RC23499 (W0501-090) January 19, 2005
Computer Science

IBM Research Report

CSR: Speaker Recognition from Compressed VoIP Packet Stream

Charu Aggarwal, David Olshefski, Debanjan Saha, Zon-Yin Shae, Philip Yu
IBM Research Division
Thomas J. Watson Research Center
P.O. Box 704
Yorktown Heights, NY 10598

Research Division
Almaden - Austin - Beijing - Haifa - India - T. J. Watson - Tokyo - Zurich

CSR: Speaker Recognition from Compressed VoIP Packet Stream

Charu Aggarwal, David Olshefski, Debanjan Saha, Zon-Yin Shae, Philip Yu

IBM T.J. Watson Research Center
{charu, olshef, dsaha, zshae, psyu}@us.ibm.com

ABSTRACT

VoIP applications require the ability to identify speakers in real time. This paper presents Compressed Speaker Recognition (CSR), an innovative approach to perform speaker recognition directly from the compressed voice packets. CSR performs online speaker recognition from live packet streams of compressed voice packets by performing fast clustering over a defined subset of the features available in each compressed voice packet. Our experimental results show that CSR is highly scalable and accurate across a broad range of speakers.

1. INTRODUCTION

As VoIP continues to grow in popularity, the need for scalable, online speaker recognition becomes paramount. ISPs require fast, online speaker authentication mechanisms for providing various interactive voice services via VoIP phones. The ability to identify speakers for security enforcement reasons is a key emerging corporate and governmental application. Accurate, online speaker recognition performed in real-time, for large numbers of users opens the door to a variety of applications simply not applicable in an offline, high latency environment.

Traditionally, voice analysis is performed using the voice signal waveform as input. A voice waveform signal not only conveys speech content, but also reveals several other features. Such analysis has demonstrated the ability to extract speaker identity, language, and violent voice tones, to name a few.

However, the majority of the compressed VoIP traffic is based on Code Excited Linear Prediction (CELP). Since the VoIP traffic is compressed at the end point device before transport, it requires decompression to obtain the voice signal waveform before traditional analysis methods can be used. Figure 1 depicts a traditional, offline, speaker recognition system. The compressed VoIP traffic is first captured to file. Then, during an offline processing phase, the file is read, decompressed and the voice signal waveform reconstructed. The voice signal is then Fourier transformed into the frequency domain. After passing through a frequency spectrum energy calculation and pre-emphasis processing, the frequency parameters are then passed through a set of Mel-Scale logarithmic filters. The output energy of each individual filter is log-scaled (e.g., via a log-energy filter), before a cosine transform is performed to obtain "cepstra". The set of "cepstra" then serves as the feature vector for a vector classification algorithm, such as the GMM-UBM (Gaussian Mixture Model-Universal Background Model) for speaker recognition [1].

This traditional approach, which needs to decompress the compressed voice packets into a voice signal waveform, does not scale in terms of CPU, disk access, and memory use for fast data streams. In addition, there are long latencies incurred while obtaining a feature vector for the analysis due to the CPU intensive operation (decompress-FFT-Mel-Scale filter-Cosine transform) [1]. Moreover, voice packets can be dropped in the network due to congestion. A compressed voice packet contains information relating itself to its predecessor and successor. A missing packet, therefore, greatly impacts the decompression algorithms' ability to recreate the original voice signal. This results in a poor voice signal not of sufficient quality for accurate voice recognition analysis.

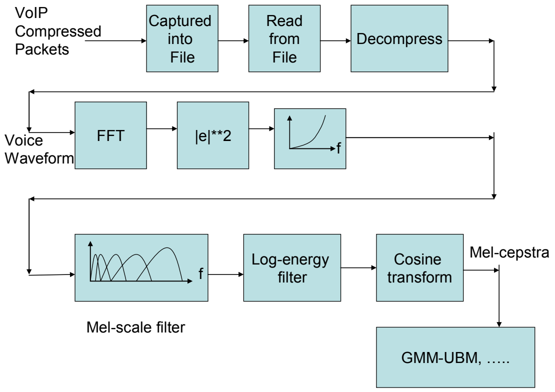

Figure 1: Traditional speaker identification system.

This paper presents Compressed Speaker Recognition (CSR), an approach for performing online speaker recognition from live packet streams of compressed voice packets by performing fast clustering over a defined subset of the features available in each compressed voice packet. The rest of this paper is organized as follows. The CSR feature vector and its discrimination power are described in section 2. The fast clustering mechanism within CSR is described in section 3. Experimental setup and results are reported in section 4 and 5. A summary is presented in section 6.

2. FEATURE EXTRACTION FROM COMPRESSED VOIP PACKETS

It is essential to derive a discriminative feature vector for speaker recognition. CSR extracts a feature vector which exhibits a powerful discrimination property directly from each compressed VoIP packet. Modern voice compression is based on a CELP algorithm (e.g., G723, G729, GSM [2]), which models the human vocal tract as a set of filter coefficients. A block diagram of a G729 compression algorithm [7] is shown in Figure 2. After pre-processing of a voice input, an LSF frequency transformation is performed. An adaptive codebook is used to model long term pitch delay information, and a fix codebook is used to model the short term excitation of the human speech. Gain block is a parameter used to capture the amplitude of the speech, and A(Z) is used to model the vocal track of the speaker.

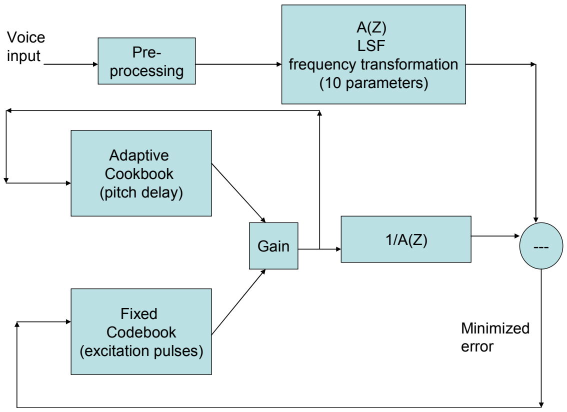

Figure 2: G729 block diagram

Consequently, each compressed voice packet explicitly carries a set of important voice characteristics (e.g., voice tract filter model parameters, pitch delay, amplitude, excitation pulsed positions for the voice residues) which can be used to create a voice feature vector for the speaker A G729 voice packet, for example, is shown in Figure 3. L0 through L3 indicate the vocal tract model of the speaker; P1, P0, GA1, GB1, P2, GA2 and GB2 capture the long term pitch information of the speaker; and C1, S1, C2, and S2 capture the short term excitation pulsed positions for the voice residues of the speech frame.

We use the vector (L0, L1, L2, L3, P1, P0, GA1, GB1, P2, GA2, GB2) as the feature vector within CSR. This set of features emphasizes pitch and vocal tract parameters over the less important excitation parameters.

	Description	Bits
L0	Switched MA predictor of LSP quantizer	1
L1	First stage vector of quantizer	7
L2	Second stage lower vector of LSP quantizer	5
L3	Second stage higher vector of LSP quantizer	5
P1	Pitch delay first subframe	8
P0	Parity bit for pitch delay	1
C1	Fixed codebook first subframe	13
S1	Signs of fixed-codebook pulses 1st subframe	4
GA1	Gain codebook (stage 1) 1st subframe	3
GB1	Gain codebook (stage 2) 1st subframe	4
P2	Pitch delay second subframe	5
C2	Fixed codebook 2nd subframe	13
S2	Signs of fixed-codebook pulses 2nd subframe	4
GA2	Gain codebook (stage 1) 2nd subframe	3
GB2	Gain codebook (stage 2) 2nd subframe	4

L0-L3: LSF
(vocal tract model)

P0, P1, P2: Voice Pitch
(specific to language and person)

C1, S1, C2, S2:
Excitation (for voice residue coding)

GA1, GB1, GA2, GB2:
fitting parameters

Total: 80 bits / frame

Figure 3: G729 packet format

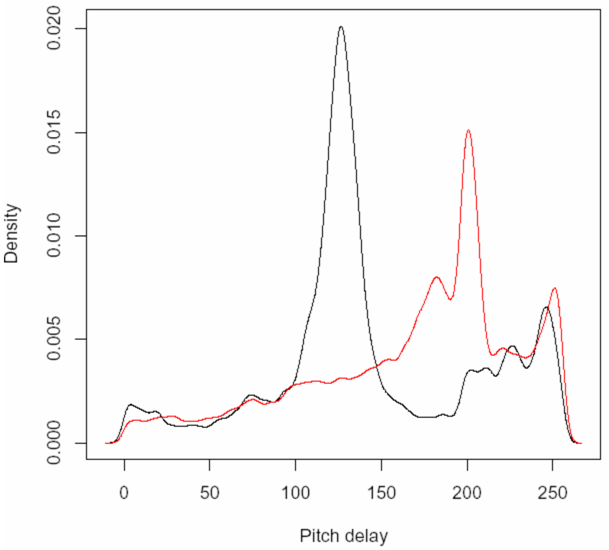

Figure 4: Density distribution of P1 parameter

We performed several experiments to verify the discrimination power of the CSR feature vector. Figure 4 compares the P1 parameter from two individual speakers, captured over a 30 second time period. It shows two very distinct density profiles, making this single parameter a useful discriminator. Other parameters also show similar discrimination properties (space limitations prevent us from presenting this information).

3. MICRO-CLUSTERING TECHNIQUE

Speaker recognition systems apply clustering algorithms to group similar feature vectors together to create unique speaker signatures. Traditionally, GMM is used in such a capacity in the Mel-Cepstra feature vector space. However, GMM requires high CPU utilization and long latencies (about 15 seconds [6]) which makes it ill-suited for deployment in a real-time environment. In addition, its performance can further degrade due to packet loss.

CSR employs a high speed, online, micro-clustering algorithm [3], which was developed specifically for high speed, evolving data streams [5]. This allows CSR to perform speaker recognition by capturing packets from a high speed, integrated network, where both data and voice are being transmitted. We refer the reader to [3] for details on the Micro-Clustering algorithm.

We briefly describe how micro-clustering is applied within CSR, which is shown in Figure 5. In the offline training phase, a micro-clustering algorithm [4] is used to create a set of multi-dimension clusters from a stream of CSR features. User signatures are created based on the distribution of the data points among the different clusters within the multi-dimensional feature space. These sets of user signatures are then used during online speaker recognition.

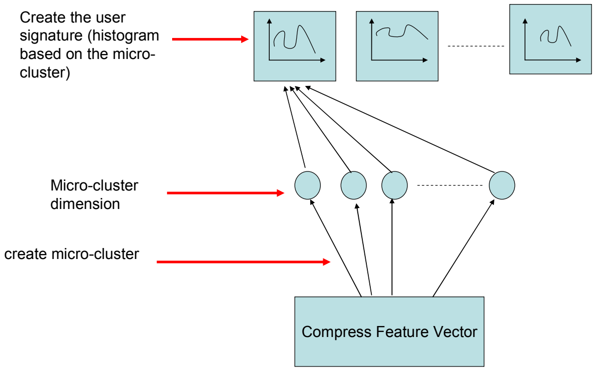

Figure 5: Micro-Clustering within CSR

4. EXPERIMENTAL SETUP

We implemented CSR and installed it on a Linux-based server to demonstrate its performance and accuracy. Our experimental testbed is shown in Figure 6.

The VoIP Traffic Generator consists of a set of processes which emulate real people by transmitting compressed speech in the form of G729 packets to a VoIP receiver (sink). The audio recordings used in our experiments were previously captured from radio and TV, and consisted of remarks made by well known politicians and newscasters. The audio for each speaker was transmitted from a separate IP address at a rate of 8000 bps.

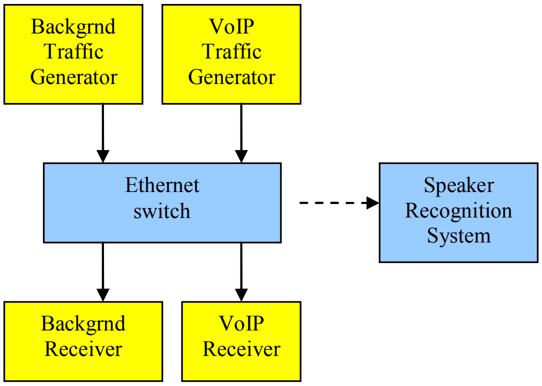

Figure 6: Experimental testbed

The Ethernet switch depicted in Figure 6 was configured to mirror all VoIP traffic between the VoIP Traffic Generator and the VoIP receiver to the Speaker Recognition System. In addition, background traffic generators where used to emulate traffic associated with various other non-VoIP protocols typically found on the Internet (HTTP, AOL chat traffic, etc).

After receiving a captured G729 packet from the switch, the Speaker Recognition first determines which flow it belongs to (based on IP addresses) and then performs feature based micro-clustering on the packet to identify the speaker talking over this flow. The confidence level for identifying the speaker is reported in addition to the actual person identified as the speaker on this flow.

5. EXPERIMENTAL RESULTS

Our results show that CSR can create user signatures with great discrimination power based on the CSR feature vector. The speaker recognition accuracy rate for CSR was 80% and CSR was three times faster than GMM. Our results also indicate that CSR is highly scalable. We predict (based on extrapolation of experimental data and up to 80% CPU utilization, as a conservative estimate) that CSR will analyze 5760 streams using one single PC.

Our experiments were based on voice data from 18 speakers captured from radio and TV. Each speaker had at least 46 seconds of speech, which was divided equally, for use in training and testing. The training set was used to construct the speaker's signature, as described in Section 3. The resulting signatures for several speakers are shown in Figure 7. The x-axis depicts the micro-cluster (from 1 to 40), and the Y-axis depicts the number of occurrences of the corresponding feature.

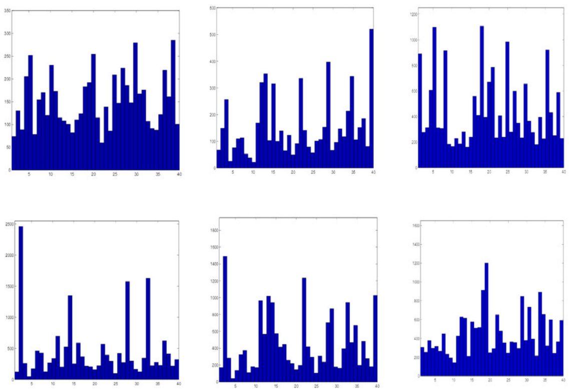

Figure 7: Examples of speaker signature histogram

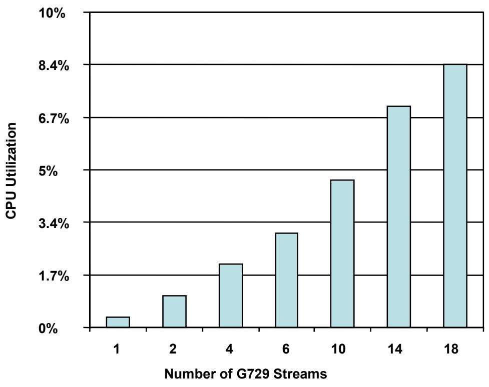

Figure 8: CSR scalability

After the offline learning phase, we conducted online experiments which involved analyzing the live compressed packet streams using micro-clustering to identify each speaker on a specific flow. During the online analysis a dynamic user signature was created for each speaker, which was then compared against the set of learned signatures for a best-fit match. CSR had an 80% success rate in correctly identifying the speaker, and could do so within 5 seconds of seeing the first packet associated with a speaker. This means CSR could be used in a real-time environment where it is imperative to identify the speaker in as few seconds as possible.

Figure 8 shows the measured CPU utilization for the micro-clustering based speaker recognition for different numbers of streams. Each stream required 0.5% of CPU utilization, on average. Using linear regression, we estimate that one such machine can support up to 160 (up to 80% utilization) streams. If true, then CSR can analyze voice traffic at a rate of 1920 (160*60/5) sessions per minute using an off the self PC. Each G729 stream has 24 kbits/sec (8 kbit data + 16 kbit overhead packet headers). Each CSR machine can process 1920 streams per minute. In order to support 5760 stream, it will take the VoIP processor 3 minutes to complete the processing. Total buffer required for the 5760 stream for 3 minutes would be 5760*24kbits*3*60=3.11 GBytes. This buffer requirement can be reduced by sub-sampling the compressed stream if needed.

6. SUMMARY

This paper presented Compressed Speaker Recognition (CSR), which is a scalable approach for performing speaker recognition directly from live, compressed VoIP packet streams. CSR creates a discriminating feature vector directly from the compressed VoIP packet. This eliminates much of the time consuming processing required by traditional approaches based on the decompress-FFT-Mel-Scale filter-Cosine transform. CSR employs a high-speed micro-clustering technique which allows it to analyze much higher bandwidth rates than any existing system. The accuracy of CSR has been examined and shown to be compatible with that reported in the NIST speech group annual evaluation report. At the same time, CSR requires less time for the speaker recognition. Importantly, CSR demonstrates good scalability in terms of CPU utilization and memory requirements. The CSR approach analyzes directly from the compressed voice packets. This will allow the compressed voice data packets to be sub-sampled and reduce the memory buffer requirement. This enhances scalability even further.

7. REFERENCES

[1] J. Douglas Reynolds, et. al., "Robust Text-Independent Speaker Identification Using Gaussian Mixture Speaker Models", IEEE Transactions on Speech and audio processing, Vol.3, No.1, Jan. 1995.

[2] Lajos Hanzo, et. al. "Voice Compression and Communications", John Wiley & Sons, Inc., Publication, ISBN 0-471-15039-8.)

[3] Charu C. Aggarwal, et. al., "A Framework for Clustering Evolving Data Stream", Proceeding of the 29th VLDB Conference, Berlin, Gemany, 2003

[4] A. Jain, R. Dubes, "Algorithms for Clustering Data", Prentice Hall, New Jersey, 1998

[5] L. O'Callaghan, et. al, "Streaming-Data Algorithms for High-Quality Clustering", ICDE Conference, 2002.

[6] Mark Prybocki, Alvin martin, "NIST's Assessment of Text Independent Speaker Recognition Performance", http://www.nist.gov/speech/publications/index.htm , 2002

[7] ITU G729 standard, "Coding of Speech at 8 kbits/s using Conjugate Structure Algebraic Code-Excited Linear Prediction (CS-ACELP)", ITU Study Group 15, 1995.