Comparison of Speech Coding Algorithms:

ADPCM, CELP and VSELP

Submitted By:

Wei Li

Anu Sridhar

Tina Teng

Project for EE 6390, Fall 1999

Instructor: Dr. Murat Torlak

University of Texas at Dallas

Abstract

The past decade has witnessed substantial progress towards the application of low-rate speech coders to civilian and military communications as well as computer-related voice applications. Central to this progress has been the development of new speech coders capable of producing high-quality speech at low data rates. Most of these coders incorporate mechanisms to: represent the spectral properties of speech, provide for speech waveform matching, and "optimize" the coder's performance for the human ear. A number of these coders have already been adopted in national and international cellular telephony standards.

In mobile communication systems, service providers are continuously met with the challenge of accommodating more users within a limited allocated bandwidth. For this reason, manufactures and service providers are continuously in search of low bit-rate speech coders that deliver toll-quality speech.

The objective of this project is to compare three commonly used algorithms in wireless communication systems: ADPCM, CELP and VSELP. The project report starts with the description of these speech coders. Then we present our implementation results and finally give concluding remarks followed by comments on future research in this area.

1. Introduction

In mobile communication systems, since service providers are continuously met with the challenge of accommodating more users within a limited allocated bandwidth, speech coders that provide toll quality speech at low bit rates are needed. The goal of all speech coding systems is to transmit speech with the highest possible quality using the least possible channel capacity. In general, there is a positive correlation between coder bit-rate efficiency and the algorithmic complexity required to achieve it. The more complex an algorithm is, the more is processing delay and cost of implementation.

In this project, we have looked at the bit-rate versus algorithmic complexity for three speech coders: ADPCM, CELP and VSELP. The motivation to considering these speech coders arises from the fact that CDMA, North America Digital Cellular Telephone Systems, etc. make use of these speech coders.

Speech coders differ widely in their approaches to achieving signal compression. Based on the means by which they achieve compression: ADPCM is classified as a waveform coder, while CELP/VSELP are known as analysis-by-synthesis coders.

Waveform coders essentially strive to reproduce the time waveform of the speech signal as closely as possible. They have the advantage of being robust for a wide range of speech characteristics and for noisy environments. All these advantages are preserved with minimal complexity, and in general these coders achieve only moderate economy in transmission bit rate.

Analysis-by-synthesis speech coders are among the newest and most effective of modern speech coders. They use most of the available information from the speech signal to improve the quality and reduce the bit-rate. In particular, they make use of aural noise-masking, aural frequency resolution, aural phase insensitivity, syllabic energy variation, the long-term vocal tract properties, and pitch information in the coding process. In general, this makes these coders among the best-quality, lowest bit-rate, and most computationally demanding of speech coding techniques.

In the following section, we discuss ADPCM, while section 3 deals with analysis-by-synthesis linear predictive coders. In particular, CELP and VSELP coders are considered. In section 4, we give the implementation details of various speech coders and the source-code of these implementations is given in the appendix. Finally in section 5, a table summarizing the performance and complexity of these speech coders is given along with the current and future research in speech coding.

2. Adaptive Differential Pulse Code Modulation (ADPCM)

Differential coding refers to coding the difference between two signals rather than the signals themselves. In differential coding, the short-term redundancy of the speech waveform is removed as much as possible. This is accomplished by forming an *error (difference) signal* by subtracting an estimate of the signal from the original signal. The estimate is generally obtained by a linear predictor that estimates the current samples from a linear combination of one or more past samples. The main source of performance improvement for DPCM coders is the reduced dynamic range of the quantizer input signal. Since the quantization noise is proportional to the step size, a signal with a smaller dynamic range can be coded more accurately with a given number of quantization levels.

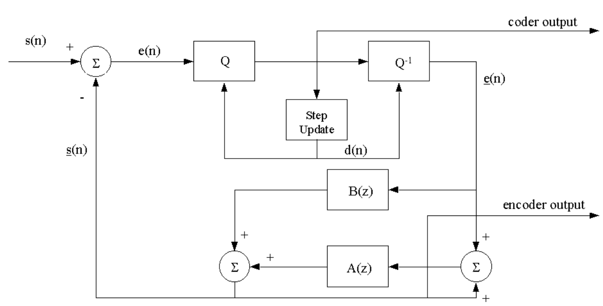


Figure 1. ADPCM G.721 Encoder

The basic issues in differential coding are the predictor properties and the quantization of the prediction error signal. DPCM with adaptation of the quantizer is known as Adaptive DPCM (ADPCM) as shown in Figure 1. The ADPCM algorithm is conceptually similar to DPCM but more sophisticated in that it uses an eight-order predictor, adaptive quantization, and adaptive prediction. Furthermore, the algorithm is designed to recognize the difference between voice or data signals and use a fast or slow quantizer adaptation mode, respectively. ADPCM provides greater levels of prediction gain than simple DPCM depending on the sophistication of the adaptation logic and the number of past samples used to predict the next sample. The prediction gain of ADPCM is ultimately limited by the fact that only a few past samples are used to predict the input and the adaptation logic only adapts the quantizer – not the prediction weighting coefficients.

3. Analysis-by-Synthesis Linear Predictive Coders

Analysis-by-synthesis speech coders, which include such important classes of coders as *code-excited linear predictive* (CELP) coders, are among the newest and most effective of modern speech coders. Note that the analysis-by-synthesis LPC is essentially a hybrid coder in the sense that it combines the features of model-based vocoders, by representing the formant and the pitch structure of speech, and the properties of waveform coders by providing for the matching of the input speech waveform. In the following subsections, we describe CELP and *vector sum excited linear predictive* (VSELP) coders.

3.1 Code Excited Linear Predictive Coder (CELP)

Medium or low bit-rate speech coders have been researched for application to mobile radio communications. Code excited linear prediction (CELP) coding is one of the most effective coding methods at low bit-rates, which was proposed in the mid-eighties by and Schroeder and

Atal [5]. CELP algorithm can produce low-rate coded speech comparable to that of medium-rate waveform coders thereby bridging the gap between waveform coders and vocoders.

CELP (as shown in Figure 2) is an efficient closed loop analysis-synthesis method for narrow and medium band speech coding systems [4-16 kbps]. In CELP coders, speech is segmented into frames (typically 10-30 ms long) and for each frame an optimum set of linear prediction and pitch filter parameters are determined and quantized. Each speech frame is further divided into a number of subframes (typically 5 ms) and for each subframe an excitation codebook is searched to find the input vector to the quantized predictor system that gives the best reproduction of the speech signal. For simplicity of design and performance robustness, the encoding of the LPC parameters is commonly achieved by the scalar quantization of the equivalent set of Reflection Coefficients, Log Area Ratios or Line Spectrum Pairs.

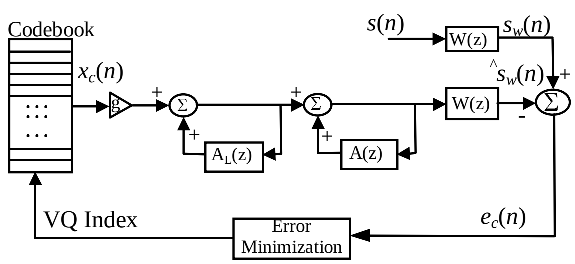


Figure 2. Analysis-by-Synthesis CELP Encoder.

CELP coders operating at low bit rates (less than 1 bit/sample) are not able to adequately reproduce the high frequency components or the transients in speech signal. The reconstructed signal suffers from a degradation that is more noticeable for high-pitched voices with strong glottal excitation pulses such as female voice. This limitation is partly due to the size of the excitation code book and its population density and partly due to fixed frame length LPC analysis where one set of LPC parameters are used to describe spectral information about a frame of speech that may contain widely different speech events. In general the reproduction accuracy improves with increasing excitation codebook size and when the code book population is a better approximation to the speech excitation distribution.

3.2 Vector Sum Excited Linear Predictive Coder (VSELP)

CELP speech coders exhibit good performance at data rates as low as 4800 bps. The major drawback to CELP type coders is their large computational requirements. This problem motivated a great deal of work focussed upon developing structured codebooks and fast search procedures.

Gerson and Jasiuk [3] proposed a vector sum excited linear predictive (VSELP) coder (as shown in Figure 3), which is associated with fast codebook search and robustness to channel errors, for use in digital cellular and mobile communications. An 8 kbps VSELP coder was selected by the Telecommunications Industry Association (TIA) as the standard for use in North American digital cellular telephone systems.

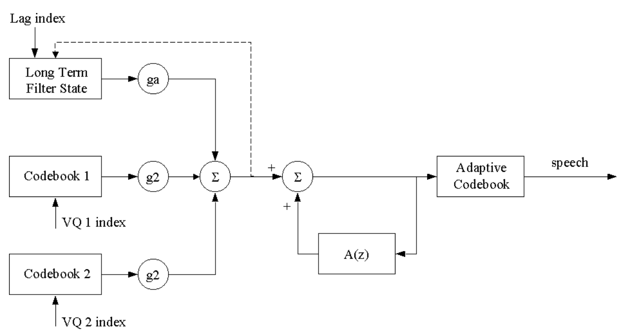


Figure 3. Analysis-by-Synthesis VSELP Encoder.

The code books in the VSELP encoders are organized with a predefined structure which significantly reduces the time required for the optimum code word search. The VSELP codec utilizes three excitation sources. The first is from the long term (“pitch”) predictor state (adaptive codebook). The second and third sources are from two VSELP excitation codebooks. Each of these VSELP codebooks contains the equivalent of 128 vectors. These three excitation sources are multiplied by their corresponding gain terms and summed. This becomes the combined excitation sequence ex(n). The synthesis filter is a direct form 10^{th} order LPC all-pole filter. The LPC coefficients are coded once per 20 ms frame and updated in each 5 ms subframe through interpolation. The excitation parameters are also updated in each 5 ms subframe. The number of samples in a subframe is 40 at an 8 kHz sampling rate. The implementation details of the algorithm, including codebook information, post filtering, etc. are presented in the IS-54 standard [2].

4. Implementation Details

We implemented PCM, non-uniform companding (μ-law), DPCM in MATLAB (illustrated in Figure 4). Further, we down loaded the kernel code of ADPCM and CELP algorithms (in C language) from the web [6]. Then, we did some extra programming to make it as a complete version executable on the SUN UltraSparc workstation. The comparison results refer to Figure 5.

For the MATLAB implementations:

(1) Input speech has been sampled at 8 kHz.

(2) For PCM, we have used a uniform quantizer with 2^8 = 256 levels. The bit-rate is 64 kbps.

(3) For non-uniform companding, we have considered μ-law. Here, $\mu = 255$ and the uniform quantizer has 2^5 = 32 levels. In this case, the bit-rate is 40 kbps.

(4) For DPCM, we have used an adaptive first-order linear predictor with coefficient a = 0.45. Again, we have a uniform quantizer with 2^5 = 32 levels. Here also, the bit-rate is 40 kbps.

5. Concluding Remarks

5.1 Summary

Table 1 summarizes the performance and complexity of the speech coders: PCM, ADPCM, CELP, and VSELP. The complexity figures, which are expressed in terms of MIPS (millions of instructions per second), have been obtained from different sources and are processor-dependent. We note that as the bit-rate goes down, the computational complexity increases on a large scale. This introduces a delay as well as an increase in the cost of implementation.

Table 1. Performance and Complexity of Algorithms.

Algorithm	Bit Rate (bit/sec)	MIPS
PCM	64 k	0.01
ADPCM	32 k	2
CELP	4.8 k	16
VSELP	8 k	13.5

5.2 Current and Future Research

Although, high-quality low-delay coding at 16 kbps has been achieved, low-delay coding at lower rates is still a challenging problem. Improving the performance of low-rate coders operating in noisy channels is also an open problem. Additionally there is a demand for robust low-rate coders that will accommodate signals other than speech such as music. Further, current research is also focused in the area of VoIP.

References

[1] T. P. Barnwell III, K. Nayebi and C. H. Richardson, "SPEECH CODING, A computer Laboratory Textbook", John Wiley & Sons, Inc. 1996.

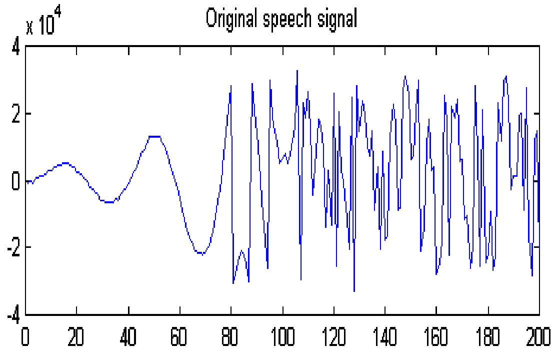
[2] EIA/TIA-PN2398 (IS-54), "The 8 kbits/s VSELP algorithm", 1989.

[3] I. Gerson and M. Jasiuk, "Vector Sum Excited Linear Prediction (VSELP) Speech Coding at 8 kbits/s", Proc. ICASSP-90, pp. 461-464, New Mexico, Apr. 1990.

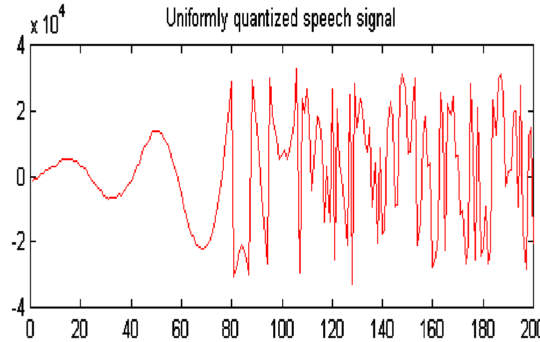
[4] T. S. Rappaport, "Wireless Communications", Prentice Hall, Inc. 1996.

[5] M.R. Schroeder and B. Atal, "Code-Excited Linear Prediction (CELP): High Quality Speech at Very Low Bit Rates", Proc. ICASSP-85, p. 937, Tampa, Apr. 1985

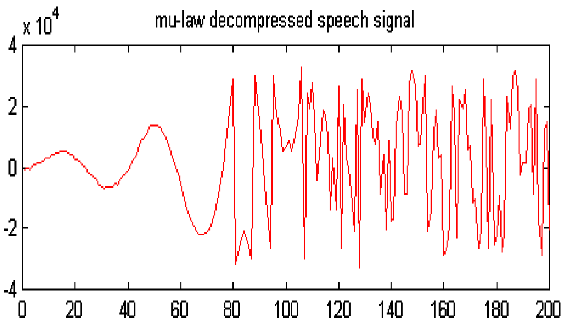
[6] ftp://ftp.cs.cmu.edu/project/fgdata/speech-compression/



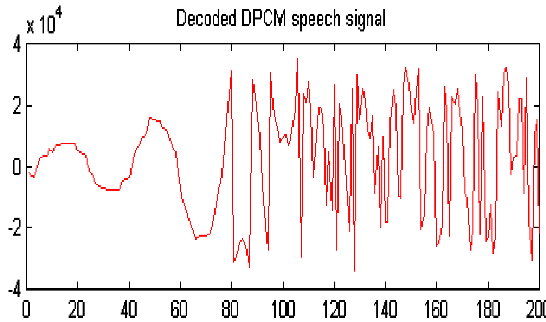
(a)



(b)

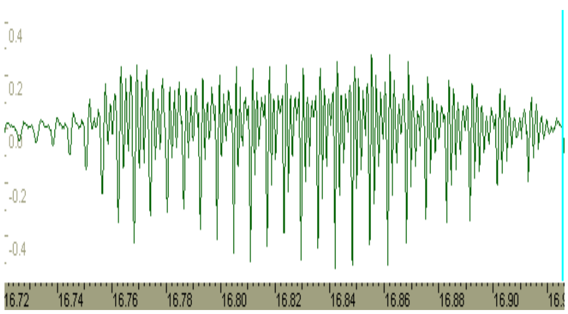


(c)

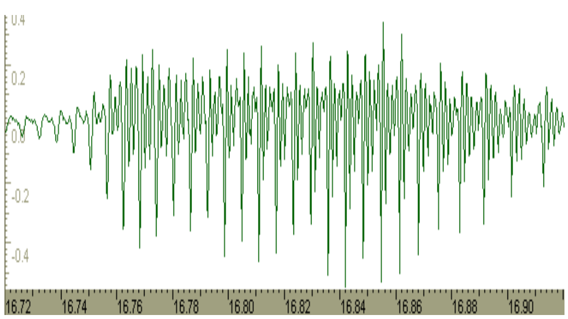


(d)

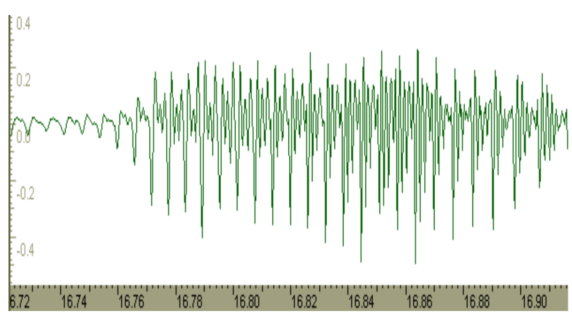
Figure 4. Matlab Simulation: (a) time domain input speech, (b) uniform quantized output waveform, (c) non-uniform quantized output waveform and (d) DPCM output waveform.



(a)



(b)



(c)

Figure 5. Simulations: (a) time domain input speech, (b) ADPCM output waveform

and (c) CELP output waveform

Appendix

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%  UNIFORM QUANTIZATION of given speech signal        %
%  s_in:  input signal;  step_size = Smax/2^n            %
%  where Smax: max signal amplitude and 2^n is        %
%  the # of levels in the uniform quantizer           %
%  operating with n bits. We have used a              %
%  mid-tread quantizer.                               %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear
%  generate the input signal
fid = fopen('hw1.raw','r');      % open given raw data file
s_in = fread(fid,'int16'); % read the given data into vector X

COUNT = 500;
Smax = 32767;           % max value of input s_in
levels = 256;           % number of levels in unif quant
nbits = 8;
step_size = (2*Smax)/levels;        % step-size of the quant

quant_in = s_in/step_size;                      % input to

quantizer
signS = sign(s_in);

% mid-tread quantizer

for I = 1:COUNT,
   S(I) = abs(quant_in(I))+0.5;
   quant_out(I) = signS(I)*round(S(I))*step_size;
end

% plot the original and the quantized signal

x = ([1:200]);
subplot(2,1,1);
plot(x,s_in([1:200]));
title('Original speech signal');

subplot(2,1,2);
plot(x, quant_out([1:200]), 'r');
title('Uniformly quantized speech signal');

fclose(fid);
clear
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%  NON-UNIFORM QUANTIZATION of given speech signal      %
%  s_in:  input signal;  step_size = Smax/2^n              %
%  where Smax: max signal amplitude and 2^n is          %
%  the # of levels in the uniform quantizer             %
%  operating with n bits. We have used mu-law for       %
%  compressing/decompressing the speech signal.         %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clear

%  generate the input signal

fid = fopen('hw1.raw','r');         % open given raw data file
s_in = fread(fid,'int16');       % read the given data into vector X

COUNT = 500;
Smax = 32767;              % max value of input s_in
levels = 256;              % number of levels in unif quant
nbits = 5;
mu = 255;
step_size = (2*Smax)/levels;        % step-size of the quant
denom = log(1 + mu);

% compress the input signal using mu-law

signS = sign(s_in);
for k = 1:COUNT,
   s_comp(k) = Smax * signS(k)*log(1+(mu*abs(s_in(k))/Smax))/denom;
end

% pass through the uniform quantizer

quant_in = s_comp/step_size;                        % input to quantizer
signX = sign(s_comp);
for I = 1:COUNT,
   S(I) = abs(quant_in(I))+0.5;
   quant_out(I) = signX(I)*round(S(I))*step_size;
end

% decompress the output signal

Signy = sign(quant_out);
y_out = Signy .* ((exp(quant_out.*denom .* Signy/Smax) - 1))*(Smax/mu);

% plot the original and the decompressed signal

x = ([1:200]);
subplot(2,1,1);
plot(x, s_in([1:200]));
title('Original speech signal');

subplot(2,1,2);
plot(x, y_out([1:200]), 'r');
title('mu-law decompressed speech signal');
fclose(fid);
clear
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%  Differential PCM (DPCM): first-order adaptive        %
%  prediction with uniform quantization.             %
%  s_in:  input signal;  step_size = Smax/2^n           %
%  where Smax: max signal amplitude and 2^n is       %
%  the # of levels in the uniform quantizer          %
%  operating with n bits. We have used a             %
%  mid-tread quantizer. Here, n=5.                   %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clear
%  generate the input signal

fid = fopen('hw1.raw','r');      % open given raw data file
[s_in,COUNT] = fread(fid,'int16');  % read the given data into vector X

COUNT = 500;
Smax = 32767;            % max value of input s_in
levels = 32;             % number of levels in unif quant
nbits = 5;
step_size = (2*Smax)/levels;        % step-size of the quant

% find the difference signal with alpha=0.45

alpha = 0.45;
diff_sig(1) = s_in(1);
for k = 2:COUNT,
   diff_sig(k) = s_in(k) - alpha*s_in(k-1);
end

% quantize the difference signal

quant_in = diff_sig/step_size;                      % input to quantizer
signS = sign(s_in);

% mid-tread quantizer

for I = 1:COUNT,
   S(I) = abs(quant_in(I))+0.5;
   quant_out(I) = signS(I)*round(S(I))*step_size;
end

% decode the signal using the quantized difference signal

s_out(1) = quant_out(1);
for k = 2:COUNT,
   s_out(k) = quant_out(k) + alpha*s_out(k-1);
end

% plot the original and the quantized signal
x = ([1:200]);
subplot(2,1,1);
plot(x,s_in([1:200]));
title('Original speech signal');
subplot(2,1,2);
plot(x,s_out([1:200]), 'r');
title('Decoded DPCM speech signal');
fclose(fid);
clear
```