REAL-TIME PITCH-SHIFTING OF MUSICAL SIGNALS BY A TIME-VARYING FACTOR USING NORMALIZED FILTERED CORRELATION TIME-SCALE MODIFICATION (NFC-TSM)

Azadeh Haghparast, Henri Penttinen, and Vesa Välimäki

Lab. of Acoustics and Audio Signal Processing
Helsinki University of Technology, Espoo, Finland
azadeh.haghparast@tkk.fi

ABSTRACT

This paper presents a high-quality real-time pitch-shifting algorithm with a time-varying factor for monophonic audio and musical signals. The pitch-shifting algorithm is based on the resampling and time-scale modification method. A new time-scale modification method has been developed which is called the Normalized Filtered Correlation Time-Scale Modification (NFC-TSM) method. It uses a ring buffer for time-scaling. The best splicing point is searched in the normalized low-pass filtered signal using the Average Magnitude Difference Function (AMDF). The new method results in low-latency and high-quality pitch-shifting of musical signals.

1. INTRODUCTION

High-quality techniques for pitch-shifting of audio and musical signals have received a lot of attention recently. In multi-track audio recording and mixing, pitch-shifting is used to match the pitches of two recorded digital audio clips [1]. Real-time pitch-shifting algorithms can be used for performing deejays [2]. In music industry, pitch-shifting is used in sampling synthesizers, sound effects for Karaoke systems [3, 4], and other musical effects.

In general, pitch-shifting algorithms can be divided into two categories; time-domain and frequency-domain techniques [5]. Time-domain techniques are simple and fast, and work fine for periodic and quasi-periodic signals. However, their quality is not good for signals which contain a lot of non-harmonic components. On the other hand, frequency-domain algorithms are more suitable for complex signals, but the price of the high-quality is the computational complexity. Additionally, frequency-domain pitch-shifting algorithms call for large delays and, thus, are not appropriate for real-time applications. Frequency-domain algorithms are usually based on the phase-vocoder [6, 7]. In the phase-vocoder technique, first the signal is converted to its frequency-domain representation using a short-time Fourier transform (STFT). After modification of the frequency-domain parameters according to the pitch-shifting factor, the signal is converted back to its time-domain waveform.

The standard time-domain pitch-shifting algorithms, commonly used in commercial applications, are based on resampling and time-scale modification [3, 8, 5]. In the standard time-domain pitch-shifting technique, for pitch-shifting the signal by a factor of α, the input signal is first resampled by a resampling factor equal to $1/\alpha$. Since resampling changes the length of the signal, a time-scale modification method is used to preserve the time duration of the original signal. The time duration of the resampled signal should be scaled by a factor equal to α. Figure 1 shows the block diagram of a time-domain pitch-shifting technique using resampling and time-scale modification. In this figure, $f_{\mathrm{s,org}}$ and $f_{\mathrm{s,replay}}$ are the sampling frequency of the original audio and that of the pitch-shifted signal, respectively.

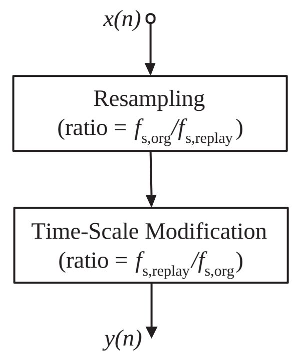

Figure 1: *Block diagram of the pitch-shifting method based on the resampling and time-scale modification.*

In general, there are two methods to implement the time-scale modification in the time domain. One of them is the ring buffer technique, which has serious quality problems [9, 10]. The other one is the overlap and add technique [5] which has many variations, developed to improve the quality of the output signal and reduce the computational complexity [1, 4, 11]. In the overlap and add method, the input signal is divided into overlapping segments which are shifted with respect to each other according to the time-scaling factor. Finally, they are added to each other to form the output signal [5]. Since overlapping and adding segments at any point breaks the continuity of the pitch and changes the spectral characteristics of the signal, the two overlapping segments have to be synchronized and cross-faded at the point of highest similarity. Several developments to find the maximum similarity points resulted in the different variations of the overlap and add method [1, 4, 11]. Some of these techniques are the synchronized overlap and add method (SOLA) [12, 13], the pitch synchronous overlap and add method (PSOLA) [5, 12, 14, 15], the waveform similarity overlap and add method (WSOLA) [16], and the global and local search time-scale modification (GLS-TSM) [17]. Recently, a comparison of the time-domain time-scale modification techniques has been presented in [18].

In this paper, a pitch-shifting method by a time-varying factor for monophonic musical tones is presented. The pitch-shifting algorithm is a standard time-domain pitch-shifting algorithm. A new time-scale modification algorithm has been designed which is called the Normalized Filtered Correlation Time-Scale Modification (NFC-TSM). The proposed time-scale modification technique enables the real-time pitch-shifting of the input signal. Moreover, the pitch of the input signal can be scaled continuously. The pitch-shifting factor can have large amounts with no serious defects in the quality of the pitch-shifted signal. The number of clicks in the pitch-shifted signal have been reduced comparing to the previous methods. A patent application has been submitted regarding this method.

In the following, first the general description of the pitch-shifting algorithm is presented. Then, the resampling process by a time-varying factor is explained. The NFC-TSM method is described next. Setting the parameters of the algorithm is discussed. Finally, the results are presented.

2. GENERAL DESCRIPTION OF THE ALGORITHM

The proposed pitch-shifting algorithm is based on the resampling and time-scale modification method. In this technique, the audio or musical signal is assumed to be periodic or semi-periodic. Moreover, there exists one pitch-shifting factor per input sample. Pitch-shifting is carried out by resampling the signal according to the pitch-shifting factor. Meanwhile, time-scale modification is performed whenever needed to preserve the duration of the original signal. In this work, to modify the time-scale of the signal, the novel NFC-TSM algorithm is used. Note that in the presented pitch-shifting algorithm, the resampling and time-scale modification operations are incorporated and can not be separated as such.

The pitch-shifting algorithm is performed on the audio signal that is stored in a ring buffer. A ring buffer, which is also called a circular buffer, is a portion of memory of fixed size into which new data is overwritten at its beginning when it is full. Two pointers to the ring buffer are defined: the input pointer and the output pointer. The input pointer is used to write into the ring buffer. That is, it is incremented by one when one input sample is received and written in the ring buffer. The output pointer is defined for resampling the audio signal in the ring buffer.

The rates at which the input and output pointers move in the ring buffer are different due to the resampling process. The rate of the input pointer is fixed, and equal to the rate of the sampling rate of the input signal, e.g. 44100 samples/sec. The rate at which the output pointer moves depends on the pitch-shifting factor. Different speeds of the input and output pointers result in their collision. After collision, either of the pointers may pass the other one. This causes discontinuity in the output signal which, in turn, results in audible artefact in the pitch-shifted signal.

In order to avoid the collision between the two pointers, the output pointer should jump backward and forward in the ring buffer to remain behind the input pointer. The best location for the output pointer to jump to, also referred to as the best splicing point, is searched using the NFC-TSM technique. In this method, the best splicing point is searched in the normalized low-pass filtered version of the ring buffer using Average Magnitude Difference Function (AMDF) as the correlation function. After having found the best splicing point, the two segments are joined to each other using a linear cross-fading function.

Next, the resampling by a time-varying factor and the NFC-TSM algorithm are explained in detail.

2.1. Resampling by a Time-Varying Factor

Resampling is the process of interpolating a signal at non-integer multiples of the sampling period. Theoretically, it can be stated as follows: first the original signal is reconstructed from a set of samples. In the next step, it is resampled at the desired locations. In practice, these two stages can be combined so that we need to find the values of the signal at the desired locations. The process of finding the signal values at arbitrary time instants from a set of samples is called interpolation. There are a variety of interpolators [19], such as truncated sinc [3, 8], linear interpolator [20], Lagrange interpolator [21], and spline interpolator [22]. In this algorithm, truncated sinc has been used to resample a signal, since it performs well for signals with rich high-frequency content.

In order to resample the digital audio stored in the ring buffer, for every input sample, the resampling factor is determined. The resampling factor is equal to the reciprocal of the pitch-shifting factor. This way the time instant at which the value of the signal is to be interpolated is found. Then, the truncated-sinc function is lined up with its peak at this time instant. The signal samples on both sides of the interpolation point are multiplied by the corresponding sinc function values and summed up to produce the pitch-shifted sample value.

The time instant at which the sinc function is lined up is pointed to by the output pointer. The speed of the output pointer depends on the resampling factor and, in turn, pitch-shifting factor. Figure 2 shows how the resampling process influences the speed of the output pointer. In Figure 2 (a), the signal is pitch-shifted down and the output pointer moves slower than the input pointer. In contrast, Figure 2 (b) shows a case in which the signal is pitch-shifted up and the speed of the output pointer is faster than the input pointer.

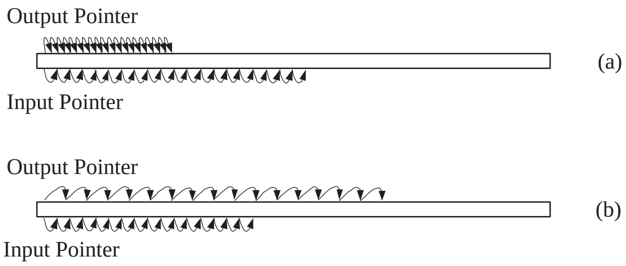

Figure 2: *Effect of resampling process on the speed of the output pointer, (a) signal is pitch-shifted down, the output pointer moves slower than the input pointer, (b) signal is pitch-shifted up, the output pointer moves faster than the input pointer.*

An important issue regarding interpolation is that the distance between the input pointer and the output pointer cannot be less than half the length of the sinc interpolator. Otherwise, the discontinuity in the sample set will result in interpolation errors.

2.2. Normalized Filtered Correlation Time-Scale Modification (NFC-TSM)

The main idea of the time-scale modification method was taken from the ring buffer method presented by Francis Lee in 1972 [10], which is based on discarding and repeating some segments of the audio signal to compress and expand the length of the signal, respectively. The conventional ring buffer technique results in audible artifacts, since the periodicity of the signal is broken and also amplitude discontinuities occur at the splicing points.

As discussed, the resampling process makes the output pointer move at a different speed from the input pointer. Different speeds of the pointers moving around a fixed-length buffer cause them to collide at some locations in the ring buffer occasionally. Collision of the input and output pointers in the ring buffer results in discontinuity in the time-scaled version of the resampled signal, which is heard as a click. To avoid this problem, in the proposed algorithm, the output pointer is handled in such a way that it never collides with the input pointer. Moreover, to maintain the time evolution of the signal, the output pointer should always keep the pace with the input pointer. This way, changes in the amplitude and frequency of the signal are followed sufficiently. Therefore, for every sample in the input, the distance between the input and output pointers is measured, d. If it is longer than, or equal to, a maximum allowed distance $d_{\max}$, the output pointer has to jump forward, behind the input pointer. On the other hand, if this distance is shorter than, or equal to, a minimum allowed distance $d_{\min}$, it is possible that soon the output pointer collides with the input pointer. Hence, it is required that the output pointer jumps backward in the ring buffer and continues resampling the signal from this point. This process has been shown in Figure 3.

Figure 3 (a) shows the case in which the input signal is pitch-shifted down and the distance between the pointers increases. Therefore, the output pointer hops forward. Figure 3 (b) shows the opposite case, when the signal is pitch-shifted up and the output pointer should jump backward to avoid collision. It is important to remember that the hop size cannot be very long, since we are aiming at following the changes in the signal.

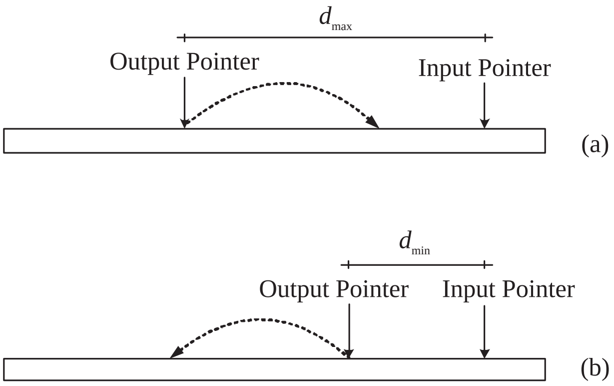

Figure 3: *Behavior of the output pointer with respect to its distance from the input pointer, d, (a) when $d > d_{\max}$, the output pointer jumps forward, (b) when $d < d_{\min}$, the output pointer jumps backward.*

The hop of the output pointer to the new location will break the periodicity of the signal. Therefore, it is required to search for the best point for the output pointer to jump so that the periodicity of the signal is maintained. This search is performed in another ring buffer of the same size as the main ring buffer. However, this second ring buffer contains the normalized low-pass filtered version of the signal stored in the main ring buffer. The best splicing point is, then, searched in the normalized low-pass filtered version of the signal using AMDF. The AMDF of two frames of length L in the signal $x(n)$ is defined as

$$D(m) = \sum_{k=0}^{L-1} |x(k+m) - x(k)|, \tag{1}$$

where m is the time lag between two frames. The maximum similarity point in the search region is the point at which AMDF is minimum for the whole search region.

The reason why the normalized low-pass filtered signal has been used in the search for the best splicing point is that we aim at preserving the continuity of the signal in its lowest partials. By normalization, the effect of the signal level alteration from one period to the other period is eliminated in the search for the best match point. The other advantage of normalization is that the amplitude changes in one period of the signal are more distinguishable with respect to the original and low-pass filtered signal. Hence, it is possible to choose a short correlation window for the AMDF search and the best splicing point can be found more accurately. Figures 4 (a)-(c) show the original signal, the low-pass filtered signal, and the normalized low-pass filtered signals, respectively.

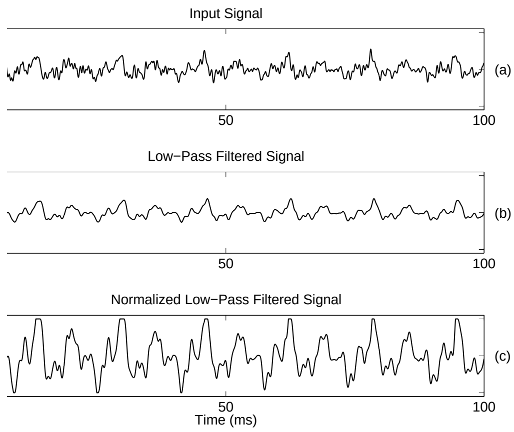

Figure 4: *(a) Original signal, (b) low-pass filtered version of the original signal, (c) normalized low-pass filtered version of the original signal.*

The choice of the correlation window and search area depends on the direction towards which the output pointer hops. Figure 5 shows two cases in which the output pointer jumps backward and forward, respectively. In Figure 5 (a), the output pointer jumps backward in the ring buffer. Therefore, the correlation window is chosen so that the output pointer points to the last sample in the correlation window. In the case the output pointer is to jump forward, the correlation window starts from the sample to which the output pointer points. This is shown in Figure 5 (b). In the figures, the best splicing points are also illustrated.

The process of finding the best splicing point is shown in Figure 6. Figure 6 (a) shows the correlation window and the search area. In Figure 6 (b), the search area has been zoomed in. The AMDF over the search area is demonstrated in Figure 6 (c). This figure shows the case the output pointer jumps forward in the ring buffer. As can be seen in the figures, the minimum AMDF results in the maximum similarity and, thus, it is highly probable that the periodicity of the signal is preserved.

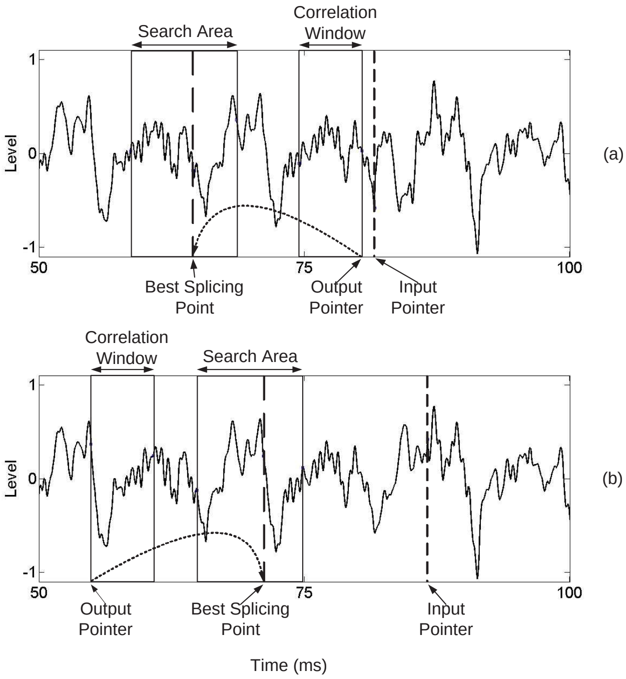

Figure 5: *Choice of the correlation window and search area in the search for the best splicing point, (a) the output pointer jumps backward, (b) the output pointer jumps forward.*

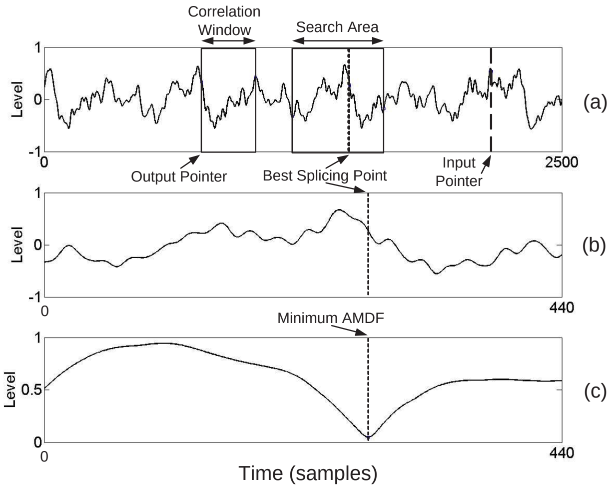

Figure 6: *Search for the best splicing point, (a) part of the ring buffer in which the correlation window and search area are placed, (b) the search area in which the best splicing point is looked for, (c) the AMDF over the search area for the shown correlation window.*

3. PARAMETER SETTING

For different input signals, the parameters of the algorithm are changed according to the period length of the lowest frequency in the frequency range of the input signal. In the following subsections, it is explained how to choose these parameters.

3.1. Maximum and Minimum Distances Between Pointers

The maximum and minimum allowed distances between the input and output pointers are essential parameters of the algorithm, which depend on the period length of the input signal. Empirically, the maximum allowed distance is chosen to be twice the longest period length of the input signal. When the distance between the output and input pointers reaches this amount, the output pointer will jump forward in the ring buffer for one period to keep the pace with the input pointer.

On the other hand, the minimum allowed distance depends on the length of the interpolator filter, the length of the cross fading region and the maximum pitch-shifting factor for upward pitch-shifting. The minimum allowed distance cannot be less than half the length of the interpolation filter. The length of the cross-fading region should be also considered. If the interpolation filter length is equal to N and the maximum pitch-shifting factor is $\alpha_{\max}$, the minimum allowed distance between input and output pointers is obtained by

$$d_{\min} = (\alpha_{\max} - 1)L_{\text{CrossFade}} + \frac{N}{2}, \tag{2}$$

where $L_{\text{CrossFade}}$ is the length of the cross-fading region. When the distance between the pointers reaches to $d_{\min}$, the output pointer jumps backward on the ring buffer for one period.

3.2. Lengths of Correlation Window and Search Area

The length of the correlation window L_{CorrWin}, and the length of the search area $L_{\text{SearchArea}}$ depend on the period length of the lowest frequency in the frequency range of the input signal T.

$$L_{\text{CorrWin}} = CT, \tag{3}$$

$$L_{\text{SearchArea}} = (1 - C)T, \tag{4}$$

where C is chosen to be $3/8$ in this algorithm, although a choice of $1/4$ of one period length of the signal for the length of the correlation window contains enough information to find its right location in the period.

The search area contains all the points in the ring buffer on which the first point of the correlation window is placed and its AMDF is computed in every iteration. Therefore, if the length of the search area is selected to be $(T - L_{\text{CorrWin}})$, the correlation window moves over one period of the signal. This is sufficient to find the best match point.

Another important parameter is the starting point of the search area. The choice of the search start point depends on the direction that the output pointer jumps in the ring buffer. However, it should be close enough to the input pointer, in order to follow changes in the signal. It is usually selected to be as far as one period length of the lowest frequency of the frequency range apart from the input pointer.

3.3. Other Parameters

There are a few other parameters that should be set in the algorithm. The first one is the size of the ring buffer. The ring buffer should have space for about four periods of the signal. Therefore, it can be chosen according to the longest period in the frequency range of the input signal.

The other parameter is the cut-off frequency of the low-pass filter, which is not very critical. However, it has to be greater than the highest fundamental frequency in the frequency range of the input signal. For example, if the fundamental frequency of the input signal ranges between 300 to 900 Hz, the cut-off frequency of the low-pass filter should be greater than 900 Hz.

The normalization is performed by multiplying the low-pass filtered signal by the reciprocal of its temporal envelope. Therefore, an envelope-following filter is used in the algorithm for which the parameters are determined empirically. However, in order to obtain a very smooth estimate of the signal amplitude, a large time constant for the envelope follower should be chosen.

4. TESTING AND RESULTS

The proposed algorithm has been tested using piano sound samples. Figure 7 shows the spectrograms of the original signal and pitch-shifted signals for three different techniques in finding the best splicing point. The signals are of duration 1.0 second and the pitch-shifting factor is equal to $+15\%$ for all the pitch-shifted signals. Figure 7 (a) shows the spectrogram of the original signal, the piano tone B1[1]. In Figure 7 (b), the best splicing point is searched in the original signal using the cross-correlation function, i.e., the standard SOLA method. Figure 7 (c) shows the case in which the best splicing point is searched in the normalized low-pass filtered version of the original signal using the cross-correlation function. In Figure 7 (d), the best splicing point is looked for in the normalized low-pass filtered signal with the AMDF. A linear-phase FIR filter of order 50 with a cut-off frequency of 70 Hz is used for low-pass filtering in both cases. Comparing the spectrogram of the original signal and those of the pitch-shifted signals reveals that the pitch-shifting process brings about discontinuities in the signal which appear as dark vertical lines in the figures (a few examples are circled in Fig. 7). These dark lines are the splicing points whose occurrence may be heard as clicks depending on the signal content at these points.

Figures 7 (b) and (c) show that when the signal is low-pass filtered and normalized before the splicing point search, the number of discontinuities and their intensities are reduced. These changes and improvements in the discontinuities are also audible in the sound samples. In the last case (Figure 7 (d)), where the NFC-TSM method has been used in the pitch-shifting process, no clicks are heard according to our informal listening tests [23]. This is because the continuity of the signal is preserved for low-frequency content of the signal. Moreover, small changes in the spectrogram have occurred, but many of the vertical lines are still visible. On the whole, searching the best splicing point in the normalized low-pass filtered signal leads to a smaller number of discontinuities than when searching in the original signal. The audibility of clicks and masking effects in the auditory system are beyond the scope of this paper, and are left for future work. In addition, the AMDF performed well in an objective evaluation of the quality of each synchronization procedure presented in [18].

Using this algorithm, real-time pitch-shifting of the input audio is possible. In addition, the pitch-shifting factor can be a time-varying function. For pitch-shifting down, there is no limitation on the amount of pitch-shifting factor. For pitch-shifting up, however, the restriction on the pitch-shifting factor is applied in the algorithm by the period length of the signal and the permitted delay. When pitch-shifting up, it is required to have at least one period of the signal in the ring buffer before the output pointer can jump backward on the ring buffer to repeat a sound segment due to the time-scale modification. This implies a delay in the system. For example, if the fundamental frequency of an input signal is equal to 63 Hz, for an allowed delay of 5 ms, the pitch-shifting up factor can have a maximum amount of $+30\%$. It should be noted that this restriction holds only at the onset of the tone, when pitch-shifting up. After having received one period of the input audio, this limitation is eliminated and, then, pitch-shifting factor can have any value.

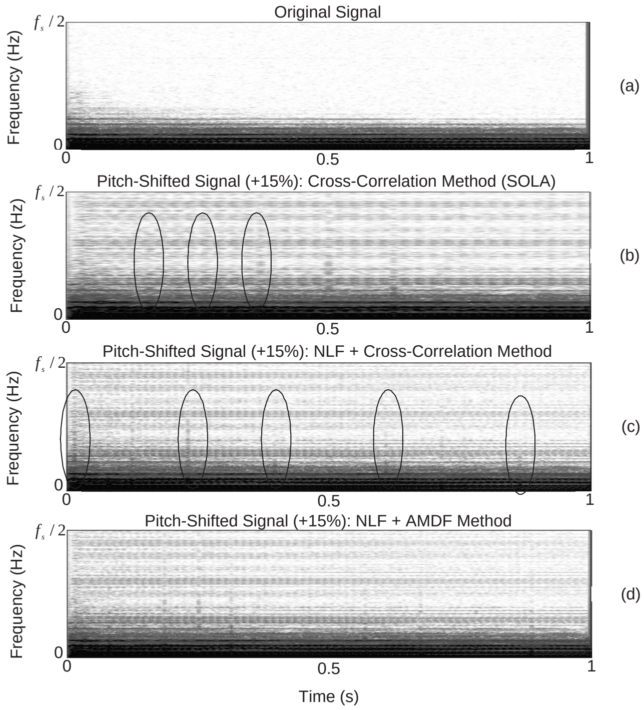

Figure 7: *Spectrograms of (a) the original signal (piano tone B1, fundamental frequency = 63 Hz) and pitch-shifted signals by $+15\%$ using different methods based on (b) search in the original signal using cross-correlation function, (c) search in the normalized low-pass filtered (NLF) version of the original signal using cross-correlation function, (d) search in the normalized low-pass filtered (NLF) version of the original signal using AMDF. The ovals in the figures indicate occurrence of audible artifacts.*

In the proposed algorithm, the AMDF calculation is computationally the heaviest part. However, the computation of AMDF is required only when the best splicing point should be found. If the maximum period length of the input signal is T samples, the number of abs, subtraction and addition operations required to find the best splicing point is equal to $C(1-C)T^2$. C is the coefficient to define the lengths of the correlation window and search region ($C = 0.375$). Moreover, $(1 - C)T$ comparisons should be performed to find the minimum AMDF. For example, when the minimum fundamental frequency in the frequency range of the input signal is 73 Hz, the lengths of the correlation window and search area are selected to be 230 and 380 samples, respectively, with the

[1]The sound sample has been taken from the McGill University Master Samples Collection.

sampling frequency of 44.1 kHz. Therefore, every time the best splicing point is searched in the ring buffer 87400 abs, subtraction, and addition operations and 380 comparisons are needed.

Since the search is performed in the normalized low-pass filtered version of the input signal, the smoothness of the normalized filtered signal can be exploited to reduce the sample rate and thus the computational load. According to the period length of the input signal, the correlation window and the search region can be downsampled when computing AMDF. The longer the period length, the greater the down-sampling factor is. For instance, in the above example a down-sampling factor of 7 can be used without any reduction in the quality of the output signal. This way the number of abs, subtraction, and addition operations can be reduced to 1815 and the number of comparisons to 55. Another method to make the implementation of the algorithm possible, regarding the heavy computational complexity of the AMDF, is to divide the computational load between a number of sound samples before the output pointer jumps in the ring buffer.

This algorithm has been implemented on a Motorola (Freescale) DSP56303 for a monophonic sound and it runs in real-time.

5. CONCLUSIONS

A real-time high-quality pitch-shifting algorithm was presented. The pitch-shifting algorithm is based on the resampling and time-scale modification method. A new method for time-scale modification of musical signals was developed which is called the NFC-TSM technique. In this technique, repeating and discarding signal segments are performed in such a way that the pitch-shifted signal has the highest similarity with the original signal in the case of the changes in the signal attributes. The best point to splice the signal segments is searched in the normalized low-pass filtered signal using the AMDF. In this algorithm, the pitch-shifting factor can have large values without any degradation in the quality of the signal. Sound samples are available at [23].

6. ACKNOWLEDGEMENTS

This work has been supported by the Foundation of Finnish Innovations and CIMO. Special thanks go to Oskari Porkka for the implementation of the algorithm on the Motorola DSP56303.

7. REFERENCES

[1] B. G. Crockett, "High quality multi-channel time-scaling and pitch-shifting using auditory scene analysis," in *Proc. 115th Convention of the Audio Engineering Society, New Yourk, USA, Preprint 5948*, October 2003.

[2] R. Mastro and D. Baldacci, "Real time pitch shift algorithm and enhanced tone controls for deejay applications," in *Proc. 106th Convention of the Audio Engineering Society, Munich, Germany, Preprint 4899*, May 1999.

[3] A. Duncan and D. Rossum, "Fundamentals of pitch shifting," in *Proc. 85th Convention of the Audio Engineering Society, Los Angeles, USA, Preprint 2714*, October 1988.

[4] G. J. Lin, S. G. Chen, and T. Wu, "High quality and low complexity pitch modification of acoustic signals," in *Proc. IEEE Intrnational Conference on Acoustics, Speech, and Signal Processing, vol. 5*, May 1995, pp. 2987–90.

[5] U. Zölzer, Ed., *DAFX - Digital Audio Effects*, J. Wiley & Sons, 2002.

[6] J. L. Flanagan and R. M. Golden, "Phase vocoder," *Bell System Technical Journal*, vol. 45, pp. 1493–509, 1966.

[7] J. Laroche, "Improved phase vocoder time-scale modification of audio," *IEEE Transactions on Speech and Audio Processing*, vol. 7, no. 3, pp. 323–32, May 1999.

[8] D. Rossum, "An analysis of pitch shifting algorithms," in *Proc. of the 87th Convention of the Audio Engineering Society, New York, USA, Preprint 2843*, September 1989.

[9] G. Fairbanks, W. L. Everitt, and R. P. Jaeger, "Method for time or frequency compression-expansion of speech," *Transactions of the Institute of Radio Engineers, Professional Group on Audio AU-2*, pp. 7–12, 1954.

[10] F. F. Lee, "Time compression and expansion of speech by the sampling method," *Journal of the Audio Engineering Society*, vol. 20, no. 9, pp. 738–42, 1972.

[11] D. Knox, N. Bailey, and I. Stewart, "A simple hybrid approach to the time-scale modification of speech," *Journal of the Audio Engineering Society*, vol. 53, no. 7, pp. 612–5, 2005.

[12] D. Dorran and R. Lawlor, "An efficient audio time-scale modification algorithm for use in a subband implementation," in *Proc. of the 6th International Conference on DAFX-03, London, UK*, September 2003.

[13] S. Roucos and A. M. Wilgus, "High quality time-scale modification for speech," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing, Florida, USA*, May 1985, pp. 493–6.

[14] E. Moulines, C. Hamon, and F. Charpentier, "An efficient high quality time-scale modification of speech signal," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing, Glasgow, UK*, May 1989, pp. 238–41.

[15] N. Schnell, G. Peeters, S. Lemouton, P. Manoury, and X. Rodet, "Synthesizing a choir in real-time using pitch-synchronous overlap add (PSOLA)," in *Proc. International Computer Music Conference, Berlin, Germany*, 2000, pp. 102–8.

[16] W. Verhelst and M. Roelands, "An overlap-add technique based on waveform similarity (WSOLA) for high quality time-scale modification of speech," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing, Minneapolis, USA*, April 1993, pp. 554–7.

[17] S. Yim and B. I. Pawate, "Computationally efficient algorithm for time-scale modification (GLS-TSM)," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing, Atlanta, USA*, May 1996, pp. 1009–12.

[18] D. Dorran, R. Lawlor, and E. Coyle, "A comparison of time-domain time-scale modification algorithms," in *Proc. 120th Convention of the Audio Engineering Society, Paris, France, Preprint 6674*, May 2006.

[19] R. W. Schafer and L. R. Rabiner, "A digital signal processing approach to interpolation," *Proc. IEEE*, vol. 61, no. 6, pp. 692–702, 1973.

[20] D. Rossum, "Constraint based audio interpolators," in *Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, New Paltz, USA*, October 1993, pp. 161–4.

[21] T. Laakso, V. Välimäki, M. Karjalainen, and U. K. Laine, "Splitting the unit delay-tools for fractional delay filter design," *IEEE Signal Processing Magazine*, vol. 13, no. 1, pp. 30–60, 1996.

[22] P. V. Sankar and L. A. Ferrari, "Simple algorithms and architectures for b-spline interpolation," *IEEE Transactions on Pattern Analysis Machine Intelligence PAMI-10*, vol. 10, no. 2, pp. 271–6, 1988.

[23] A. Haghparast, H. Penttinen, and V. Välimäki, "Sound samples," April 2007, http://www.acoustics.hut.fi/publications/papers/dafx07-pitch-shifting.

[24] J. Laroche, "Autocorrelation method for high-quality time/pitch-scaling," in *Proc. IEEE Workshop Application of Signal Processing to Audio and Acoustics, New York, USA*, October 1993, pp. 131–4.

[25] J. B. Hong, "An efficient high quality time-scale modification of speech signal," in *Proc. Finnish Signal Processing Symposium*, May 1997, pp. 139–42.