EFFICIENT, HIGH-QUALITY TIME-SCALING OF AUDIO SIGNALS

Markus S. Schlosser

Digital Audio Lab, Thomson Corporate Research
Karl-Wiechert-Allee 74, 30625 Hannover, GERMANY
phone: + (49) 511 418 1362, fax: + (49) 511 418 2483, email: markus.schlosser@thomson.net
web: www.thomson.net

ABSTRACT

In this paper, enhancements to the classical Waveform Similarity Overlap-Add (WSOLA) algorithm are proposed. As a time-domain approach, it works best for small speed changes and quasi-periodic, monophonic signals. Some of our enhancements are especially effective for small, others for large speed changes. As a consequence, significant improvements for all scaling factors are achieved extending the usability of the new scheme to larger speed changes and more complex signal characteristics. The reduction in computational complexity is analyzed by comparing the number of splice points needed to time-scale the input signal. As will be shown, these are the only points where real signal processing is performed. Therefore, a reduction in their number results in an equivalent decrease in computational demand. Additionally, they are also the only points where artifacts may arise so that, in many cases, a reduction in their number can serve as an indicator for improvements in the signal quality, too.

1. INTRODUCTION

Time-scaling denotes the classical problem of changing the play-out speed (or, equivalently, the duration) of an audio signal without altering its perceived frequency characteristics, i.e., its pitch and timbre. In our case, the objective was to design an efficient algorithm for typical frame rate conversions found in the film industry. Another major application in the professional domain is the generation of sound effects for music productions. Time-scaling capabilities are, however, also of increasing interest for consumer electronics products, like e.g., for audio play-out during fast forward. A more complete list of applications can be found in [2].

In America, feature films are typically recorded with a frame rate of 24 fps. To change this rate to the (nominal) 30 fps of NTSC for television, four film frames are converted into five video frames with a process called "3:2 pull-down" [4]. To this end, the film frames are copied in alternation into two or three video fields, which are one half of a video frame. Finally, these ten video fields are recombined into five complete video frames. This simple procedure results in a sawtooth pattern for moving objects and jitter. Both artifacts are, however, typically not perceived by the viewers. Furthermore, the overall duration is kept unchanged so that the audio signal can stay untouched. The final adjustment of the play-out speed to the actual 29.97 fps of NTSC is neither perceived optically nor acoustically.

The procedure for converting the frame rate from 24 fps to the 25 fps found in Europe is even more straightforward: The film is simply sped up by these approximately 4 % so that all the action happens just a little bit faster. On the video side, this is hardly perceived. Speeding-up the audio signal by a factor of $\frac{25}{24}$, however, results in an audible pitch shift by about two thirds of a semitone. Especially male actors are often not willing to accept this alteration of their voice so that time-scaling techniques need to be employed instead.

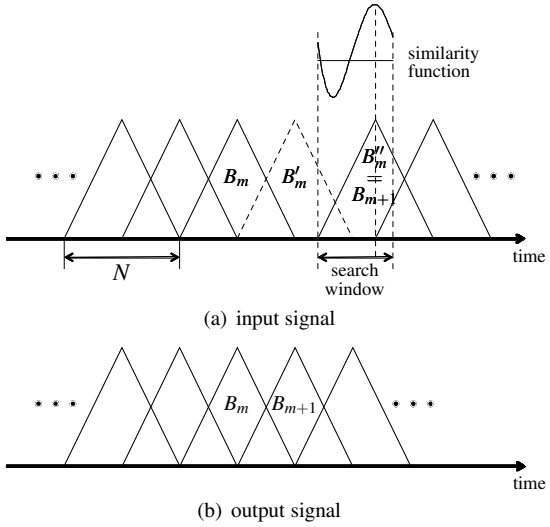


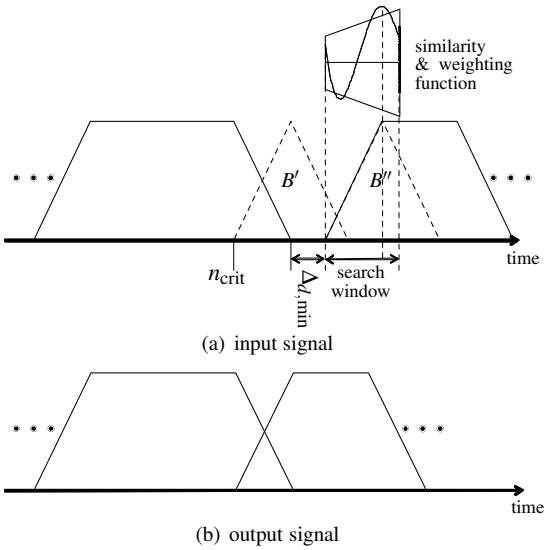
Figure 1: Illustration of the block-based processing for the classical WSOLA approach proposed in [6]

Time-scaling algorithms can broadly be categorized into those operating in the time-domain and those operating in the frequency-domain [2]. Time-domain techniques are computationally more efficient but typically restricted to quasi-periodic signals, like speech or monophonic music. The underlying idea is to detect individual periods in the signal that can then be repeated or discarded as needed. As long as the speed change is small, these time-domain techniques are, however, also capable of producing high-quality results for more complex signals. This is the reason why we opted for such a time-domain approach.

The paper is organized as follows: Sect. 2 introduces the classical Waveform Similarity Overlap-Add (WSOLA) approach, which forms the basis for our enhanced scheme detailed in Sect. 3. Sect. 4 presents some quantitative and qualitative results for our enhancements. A further extension of the new scheme to take the underlying signal characteristics into account is discussed in Sect. 5. Finally, Sect. 6 provides some concluding remarks.

2. CLASSICAL WAVEFORM SIMILARITY OVERLAP-ADD METHOD

Figure 1 illustrates the so-called Waveform Similarity Overlap-Add (WSOLA) approach as proposed in [6]. In this



(a) input signal

(b) output signal

Figure 2: Illustration of the proposed enhancements to the classical WSOLA approach

time-domain approach, the output signal is constructed from blocks B_m of a fixed length N (typically around 20 ms). As indicated by the triangles, these blocks overlap by 50 % so that a fixed cross-fade length is guaranteed. In our opinion, this is one of the reasons why the time-scaled signal is of higher quality than with other Synchronous OverLap-Add (SOLA) methods.

Another reason is that the best interval for a cross-fade is not only determined based on the interval itself but also takes the future development of the signal into account. If we assume that the block B_m has just been appended to the output signal, the next block $B_{m+1} = B''_m$ is determined as follows: B''_m is the block that is, first, most similar to the block B'_m that would normally follow the current block B_m and that, second, lies within a search window around the ideal position (as determined by the scaling factor). The deviation from the ideal position is thereby typically restricted to be less than 5 ms resulting in a search window of 10 ms in size.

3. ENHANCEMENTS

The classical WSOLA approach described in the last section can be enhanced in several ways. These enhancements are depicted in Fig. 2 and will be detailed in the following subsections.

3.1 Splice Point Prediction

The first enhancement consists in realizing that it is not always necessary to find the next block B_{m+1} by performing a computationally expensive template matching (i.e., by maximizing a similarity function). Especially if the scaling factor is close to one, the signal block B'_m whose most similar counterpart is searched for will often simply lie itself within the search window. For any sensible similarity function, this block should represent the best match, i.e., $B_{m+1} = B''_m = B'_m$.

Using the deviation from the ideal position of the last best match and the additional deviation caused by appending a consecutive block, it can easily be checked if this is the case and the position of the block determined. It can even be predicted for how many consecutive blocks this is going to be the case and, thus, an input signal segment of corresponding length can simply be appended to the output signal. This procedure not only avoids the computations for the template matching but also those needed to perform the cross-fades.

Let us denote the current deviation from the ideal position with $d(m)$ and the scaling factor with α, where $\alpha > 1$ shall indicate a stretching of the input signal. Performing the cross-fade between consecutive blocks is equivalent to adding $\frac{N}{2}$ further samples to the output signal. On the input side, $\frac{N}{2\alpha}$ samples should, however, have been used instead. This results in an additional deviation

$$\Delta_d = \frac{N}{2}\frac{\alpha - 1}{\alpha}. \tag{1}$$

As a consequence,

$$N_{\text{blocks}} = \text{floor}\left(\frac{\text{sign}(\alpha-1)d_{\max} - d(m)}{\Delta_d}\right) \tag{2}$$

consecutive blocks may be copied without surpassing the maximum allowed deviation $d_{\max}$. The factor "sign$(\alpha-1)$" takes into account that the deviation tends towards negative values for scaling factors smaller than one.

This enhancement can even be taken one step further by removing the restriction that the appended signal segment needs to consist of entire blocks. Instead of determining the block for which the maximum allowed deviation is surpassed, this can just as well be done on a sample by sample basis. This approach leads to calculating the critical sample n_{crit} (indicated in Fig. 2) for which the deviation $d(n_{\text{crit}})$ from the ideal position reaches the limit $\pm d_{\max}$. Analogous to Eq. 2, the maximum number of samples to be copied $N_{\text{samples}} = n_{\text{crit}} - n$ may be calculated instead as

$$N_{\text{samples}} = \text{floor}\left(\frac{\alpha}{\alpha-1}\big(\text{sign}(\alpha-1)d_{\max} - d(n)\big)\right). \tag{3}$$

This first enhancement can be summarized as realizing that the processing can be divided into two phases. In the first phase, samples are simply copied to the output signal as long as the deviation from the ideal input sample induced by this procedure stays in a predefined range. When the limit is reached, the signal is spliced and a template matching is performed in the second phase to find the most similar block within the allowed range.

3.2 Weighted Similarity Function

The second enhancement shown in Fig. 2 consists in weighting the similarity function. If there are similarly good matches within the search window, the candidate closest to the opposite end $\mp d_{\max}$ should be chosen. As described by Eq. 1, this results in longer signal segments being appended next so that the number of splice points is reduced. In addition to reducing the amount of template matchings, this also reduces the points were artifacts may develop. Consequently, an appropriately chosen bias will speed up the processing as well as improve the signal quality. The bias should, however, also not be chosen too high to avoid cross-fading between dissimilar signal segments.

3.3 Lower Limit for Minimum Hop Distance

The deviation from the ideal position at the critical sample $d(n_{\text{crit}})$ equals $\pm d_{\max}$. As described by Eq. 1, the subsequent cross-fade amounts to the signal block B'_m(whose most

similar counterpart is searched for) lying exactly Δ_d samples outside of the search window. Therefore, Δ_d may be denoted the minimum hop distance. It can be very small if the scaling factor α is close to one. In our case of increasing the play-out speed from 24 fps to 25 fps and a block length of $N = 20$ ms, the minimum hop distance becomes $\Delta_d = 10\,\text{ms} \cdot \frac{24-25}{24} \approx 0.4\,\text{ms}$.

This is problematic as the energy of audio signals tends to be concentrated in the low-frequency range so that any self-similarity function will have a broad peak around zero. The fundamental frequency of human speech typically lies between 80 Hz for deep male voices and 350 Hz for children [3]. Assuming a pure sine, this would be equivalent to the first zero-crossing of the auto-correlation function lying between $\frac{1}{4 \cdot 350\,\text{Hz}} = 0.7\,\text{ms}$ and $\frac{1}{4 \cdot 80\,\text{Hz}} = 3.1\,\text{ms}$. Now, if Δ_d is smaller than the peak, the template matching is likely to decide for the border of the search window being closest to the ideal point, i.e., to jump by the minimum hop distance. If, as in our case, Δ_d is even a lot smaller than the peak, the template matching will decide for the minimum hop distance several times in a row until the accumulated sum of Δ_d surpasses the width of the peak.

Using the terminology distinguishing two processing phases, there will be no copying phase. The output signal will contain a concatenation of cross-fades between signals being spaced Δ_d apart. This not only increases the computational demand dramatically, it also results in poor audio quality due to the induced quasi-periodic artifacts.

The weighting of the similarity function described in Sect. 3.2 can only lessen this effect but not remove it completely. Therefore, the third enhancement consists in introducing a lower limit $\Delta_{d,\min}$ for the minimum hop distance, i.e., artificially increasing the minimum hop distance if $\alpha \approx 1$. For our implementation distinguishing two processing phases, this can easily be achieved by increasing the maximum allowed deviation from the ideal point during the copying phase by the corresponding difference d_x, i.e.,

$$N_{\text{samples}} = \frac{\alpha}{\alpha - 1}\big(\text{sign}(\alpha - 1)(d_{\max} + d_x) - d(n)\big) \qquad (4)$$

where

$$d_x = \min(0, \Delta_{d,\min} - |\Delta_d|)\,. \qquad (5)$$

4. RESULTS

The first aspect of the enhancement described in Sect. 3.1 consists in separating the block-based processing into a copying and a template matching phase. This enhancement merely reduces the computational load without affecting the output signal. The computational savings can, however, be significant if, as in our case, the speed change is small so that the copied signal segments are long. As the required number of splice points for this approach equals the effective number of splice points in the classical WSOLA, it shall serve as a reference in the sequel (cf. column "Sect. 3.1a" in Table 1).

On the other hand, the second aspect of moving from a block-based to a sample-based copying of signal segments guarantees that the allowed deviation is consistently exploited to its maximum. As can be seen in column "Sect. 3.1b" Table 1, the reduction in the number of splice points is most prominent for large speed changes, i.e., when Δ_d in Eq. 1 is large. In this case, only few or even no complete block may be copied so that the introduction of fractional blocks makes a huge difference. As artifacts may only arise at splice points, reducing their number improves the audio quality accordingly.

Table 1: Comparison between required number of splice points for…

(a) 18 s of male speech

α	Sect. 3.1a	Sect. 3.1b	Sect. 3.2	Sect. 3.3
0.5	896	600	566	–
0.67	834	627	557	–
0.8	675	518	457	426
0.96	653	633	404	83
1.04	718	642	406	84
1.25	929	718	632	533
1.5	1441	1168	1028	–
2	2440	1918	1713	–

(b) 27 s of pop music

α	Sect. 3.1a	Sect. 3.1b	Sect. 3.2	Sect. 3.3
0.5	1345	875	821	–
0.67	1335	1025	967	–
0.8	1227	994	914	829
0.96	1135	1090	798	157
1.04	1206	1132	705	160
1.25	1666	1338	1198	1014
1.5	2470	2099	1896	–
2	3846	3006	2806	–

As already detailed in Sect. 3.2, weighting the similarity function offers a trade-off between the number of splice points and the similarity of the cross-faded signal blocks. As a consequence, the number of points where artifacts may arise is reduced but the severeness of the artifacts increased. In informal listening tests, a linear weighting function where the similarity values at the unfavorable border of the search window are reduced by 30 % resulted in a notable reduction in the number of splice points (cf. column "Sect. 3.2" in Table 1) while at the same time improving the audio quality slightly.

Finally, Table 1 also shows the results for limiting the minimum hop distance of Sect. 3.3. The exact value of the limit is uncritical. It should just neither be chosen too small to be effective nor too large to avoid jumps by two periods. Similar considerations to the comparison with the typical range of pitch periods in Sect. 3.3 led us to limiting the minimum hop distance to $\Delta_{d,\min} = 3\,\text{ms}$. As this lower limit is only effective for $\alpha \in [\frac{10}{13}, \frac{10}{7}]$ (cf. Eq. 1), the remaining cells in column "Sect. 3.3" of Table 1 are empty. For our target application of changing the play-out speed from 24 fps to 25 fps (i.e., $\alpha \approx 1.04$), introducing this limit for the minimum hop distance reduces the number of splice points dramatically. In this case, the final enhancement has a huge effect on computational efficiency as well as audio quality.

5. SIGNAL-AWARE TIME-SCALING

As already detailed in the introduction, time-domain approaches work best for quasi-periodic, monophonic signals where sequences of similar signal segments exist so that individual segments may be repeated or discarded as needed. Due to the lack of temporal structure, noise-like and silent signals are well suited to be time-scaled, too. On the other hand, transients should be kept unchanged as no similar sig-

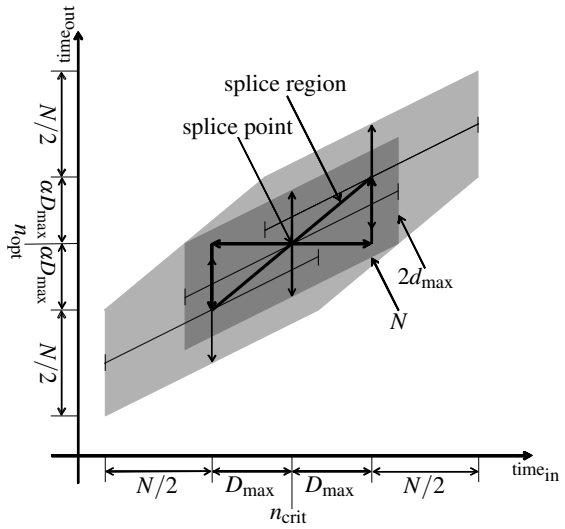


Figure 3: Amount of sample-wise similarities needed to determine the best match a) for one single point (dark gray area) and b) within a whole region (light gray area)

nal segments can be found by nature [5].

The classical WSOLA method is, however, agnostic of the underlying signal characteristics. Splice points are merely determined by constraining the allowed temporal deviation. For this reason, it was already proposed in [1] to determine the type of a signal segment before time-scaling it with an adjusted scaling factor. The main disadvantages of this approach are that signal segmentation is itself not a trivial task and that the difference in treatment of the various segments needs to be tuned manually.

Our approach distinguishing two processing phases opens up a straightforward alternative where the splicing of the signal is inherently restricted to positions where a good match can be found. Instead of determining one critical sample n_{crit} where the signal is spliced and a template matching performed, the best match may be searched for within a small region around this critical sample. This provides the necessary flexibility to cope with instances where n_{crit} happens to fall in the middle of a transient so that looking for a similar match is futile. In such a case, a sample just before or after the transient should be chosen instead.

This procedure not only avoids both of the disadvantages mentioned above, the similarity between signal segments should also be the single most important factor in determining where a signal is spliced. In addition to this, the splice points merely need to be restricted to an approximately uniform distribution in time to avoid audible speed changes, i.e., flutter.

Finding the best match within a whole region instead of for one single point sounds like a dramatic increase in computational complexity. At first sight, a one-dimensional search is replaced by a two-dimensional one. On the other hand, practically any similarity function between audio signals is based on averaging sample-wise similarities. As indicated by the thick arrows starting at the "splice point" in Fig. 3, moving from one sample to the next will typically simply mean to remove one of these sample-wise similarities while adding another one. As a consequence, the complexity does not increase quadratically but merely linearly.

In Fig. 3, n_{crit} is the critical sample and n_{opt} the corresponding optimal sample. $D_{\max}$ denotes the maximum allowed deviation from the critical sample, i.e., the given splice region is of length $2D_{\max}$. Finally, shifting n_{crit} by $D_{\max}$ corresponds to shifting n_{opt} by $D'_{\max} = \alpha D_{\max}$. As a consequence, the size of the dark gray area is $A = N \cdot 2d_{\max}$ and that of the light gray area $B = N \cdot (2d_{\max} + 2D_{\max}\sqrt{1+\alpha^2})$. Therefore, the computational demand only increases by $\frac{B}{A} = 1 + \frac{D_{\max}}{d_{\max}}\sqrt{1+\alpha^2} \approx 1+\sqrt{2}$ (for $D_{\max} = d_{\max}$ and $\alpha \approx 1$).

6. CONCLUSIONS

A thorough analysis of the Waveform Similarity Overlap-Add (WSOLA) method leads to the development of a new scheme distinguishing two processing phases: a copying and a splicing phase. This shift in point of view paves the way for several improvements.

First, the block-based processing is switched to a sample-based one resulting in a consistent exploitation of the maximum allowed temporal deviation. The effect of this improvement is especially pronounced if the typical length of copied signal segments is short, i.e., for large speed changes.

Second, a weighting of the similarity function is introduced to achieve a bias towards long copying phases. This allows for a trade-off between the number of points where artifacts may occur and the severeness of these artifacts.

Third, the minimum hop distance is restricted to stay above a lower limit to avoid quasi-period artifacts for small speed changes. This improvement is especially important for our target application as it not only improves the signal quality significantly but also has a dramatic effect on the computational complexity in this case.

Finally, our new scheme also allows for an alternative approach to make the time-scaling algorithm aware of the underlying signal characteristics. Instead of an explicit segmentation of the input signal, the signal characteristics are taken into account implicitly during the splicing phase by maximizing the similarity function for a whole splice region. As the computational demand stays manageable, this straightforward approach is intuitively appealing. Cross-fading between similar signal segments is the crucial factor in minimizing audible artifacts.

REFERENCES

[1] M. Demol, W. Verhelst, K. Struyve, and P. Verhoeve. Efficient non-uniform time-scaling of speech with WSOLA. In *Int. Conf. on Speech and Computers (SPECOM)*, pages 163–166, 2005.

[2] D. Dorran, R. Lawlor, and E. Coyle. A comparison of time-domain time-scale modification algorithms. In *120th AES Convention*, pages 6674–6691, 2006.

[3] K. Fellbaum. *Sprachverarbeitung und Sprachübertragung*. Berlin: Springer, 1984.

[4] T. Holman. *Sound for Film and Television*. Oxford: Focal Press, 2nd edition, 2002.

[5] S. Lee, H. D. Kim, and H. S. Kim. Variable time-scale modification of speech using transient information. In *Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 2, pages 21–24, 1997.

[6] W. Verhelst, D. Compernolle, and P. Wambacq. A unified view on synchronized overlap-add methods for prosodic modifications of speech. In *6th Int. Conf. on Spoken Language Processing (ICSLP)*, volume 2, pages 63–66, 2000.