Speech Compression and Effects on Speech Recognition

Course Project for CSC 561: Multimedia Systems

University of Victoria - July, 2018

Table of Contents:

1. Introduction:

Speech compression has been very important since the development of digital communication, reducing the required bandwidth used for transmission and therefore improving the quality for audio calls. Today speech compression continues to play a very important role in allowing applications such as Mobile Telephony and VoIP to work as we know them. This project aims to give an introduction to the field and analyze the state of the art of speech compression. We'll also look into what are the effects that compression have on speech recognition. Finally, an implementation of speech compression with focus on speech recognition using latest algorithms is to be developed.

Speech can be compressed using the standard audio compression techniques available, however we want to achieve better compression rates given the importance of the applications. For that end, speech compression focuses on encoding only information that is relevant for the human ear (not all sounds present in the recording). Since speech is a simpler signal than most other audio signals and because we know the properties of speech and how is produced, we can create stricter encoders/decoders to model speech. The next section describes what are the known characteristics of speech used in compression systems, Section 3 introduces the types of speech compression and finally Section 4 presents the speech compression standards used today.

After having a clear look of how the field of speech compression looks like, we focus on the relation between compression and recognition of speech. As expected with lossy compression, part of the audio data is not recovered when decompressed. Although speech compression (mostly) prevents loss of data to be perceived by the human ear, it is interesting to analyze what are the effects on speech recognition algorithms. Section 5 describes these effects and what are the possible adjustments that can be made to speech recognition systems in order to deal with such changes.

2. Basic Concepts

Speech is digitalized by the process of sampling, quantization and coding. Sampling is relatively simple, usually based on the Nyquist frequency of the voice signal to be sampled to allow the recovery of the analog signal. For example, for narrow-voice band (4 kHz) the sampling frequency would be 8 kHz. According to [1] speech amplitude is not evenly distributed, uniform quantization creates high levels of noise. Therefor, non-inform quantization is usually used for speech and with finer quantization applied to

low speech signals. Coding is the process of representing the sampled and quantized values with bits, ideally using the minimum amount possible. We'll expand on this topic in Section 3 and 4.

Digitalized speech signals have some characteristics that are explored during coding and they occur due to how human speech is produced. The main characteristic used in coding is the fact that certain sounds produce quasi-periodic patterns in the signal (voiced sounds) due to how the vocal cords vibrate during the speech process. Other sounds that don't depend on the vibration of vocal cords produce a signal that resembles noise. These differences are used in frame-based speech coding.

The next section explains the main types of compression techniques, some of which use the characteristics described above to enhance compression.

3. Compression Techniques

There are three basic types of speech encoding. First we'll look into Waveform compression which is a simple approach with low compression rate. Secondly we'll analyze Parametric-based compression that uses knowledge of how speech is produced to generate a set of parameters to represent each speech segment. The decoder in turn would be able to reconstruct the speech based on these parameters. Finally the last type called Hybrid compression aims to unite the best parts of each method and has proven to generate very effective speech compression.

3.1. Waveform Compression

The most common type of waveform compression is called Pulse Code Modulation (PCM) [3] which removes correlation between samples as it digitizes analog signals. Compression in PCM happens by quantization of the amplitude of a sample. Research has shown that lower audio signals have more information about speech than higher audio signals [1], and because of that PCM uses non-uniform quantization, with focus on low signals thus providing better signal-to-noise ratio (SNR).

An improvement of PCM is called Adaptive Differential Pulse Code Modulation (ADPCM) which further compresses the output of PCM, it does that by incorporating a predictor component and a adaptive quantizer in the encoder. The predictor generates an estimate of the next speech segment based on the previous speech segment, then only the prediction error is passed to the quantizer and coded. Since the quantized

prediction error is smaller than the original PCM then ADPCM uses less bits to represent a speech segment.

PCM and ADPCM were initially designed for telephony but these techniques are also used in voice over IP communications.

3.2. Parametric Compression

Waveform compression exploits the similarities between samples and makes use of a prediction algorithm, one on encoder and one on decoder, in order to encode only the prediction errors. This makes audio signals smaller, but to a certain limit. For cases where very little bandwidth is available and we need to achieve more compression ratios, parametric compression might be the solution.

As opposed to waveform, parametric compression makes use of the knowledge we have about how human speech is produced. This encoder works by extracting speech parameters like pitch and gain (and others) for every given interval and encoding a sequence of these parameters. The decoder in turn should be able to reconstruct the speech based on these parameters using a voice synthesizer. Because only parameters are encoded, this method can achieve high levels of compression, but at the cost of speech quality. Although the decompressed result is still understandable, the use of a synthesizer makes the voice sound robotic and mechanical.

A common parametric encoder is Linear Prediction Coding (LPC), sometimes called a vocoder (voice encoder). Based on statistical information from speech LPC assumes that the speech signal is stable for periods of 20ms, the encoder then extracts information about each of these segments. Because of how the vocal tract works, these segments can be separated in two classes for easier parameters extraction and speech reconstruction. Segments generated by vocal tract and vibration of vocal cords are considered Voiced, while segments that use only the vocal tract are called unvoiced.

As mentioned before, the encoder extracts the pitch, the gain (power of the segment), the type of segment and vocal tract coefficients (given by a linear filter that represents the spectral envelope of the speech). All these parameters are quantized and sent to the decoder. The decoder feeds these parameters to a voice synthesizer which uses a period pulse train using pitch (for voiced segments) and white noise (for unvoiced segments) to generate a base wave, then finally the decoder introduces the gain and vocal tract coefficients to the wave in order to simulate the original speech.

3.3. Hybrid Compression

The previous section shows how parametric compression is able to expand the compression ratio of speech signal by coding only speech parameters used to recreate the signal on the decoder. However, this comes with a heavy price as the quality of the reconstructed speech is much degraded (but still understandable).

One of the limitations of parametric compression is the usage of a period pulse train to reconstruct speech. The main issue is that the signal generated is simplistic and different from the actual speech waveform. This becomes apparent when we remove the LPC estimation from the original speech, resulting a complex residual signal that still has the pulse patterns of speech.

In order to improve quality hybrid compression tries to find a “excitation signal” that would closely represent the LPC residual signal. For that a synthesizer is included in the encoder side so that a closed-loop search can be performed to find the best excitation signal. Similar to waveform compression, now some error (residual) is coded with a prediction algorithm in both encoder and decoder. But like parametric compression, this residual signal is represented by a excitation signal parameter that is used to reproduce the residual signal on the decoder side, other parameters like pitch, gain and voice/unvoiced decision are still sent along.

The differences between hybrid compression systems remains mostly on how the excitation signal is predicted. The most used type is called Code-Excitation Linear Prediction (CELP) [2] which uses a local list of up to 1024 potential excitation signals at both sender and receiver. The close-loop search in the encoder finds the best excitation signal and sends only the index to the decoder, since the lists are identical the decoder needs only the index to reproduce the signal. This allows hybrid compression to achieve relatively good quality (enough for regular communications) while maintaining low bit rates.

4. Compression Standards

4.1. Narrowband, Wideband, Super-wideband and Fullband

The standard range of frequencies used for speech compression used to be around 4 kHz (with 8 kHz sampling) because most properties of voice are present this in range, compression methods that uses this range for the speech spectrum are called **Narrowband**. However human speech is capable of producing frequencies from -30 Hz up to 18 kHz, so there are some frequencies which are lost when using narrowband

compression. Many new speech compression applications opt to follow more modern standards of frequency range such as **Wideband** which has speech spectrum range of 7 kHz, **Super-wideband** which has 14 kHz range and **Fullband** that has 20 kHz, covering the entire range of possible frequencies generated by human speech. In the following sub-section we'll describe some common standards of speech compression with a simple overview of their architecture and their choice of speech spectrum range (we'll focus on the standards that are later used on our tests).

4.2. Standards

G.726 ADPCM is a narrowband compression method with 8000 Hz sampling rate. As the name suggests these standard is based on ADPCM (Waveform Compression). It is an improvement over G.711 and G.723 and it has four bitrates 16, 24, 32 and 40 kbit/s. Originally designed for Digital Circuit Multiplication Equipment (DCME) but the main applications that use G.726 are international trunks in phone networks and VoIP.

G.729 (ACELP) [14] is a narrowband vocoder based on the hybrid compression method Algebraic Code Excited Linear Prediction (ACELP) which is an improvement over regular CELP by incorporating an algebraic codebook which lowers computational complexity. We note that although standard G.729 is royalty-free, ACELP is parented. Because it operates with low bitrate of 8 kbit/s, one of the main applications of G.729 is VoIP application for wireless or bandwidth-limited systems.

GSM is one of the most used speech codecs standards, used in more than 190 countries. There's are various sub-standards but we'll focus on the basic version called Full Rat which operates at 13kb/s and is based on Regular Pulse Excitation/Lord Term Prediction (RPE/LTP). Compared with current standard it has low voice quality but it was still a great achievement in its time. Even though it is still used today on some networks it has been slowly replaced by standard like Enhanced Full Rate and Adaptive Multi-Rate.

MELPe [15] is a standard of hybrid speech compression used by the U. S. Military in secure, satellite and radio communications. It operates at 2400, 1200 and 600 bit/s and it's best fitted for applications that require low bandwidth.

5. Effects of Audio Compression on Speech Recognition

Now we have an overview of how speech compression works and what are the standards used today. This section will look into what is the effect of joining speech

compression and recognition, why is it necessary and what are the ways to structure systems that rely on both this mechanisms so that recognition accuracy degradation remains within an acceptable level.

Speech recognition has become widely popular and it is very common to be used as an add-on for various applications (often mobile) that use wireless communications. Although certain speech recognition tasks can be performed by the device itself, complex cases usually require lots of memory and power so speech recognition servers are used, since this server is usually used by various applications at the same time then speech compression is necessary to reduce the bandwidth consumed. In this report we'll focus on the client-server approach which adds some constraints to the amount of data that can be send over the networks.

If the client application can be complex, certain systems rely on the client extracting the necessary acoustic features of speech, thus compressing and sending only these relevant parameters. However, In certain cases the client has to be simple and cannot extract such information, so the entire speech signal has to be transmitted to the server. Because of bandwidth and delay limits, the signal usually has to be compressed which somewhat alters the original signal.

In the next sections we'll look at some result that outline the effects compression can have on speech recognition and we'll analyze what methods exist to prevent compression from reducing the accuracy of these systems.

5.1. Speech Signal Compression

First lets analyze the effects that compression can have on speech recognition systems that deal directly with speech signal and not only features, which can tells us if there is an actual problem when merging compression and recognition.

In [4], the authors used around 400 samples of compressed speech signals, using standard methods GSM, MPEG, G.711 and G.723.1, on a French speech recognition system. Based on the Word Accuracy measure they found that GSM encoding had little or no effect on the accuracy of the system. Several variations of MPEG encoding were also tested and they found that MPEG operating lower than 32 kbits/s greatly degrades accuracy. G.711 and G.723 have shown very little degradation and no accuracy difference between the standards.

On other research [16] Muthusamy, Gong and Gupta shown degradation of speech recognition on more recent encoders such as G.726, G.729, GSM-FR and GSM-EFR. G.726 and G.729 appear to cause considerable degradation, specially on noisy

speech. GSM-FR and GSM-EFR, similar to results of [4], show very little degradation and, interestingly, they also found that for noisy speech samples GSM encoding shows accuracy improvement because of its noise removing properties.

Certain encoding techniques appear to degrade the accuracy of speech recognition while other have no effect to even can cause improvements. It is up to system architects to understand the potential effects and take measure if necessary. In the case where speech features can not be extracted before compression there are still way to prevent degradation for being too severe. Authors of [4] show one example where their system was trained using encoded speech instead of plain speech and with this new training they were able to recover acceptable speech recognition accuracy.

5.2. Acoustic Features Compression

As mentioned above, compressing the entire audio signal can lead to lower speech recognition accuracy as relevant speech properties can not be removed on decompression. In some cases it is not possible to adjust the speech recognition system to compensate for this loss (example: by training with compressed data as mentioned on the previous section) so we need an alternative.

In cases where the client is expected to have a certain degree of complexity, it is possible to envision a speech recognition system where the client collects the speech data, extracts speech relevant features (as describe by Bahl in [17] using MFCCs or similar), compresses the features and transmits the encoded data to the server. The server in turn will receive said data, decompress it and use it as input for the speech recognition engine. In this system the compression happens after extraction so, as long as the compression of the features has acceptable noise, the relevant information is kept intact.

Ramaswamy and Gopalakrishnan in [5] describe a compression algorithm specific for encoding acoustic features which has low computational complexity and low memory usage. The algorithm described takes in 13-dimensional MFCCs features vectors for each 10ms speech segment, then differences between adjacent frames are calculated to take advantage of the correlation in time between samples, this forms a linear prediction step. After this step the resulting error is quantized in a 2-step process and ready to be transmitted and decoded.

Evaluation of the compression algorithm was done using a dataset of 6144 uttered words from 9 different speakers. The encoder is able to operate at a rate of 4 kbits/s while actually increasing accuracy of the system by a slight margin.

6. Implementation and Tests

In this section we show our tests were used various speech recognition systems from a library to measure the effects compressed speech have on their accuracy.

6.1. Dataset

In order to better understand the effects of compression on speech recognition we collected a set of speech samples with different levels of compression to be used for testing. These samples are from a dataset used as benchmark by Signalogic [6] which contains speech samples of male and female encoded speech. Our tests used the MELPe-Plus, G726 and G729 samples. Some of the speech recognition systems we considering don't handle long inputs well (Wit.ai), and since some of these speech samples are relatively long we've also broken these samples into smaller ones (making sure that not uttered word were cut off).

We also wanted to generate our own compressed samples from a modifiable project in order to analyze possible ways to make changes to it and measure the improvements, but because many speech compression algorithms are proprietary it is difficult to find a open source implementation. We settled on using World Vocoder [7], an open source high-quality vocoder. More specifically, we used PyWorldVocoder [8], a wrapper for World Vocoder.

6.2. World Vocoder Analysis

World Vocoder was developed with the goal of being used in real-time applications (example: real-time transcription from speech streaming). Various other vocoder (hybrid based) have been able to achieve high-quality speech, but at the cost of high computational complexity. In order to allow World Vocoder to be used in real-time applications the focus of the project was on improving the speed of the vocoder, for that three known feature extraction algorithms were used: **Fundamental Frequency** is estimated by using DIO [11] which dispenses the necessity of the often used Intensive Short-Time Fourier Transform for each frame and instead works on the entire speech signal, **Spectral Envelope** is estimated with CheapTrick [12] which uses only one power spectra and **Excitation Signal** is estimated using PLATINUM [13] which has not need for post-processing unlike similar algorithms.

6.3. Speech Recognition

Next we needed a speech recognition system in order to run our samples. Python's SpeechRecognition [9] library offers a wide range of speech recognition engines, which helps us achieve more reliable results by running tests on multiple speech recognition systems. For our tests we used Google Speech Recognition, Sphinx, Wit.ai and Houndify. Speech recognition systems are usually evaluated based on accuracy and computation complexity, since our goal is analyze the effects of compression on recognition we'll focus on the accuracy measure. There are various methods for measuring accuracy of speech recognition transcriptions such as Word Error Rate (WER) [10] or Single Word Error Rate (SWER). For our tests, for each speech recognition system we first computed the WER using uncompressed speech samples and then computed WER for compressed samples, finally we represent the speech recognition accuracy degradation by the percentage of WER lost.

6.4. Results

Table 1 shows the degradation of accuracy that we found for each combination of speech recognition system and speech compression method. A few notes about the tests:

- Wit.ai had problems recognizing certain MELPe-Plus samples because certain levels of degradation cause the algorithm to stop (not designed for long sentences). We still used these long samples when reporting the degradation of Wit.ai to maintain consistency but we note that when breaking up samples into smaller subsets the degradation caused by compression on the Wit.ai system becomes minimal.

Percentage of Speech Recognition Degradation (less is better, 0 is optimal)	**MELPe-Plus**	**G.726**	**G.729**	**World Vocoder Output**
Google Speech Recognition	51.8%	39.9%	25.9%	35.5%
Sphinx	51.2%	49.6%	48.3%	35.4%
Wit.ai	61.6%	15.4%	23.8%	35.0%
Houndify	42.0%	29.4%	21.7%	18.1%

Table 1: Speech recognition degradation for compressed samples

We can see a general degradation that occurs when speech recognition systems handle compressed speech, most drastically on MELPe-Plus which is expected since the bitrate of this method is lower than the others. The degradation levels found for G. 726 and G.729 are consistent with the results found by Muthusamy, Gong and Gupta [16]. On average MELPe-Plus suffers 51% degradation, G.726 suffers 33.57% and G. 729 suffers 29.9% degradation. The most affected system seems to be Sphinx with average 46.1% degradation of accuracy when facing compressed speech and Houndify has been the most robust system suffering only 27.8% average degradation of accuracy.

The samples generated by World Vocoder have much lower quality of speech with robotic and mechanic sounds, however it doesn't seem to suffer so much in terms of accuracy degradation, specially when compared with MELPe-Plus which still has relatively good quality but affects speech recognition very strongly. Our focus of evaluation have been only on accuracy but we note that Sphinx and Wit.ai required the most processing time when encoding the plain samples of speech.

7. Conclusion

In this report we've looked at the basic concepts of how speech in produced and what are the effects on digitalization of speech and feature extraction. Speech compression makes use of these concepts to achieve better results. We've described the three main types of speech compression. Waveform compression performance relies mostly on the prediction component and on the quantization step and achieves high speech quality but low compression ratio. Parametric compression extracts certain features that represent speech, encoding only these parameters so that speech can me recreated by the decoder, it achieves low quality of speech but high compression ratios. Hybrid compression joins these two method by still using parametric extraction but incorporates a predictor and instead of sending only parameters it also sends the difference between original signal and the predicted speech wave.

We've shown that for a common architecture of client-server speech recognition the use of compression is required, and this compression can affect the accuracy of speech recognition systems. Certain compression methods have stronger effects while other may actually increase accuracy by removing noise from speech. For the cases where compression becomes a problem for recognition there are measures that can be taken to improve accuracy such as using compressed samples as part of the training

dataset or extracting important speech features and sending this features instead of the actual speech signal to the speech recognition system.

Along with this report some tests were performed to better understand the effects of compression on speech recognition. We've tested different combinations of compression methods and speech recognition systems to prevent bias results, and the results we found support the results from other papers presented in section 5. It is clear that when implementing speech recognition systems one must take the necessary steps to prevent compression from reducing accuracy.

8. References

1. L. Mkwawa, E. Jammeh, E. Ifeachor. Guide to Voice and Video over IP. Springer-Verlag London 2013.
2. A. Gersho. Advances in speech and audio compression. Proceedings of the IEEE (Volume: 82, Issue: 6, Jun 1994).
3. Y.Yatsuzuka, Highly Sensitive Speech Detector and High-speed Voiceband Data Discriminator in DSI-ADPCM Systems," IEEE Trans Commun., vol. COM-30, pp. 739-750, Apr. 1982.
4. L. Besacier, C. Bergamini, D. Vaufreydaz, E. Castelli. THE EFFECT OF SPEECH AND AUDIO COMPRESSION ON SPEECH RECOGNITION PERFORMANCE. IEEE Multimedia Signal Processing Workshop, Oct 2001, Cannes, France. pp. 301-306, 2001.
5. G.N. Ramaswamy, P.S. Gopalakrishnan. COMPRESSION OF ACOUSTIC FEATURES FOR SPEECH RECOGNITION IN NETWORK ENVIRONMENTS. IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 1998.
6. Signalogic - DSP Real-Time Algorithms: http://www.signalogic.com/index.pl?page=dsprt.
7. M. Morise, F. Yokomori, K. Ozawa. WORLD: A Vocoder-Based High-Quality Speech Synthesis System for Real-Time Applications. IEICE TRANSACTIONS on Information and Systems, 2016.
8. PyWorldVocoder - A Python wrapper for World Vocoder. https://github.com/JeremyCCHsu/Python-Wrapper-for-World-Vocoder.
9. SpeechRecognition - Library for performing speech recognition, with support for several engines and APIs, online and offline: https://pypi.org/project/SpeechRecognition/.
10. D. Klakow, J. Peters. Testing the correlation of word error rate and perplexity. Speech Communication - Volume 38, Issues 1–2. September 2002, Pages 19-28.
11. M. Morise, H. Kawahara, H. Katayose, Fast and reliable f0 estimation method based on the period extraction of vocal fold vibration of singing voice and speech in Proc. AES 35th International Conference, CD-ROM Proceedings, 2009.
12. M. Morise, Cheaptrick, a spectral envelope estimator for high-quality speech synthesis Speech Communication, vol.67, pp.1–7, 2015.

13. M. Morise, Platinum: A method to extract excitation signals for voice synthesis system. Acoust. Sci. & Tech., vol.33, no.2, p.123–125, 2012.

14. C. Yeh, C. Zhuo. An efficient complexity reduction algorithm for G.729 speech codec. Computers & Mathematics with Applications. Volume 64, Issue 5, September 2012, Pages 887-896.

15. U. S. Department of Defense, Specifications for the Analog to Digital Conversion of Voice by 2400 bit/second mixed excitation linear prediction.

16. Y. Muthusamy∗, Y. Gong, R. Gupta. The Effects of Speech Compression on Speech Recognition and Text-to-Speech Systhesis - Speech Technologies Lab, DSP Solutions R&D Center, Texas Instruments, Dallas, Texas, US.

17. L. Bahl. Performance of the IBM Large Vocabulary Continuous Speech Recognition System on the ARPA Wall Street Journal Task. Proceedings of the IEEE ICASSP, Detroit, pp. 41-44, May 1995.