Efficient Technique for Quantization of Pitch Contours

Jani Nurminen, Sakari Himanen, and Anssi Rämö

Multimedia Technologies laboratory
Nokia Research Center, Tampere, Finland
{jani.k.nurminen, sakari.himanen, anssi.ramo}@nokia.com

Abstract

This paper introduces an efficient technique for pitch contour quantization designed mainly for applications that require storage of speech or prosodic information at a high compression ratio. Instead of quantizing the estimated pitch values directly, the proposed technique forms and quantizes a simplified model of the pitch contour. The simplified contour is constructed in such a manner that the amount of information needed for describing it is minimized. At the same time, the deviation from the original contour is maintained below a predetermined limit. In addition to the high compression ratio, the contour representation offers benefits in pitch-synchronous decoding. The proposed technique is implemented and evaluated in a practical storage speech coder. According to the evaluation, the performance of the quantization technique is very promising as it achieves perceptually satisfactory quality at an average bit rate of about 100 bits per second.

1. Introduction

Quantization of pitch contours is a task that is required in almost all practical speech coders and also in many text-to-speech (TTS) systems. During voiced speech, the pitch parameter corresponds to the fundamental frequency and can be perceived as the pitch of speech. During purely unvoiced speech, there is no fundamental frequency in a physical sense and the concept of pitch is vague. In most speech coders, however, "pitch information" is also needed during unvoiced speech. For example, in coders based on the well-known code excited linear prediction (CELP) approach, the long-term prediction lag (roughly corresponding to pitch) is also quantized during unvoiced portions of speech.

Typically, the coding of the pitch information has been handled using straightforward quantization approaches such as memoryless scalar quantization. More advanced techniques have also been considered, mainly to be able to exploit the redundancies between successive pitch values. For example, predictive quantization approach was used in [1], and in [2] the pitch values were coded using a shaped lattice quantizer. In [3], the exploitation of redundancies was taken one step further: the pitch information was orthogonally transformed and vector quantized by taking into account the tonal nature of Mandarin speech.

In this paper, we introduce a simple but efficient technique for pitch contour quantization. In contrast to the conventional methods, the key idea in the proposed technique designed mainly for storage applications is to construct a simplified model of the pitch contour that is quantized instead of the original contour. The simplified contour is formed in such a manner that the amount of information to be coded is minimized while the deviation from the original contour is kept below a predetermined limit. This main idea can be implemented in several ways, e.g. using splines, but in this paper we concentrate on the case of a piece-wise linear pitch contour model. The main advantages of this model are that it is very simple and that the related optimizations can be performed using computationally efficient techniques. Furthermore, with this model it is easy to handle the pitch information in pitch-synchronous decoding.

The primary target application for the proposed quantization technique is speech storage. In this context, the proposed approach is very efficient and offers many benefits. The proposed technique can also be utilized in other applications, e.g. in the compression of F0 contours in TTS systems. It should be noted, however, that the technique should not be used in applications that are very sensitive to small changes in pitch (e.g. in CELP based speech coders).

The rest of this paper is organized as follows. First, Section 2 formulates the construction of the simplified contour as an optimization problem. Then, an efficient solution for this problem is given in Section 3. In Section 4, we discuss the benefits that the proposed quantization approach offers in pitch-synchronous parametric decoding. In Section 5, a practical implementation of the proposed quantization approach is described and evaluated in experiments. Finally, conclusions are given in Section 6.

2. Problem formulation

The search for the optimal simplified model of the pitch contour can be formulated as a mathematical optimization problem. Let $f(t)$ denote the function that describes the original pitch contour between the time instants 0 and $t_{\max}$, and $g(t)$ denote the simplified pitch contour. Furthermore, let $d(f(t), g(t))$ measure the deviation between the two contours at the time instant t. Now, after estimating $f(t)$, the problem to be solved is to find $g(t)$ that satisfies two optimality conditions:

(i) The number of bits needed for describing the contour $g(t)$ is minimized.

(ii) $d(f(t), g(t)) \le h(f(t))$ for all $0 \le t \le t_{\max}$, where $h(\cdot)$ defines the maximum allowable deviation from the original pitch contour.

From the set of contours that satisfy both conditions, the contour function that minimizes the maximum deviation,

$$D_{\max} = \max(d(f(t), g(t))),\ t \in [0, t_{\max}], \quad (1)$$

is selected as the final simplified contour. The pitch values can be represented either in time or frequency domain.

In general, the above optimization problem is unsolvable. However, the problem can be solved quite easily if its generality is reduced by fixing the pitch contour model. As discussed in Section 1, this paper concentrates on the piece-wise linear model. With this model, the function $g(t)$ can be described using the points in which the derivative of $g(t)$ changes. Let q_n and t_n denote the coordinates of the nth such

point ($1 \le n \le N$, where N is the number of these points in the piece-wise linear model). The simplified contour can be defined in N–1 linear pieces as

$$g(t) = q_{n-1} + \frac{t - t_{n-1}}{t_n - t_{n-1}}(q_n - q_{n-1}) \text{ for } t_{n-1} \le t \le t_n, \qquad (2)$$

where $2 \le n \le N$. To make the definition complete, we require that $t_{n-1} < t_n$, and that $t_1 = 0$ and $t_N = t_{\max}$. In addition, we require that all values of q_n are within the finite range from $q_{\min}$ to $q_{\max}$. With this model, the optimization problem reduces to the search for the set of points (t_n, q_n) describing the contour $g(t)$ that satisfies the conditions (i) and (ii) and minimizes the maximum deviation in Equation (1). Now, by making the reasonable assumption that the point coordinates can only be represented with a limited resolution, the problem becomes solvable, at least in theory, since the points are located in a grid with a finite number of possible point locations and it is possible to compute the bit usage. This assumption does not reduce the generality of the formulation since the finite accuracy follows directly from the optimality condition (i).

3. Solution for the problem

First, let us assume that the pitch values q_n are coded into bits using a scalar quantizer with a codebook $C = \{c_1, c_2, ..., c_M\}$, and that the time indices t_n are integer multiples of some time unit T. Furthermore, we assume that both C and T are selected in such a manner that a solution exists, and make the reasonable assumption that the number of bits needed for describing the contour can be minimized by minimizing N (the number of points needed for defining the simplified contour). Finally, we require that an application-specific maximum limit is set for $t_{\max}$ to achieve a reasonable buffer size for processing.

In theory, a globally optimal solution for the optimization problem presented in Section 2 can always be found using a brute force algorithm that goes through all possible pitch contour candidates and selects the one that provides the best match to the original contour. However, the complexity of this approach grows extremely fast with increasing problem size. More precisely, we can state that in the worst case the number of different contour candidates to be checked is

$$w = \sum_{j=0}^{m} \frac{b^{j+2} m!}{j!(m-j)!}, \qquad (3)$$

where $m = (t_{\max} / T) - 1$ and b denotes the maximum number of codebook entries that can satisfy the condition

$$d(f(t_n), q_n) \le h(f(t_n)). \qquad (4)$$

Consequently, techniques with a much lower computational complexity must be considered in practical applications.

In our implementation and experiments, we have successfully used a simplified approach in which the main idea is to go through the optimization process one linear piece at a time. For each linear piece, the maximum length line that can keep the deviation from the true contour low enough is searched without using knowledge of the contour outside the boundaries of the linear piece. Within this optimization technique, there are two cases that have to be considered separately: the first linear piece and the other linear pieces.

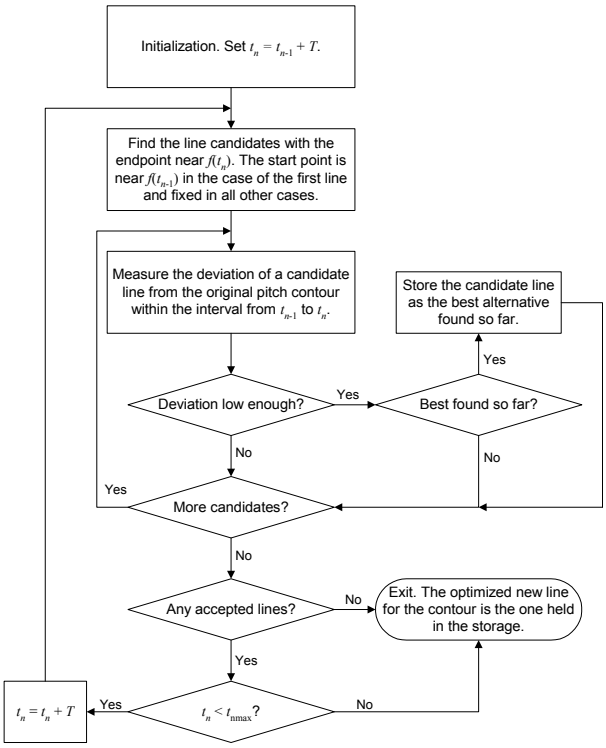

Figure 1. Optimization process for one linear piece.

The case of the first linear piece occurs at the beginning when the encoding process is started. In addition, if no pitch values are transmitted for inactive or unvoiced speech, the first linear pieces after these pauses in the pitch transmission fall into this category. In both situations concerning the first linear piece, both ends of the line are optimized. All other cases fall into the second category in which the starting point for the line has already been fixed in the optimization of the previous linear piece and thus only the location of the end point is optimized.

The optimization of one linear piece is illustrated in Figure 1. The process starts by finding the potential line candidates for the situation in which a time distance T separates the ends of the current linear piece, i.e. $t_n = t_{n-1} + T$. The candidates for the end point are the quantized pitch values that are close enough to the original pitch value at t_n such that the criterion for the desired accuracy given in Equation (4) is satisfied. Similarly, in the case of the first linear piece, the candidates for the start point are the quantized pitch values close enough to $f(t_{n-1})$. For other linear pieces, the start point has already been fixed and thus there is only one start point candidate. After the candidates have been found, all the possible start point and end point combinations are tried out: the accuracy of the linear representation is measured in the time interval between t_{n-1} and t_n, and the candidate line can be accepted as a part of the piece-wise linear contour if the accuracy criterion is satisfied. Furthermore, if the deviation from the original contour is smaller than with the other lines accepted during this iteration step, the line is selected as the best line found so far. If at least one of the candidates was accepted, the iteration is continued by repeating the process after increasing t_n by a step of size T. If none of the lines was accepted, the optimization process is terminated and the best line found

during the previous iteration is selected as the linear representation of the current piece.

The iterative process described above can be finished prematurely for two reasons. First, the process is terminated if t_n cannot be increased because the original pitch contour ends before $t_n + T$. This may happen if the whole lookahead buffer has been used, if the speech signal to be encoded has ended, or if the pitch transmission has been paused during inactive or unvoiced speech. Second, it is possible to limit the maximum length of a single linear part in order to limit the size of the lookahead and/or to code the time indices more efficiently. For both cases, these issues can be taken into account by setting a limit t_{nmax} based on the duration of the available pitch contour and on the maximum time-distance between the ends of the line. This approach has also been employed in Figure 1.

The complexity of the optimization procedure described above is quite small in practical situations and grows only linearly with increasing problem size. However, it is still possible to reduce the average computational load, if necessary, for example by using binary search for finding the optimal line length. An additional benefit of the proposed solution, for example when compared to the storage coding approaches presented in [4] and [5], is that only a relatively small lookahead is required as the whole pitch contour is not processed at once.

4. Benefits in pitch-synchronous decoding

In addition to enabling efficient compression, the proposed contour-based approach offers benefits in pitch-synchronous parametric decoding. Firstly, due to the contour representation, it is straightforward to compute a parameter value for any time instant. Secondly, it is very convenient to incorporate the possibility for high-quality playback speed modifications into this framework. This can be achieved by requiring at the decoder that the center of the pitch cycle, t_{center}, must satisfy the condition

$$2 \cdot (t_{\mathrm{center}} - t_{\mathrm{boundary}}) = p(t_{\mathrm{center}}) \cdot v \,, \tag{5}$$

where v is the relative playback speed, $p(t)$ denotes the time-domain pitch parameter at the time instant t, and t_{boundary} denotes the "virtual boundary" of the previous pitch cycle (at the beginning of the synthesis process t_{boundary} is 0). After a center that satisfies the above condition is found, t_{boundary} is updated using

$$t_{\mathrm{boundary}} = t_{\mathrm{boundary}} + 2 \cdot (t_{\mathrm{center}} - t_{\mathrm{boundary}}) \,, \tag{6}$$

before finding the next pitch cycle center.

It is easy to see that if the pitch contour is continuous, there always exists a center satisfying the condition in Equation (5). If the contour is only piece-wise continuous, the discontinuity points have to be considered as special cases. With the proposed piece-wise linear quantization approach, the search for the next center should be done one linear piece at a time. Furthermore, t_{center} can be found by solving

$$t = \frac{\left(\frac{2t_{\mathrm{boundary}}}{v} + p(t_{n-1})\right)(t_n - t_{n-1}) - t_{n-1}\left(p(t_n) - p(t_{n-1})\right)}{\frac{2(t_n - t_{n-1})}{v} - \left(p(t_n) - p(t_{n-1})\right)} \tag{7}$$

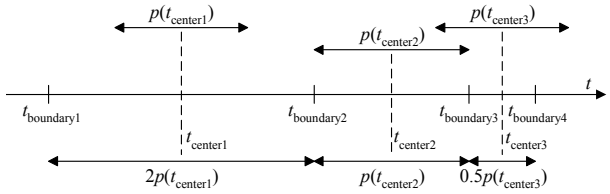

Figure 2. Three examples illustrating the use of different playback speeds.

for each interval and by checking whether the obtained t_{center} candidate t is in the current interval between t_{n-1} and t_n. If it is, then $t_{\mathrm{center}} = t$. Otherwise, n is increased by 1 and the equation is solved for the next interval etc.

The process of finding the center of the next pitch cycle using different playback speeds is demonstrated in Figure 2. The figure illustrates the center locations found in three different situations:

1. Playback using double speed ($v = 2$)
2. Normal playback speed ($v = 1$)
3. Half speed ($v = 0.5$)

In the figure, these situations are denoted with the corresponding subscript. For example, $t_{\mathrm{boundary1}}$ denotes the value of t_{boundary} when searching for the center t_{center1}.

5. Practical implementation and results

The pitch contour quantization technique introduced in this paper was included in a practical speech coder designed for storage applications. The coder operates at very low bit rates (about 1 kbps) and processes the 8 kHz input speech in segments of variable duration (between 10 and 640 ms). This section introduces the relevant details of the practical implementation and describes the results achieved in our experiments.

5.1. Implementation

In the practical implementation of the proposed quantization technique, the approach described in Section 3 was used with time domain pitch values and only the pitch contour located in the current segment was considered in the optimization. During unvoiced or inactive segments, no pitch information was coded. The variable T was set to 10 ms that was equal to the pitch estimation interval. Furthermore, the continuous pitch contour was approximated using the discrete contour formed by the estimated pitch values p_k. Consequently, the optimality condition (ii) was changed into

$$d(p_k, g(kT)) \le h(p_k) \text{ for all } 0 \le k \le t_{\max} / T. \tag{8}$$

In addition, d was defined as

$$d(x,y) = |x - y| \,, \tag{9}$$

and the maximum distortion in Equation (1) was only measured at the corresponding discrete time instants similarly as in Equation (8).

The function h that defines the maximum allowable coding error for a given pitch value was determined as

$$h(p_k) = \max(2,\, 480\, p_k / 8000) \,. \tag{10}$$

The same function was also employed in the generation of the codebook C used in the scalar quantization of the pitch values

q_n. The entries of the 32-level (5-bit) codebook C were computed using $c_j = c_{j-1} + h(c_{j-1})$ with $c_1 = 19$. This codebook covers the pitch period range used in the coder and is quite consistent with the experimental findings reported in [1]. Moreover, this codebook and the function h approximately follow the theory of critical bands [6] in the sense that the frequency resolution of the human ear is assumed to decrease with increasing frequency. To further enhance the perceptual performance, the quantization was done in logarithmic domain.

The time indices were coded for one segment at a time using differential quantization, with the exception that the time-distance was not coded at all for the first point of each segment since t_1 was always 0. In the differential coding scheme, a given time index was coded using the time-distance between it and the previous time index in steps of size T. More precisely, the value of a given t_n was coded by converting $((t_n - t_{n-1}) / T) - 1$ into the binary representation containing $\lceil \log_2((t_{\mathrm{nmax}} - t_{n-1}) / T) \rceil$ bits. One additional trick was used in our implementation to increase coding efficiency: If the number of time indices to be coded was more than half of the number of pitch estimation instants in the segment, the "empty" time indices were coded instead of the time indices t_n (and one bit was used to indicate which coding scheme was used). However, it should be noted that the efficiency of this scheme is enabled by the segmental processing used in the storage coder implementation. In a general case with continuous frame-based processing, a better way would be to use some lossless coding technique, such as Huffman coding, directly on the time distance values.

5.2. Test results

The implementation described in the first part of this section was tested with large amounts of speech data. Based on these experiments, it can be concluded that the proposed method is capable of coding the pitch contour (containing 100 pitch values per second) with the average bit rate of approximately 100 bps in such a manner that the deviation from the original contour remains below the maximum allowable deviation defined in Equation (10). Despite the very low bit rate, the coded pitch contour is quite close to the original contour. With an exemplary collection of speech data, the average and the maximum absolute coding errors were about 1.16 and 5.12 samples, respectively, at 99 bps. For comparison, the very efficient technique presented in [1] achieved the average and maximum errors of 0.90 and 12.30 samples at 235 bps with the same speech data.

When judged by expert listeners, the contour coded using the proposed technique could be easily distinguished from the original contour but the coding error was not particularly annoying. When compared to the technique proposed in [1], the coded contour sounded slightly more pleasant despite the small increase in the average coding error. The most probable reasons for this are the facts that the maximum error was smaller and the pitch contour was smoother with the proposed technique. The speech quality with different playback speeds between half speed and double speed was found to be very high when compared to the compressed speech with the original playback speed. In general, the speed modifications did not cause audible distortions.

The proposed pitch quantization approach has not been tested explicitly with naive listeners. However, a formal listening test indicated that the storage coder containing the proposed pitch quantization technique outperformed a 1.2 kbps state-of-the-art reference coder by a wide margin despite the average bit rate reduction of more than 200 bits per second [7]. For the pitch parameter, the achieved bit rate reduction was close to 70 bits per second.

6. Conclusions

An efficient technique for pitch contour quantization has been introduced. The main idea in the technique is that a simplified model of the pitch contour is formed and quantized instead of the original contour. To make the compression as efficient as possible, the simplified contour is constructed such that the amount of information to be coded is minimized while still keeping the deviation from the original contour low enough. In this paper, the simplified contour was constructed using a piece-wise linear model. The proposed pitch quantization approach was also implemented and evaluated in a practical speech coder designed for storage applications. Based on the evaluation, the proposed approach can be considered very promising, as the practical implementation was capable of achieving quite good perceptual quality at the average rate of about 100 bits per second. In addition to the high compression rate, the proposed technique offers advantages, such as convenient playback speed control, in pitch-synchronous decoding.

7. References

[1] Eriksson, T. and Kang, H.-G., "Pitch quantization in low bit-rate speech coding", *Proc. of 1999 IEEE International Conference on Acoustics, Speech, and Signal Processing*, Phoenix, Arizona, Mar 15-19, 1999, pp. 489-492.

[2] Heikkinen, A., Ruoppila, V.T., and Pietilä, S., "A shaped lattice quantizer for successive pitch periods", *Proc. of Eurospeech 2001 - 7th European Conference on Speech Communication and Technology*, Aalborg, Denmark, Sep. 3-7, 2001, pp. 1965-1968.

[3] Chen, S.-H. and Wang, Y.-R., "Vector quantization of pitch information in Mandarin speech", *IEEE Transactions on Communications*, Vol. 38, No. 9, 1990, pp. 1317-1320.

[4] Roucos, S., Schwartz, R., and Makhoul, J., "Segment quantization for very-low-rate speech coding", *Proc. of 1982 IEEE International Conference on Acoustics, Speech, and Signal Processing*, Paris, France, May 3-5, 1982, pp. 1565-1568.

[5] Lee, K.S. and Cox, R.V., "A very low bit rate speech coder based on a recognition/synthesis paradigm", *IEEE Transactions on Speech and Audio Processing*, Vol. 9, No. 5, 2001, pp. 482-491.

[6] Zwicker, E. and Fastl, H., *Psychoacoustics*, Springer, Berlin, 1990.

[7] Rämö, A., Nurminen, J., Himanen, S., and Heikkinen, A., "Segmental speech coding model for storage applications", *Proc. of Interspeech 2004 - 8th International Conference on Spoken Language Processing*, Jeju Island, Korea, Oct. 4-8, 2004, pp. 2677-2680.