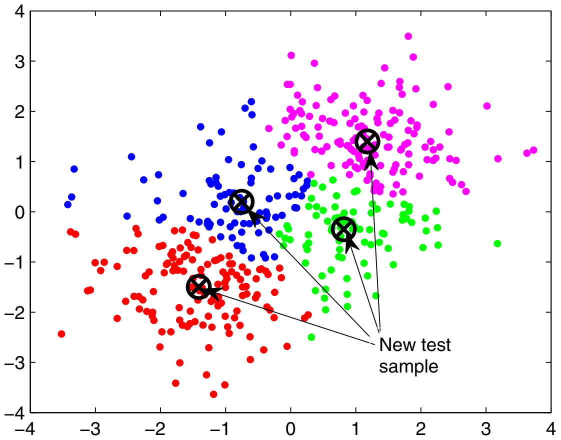

Figure 4. Centroid Comparison with Test Sample

To compute the recognition rate of the clustering algorithm, fifteen test samples of same type of audio were used for classification. The number of times it was correctly classified over the total number of test samples was used for calculating the recognition rate. The overall recognition rate was an average of the individual recognition rates of each audio scene. Table I shows the confusion matrix and the recognition rates of the individual samples.

Table I
CONFUSION MATRIX OF ACOUSTIC ENVIRONMENT CLASSES

Scene	Bus	Lab	Gym	Class	Car	Percentage
Bus	15	0	0	0	0	100
Lab	0	14	0	1	0	93.3
Gym	0	0	15	0	0	100
Class	0	3	0	12	0	80
Car	2	0	0	0	13	86.6

We can see from the table that out of the 15 test samples of bus used for classification, all were correctly classified giving a 100% recognition rate. The same procedure was applied to calculate the recognition rate for the other audio samples. An overall recognition rate of 91% was obtained for the 5 classes of audio defined

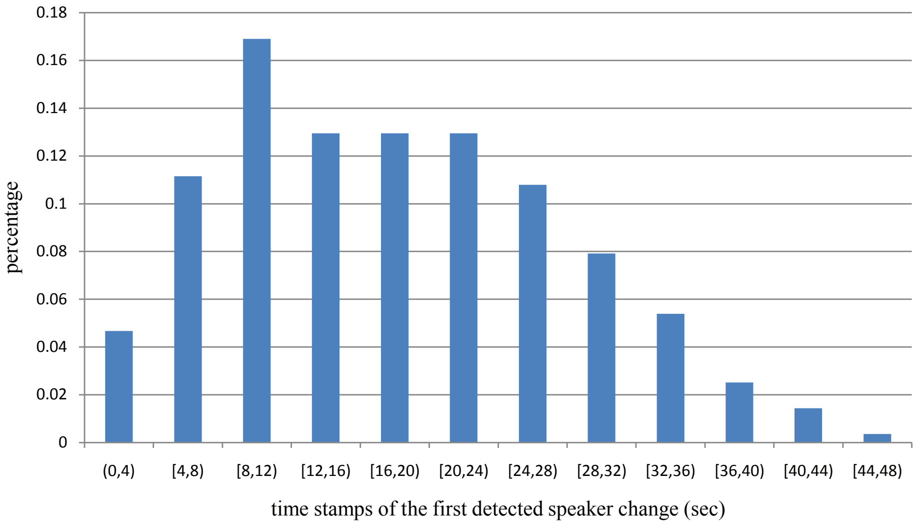

Figure 5. Histogram of Time Stamp Distribution of the First Detected Speaker Change. (The plot shows percentage of first detected speaker change that happens during each time interval. For instance, in most of the tested clips, the first changing point happens during the 8-12 second interval, indicating that speech turn occurs about every 8-12 seconds statistically.)

in this work.

As for the second phase, the audio samples were collected in speech scenarios such as classroom lecture, group discussion, casual two-party conversation with quiet background. No constraint was imposed on the uttering fashion, and all speech process turns happened naturally. The mobile device was located either in the pocket, or on a solid furniture surface close by, as where most users will habitually place it. The device was not always close to the speaker at the time of recording, but within the user's hearing reception distance. To summarize it, the recording environments were perfectly realistic. A total length of about 12 hours of audio recordings were taken. The original audio documents were segmented into one-minute clips, each of which was manually labeled as speaker change or non speaker change. Mostly there are multiple speaker changes in one clip. However, given the objective of this project, the detection of every turn is unnecessary, and computationally expensive considering the capabilities of mobile devices. Thus, the detection rate takes account of recognizing at least one change. Detection was performed on 350 clips. Experimental results show that about 79.4% of all audio clips with speaker changes tested can be correctly categorized.

Furthermore, Fig. 5 shows percentage of first detected speaker change that happens during each time interval. For instance, in most of the tested clips, the first changing point happens during the 8-12 second interval, indicating that speech turn occurs about every 8-12 seconds statistically. It is shown in Fig. 6 that more than 50% of the first changing point occur within 20 seconds, and more than 90% occur within 35 seconds of the audio clips. This information can be quite instructive to set the termination condition such that the detecting program reacts responsively.

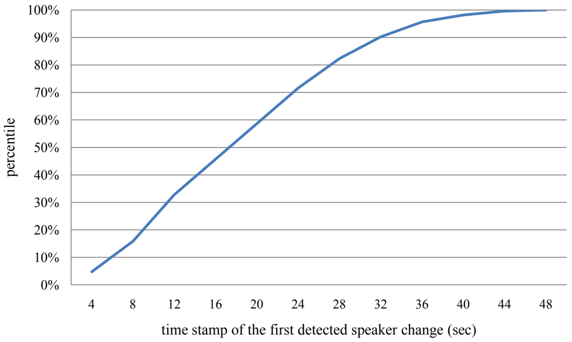

Figure 6. Percentile of Time Stamps of the First Detected Speaker Change. (It shows that more than 50% of the first changing point occur within 20 seconds, and more than 90% occur within 35 seconds. This information can be instructive in that the mobile device can react based on the detection results obtained on the audio samples within 35 seconds to be more responsive.)

VI. Conclusions

By adopting a two step approach, the authors exploited background sound and multi party speech for achieving presence detection in environments like bus, lab, office, car, etc. The Vector Quantization method of audio classification yielded recognition rates ranging from 86% to 100% for individual classes and 91%

overall. For speaker change detection via Bayesian Information Criterion, about 79.4% of all 350 test audio clips was correctly categorized. Moreover, using additional sensors like GPS and accelerometers along with microphones applications that include avoiding interruptions and unwanted calls can be developed.

REFERENCES

[1] S. E. Tranter and D. A. Reynolds, "An overview of automatic speaker diarization systems," *IEEE Trans. Audio Speech Lang. Process.*, vol. 14, no. 5, pp. 1557–1565, Sept. 2006.

[2] V. Peltonen, J. Tuomi, and A. Klapuri, "Computational Auditory Scene Recognition," in *IEEE Intl Conf. on Acoustics, Speech, and Signal Processing*, 2002, pp. 1941–1944.

[3] N. Sawhney and M. Pattie, "Situational awareness from environmental sounds," Tech. Rep., 1997, http://web.media.mit.edu/˜nitin/papers/Env_Snds/EnvSnds.html.

[4] L. Ma, D. Smith, and M. Ben, "Environmental Noise Classification for Context-Aware Applications," in *Proc. Int. Conf. DEXA*, 2003, pp. 360–370.

[5] B. Clarkson, N. Sawhney, and A. Pentland, "Auditory context awareness via wearable computing," in *Workshop on Perceptual User Interfaces (PUI98)*, 1998.

[6] N. Sawhney and C. Schmandt, "Nomadic radio: speech and audio interaction for contextual messaging in nomadic environments," *ACM Trans. Comput.-Hum. Interact.*, vol. 7, no. 3, pp. 353–383, 2000.

[7] X. Anguera, "Robust speaker diarization for meetings," Ph.D. dissertation, Universitat Politècnica de Catalunya, Oct. 2006.

[8] R. Sinha, S. E. Tranter, M. J. F. Gales, and P. C. Woodland, "The Cambridge University March 2005 Speaker Diarisation System," in *Proc. European Conf. Speech Comm. Tech.*, Lisbon, Portugal, Sept. 2005, pp. 2437–2440.

[9] W. H. Tsai, S. S. Cheng, and H. M. Wang, "Automatic speaker clustering using a voice characteristic reference space and maximum purity estimation," *IEEE Trans. Audio Speech Lang. Process.*, vol. 15, no. 4, pp. 1461–1474, May 2007.

[10] D. Lilt and F. Kubala, "Online speaker clustering," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, vol. 1, May 2004, pp. I–333–6 vol.1.

[11] S. S. Chen and P. S. Gopalakrishnan, "Speaker, environment and channel change detection and clustering via the bayesian information criterion," in *Proc. DARPA Broadcast News Transcription and Understanding Workshop*, Lansdowne, Virginia, USA, Feb. 1998. [Online]. Available: http://www.nist.gov/speech/publications/darpa98/pdf/bn20.pdf

[12] "Android audio capture," http://developer.android.com/guide/topics/media/index.html.

[13] J. Makhoul, S. Roucos, and H. Gish, "Vector quantization in speech coding," *Proceedings of the IEEE*, vol. 73, no. 11, pp. 1551–1588, Nov. 1985.

[14] D. N. Minh, "An automatic speaker recognition system," http://www.ifp.uiuc.edu/˜minhdo/teaching/speaker_recognition/.

[15] Y. Linde, A. Buzo, and R. Gray, "An algorithm for vector quantizer design," *Communications, IEEE Transactions on*, vol. 28, no. 1, pp. 84–95, Jan 1980.