INCORPORATING ENVELOPE INFORMATION FOR LOW BIT RATE VOCODERS

SUN Xun, WANG Hongqiang, HU Hongtao and Limin DU
Lab for Interactive Information Systems, P.O box 2712, Beijing, 100080, P.R.China
E-mail: sx@cenpok.net, Tel: +86-10-62627570

ABSTRACT

This paper presents an approach of incorporating speech envelope in low bit rate speech compression algorithm. The speech envelope is extracted by a specially designed peak-picking and interpolation scheme and quantified with 16 models that were acquired by a cluster analysis method. Experiments showed an improvement through traditional vocoders.

1. INTRODUCTION

Low bit rate speech coding is more and more important as the fast development of digital communication and multimedia network. When bit rate goes down below 4 KBPS, the timber of vocoders has apparent changes from the original speech signal, although some algorithms can give an intelligible output. Upon review of thousands of original speech waveforms and the output of vocoders, we found that the speech envelope in time domain had obvious difference between original speech and the synthesized output of vocoders, even the LPC spectrum can be matched perfectly. This leads to the idea that incorporating envelope parameters in vocoder may improve its performance. In this letter, we quantify the envelope with only 4 bits (16 models) and use it to adjust the amplitude of the output of linear prediction synthesizer. Listening tests showed that this approach got a better result than traditional vocoders.

2. ENVELOPE EXTRACTING

In this letter speech envelope is defined as the contour of peaks in each pitch period. We assume the positive part and negative part of speech envelope is approximately symmetrical. Under this presumption, only the positive part of envelope needs to be extracted. Because direct current bias may be introduced at recording, we must first remove the direct current bias of input speech otherwise the presumption of symmetry can not be satisfied. After the DC bias correction, the negative part of speech is then cut off. Let us assume the speech signal now is $x(n)$. Peak picking algorithm is now used to find the peaks of each pitch period. This is accomplished through the following steps: First we find the greatest value within the current frame, and then a rectangle window is applied to exclude all the points within the window. This is necessary because many points around the greatest value have amplitudes greater than the greatest value of other pitch periods in the same frame. One key parameter of rectangle window is its width. In our experiments we found that a constant width could not fit the variance of speech waveform. A simple way to conquer this problem is to make the width pitch depended. Experiments results showed that the optimum width of rectangle window was 0.75 multiplying the pitch period. By applying a rectangle window, one frame is divided into three parts. The first part is from the beginning point of the frame to the left bounder of the window, the second part is within the window and the third part is from the right bounder of the window to the end point of the frame. The second part is discarded. Pitch-picking algorithm is applied to the first part and the third part respectively. After find the greatest values of these two parts, two rectangle windows are applied to the two parts respectively. Such a recursive procedure continues until all the points in current frame are discarded. Then we get the peak points and their values of the current frame.

In order to get the envelope, we interpolate the peaks using the following method. Given n points x_i (i=0, 1, ⋯, n−1), we compute the functional value z of the interpolation point t with a seven order Lagrange interpolation. Eight points are selected automatically under the condition that they should make the interpolation points t just at the middle of them. That is, these eight points satisfy $x_k < \cdots < x_{k+3} < t < x_{k+4} < \cdots < x_{k+7}$

The functional value z of the interpolation point t is computed as following:

$$z = \sum_{i=k}^{k+7} y_i \prod_{\substack{j=k \\ j \neq i}}^{k+7} [(t - x_i) / (x_i - x_j)]$$

where y_i is the functional value of point x_i.

The above procedure can get a good result for singe frame speech. But when continuous speech frames are considered, the envelopes extracted using the above scheme are not continues as they should be for two continuous frames. To conquer this problem, we save the last two peak values of previous frame and used them together with the peak values of the current frame when interpolating peaks to get the envelope. Because of taking use of the previous two peaks, the envelope must be output with one frame delay.

3. ENVELOPE QUANTIFICATION

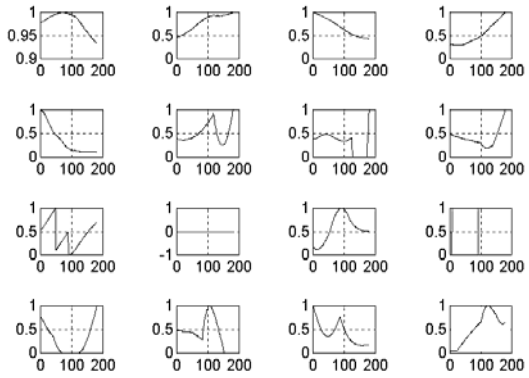

Fig.1 Models of speech envelope. The x-axis represents time, the y-axis represents relative amplitude.

Envelope is quantified once a frame. An envelope has two parameters: one is its shape, the other is its amplitude. After an envelope is extracted, we normalize it, that is, multiply the envelope with a factor to make its greatest value equal to 32767. The multiplication factor is then quantified with 6 bits using a log table whose step is 0.773db. Taking both the bit rate and quantification error into account, we use 4 bits to quantify envelopes. The quantification table is acquired by cluster analysis. Let us first define some terms used below. An element in this paper means an single frame envelope. Because in our program one frame is chose to be 180 samples (22.5ms) in length, an envelope has 180 samples. A class means a set of envelopes that satisfy the same criterion. Inter-element distance $d(x_1, x_2)$ is defined as $d(x_1, x_2) = \sqrt{\sum_{i=1}^{N} (x_1(i) - x_2(i))^2}$, where x_1, x_2 are two elements and N is the number of samples in each element. Inter-class distance is defined as the minimum element distance between each element in one class and each element in other class. The scheme of the cluster analysis algorithm of waveform envelopes is as following:

initial;
while (number of classes greater than 16)
do
  compute minimum distance between every two classes;
  incorporate classes;
  number of classes minus one
done
compute average of each class;
output average of each class;

We define the initial number of classes equal to the number of elements. Distance between each two classes is computed and the two classes with minimum distance are incorporated. This procedure continues until the number of classes reaches the pre-defined value, i.e. 16(for 4-bit quantification). For each of the 16 classes, we calculate the arithmetic average of elements in it respectively and use the average as codebook to quantify speech envelope. In this letter 1970 envelopes were used to get the final 16 models. Fig.1 shows the result of the 16 models.

An additional bit rate of $\frac{1000}{22.5} \times 0.004 \approx 0.18$ kbps will be introduced to quantify the shape of envelope. The multiplication factor is quantified for voiced frame instead of the RMS energy value, which is used for unvoiced frame in this scheme. So the quantification of multiplication factor will introduce no additional bit.

4. USING ENVELOPE PARAMETERS IN DECODER

In decoder, we decode the envelope parameters transmitted by the encoder and use the decoded envelope as the standard envelope. Speech must be synthesized once a pitch period instead of once a frame in order to use the envelope parameters. So LPC and voice/unvoice decision must be interpolated from once a frame to once a pitch block. We exploit white noise as excitation signal for unvoiced speech. For voiced speech, the excitation signal takes the following form[1]:

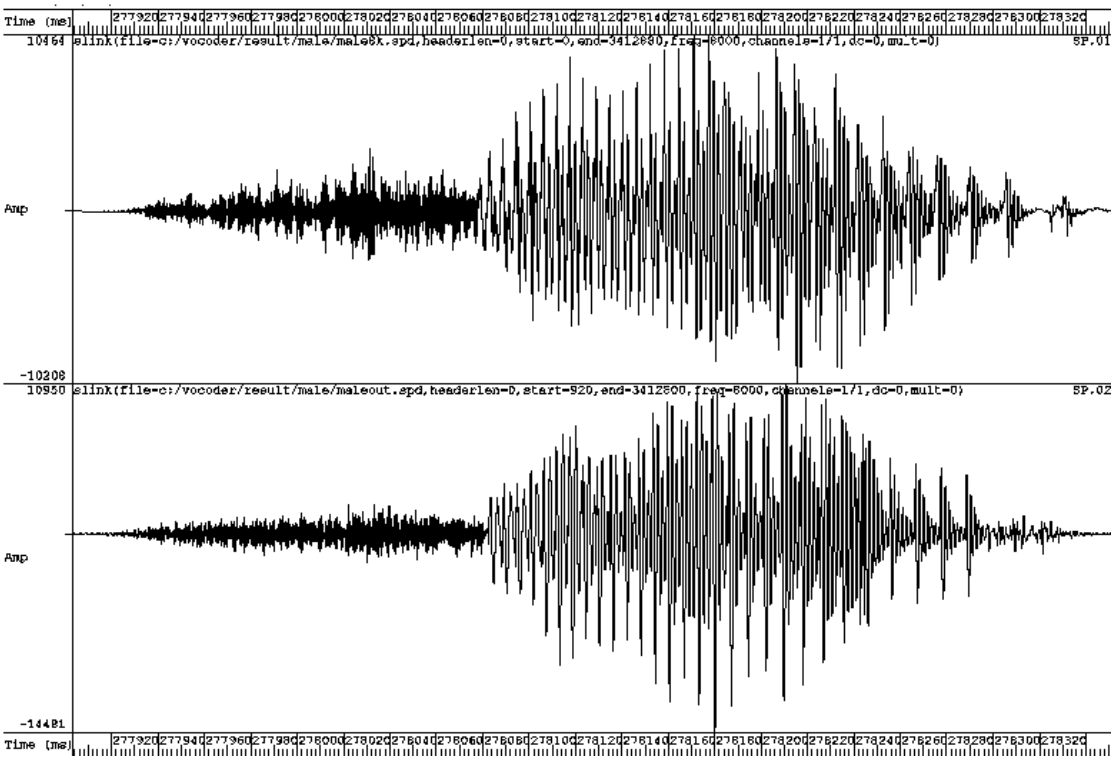

Fig.2 The top one is original speech; the bottom one is the synthesized speech of vocoders incorporating envelope information

$$e(n) = \{8,-16,26,-48,86,-162,294,-502,$$
$$718,-728,184,672,-610,-672,184,728,$$
$$718,502,294,162,86,48,26,16,8\}$$

The envelope parameters are used to adjust the amplitude of the output of linear prediction synthesizer. Let the output of linear prediction synthesizer be $x(i)$, the standard envelope is env(i),i=1,2,……, pitch period, the criterion of amplitude adjustment is as following:

(1) If the current pitch block is voiced, then make the greatest value of x(i) reach the standard envelope, i.e. $G = \dfrac{env(k)}{x(k)}$ where k satisfies $x(k) = \max\limits_{i=1,2,...,pitch} (abs(x(i)))$.

(2) If the current pitch block is unvoiced, then adjust the output of linear prediction synthesizer with the RMS, i.e.,

$$G = \frac{RMS}{\sqrt{\sum_{i=1}^{pitch} x(i)^2}}.$$

A segment of speech synthesized by this approach is showed in Fig.2. We held clarity tests[2] and MOS test for this vocoder. In clarity test, 74 Chinese words with a five seconds pause between adjacent words are used as the listening material and 5 people were asked to choose the words they heard from one of five candidates. In MOS test, 21 people were asked to evaluate the synthesized speech of this vocoder. Test result shows that the clarity of this vocoder is 91.8% and the MOS is 3.6 for male and 3.2 for female.

5. CONCLUSION

Through comparing the output of vocoders with the original speech we found only extracting energy parameter was not enough to synthesize high quality speech. A significant difference between the synthesized speech and the original one is their envelope. This letter incorporated the speech envelope

information in traditional compression algorithm. Simulation results revealed that with a cost of only adding 0.18kbps the clarity can reach 91.8% and the MOS can reach 3.6 and 3.2 for male and female speech respectively.

6. REFERENCE

[1] George S. Kang, Stephanie S. Everett, “Improvement of the Excitation Source in the Narrow-Band Linear Prediction Vocoder”, IEEE Trans. On ASSP, Vol. 33, April 1985, pp.377-386.

[2] Chen Yong Bin, Wang Ren Hua, “Speech Signal Processing”, University Press of Science and Technology of China, He Fei, China, 1990.