Efficient Implementation of WSOLA

Matt Flax <flatmax at ieee dot org>

1 Introduction

WSOLA [1] is an algorithm for the time-scale modification of audio. Simplistic in nature, WSOLA copies required information in the time domain from the original audio waveform to the output or desired audio waveform. This copy process operates by firstly segmenting the input audio waveform into overlapping frames. Assume the algorithm starts with the input audio frame x_i. The output (time-scaled) waveform is created by copying a windowed version of this input frame $y_i = x_i.w$ as its output frame (where w is the windowing function and $a.b$ is inner product of the two vectors a and b). The algorithm now requires the output frame y_{i+1}.

To preserve pitch and auditory features, it is generally assumed that vector re-sampling is not allowed. Consequently the vector lengths of frames y_i, x_i and the window w are all the same and denoted m. Essentially this suggests that there is a relation for finding subsequent output frames where

$$y_{i+1}.w \,(=)\, x_{i+1}.w \tag{1}$$

where $(=)$ defines the maximum similarity operator. This may be re-written as

$$y(\frac{\tau im}{2} + n).w(n) \,(=)\, x(\frac{im}{2} + n).w(n) \tag{2}$$

where τ denotes the scaling factor and the factor of a half is due to the frame overlap (assuming 50% overlap).

The purpose of this article is to define an efficient implementation of the similarity operator as originally defined in [1].

2 Efficient implementation of the similarity operator.

[1] defines three similarity estimation operators. These are

1. Cross-correlation coefficient

$$C_c(m, \delta_{i+1}) = \sum_n x_{i,\delta_i}.x_{i+1,\delta_{i+1}} \tag{3}$$

where δ_{i+1} is the window sample delay variable.

2. Normalised cross-correlation coefficient. (A normalised version of Equation 3)

3. Simple Euclidean distance

$$C_A(m, \delta_{i+1}) = \sum_n |x_{i,\delta_i} - x_{i+1,\delta_{i+1}}| \tag{4}$$

It is obvious that in the time domain, both equations 3 and 4 require the same number of operations. In the frequency domain however, this is not the case.

In the frequency domain equation 3 becomes

$$\mathcal{F}\{C_c\} = \mathcal{F}\{x_{i,\delta_i}\}.\overline{\mathcal{F}\{x_{i+1,\delta_{i+1}}\}} \tag{5}$$

where $\mathcal{F}\{a\}$ denotes the discrete Fourier transform of vector a and $\overline{a}$ denotes the complex conjugate of a. Equation 4 becomes

$$C_A(m, \delta_{i+1}) = \sum_n \mathcal{F}^{-1}\{\mathcal{F}\{|x_{i,\delta_i}|\} - \mathcal{F}\{|x_{i+1,\delta_{i+1}}|\}\} \tag{6}$$

where $\mathcal{F}^{-1}\{A\}$ denotes the inverse discrete Fourier transform of the Fourier spectrum A.

It is apparent that equation 6 incurs a larger number of numerical operations then 4. Equation 5 reduces the number of numerical operations when compared to equation 3 (when an efficient implementation of the discrete Fourier transform is used).

3 Implementation and Conclusion

An implementation of [1] using a fast Fourier transform to replace the discrete Fourier transform shows that replacement of equation 4 by equation 5 speeds the operation of the WSOLA algorithm. Experimental measurements[1] show that stereo audio of duration 427.52s (sampled at 44.1 kHz using 16 bit samples) is processed using equation 4 in 11 414.64 s and equation 5 in 113.72 s. This is an improvement of 10 000%.

References

[1] W. Verhelst and M. Roelands. An overlap-add technique based on waveform similarity (wsola) for high quality time-scale modification of speech. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 2, pages 554 –557, April 1993.

[1] Using GPL software on an 800 MHz CPU with math co-processor. Software available at http://mffmtimescale.sourceforge.net/