PERFORMANCE ANALYSIS OF A DECODER-BASED TIME SCALING ALGORITHM FOR VARIABLE JITTER BUFFERING OF SPEECH OVER PACKET NETWORKS

Philippe Gournay, Kyle D. Anderson

VoiceAge Corporation
750, Chemin Lucerne
Montréal (Québec) Canada H3R 2H6

Philippe.Gournay@USherbrooke.ca

ABSTRACT

This paper describes how a CELP speech decoder such as the VMR-WB decoder can be modified in order to deliver decoded speech frames of variable length, a feature that is required for adaptive jitter buffering. This method is shown to be successful for playout delay adaptation in terms of both subjective quality and reactivity. Moreover, it requires almost no additional complexity provided some clever limitations are imposed on time scaling.

1. INTRODUCTION

Speech transmission over packet networks is characterized by variations in the time that packets take to transit through the network. VoIP receivers generally rely on a "jitter buffer" to control the effects of jitter (which is the difference between the actual arrival time of the packets and a reference clock at the normal packet rate). This jitter buffer works by introducing an additional "playout" delay (which is defined with respect to the reference clock that was, for example, started at the reception of the first packet) in order to transform the uneven flow of arriving packets into a regular flow of packets, so the decoder can provide a sustained flow of speech to the listener. The efficiency of the jitter buffer is thus determined by two interdependent parameters: the additional delay it introduces, and the percentage of frames that will arrive after this delay and will therefore be considered as lost.

Several strategies [1] can be used to achieve the best compromise between these two parameters. The simplest one is the "fixed jitter buffering" strategy, in which a certain playout delay is applied at the beginning of the conversation and maintained afterwards. In the "adaptive jitter buffering" strategy, which is more efficient when network characteristics are varying in time, the playout delay is adapted at the beginning of each received talkspurt based on previous jitter statistics. Since the playout delay is changed only during silence or background noise, just removing (in order to decrease the playout delay) or inserting (in order to increase the playout delay) a certain number of samples does the trick. Even better results are obtained when the playout delay is also adapted during active speech [2]. As shown in Table 1, this latter strategy requires a means for time scaling decoded speech frames: playing out a longer frame *i* increases the playout delay P_{i+1}, while playing out a shorter frame decreases that delay.

1. Using past jitter values, estimate the "ideal" playout time $\hat{P}_{i+1}$ of frame number *i*+1.
2. Send frame number *i* to the decoder, requesting it to generate an output frame of length $\hat{T}_i = \hat{P}_{i+1} - P_i$.
3. The actual playout time of packet *i*+1 is $P_{i+1} = P_i + T_i$, where T_i is the actual length of frame *i*. Iterate from step 1.

Table 1: The algorithm for playout delay adaptation (the playout delay for packet *i* is the difference between P_i and the reference clock at the normal frame rate)

Playout delay adaptation is generally done outside the decoder (on the PCM signal) using a technique such as PSOLA [3] (Pitch Synchronous Overlap Add) or TDHC [4] (Time Domain Harmonic Scaling). It is clear however that doing it inside the decoder, in the excitation domain, has many advantages in terms of complexity:

- Working inside the decoder makes it possible to use the internal parameters of the codec. The pitch values and gains, and the voicing classification in the case of the VMR-WB codec [5], are particularly useful parameters for time scaling.
- Working inside the decoder also regulates the processor load (i.e., the number of arithmetic and logical operations needed to decode one frame divided by the duration of that frame). This is particularly true when the frame is shortened since in that case the processor load tends to increase as the frame length decreases even if the number of operations remains the same. Working in the excitation domain, for example, saves some complexity because the synthesis operation (which, in the case of VMR-WB, includes LP synthesis, post-filtering, up-sampling and high-frequency generation) is performed on the shortened frame instead of the normal frame.

It is also potentially advantageous in terms of quality, as the smoothing performed by the synthesis filters of the decoder is supposed to be beneficial to quality.

1-4244-0469-X/06/$20.00 ©2006 IEEE

This paper describes how a CELP speech decoder such as the VMR-WB decoder can be modified in order to deliver speech frames of variable length. The common principles and limitations are explained in section 2. Depending on the voice activity indication and on the voicing classification, which are internal parameters of the VMR-WB codec[1], inactive, voiced and unvoiced frames are processed as described in sections 3, 4 and 5, respectively. The performance of the method in terms of quality and reactivity (percentage of frames that can be modified, amount of scaling achievable in one frame, and time needed to achieve a 50-ms increase of the playout delay) is studied in section 6. Finally, conclusions are drawn in section 7.

2. PRINCIPLES AND LIMITATIONS

For active speech frames, time scaling is applied to both the "raw" and "post-processed" excitations, with the first one being used to update the adaptive codebook and the second for the synthesis. The reason for this is that the low-frequency pitch enhancer of VMR-WB uses both signals and requires them to be perfectly in phase. To keep the encoder and decoder synchronized, however, the adaptive codebook is updated before time scaling.

During the synthesis, we take care to apply the same filter to the same samples as for unmodified speech. This requires providing the modified length of each subframe to the synthesis function.

The VMR-WB decoder operates on 20-ms frames. Time scaling permits to generate shorter or longer frames. As will be explained in sections 3 to 5, some frames (onsets, plosives, etc.) are not modified for fear of degrading quality. But we also imposed the following limitations to scaling:

- Stretched frames are limited to 40 ms (twice the standard length) in order to keep the memory requirements of the modified decoder within acceptable limits.
- Preserving the periodic nature of voiced speech requires only adding or removing an integer number of pitch cycles to voiced frames. The requested modification is in fact rounded *up* to the nearest multiple of the pitch period (without exceeding the 0-to-40-ms range) in order to maximize the "reactivity" of the adaptation.
- For unvoiced frames, a lower limit is set to 10 ms in order to minimize the impact on quality.
- For CNG (Comfort Noise Generation) frames, the lower limit is set to 0.
- In all cases, the frame length is forced to be a multiple of 5 samples in the synthesis domain (16 kHz), which corresponds to a multiple of 4 samples in the excitation domain (12.8 kHz). This limitation is imposed by the 4/5 up-sampling filter that is used to go from the excitation domain to the synthesis domain. We verified that loosing 1 to 3 samples from time to time, though it creates a small "pitch jitter", does not overly degrade quality.

3. TIME SCALING OF INACTIVE FRAMES

The pseudo-random number generator used to build the excitation signal of CNG frames is simply run for the requested number of samples.

4. TIME SCALING OF VOICED FRAMES

Voiced frames that are classified as "onsets" by the decoder and voiced frames for which the maximum pitch gain over the four subframes is less than 0.50 are not modified. The remaining voiced frames are processed as follows.

4.1. Lengthening voiced frames

The procedure is illustrated in Fig. 1.The subframe with the highest pitch gain is selected. Within that subframe, a sliding window of 20 samples is used to find the minimum-energy point (i). Some processing is performed on the four pitch values of the frame in order to correct any obvious pitch error (specifically, to avoid using a submultiple of the real pitch value). The difference between the desired frame length and the standard frame length, rounded up to the nearest integer multiple of the pitch period of the subframe, gives the number of samples that will be added to the excitation. A space is created in the excitation signal right after the minimum-energy point to receive the extra pitch cycles. The long-term prediction function of the decoder is used to replicate the pitch cycle that immediately precedes the minimum-energy point, thus filling in the space.

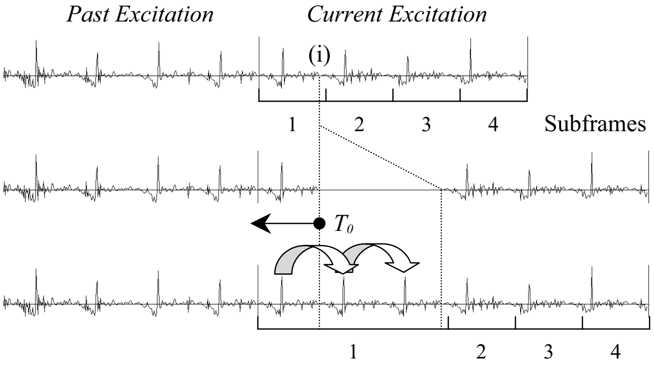

Fig. 1: Voiced frames are lengthened by repeating some pitch cycles (here, only the first subframe is modified)

4.2. Shortening voiced frames

Voiced frames are shortened by removing some pitch cycles from the excitation signal. Since it is not possible to dip into the past excitation, we have chosen to always remove pitch cycles from the last subframe backwards. The minimum-energy point in that subframe is found as in section 4.1. The pitch value is corrected in order to avoid any obvious pitch error. The difference between the requested and standard frame lengths, rounded up to the nearest multiple of the pitch period when possible and rounded down otherwise, gives the number of samples to be removed before the minimum-energy point.

[1] We also implemented that method in the AMR-WB decoder. This required either to perform the voicing classification at the decoder, or to send it along with the bitstream as additional information.

5. TIME SCALING OF UNVOICED FRAMES

Plosive frames and frames that seem “too voiced” to be handled as unvoiced are not modified. Plosive frames are those for which one subframe is more than 9 dB higher than the preceding subframe. “Too voiced” frames are those for which the average pitch gain is above 0.45. The remaining unvoiced frames are processed as follows.

5.1. Lengthening unvoiced frames

Unvoiced frames are lengthened by inserting the necessary number of zeroes between the original excitation samples. Those zeroes are distributed “as uniformly as possible” throughout the frame. A weighting factor equal to the square root of the ratio between the requested and standard frame lengths is further applied to the modified excitation in order to preserve the average energy per sample.

5.2. Shortening unvoiced frames

Unvoiced frames are shortened by removing the necessary number of samples from the excitation signal. These samples are removed from either the beginning or the end of the frame, depending on what the previous frame was. If it was an unvoiced frame, the samples are removed from the beginning of the frame. If the previous frame was not an unvoiced frame, then they are removed from the end of the frame. This protects against removing any possible weak voiced component from an unvoiced frame, such as the transition from an unvoiced sound to a voiced sound (voiced onset) or vice versa (voiced offset).

6. EVALUATION RESULTS

The following experiments were conducted on clean speech using mode 2 of the VMR-WB codec (Average Data Rate of 4.96 kbits/s). Audio demo files are available on request.

6.1. Subjective results

When time scaling is used to perform what is known as “fast playback,” the subjective quality remains quite good down to an average speed factor of 50% (speech played at twice its normal rate).

When time scaling is used to perform “slow playback,” the quality of CNG and unvoiced frames also remains very good up to a speed factor of 200% (speech played at half its normal rate). Lengthening voiced frames, however, occasionally produces some artifacts (a very few “clacks”). Voiced speech also tends to sound less natural (more “robot”-like) when substantially longer frames (35 ms or more) are always requested from the decoder.

Overall however, this method is very efficient at adding or removing a few milliseconds (e.g. 20 ms) to the playout delay from time to time. It is at least equivalent to PSOLA in terms of quality. The potential quality degradation is in any case nothing compared to losing the next frame or few frames because of an insufficient playout delay.

6.2. Objective results

The percentage of frames that can be modified (regardless of the impact on quality) is around 82%. Note that this does not depend on the requested amount of scaling. Frames that cannot be modified are distributed as shown in Table 2.

Total number of frames:	22803	
Number of active speech frames:	13668	(60% of 22803)
Number of unchanged frames:	4085	(18% of 22803)
Distribution of unchanged frames:		
1. Voiced frames		
Onsets:	814	(20% of 4085)
Not voiced enough:	935	(23% of 4085)
2. Unvoiced frames		
Plosive:	1850	(45% of 4085)
Too voiced:	486	(12% of 4085)

Table 2: Distribution of unmodified frames

As mentioned in section 4, while modified unvoiced frames always have the requested frame length rounded to the nearest multiple of 5, voiced frames can only be modified by an integer multiple of the pitch period. Fig. 2 shows the length distribution of modified voiced frames, when the requested frame length is twice the standard length.

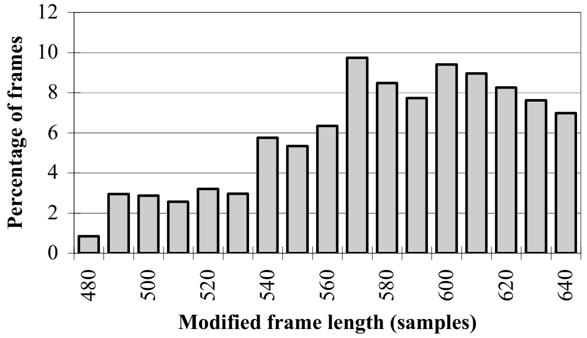

Fig. 2: Frame length distribution for modified voiced frames (Desired frame length was 640 samples = 40 ms)

In order to evaluate the reactivity that the proposed time scaling method would bring to an adaptive jitter buffer, we measured the time needed to increase the playout delay by 50 ms. The adaptation time is the sum of the durations of the frames (be they modified or not) that must be played out before the desired playout delay is achieved. The duration of the last modified frame is not taken into account in the adaptation time, as the desired playout delay can be considered as already achieved when playing out the first sample of that frame. For example, if the request to increase the playout delay arrives just before decoding frame *n*, and if frames *n*, *n*+1 and *n*+2 are stretched to 40 ms, 40 ms and 30 ms, respectively, then the adaptation time is equal to 80 ms (sum of the modified durations of frames *n* and *n*+1).

As shown in Fig. 3, the adaptation time is always between 70 ms and 300 ms with very few cases above 200 ms (121 cases out of 8000). This is well below the average duration of a talkspurt, which is rather on the order of one second. The shortest adaptation time is 70 ms (14 cases) and is achieved when the adaptation is done in 3 frames, the first two frames being stretched to a total of 70 ms and the third frame being 20 ms longer than usual. When the adaptation is performed during purely unvoiced segments (containing no onsets or plosives), the adaptation time is exactly 80 ms (which corresponds to the example given above). The adaptation time is never between 80 ms and 90 ms, because it cannot be more than 80 ms when done in 3 frames nor less than 90 ms when done in 4 frames (three 30-ms frames followed by one 40-ms frame).

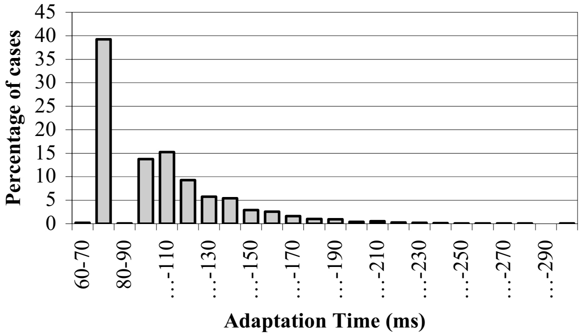

Fig. 3: Time required for a 50-ms increase of the playout delay (experiment done for 8000 different active speech frames)

6.3. Complexity measures

We instrumented the VMR-WB decoder in order to count the number of operations performed for each frame. We then ran it on the same test file as above, asking it to scale all the frames by a certain amount. The complexity is given in weighted MOPS (Million Operation Per Second) where the operations are weighted according to their respective complexity and the actual frame duration is used (Table 3).

When no time scaling is requested (standard 20-ms frames), the maximum decoding complexity is 6.68 WMOPS. When the decoder is requested to double the frame length, the highest complexity is just slightly more (6.69 WMOPS). But the corresponding frame length is in fact 20 ms, which indicates that the frame was in fact not modified. Lengthening speech frames therefore never increases the decoding complexity. When the decoder is asked to halve the frame length, the highest complexity is 33.90 WMOPS for a frame length of 1.875 ms. When an additional constraint is set so that the length of modified voiced frames never goes below 10 ms, the highest complexity is 9.57 WMOPS. This renders the adaptive jitter buffer less "reactive" for decreasing the playout delay but is much more reasonable from a complexity point of view. As a comparison, the complexity of a standard decoder followed by an external time compression with a factor of 50% would be at least twice that of the decoder (6.68 WMOPS*20 ms/10 ms = 13.36 WMOPS) plus that of the time scaling operation.

Requested frame length	Standard (20 ms)	Longer (40 ms)	Shorter (10 ms)	Shorter[1] (10 ms)
Maximum complexity	6.68 WMOPS	6.69 WMOPS	33.90 WMOPS	9.57 WMOPS
Corresp. length	20 ms	20 ms	1.875 ms	10 ms

[1] Modified voiced frame *not* allowed to be less than 10 ms.

Table 3: Maximum complexity and corresponding frame length

We noted that generating shorter frames is more complex during active speech. In that case, the whole excitation signal must be decoded before time scaling. The resulting complexity (number of operations divided by the frame duration) goes up even though some processing is saved during the synthesis. In contrast, time scaling does not really affect the complexity of CNG frames (the pseudo-random number generator is run only for the requested number of samples, and the synthesis is done only for the actual frame length). Therefore, if complexity is an issue, a good solution is: 1. to increase the playout delay by lengthening speech frames as soon as it is necessary to avoid late frames, and 2. to decrease the playout delay by shortening speech frames only during silences. In that case, playout delay adaptation requires almost no additional complexity and does not unduly degrade quality.

7. CONCLUSION

This paper described how a CELP speech decoder such as the VMR-WB decoder can be modified in order to deliver speech frames of variable length, a feature that is required for adaptive jitter buffering. This method is shown to be successful for playout delay adaptation in terms of both subjective quality and reactivity. Moreover, it requires almost no additional complexity provided some clever limitations are imposed on scaling.

8. REFERENCES

[1] A. Ramjee, J. Kurose, D. Towsley, and H. Schulzrinne, "Adaptive playout mechanisms for packetized audio applications in wide-area networks." In *Proceedings of the Conference on Computer Communications* (IEEE Infocom), Toronto, Canada, June 1994.

[2] Y.J. Liang, N. Färber, and B. Girod, "Adaptive playout scheduling using time-scale modification in packet voice communications," in *Proc. IEEE ICASSP'2001*, pp. 1445-1448, May, 2001.

[3] H. Valbret, E. Moulines and J.-P. Tubach, "Voice Transformation Using PSOLA Technique," *Speech Communication* 11 (1992), pp. 175-187.

[4] D. Malah, "Time-domain algorithms for harmonic bandwidth reduction and time scaling of speech signals," *IEEE Trans. Acoust. Speech Signal Process.*, Vol. ASSP-27, No. 2, pp. 121-133, Apr. 1979.

[5] "Source-Controlled Variable-Rate Multimode Wideband Speech Codec (VMR-WB) - Service Option 62 for Spread Spectrum Systems," 3GPP2 TSG-C specification C.S0052-0, July 2004.