ResearchGate

See discussions, stats, and author profiles for this publication at: https://www.researchgate.net/publication/242405491

Audio time-scale modification

Article · January 2005

CITATIONS
17

READS
532

1 author:

David Dorran
Technological University Dublin - City Campus

34 PUBLICATIONS **171** CITATIONS

SEE PROFILE

All content following this page was uploaded by David Dorran on 24 January 2014.

The user has requested enhancement of the downloaded file.

Audio Time-Scale Modification

PhD Thesis

2005

David Dorran

School of Control Systems and Electrical Engineering

Dublin Institute of Technology

Research Supervisors: Dr. Robert Lawlor

Dr. Eugene Coyle

Prof. Anthony Fagan

I certify that this thesis which I now submit for examination of the award of PhD is entirely my own work and has not been taken from the work of others save and to the extent that such work has been cited and acknowledged within the text of my work.

This thesis was prepared according to the regulations for postgraduate study by research of the Dublin Institute of Technology and has not been submitted in whole or in part for an award in any other Institute or University.

The work reported on in this thesis conforms to the principles and requirements of the Institutes for ethics in research.

The Institute has permission to keep, to lend or to copy this thesis in whole or in part, on condition that any such use of the material of the thesis be duly acknowledged.

Signature

Date

Abstract

Audio time-scale modification is an audio effect that alters the duration of an audio signal without affecting its perceived local pitch and timbral characteristics. There are two broad categories of time-scale modification algorithms, time-domain and frequency-domain. The computationally efficient time-domain techniques produce high quality results for single pitched signals such as speech, but do not cope well with more complex signals such as polyphonic music. The less efficient frequency-domain techniques have proven to be more robust and produce high quality results for a variety of signals; however they introduce a reverberant artefact into the output.

This dissertation focuses on incorporating aspects of time-domain techniques into frequency-domain techniques in an attempt to reduce the presence of the reverberant artefact and improve upon computational demands.

From a review of prior work it was found that there are a number of time-domain algorithms available and that the choice of algorithm parameters varies considerably in the literature. This finding prompted an investigation into the effects of the choice of parameters and a comparison of the various techniques employed in terms of computational requirements and output quality. The investigation resulted in the derivation of an efficient and flexible parameter set for use within time-domain implementations.

Of the available frequency-domain approaches the phase vocoder and time-domain/subband techniques offer an efficiency and robustness advantage over sinusoidal modelling and iterative phase update techniques, and as such were identified as suitable candidates for the provision of a framework for further investigation. Following from this observation, improvements in the quality produced by time-domain/subband techniques are realised through the use of a bark based subband partitioning approach and effective subband synchronisation techniques.

In addition, computational and output quality improvements within a phase vocoder implementation are achieved by taking advantage of a certain level of flexibility in the choice of phase within such an implementation. The phase flexibility established is used to push or pull phase values into a phase coherent state. Further improvements are realised by incorporating features of time-domain algorithms into the system in order to provide a ‘good’ initial set of phase estimates; the transition to ‘perfect’ phase coherence is significantly reduced through this scheme, thereby improving the overall output quality produced. The result is a robust and efficient time-scale modification algorithm which draws upon various aspects of a number of general approaches to time-scale modification.

Acknowledgments

The process of PhD research is a difficult task which is made possible by the support, cooperation and encouragement of a number of individuals. I wish to express my sincere gratitude to Dr. Bob Lawlor, both for his enthusiasm and for providing the guidance necessary to complete this work. I extend this gratitude to Dr. Eugene Coyle, firstly, for providing me with this research opportunity, and secondly, for his unwavering optimism and eagerness to help in any way he can.

Thanks to Prof. Anthony Fagan for his insights into the research process.

Thanks to Dan Barry and Mikel Gainza for their friendship and countless discussions on audio processing and politics.

Thanks to Aileen, Jane, Charlie, Derry, Cera, Ciarán and Dermot for their discussions on audio and research in general; To everyone in the Digital Media Centre for making my time there enjoyable and for undertaking all the listening tests; To my family and friends, for their support, and also for participating in listening tests.

A special word of thanks to my parents who have always been a source of encouragement and support.

Finally, and most importantly, I would like to thank my future wife, Áine, for believing in me and making me so happy over the last four years.

Table of Contents

Table of Figures .... **ix**

1 Introduction **1**

1.1 Audio Time-Scale Modification 1

1.2 Scope of Work 4

1.3 Dissertation Layout 5

1.4 A Note on Pitch/Frequency-Scaling 6

2 Review of Existing Approaches **9**

2.1 Time-Domain Overlap-Add Techniques 9

2.1.1 SOLA 12

2.1.2 GLS-TSM 16

2.1.3 EM-TSM 18

2.1.4 PSOLA 21

2.1.5 WSOLA 23

2.1.6 Laroche Implementation 24

2.1.7 Suzuki Implementation 25

2.1.8 AOLA 26

2.1.9 Time-Domain Conclusion 29

2.2 Time-Domain/Subband Techniques 31

2.2.1 Time-Domain/Subband Conclusion 33

2.3 Iterative Frequency-Domain Techniques 35

2.3.1 The Discrete Fourier Transform 35

2.3.2 Signal Estimation from a Modified Short Time Fourier Transform 37

2.3.3 Iterative Frequency-Domain Conclusion 40

2.4 Phase Vocoder Techniques 41

2.4.1 Filter Bank Interpretation of the Phase Vocoder... 41

2.4.2 Fourier Transform Interpretation of the Phase Vocoder 43

2.4.3 TSM using FFT Phase Vocoder Interpretation 44

2.4.4 Phase Locked Vocoder.. 49

2.4.5 Multi-resolution Implementations... 53

2.4.6 Time-scaling via Spectral Expansion 59

2.4.7 Phase Vocoder Conclusion 61

2.5 Sinusoidal Modelling Techniques 63

2.5.1 Analysis... 64

2.5.2 Synthesis 69

2.5.3 Time-Scaling using the Sinusoidal Model 71

2.5.4 Sinusoidal plus Stochastic Modelling 72

2.5.5 Shape Invariant Techniques 75

2.5.6 Multi Resolution Implementations.. 77

2.5.7 Sinusoidal Modelling Conclusion... 78

3 Review Summary and Conclusions 80

4 Analysis of SOLA Parameters 85

5 A Comparison of Synchronisation Procedures 92

5.1 Comparison of Computational Requirements 92

5.1.1 Basic Unbiased Correlation ... 92

5.1.2 Normalised Correlation ... 93

5.1.3 Simplified Normalised Correlation ... 94

5.1.4 Average Magnitude Difference Function ... 95

5.1.5 Mean Square Difference ... 95

5.1.6 Envelope Matching-TSM ... 96

5.1.7 Modified Envelope Matching - TSM ... 98

5.1.8 GLS-TSM ... 99

5.1.9 Peak Alignment ... 101

5.2 Computational Comparison Summary ... 102

5.3 Objective Output Quality Evaluation ... 104

5.3.1 An Objective Quality Measure ... 107

6 Time-Domain/Subband Improvements ... 112

6.1 Bark Subband Approach to TSM of Music ... 112

6.1.1 The Relationship between Bark Bands and Music ... 112

6.1.2 Choice of SOLA Parameters ... 121

6.1.3 Subjective Output Quality Comparison ... 123

6.2 Improved Subband Synchronisation ... 125

6.2.1 Choice of Subband Offsets ... 125

6.2.2 Subjective Output Quality Comparison ... 128

7 Phase Vocoder Improvements ... 134

7.1 Flexibility of Horizontal Phase Coherence ... 136

7.2 Reduction in Phasiness and Computations.. 143

7.3 Improving Initial Phase Estimates.. 146

7.4 Considerations for Stereo Recordings.. 151

7.5 Subjective Output Quality Comparison 151

7.6 Discussion 155

8 Summary and Future Work.. 157

8.1 Future Work 159

9 Appendix A 161

10 Appendix B 162

11 Appendix C 167

12 References 169

Table of Figures

Figure 1-1 Time-domain and frequency-domain representations of the source-filter model from [Lawlor '99].. 8

Figure 1-2 Formant preservation for a pitched-scaled signal....................................... 8

Figure 2-1 'Cut and splice' method for time-scale compression and expansion. 9

Figure 2-2 Artefacts arising from 'cut and splice' implementations. 10

Figure 2-3 Synchronised overlap removes artefacts associated with 'cut and splice' methods. 11

Figure 2-4 Overlap prediction within the SOLA algorithm....................................... 15

Figure 2-5 Samples for feature vector used to identify a suitable frame overlap position from [Yim '96]. 17

Figure 2-6 Calculation of analysis pitch marks within a PSOLA implementation..... 22

Figure 2-7 Mapping of analysis pitch marks to synthesis pitch marks within a PSOLA implementation.. 22

Figure 2-8 Operation of WSOLA from [Verhelst '93]. 23

Figure 2-9 Operation of the AOLA algorithm from [Lawlor '99b]............................ 27

Figure 2-10 AOLA expansion process... 28

Figure 2-11 Overview of the time-domain/subband approach................................... 31

Figure 2-12 The application of a windowing function to an audio waveform............ 37

Figure 2-13 Phase vocoder overview... 41

Figure 2-14 Filterbank interpretation of the phase vocoder....................................... 42

Figure 2-15 Removal of discontinuity using a phase update procedure. 45

Figure 2-16 Analysis to synthesis frame mapping used in [Bonada ‘00]. 48

Figure 2-17 Multi-resolution Phase Vocoder block diagram from [Bonada ‘00]....... 53

Figure 2-18 Time-varying cutoff frequency within a multi-resolution phase vocoder from [Bonada ‘00]. 54

Figure 2-19 Compression and translation of the mother wavelet. 56

Figure 2-20 Frequency response of Morlet Wavelet at scales 8, 4, 2 and 1 i.e. dyadic scales from [Gersem ‘97]. 57

Figure 2-21 Overview of time-scaling by spectral expansion from [Ferreira ‘99]. 60

Figure 2-22 Parabolic frequency interpolation to obtain an accurate frequency estimation from a DFT. 65

Figure 2-23 Sinusoids plus noise analysis block diagram from [Serra ‘90]. 74

Figure 2-24 Sinusoids plus noise synthesis block diagram from [Serra ‘90].............. 74

Figure 4-1 Scenarios involving overlapping synthesis frames................................... 87

Figure 5-1 Case where peak alignment procedure would result in poor synchronisation.. 105

Figure 6-1 Tonal consonance as a function of critical bandwidth separation from [Plomp ‘65]. 113

Figure 6-2 Relationship between tonal consonance and western musical frequency ratios from [Plomp ‘65]... 114

Figure 6-3 Mean ‘maximum complexity’ of subbands for uniform (continuous line) and bark based (dotted line) partitioning schemes. 117

Figure 6-4 Mean ‘average complexity’ of subbands for uniform (continuous line) and bark based (dotted line) partitioning schemes... 118

Figure 6-5 Mean ‘average complexity’ of subbands for bark (continuous line), constant-Q (dotted line) and ERB (dashed line) partitioning schemes. 119

Figure 6-6 A masked/silent region within a synthesis frame. 127

Figure 6-7 The effects of subband synchronisation on an oboe signal. 132

Figure 6-8 The effects of synchronisation on a guitar and castanets signal. 133

Figure 7-1 The effect of the loss of horizontal phase coherence upon a pure sinusoid within a phase vocoder frame. 136

Figure 7-2 Vector representation of Figure 7-1. 137

Figure 7-3 Frame mapping required to achieve the desired time-scaling. 144

Figure 7-4 Copying a time-domain segment to the output. 145

Figure 7-5 Ratio of computations required for the improved phase vocoder implementation to the number of computations required using the phasiness reduction technique. 146

Figure 7-6 An iteration of the hybrid algorithm. 148

Figure 7-7 The effects of the reduction of phasiness using the hybrid implementation on the utterance ‘She had your dark’. 156

Figure 10-1 Vector representation of 75% STFT analysis overlap. 162

1 Introduction

1.1 Audio Time-Scale Modification

Time-scale modification of audio alters the duration of an audio signal while retaining the signal's local frequency content, resulting in the overall effect of speeding up or slowing down the perceived playback rate of a recorded audio signal without affecting the perceived pitch or timbre of the original signal. In other words, the duration of the original signal is increased or decreased but the perceptually important features of the original signal remain unchanged; in the case of speech, the time-scaled signal sounds as if the original speaker has spoken at a quicker or slower rate; in the case of music, the time-scaled signal sounds as if the musicians have played at a different tempo.

Transforming audio to an alternative time-scale is a popular and useful digital audio effect that has become a standard tool within many audio multi-processing applications. Some particular uses of this effect are:

Music and foreign language learning/teaching: It is often beneficial for students of music to play along with an accompaniment while practicing. A live accompaniment is not always available and a recording can be used instead[1]. However, the recorded accompaniment is recorded at a certain tempo. Time-scale modification algorithms provide independent control of the playback rate to suit a student's ability. In a similar manner, a foreign language student could control the rate of articulation of a recorded native speaker, which could be increased as the students comprehension improves [Erogul '98], [Donnellan '03], [Demol '04].

Fast/slow playback for telephone answering machines and dictaphones: For a telephone answering machine, fast playback allows a listener quickly scan through messages and slow playback allows the listener understand messages that were spoken too quickly e.g. a contact telephone number [Hejna '90]. Also, a secretary can

[1] Music Minus One (http://www.musicminusone.com) recognised this problem over 50 years ago.

adjust the playback rate of recorded speech on a voice recorder to suit his/her typing skills.

Fast video browsing: Control of the playback rate allows users quickly scan through a video archive [Amir ‘00].

Fast browsing of speech material for digital library and distance learning: Allows a user quickly scan through a vast amount of speech data [Wong ‘98], [Omoigui ‘99].

Video-cinema standards conversion: Video uses a standard of 25 frames per second, while cinema uses a standard of 24 frames per second. Conversion between these two standards requires the use of time-scale modification algorithms [Pallone ‘99].

Audio Watermarking: A recent paper proposed the use of time-scale modification algorithms for the purpose of audio watermarking [Mansour ‘01].

Accelerated aural reading for the blind: Control of the playback rate allows visually impaired people quickly scan through passages in much the same way as visual readers can scan a text [Arons ‘97].

Music composition: Composers working with pre-recorded audio can modify the tempo of the recorded musical pieces to ‘fit in’ with their composition [Bonada ‘02]. Non-linear time-scale modification can also be applied to achieve artificial rubato [Barry ‘04].

Audio-video synchronisation: Video scenes are often accompanied with music so as to create a certain mood in the audience. The tempo of the visual scenes and the music should be synchronised with each other; time-scale modification facilitates this requirement without the need for re-recording a live orchestra/band [Bonada ‘02].

Audio data compression: Audio is first time-scale compressed, transmitted and then re-expanded at the receiver [Dudley ‘38], [Malah ‘79], [Wayman ‘88].

Diagnosis of cardiac disorders: In [Telatar ‘03] time-scale modification techniques are used to help medical personnel diagnose cardiac conditions in patients by slowing down the signal produced by the heart monitor.

Editing audio/visual recordings for allocated time-slots within the radio/television industry: In the television and radio industry airtime is segmented into time-slots, with programs and advertisements filling these slots. Time-scale modification of recorded audio (together with a speed-up/slow-down of the visual frame rate for the television industry) allows producers compress or expand programs and advertisements to fit their allotted time-slots [Rodriguez-Hernandez ‘94].

Voice gender conversion: Voice gender conversion algorithms can be implemented using a combination of time-scale modification algorithms with linear predictive analysis/synthesis [Lawlor ‘99].

Text-to-speech synthesis: Natural sounding text-to-speech synthesis can be achieved by using time-scale modification techniques to alter prosody [Moulines ‘88], [Charpentier ‘90], [Macon ‘96].

Lip synchronisation and voice dubbing: Time-scale modification can be used to synchronise recorded speech with corresponding video [Verhelst ‘03].

Prosody transplantation and karaoke: By making use of time-scale modification algorithms the prosody of a speech signal can be altered to match that of a target speaker. This facility would be of particular use in karaoke machines whereby the musical expression of the user’s voice is corrected to match that of the original artist [Verhelst ‘03].

Existing time-scaling algorithms differ in their computational requirements and/or quality of output. Assessment of the output quality is difficult to quantify, due to its subjective nature, and also due to the fact that applications differ in their output quality assessment criteria. For example, accelerated aural reading for the blind may focus on the intelligibility of the output more than the naturalness but for the purpose of film synchronisation, naturalness may be as important as intelligibility. As the computational power of microprocessors increases, algorithm efficiency becomes less important. However, efficiency will always be an issue, albeit to various degrees, depending on the application; for example, the provision of a speed-up/slow-down facility in today's digital dicataphones would require algorithm efficiency to be taken into consideration to a greater degree than that of a PC based application.

1.2 Scope of Work

The principle objective of this work is to develop time-scale modification algorithms which improve upon the quality and/or efficiency of existing approaches. The algorithms developed should ideally be capable of time-scaling speech and music; however, due to the fact that the majority of applications outlined above deal with either speech or music separately, algorithms that are directly applicable to speech or music are also considered. In addition, both English speech and Western tonal music receive particular attention.

To achieve this objective, a review of existing approaches is first undertaken so as to identify the state-of-the-art in the area. The broad findings of the review are that the computationally efficient time-domain implementations offer the best trade-off in terms of quality and computational requirements when time-scaling single pitched sounds, such as speech and monophonic music. However, for more complex signals, such as polyphonic music, alternative techniques should be used. Of the available alternatives, the phase vocoder and time-domain/subband implementations are identified as being the best, due to the robustness they provide over other candidates.

Within time-domain techniques a number of different implementations exist; however, a comparison of these alternatives, in terms of quality and computational load, is not available. The first contribution of this work is the provision of such a comparison.

In a second contribution, a set of equations that provide the basis of computationally efficient time-domain implementations is derived.

Time-domain/subband approaches partition the complex broadband input into less complex subbands, which can introduce subband synchronisation issues, as explained in section 2.2. The third contribution of this work is the provision of improved subband partitioning and synchronising procedures, resulting in an increase in the quality of output produced.

Phase vocoder techniques suffer from a reverberant (or phasy) artefact being introduced into the time-scaled output, due to a loss of phase coherence between subband components as explained in section 2.4. It should be noted that the artefact is similar to reverberation in that small delays are effectively introduced between subband components by the time-scaling process; while with true reverberation, delays are introduced between reflections of the entire audio signal. In a fourth contribution, a phase vocoder implementation is developed which reduces the effects of this artefact for moderate time-scaling. In addition, the novel implementation is more efficient than existing phase vocoder implementations.

1.3 Dissertation Layout

Frequency-scale modification (also referred to as pitch-scale modification) of audio alters the local pitch of an audio signal without altering the duration of the original signal. For example, a frequency-scaled version of a music signal sounds like the music was played in a different key. Time-scale and frequency-scale modification are closely related; frequency-scale modification of a sound can be achieved by first applying time-scale modification and then simply resampling the resulting time-scaled signal [Ellis ‘92]. There are, however, additional considerations, when pitch-scaling, that lie outside the scope of time-scaling implementations. Owing to the close relationship between the two audio effects, it is appropriate to briefly discuss these additional considerations; such a discussion is provided in section 1.4.

Chapter 2 reviews existing time-scale modification techniques. This chapter is divided into five distinct categories i.e. time-domain, time-domain/subband, phase vocoder, iterative frequency-domain and sinusoidal modelling approaches.

Chapter 3 provides a summary and discussion of the review given in chapter 2. Following from the discussion, a number of areas are identified for future work, thus providing a framework for the remainder of this dissertation.

Chapter 4 presents an analysis of the time-domain synchronised overlap-add algorithm, whilst chapter 5 compares a number of synchronisation techniques employed within time-domain algorithms.

Chapter 6 suggests the use of a subband partitioning approach based on the bark scale within time-domain/subband implementations, as an improved alternative, in terms of output quality, to the uniform width subbands suggested previously in the literature. In addition, a number of subband synchronisation schemes are presented, resulting in a further improvement in the quality produced.

Chapter 7 presents a computationally efficient phase vocoder based scheme that reduces the effect of the reverberant/phasy artefact associated with existing phase vocoder implementations.

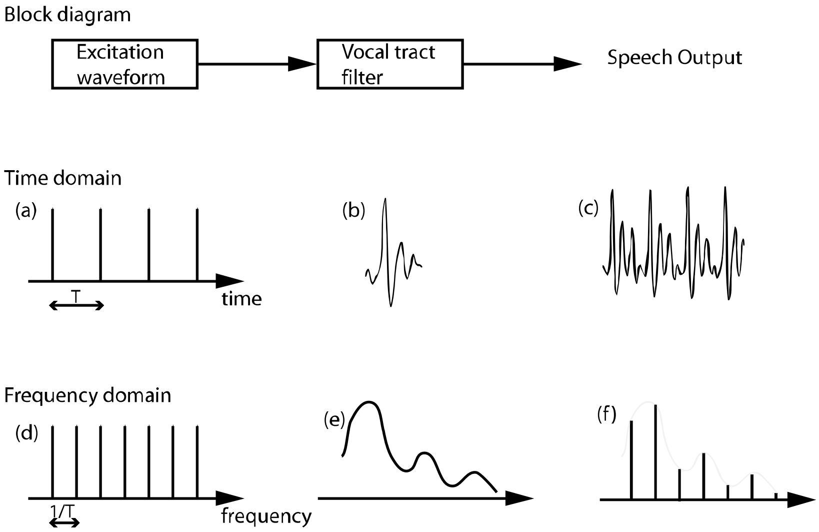
Chapter 8 provides a summary of the main contributions of the dissertation and suggests a number of possible routes for future work.

1.4 A Note on Pitch/Frequency-Scaling

Whilst achieving pitch-scaling by first time-scaling and then resampling provides an efficient solution to the pitch-scaling problem it fails to preserve the frequency envelope structure (formant structure for speech) [Bristow-Johnson ‘95], which can result in unnatural sounding modifications being produced. This problem can be understood by considering how the speech production system basically operates i.e. a stream of air, from the lungs, is passed through the vocal cords, generating an excitation waveform, which is then passed through the vocal tract and finally out through the mouth [Dudley ‘39], [Flanagan ‘65]. The excitation waveform is often modelled as a stream of impulses as shown in Figure 1-1 (a), with the corresponding frequency representation of the impulse stream shown in Figure 1-1 (d).

As the excitation waveform is passed through the vocal tract certain frequencies are amplified as they resonate with the vocal tract. Certain frequencies will also be attenuated, but the attenuation effects (modelled by zeros; resonances are modelled by poles) are often ignored when modelling the vocal tract because they are perceptually less relevant [Fant '60]. An example of the frequency response of the vocal tract is shown in Figure 1-1 (e), and the spectrum of the speech output is shown in Figure 1-1 (f).

Now, consider the case where the pitch of the excitation, i.e. the duration between impulses of the modelled excitation waveform, is modified. The frequency response of the vocal tract will not change as the pitch changes and the expected frequency response of the output speech signal is shown in Figure 1-2 (c). It should be noted that the spectral envelope of the pitch modified signal is the same as the original as shown in Figure 1-1(f). However, if pitch-scale modification is achieved through the combination of time-scale modification and resampling, the frequency response of the resulting speech signal would be similar to that shown in Figure 1-2 (f) i.e. the format structure has also been shifted. This modification of the spectral envelope (formant structure) of the speech signal results in the pitch-scaled output losing the naturalness of the original signal and taking on what is commonly referred to as a 'chipmunk' like sound effect[2].



[2] After the 'Chipmunk Song' by Ross Bagdasarian, 1958. The song was created by speeding up an original recording i.e. resampling. The song eventually lead to the creation of an animated television series in the 1961 entitled 'The Alvin Show' and a subsequent series entitled 'Alvin and the Chipmunks' in the 80's.

Figure 1-1 Time-domain and frequency-domain representations of the source-filter model from [Lawlor '99].))

It should be noted that for pitch-scale modification of music, the spectral envelope shape should be maintained for instruments that have resonating bodies that do not significantly change when different notes are played e.g. the guitar and piano. The resonating frequencies of instruments such as the flute and tin whistle are constantly changing as different notes are being played (the effective length of the resonating body is being altered for these cases), therefore high quality pitch-scale modifications would require that the spectral envelope for each note be known prior to the application of pitch-scale modification algorithms. For recordings with a number of instruments present, each instrument should first be identified and extracted/separated from the original recording and then individually pitch-scaled for high quality pitch-scale modification to be achieved. Source separation algorithms [Fitzgerald '04] have yet to achieve extremely high quality results and artefacts introduced by them are likely to be more objectionable than the unnaturalness introduced by an alteration in the spectral envelope, therefore the time-scaling plus resampling approach currently appears to be the best option for multi-instrument recordings.

The interested reader can refer to [Acero '98], [Ansari '97], [Ansari '98] and [Jiang '01] for implementation specific details on formant preserving pitch modification algorithms.

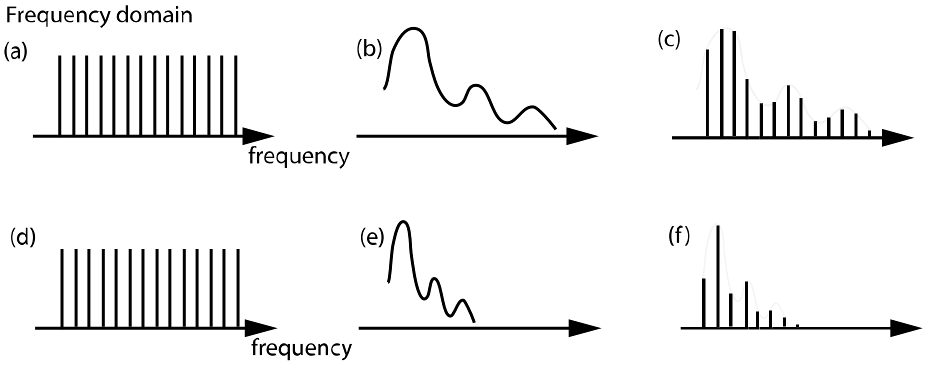
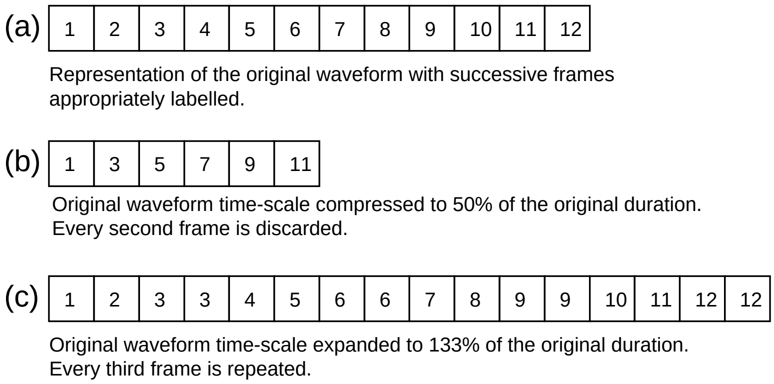


Figure 1-2 Formant preservation for a pitched-scaled signal.

2 Review of Existing Approaches

2.1 Time-Domain Overlap-Add Techniques

Time-domain techniques operate by discarding or repeating suitable segments of the input waveform. One such time-domain approach is commonly referred to as ‘cut and splice’ and its concept is illustrated in Figure 2-1, where (a) represents the audio input that has been segmented into non-overlapping frames; (b) represents a 50% time-scale compressed version of the input; and (c) represents a 133% time-scale expanded version of the input. [Portnoff ‘81] notes that the length of each discarded/repeated segment should be longer than one pitch period but shorter than the length of a phoneme.



Representation of the original waveform with successive frames appropriately labelled.

Original waveform time-scale compressed to 50% of the original duration. Every second frame is discarded.

Original waveform time-scale expanded to 133% of the original duration. Every third frame is repeated.

Figure 2-1 ‘Cut and splice’ method for time-scale compression and expansion.))

Gabor first implemented this type of approach in [Gabor ‘44] using an electro-mechanical device. Later, [Garvey ‘53] manually spliced segments of an audio magnetic tape to achieve a similar effect and Fairbanks used a modified tape recorder with four rotating playback heads to effectively automate Garvey’s process in [Fairbanks ‘54]. A digital implementation of the ‘cut and splice’ process is given in [Lee ‘72]. Scott presents a variant of the ‘cut and splice’ approach in [Scott ‘67] for the purpose of audio data compression, whereby the input signal is once again segmented into non-overlapping frames, however, instead of simply discarding every odd (or even) frame, the odd frames (or even) are presented to the left ear and the even (or odd) frames are presented to the right ear. This approach is known as ‘dichotic speech-time compression’ and [Arons ‘92] notes that “Listeners reported a switching of attention between ears, but they quickly adjusted to this unusual sensation”. Arons also notes that most test subjects preferred the dichotic approach to the standard cut and splice approach.

Whilst the process described above is efficient and relatively straightforward to implement, it does, however, introduce artefacts into the time-scaled output. These artefacts are the result of discontinuities and pitch distortions [Lee ‘72], and their origins can be understood by considering the example shown in Figure 2-2. As can be seen from the figure, the simple repetition of a frame can result in a discontinuity in the synthesised waveform together with some distortion of the pitch, which results in objectionable artefacts being perceived. One method of reducing the effects of the discontinuity is to gradually cross-fade segments together rather than simply appending synthesis frames in a hard splice manner [Gabor ‘44], [Lee ‘72], however this technique has a limited effect, since pitch distortions remain, suggesting that more intelligent methods of frame repetition are required.

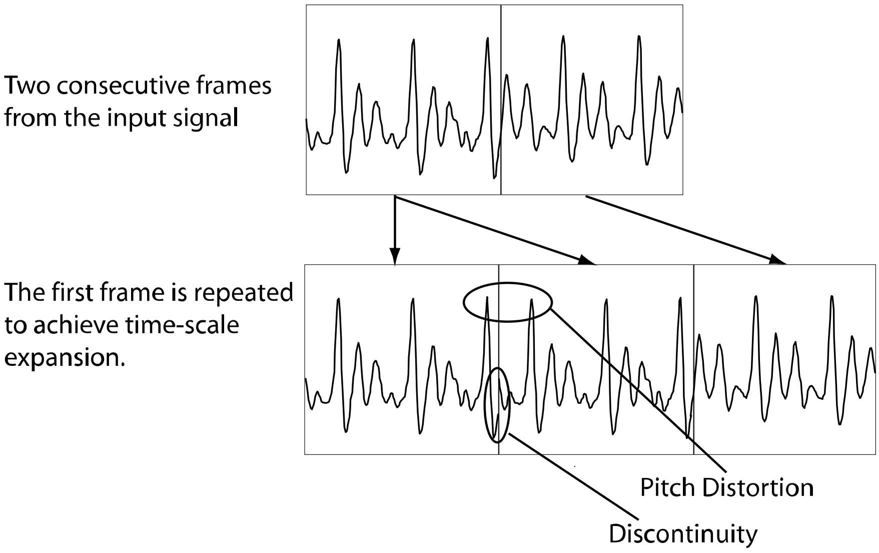

Two consecutive frames from the input signal
The first frame is repeated to achieve time-scale expansion.
Pitch Distortion
Discontinuity

Figure 2-2 Artefacts arising from ‘cut and splice’ implementations.))

A solution to these problems is proposed in [Roucos ‘85], whereby the artefacts introduced by discarding/repeating frames are significantly reduced by overlapping synthesis frames in regions of similarity. As an example, consider the case illustrated in Figure 2-3, i.e. a re-examination of Figure 2-2; by overlapping the repeating frame in a synchronous manner i.e. in a region where the frames are similar, the effects of discontinuities and pitch distortion are removed. This process essentially equates to discarding/repeating segments of the input that are integer multiples of the local pitch period.

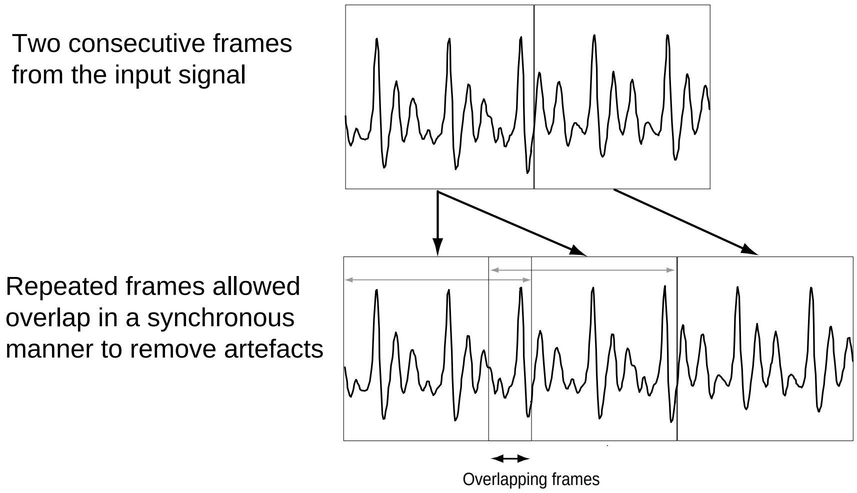

Two consecutive frames from the input signal
Repeated frames allowed overlap in a synchronous manner to remove artefacts
Overlapping frames

Figure 2-3 Synchronised overlap removes artefacts associated with 'cut and splice' methods.))

Work prior to [Roucos '85] suggests a similar solution. The earliest referenced work relating to time-scale modification, i.e. [Dudley '38], used a pitch synchronous splicing approach to time-scale compress an audio signal for the purpose of bandwidth reduction during data transmission. Dudley states: "Instead of transmitting a sustained sound directly, then, it would be transmitted as satisfactorily by transmitting a properly chosen small segment of it and then repeating this transmitted segment over and over to get the complete signal. This small segment would be a fundamental period when it exists". For the case where a fundamental period does not exist, e.g. unvoiced regions of speech, Dudley suggests that a mean pitch period be used. Later work in [Scott '67] is often incorrectly attributed as being the pioneering work of pitch synchronous cut and splice. There are limitations with Scott's approach since it requires that the location of the pitch pulses (instants of glottal closure) be determined, which is a much more difficult problem to determining the pitch period [Neuberg '78]. Neuberg showed that only the pitch period is required to determine a suitable splice position, supporting Dudley's approach.

Whilst several pitch synchronous approaches were developed prior to the synchronised overlap-add (SOLA) [Roucos '85] algorithm, it was the first time-domain pitch-synchronous technique that allowed for flexible and robust modifications since there is no requirement for explicit pitch extraction [Arons '92]. The following section provides a closer examination of the SOLA technique together with a brief outline of SOLA variants and alternative time-domain approaches. This is followed by further examination of some subsequent alternative approaches identified as being of particular importance to the time-scaling problem.

2.1.1 SOLA

The SOLA algorithm [Roucos '85] segments the original waveform into overlapping frames of length *N*. Frames are spaced S_a samples apart, where S_a represents the analysis step size. The time-scaled output *y* is synthesised by overlapping successive frames with each frame a distance of $S_s + k_m - k_{m-1}$ samples apart. S_s is the synthesis step size, and is related to S_a by $S_s = \alpha S_a$, where α is the time scaling factor. k_m is a deviation allowance and is chosen so as to allow frames overlap in a synchronous manner. SOLA uses correlation to identify the 'best' point at which to overlap frames and k_m is chosen such that

$$R_m(k) = \frac{\sum_{j=0}^{L_k-1} y(mS_s + k + j)x(mS_a + j)}{\sqrt{\sum_{j=0}^{L_k-1} x^2(mS_a + j)\sum_{j=0}^{L_k-1} y^2(mS_s + k + j)}} \tag{2-1)}$$

is a maximum for $k = k_m$, where *m* represents the m^{th} input frame and L_k is the length of the overlapping region i.e.

$$L_k = N - S_s + k_{m-1} - k \tag{2-2}$$

k_m is in the range $k_{min} \leq k \leq k_{max}$.

$R_m(k)$ is a correlation function that ensures that successive synthesis frames overlap at the 'best' location i.e. that location where the overlapping frames are most similar. Having located the 'best' position at which to overlap, the overlapping regions of the frames are weighted prior to combination, generally using a linear or raised-cosine function. The output is then given by

$$y(mS_s + k_m + j) := (1 - f(j))y(mS_s + k_m + j) + f(j)x(mS_a + j), 0 \leq j \leq L_k - 1 \qquad (2\text{-}3)$$

$$y(mS_s + k_m + j) = x(mS_a + j), L_k \leq j \leq N - 1 \qquad (2\text{-}4)$$

where *:=* in equation 2-3) means 'becomes equal to' and *f(j)* is a weighting function such that $0 \leq f(j) \leq 1$.

A linear weighting function can be expressed, from [Lee '97], as

$$f(j) = 0, j < 0 \qquad (2\text{-}5)$$

$$f(j) = j / (L_k - 1), 0 \leq j \leq L_k - 1 \qquad (2\text{-}6)$$

$$f(j) = 1, j > L_k - 1 \qquad (2\text{-}7)$$

Typically, N is in the range of 20 ms to 30 ms, S_a is in the range of *N/3 to N/2* samples, k_{min} is *-N/2* and k_{max} is *N/2*. [Hardam '90] reports that k_{min} can be set to 0.

Many variations of SOLA exist that suggest various parameter sizes and alternative techniques to determine the optimum synthesis overlap. Makhoul, whose involvement in the development of SOLA is acknowledged in [Roucos '85], in his experiments on the use of SOLA for data compression, suggests that the analysis step size be fixed at *N*/2 for compression and N/2α for expansion in [Makhoul '86]. This parameter setting for S_a is also used in [Tan '00], however, in [Wong '02] S_a is set to *N*/4α for all time-scale factors.

Makhoul also suggests that a minimum synthesis overlap should exist, so that some minimum level of cross-fading will always exist and thereby reducing perceptual distortion; Wong set the minimum overlap to *N*/4 in [Wong '02] and Laroche also provides for a minimum synthesis overlap in his implementation [Laroche ‘93a].

Makhoul notes that both linear and raised-cosine cross-fading functions were used during testing and that it was found that the more complex raised-cosine function offered no advantage and linear cross-fading was used during final testing of the algorithm.

A number of papers propose the use of alternatives to correlation to determine the ‘best’ synthesis overlap position; [Bialick ‘89] uses an average mean difference function (AMDF); [Hardam ‘90] suggests that the denominator of the normalised correlation function need not be calculated when the numerator is negative; Laroche’s approach does not normalise the correlation function in [Laroche ‘93a], significantly reducing the computational load, and suggests the use of the efficient fast Fourier transform (FFT) to calculate the correlation function for further efficiency; [Yim ‘96] and [Crokett ‘03] use a coarse search for the best overlap position followed by a fine local search; [Wong ‘98] and [Lee ‘02] reduce the complexity of the correlation function by quantising the overlapping segments to one bit values only (+1 for positive values and -1 for negative, see section 2.1.3 for more detail), therefore only +1 and -1 are used in determining the correlation function, providing significant computational savings.

[Hejna ‘90] and [Wong ‘03] show that the optimum overlap can be ‘predicted’ in certain cases, thereby removing the need for determining the correlation function. The basis of ‘overlap prediction’ can be understood by considering the situation illustrated in Figure 2-4; for the $(m-1)^{th}$ iteration an offset of k_{m-1} was determined in the usual manner i.e. through an evaluation of the correlation function described by equation 2-1); however, in the m^{th} iteration it can be seen that a common segment (the shaded regions) within the synthesis frames exists. Maximum correlation will occur for the synthesis overlap at which the common segments are aligned; therefore the synthesis frames will be overlapped at this position.

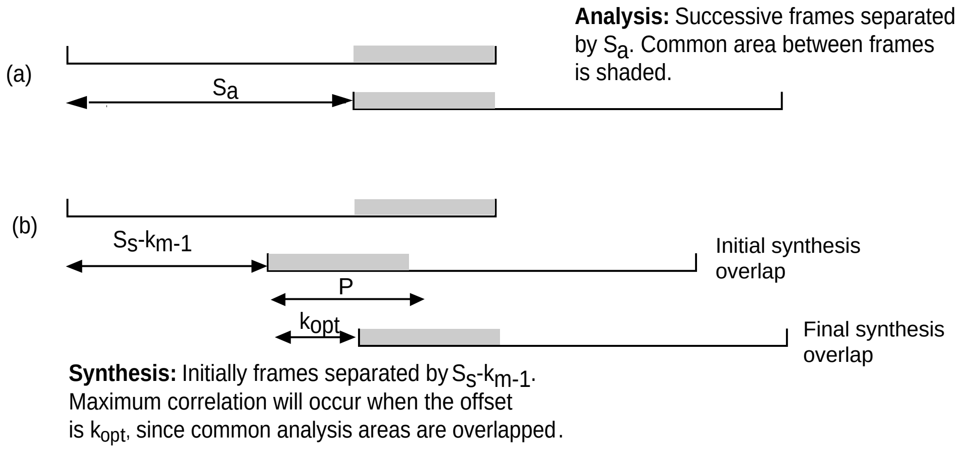

(a)
S_a
Analysis: Successive frames separated by S_a. Common area between frames is shaded.
(b)
S_s-k_{m-1}
Initial synthesis overlap
P
k_{opt}
Final synthesis overlap
Synthesis: Initially frames separated by S_s-k_{m-1}. Maximum correlation will occur when the offset is k_{opt}, since common analysis areas are overlapped.

Figure 2-4 Overlap prediction within the SOLA algorithm.

It can be shown [Wong '03] that this type of situation occurs when

$$k_{min} \le k_{m-1} + S_a - S_s \le k_{max} \tag{2-8}$$

If this condition occurs then there is no need to calculate the correlation function and the offset, k_m, can simply be set equal to

$$k_m = S_a - S_s + k_{m-1} \tag{2-9}$$

Suzuki bypasses the prediction stage by effectively skipping past those analysis frames that have predictable synthesis overlap positions in [Suzuki '92a] and Laroche achieves the same effect by monitoring how close, in duration, the time-scaled output is to the desired time-scaled output after each iteration of the algorithm, and only discards/repeats a segment if the difference between the two durations is greater than a preset threshold of, usually, 40 ms [Laroche '93a]. Laroche notes that a threshold of 40 ms will ensure distortions due to 'time-warping' will not be audible.

In [Hejna '92] an alternative to SOLA, SOLA fixed-synthesis (SOLAFS), is described, in which the synthesis step size is kept constant and the analysis step size is allowed vary. [Verhelst '93] also suggests a similar implementation in the form of a waveform similarity overlap-add algorithm (WSOLA).

Due to the discard/repeat nature of time-domain algorithms, they can easily discard or repeat transients within the signal to be time-scaled, resulting in objectionable artefacts being introduced into the time-scaled output. [Lee ‘97], [Wong ‘97] and [Laroche ‘00a] recognised this problem and proposed that transient sections be detected and be translated, without modification, to their time-scaled position, while the non-transient parts of the signal are time-scaled using the standard approach. Wong’s approach is primarily concerned with time-scale compression and also suggests the removal of silence to achieve greater time-scale compression. In an extension to the idea of preserving transients [Crokett ‘03] segments the input into auditory scenes and time-scales each auditory scene independently. Auditory scenes are detected in a similar way to that outlined in [Duxbury ‘03] i.e. by detecting significant changes in the local frequency content of the signal.

[Quatieri ‘92], [Ellis ‘92], [Covell ‘98], [Kapilow ‘99], [Donellan ‘03] and [Demol ‘04] suggest that different regions of the speech input be time-scaled by varying time-scale factors for the purpose of improved naturalness in the time-scaled output; e.g. Covell suggests that consonants be compressed more than vowels.

2.1.2 GLS-TSM

The global and local search time-scale modification (GLS-TSM) algorithm, [Yim ‘96], is an overlap-add technique similar to the SOLA algorithm. The difference lies in the method by which the amount of deviation/tolerance required to locate the optimum point of alignment of each input frame and the current output is calculated. SOLA uses a correlation technique to calculate the amount of deviation, whilst GLS-TSM uses a two-stage approach. The first stage is a preliminary global search, and the second is a refined local search to find the optimum point of alignment.

The global search consists of a search for a deviation, $k_{globalmin}$, which is chosen such that the number of zero-crossings in the output waveform and the analysis frame are most similar, within the overlapping region. $k_{globalmin}$ lies between k_{min} and k_{max}, the minimum and maximum amount of deviation allowed, respectively.

Having found $k_{globalmin}$, the next step is to locate the zero crossings on the output waveform, which lies within the global overlapping area, $L_{globalmin}$, and is most similar to $Z_{max\ slope.}$ $Z_{max\ slope}$ is the zero crossing, taken from the analysis frame, that has the largest slope, and is chosen since 'It is observed that a wrong match at a zero cross point with a greater slope has a more pronounced effect than a zero crossing with a smaller slope', [Yim '96].

Yim defines an eleven dimensional feature vector, *f*, which is used to represent local information in the region of a zero cross over point. If a zero crossing occurs between *x*[*i*] and x[*i*+1], as in Figure 2-5 below, the eleven feature vector components are given by:

$f_1 = x[i] - x[i+1]$ $f_7 = |x[i+3]|$

$f_2 = |x[i]|$ $f_8 = (x[i-1] - x[i+1])/2$

$f_3 = |x[i+1]|$ $f_9 = |x[i-1]|$

$f_4 = (x[i] - x[i+2])/2$ $f_{10} = (x[i-2] - x[i+1])/3$

$f_5 = |x[i+2]|$ $f_{11} = |x[i-1]|$

$f_6 = (x[i] - x[i+3])/3$

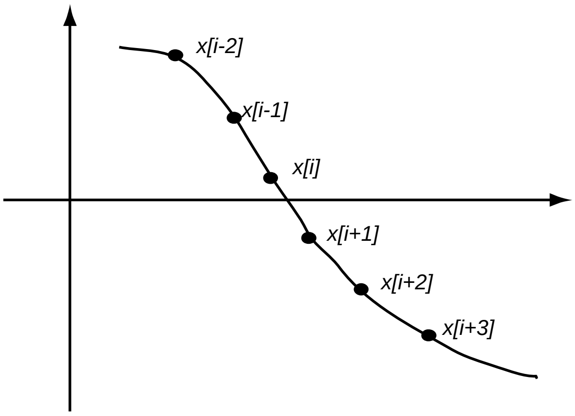


Figure 2-5 Samples for feature vector used to identify a suitable frame overlap position from [Yim '96].

The eleven components were 'determined based on some criteria that we have defined and experimented', [Yim '96]; details of the criteria used are not provided.

The feature vector of $Z_{max\ slope}$ is compared to the feature vectors of the entire zero cross over points of the output waveform that lie within the overlapping region $L_{globaladmin}$, using the following equation:

$$d_i = \frac{1}{11}\sum_{j=1}^{11}\left|f_x[j] - f_{y,i}[j]\right| \tag{2-10}$$

where,

$f_x[j]$ is the j^{th} feature vector component of $Z_{max\ slope}$, the zero crossing with the largest slope in the input frame, $f_{y,i}[j]$ is the j^{th} feature vector component of the i^{th} zero crossing of the output waveform.

The zero crossing of the output waveform that produces the smallest distance measure, d_i, is the most similar to the zero crossing with the largest slope taken from the input waveform. Having found the most similar zero crossing, the output waveform and the input frame are then aligned so that these points overlap.

Yim reports a reduction in the computational requirements of approximately a factor of 40, for a sampling rate of 44.1kHz; details of the criteria used to determine this figure are not provided.

2.1.3 EM-TSM

The envelope matching time-scale modification (EM-TSM) technique [Wong '98], [Wong '03] is a modification of the SOLA algorithm, in which the overlapping synthesis segments are transformed into one-bit functions prior to correlation. In [Wong '03] the following variables are defined

$$x_2(j) = sign(x(mS_a + j)) = \begin{cases} 1 & if\ x(mS_a + j) \geq 0 \\ -1 & if\ x(mS_a + j) < 0 \end{cases} \tag{2-11}$$

$$y_2(j) = sign(y(mS_s + j)) = \begin{cases} 1 & if\ y(mS_s + j) \geq 0 \\ -1 & if\ y(mS_s + j) < 0 \end{cases} \tag{2-12}$$

$$x_{2,k}(j) = x_2(j + k) \tag{2-13}$$

$$y_{2,k}(j) = y_2(j+k) \tag{2-14}$$

where *k* is the offset described in section 2.1.1.

The similarity function given by equation 2-15) is then used to determine a suitable synthesis offset

$$R_m(k) = \frac{\sum_{j=0}^{L_m(k)-1} y_{2,k}(j)x_{2,k}(j)}{L_m(k)} = \frac{\beta_{z,k}}{L_k}\left(2\sum_{j=1}^{M_k+N_k-2r_k}(-1)^{j+1}C_k(j)+(-1)^{M_k+N_k}L_k\right) \tag{2-15}$$

where

$$\beta_{z,k} = y_{2,k}(0)x_{2,k}(0)$$

$$A_k = \{j : x_{2,k}(j-1)x_{2,k}(j) = -1, for\ 0 < j < L_k\}.$$

Therefore, A_k is the set of locations of the zero crossing points in $x_{2,k}$. Also, M_k is the cardinality of A_k i.e. the number of zero crossings in $x_{2,k}$.

$$B_k = \{j : y_{2,k}(j-1)y_{2,k}(j) = -1, for\ 0 < j < L_k\}.$$

Therefore, B_k is the set of locations of the zero crossing points in $y_{2,k}$. Also, N_k is the cardinality of B_k i.e. the number of zero crossings in $y_{2,k}$.

r_k is the number of common zero crossings in $x_{2,k}$ and $y_{2,k}$ i.e. the cardinality of $A_k \cap B_k$.

$C_k = A_k \oplus B_k$, where $\oplus$ is the Exclusive OR operator

Equation 2-15) should be calculated for each offset, *k*, in the range k_{min} to k_{min}, which still represents a significant number of computations [Wong '03]. However, it is shown in [Wong '03] that equation 2-15) need only be evaluated for a smaller subset of all possible offsets, K_o, where

$$\begin{aligned} K_o = \{k : A_k \cap B_k \neq \phi\} \cup \{k : b_{k-1,1} = 1 \cup b_{k+1,N_k} = N-1\} \cup \\ \{k : a_{k-1,M_k} = L_{k-1} - 1 \cap L_k < N\} \cup \{k_{\min}, k_{\max}\} \end{aligned} \tag{2-16}$$

where A_k and B_k are defined above and are also represented by

$$A_k = \{a_{k,1}, a_{k,2}, a_{k,3} ......, a_{k,M_k}\} \tag{2-17}$$

$$B_k = \{b_{k,1}, b_{k,2}, b_{k,3} ......, b_{k,N_k}\} \tag{2-18}$$

Given that $K_o = \{k_1, k_2, k_3, ..., k_{Q+1}\}$, it is also shown in [Wong '03] that $R_m(k_{i+1})$ can be found iteratively from

$$R_m(k_{i+1}) = \frac{L_{k_i}}{L_{k_i} - (k_{i+1} - k_i)} R_m(k_i) + (k_{i+1} - k_i) \frac{2\beta_{z,k_i}\xi_{k_i} + \beta_{z,k_i}(-1)^{M_{k_i}+N_{k_i}+1}}{L_{k_i} - (k_{i+1} - k_i)} \tag{2-19}$$

where $\xi_{k_i} = \sum_{j=1}^{N_k} (-1)^{g(k_i,j)}$ and $g(k_i, j)$ is the location of $b_{k_i,j}$ in the set $C_{k_i,j}$.

Equation 2-19) is significantly less complex than equation 2-15) and equation 2-15) needs only be evaluated for k_1 and all other elements of K_o can be found iteratively and efficiently from equation 2-19).

In [Wong '02] two refinements to the EM-TSM synchronisation procedure of [Wong '98] are proposed. Having obtained the envelope matching function, the offsets that correspond to the M largest magnitudes of the EMF, given by $\{k_{c,1}, k_{c,2}, k_{c,3}, ..., k_{c,M}\}$, are re-evaluated using the following decimated normalised correlation function

$$R_{m,2}(k) = \frac{\sum_{j=0}^{\frac{L_k}{q}-1} y(mS_s + k + q.j)x(mS_a + q.j)}{\sqrt{\sum_{j=0}^{\frac{L_k}{q}-1} x^2(mS_a + q.j) \sum_{j=0}^{\frac{L_k}{q}-1} y^2(mS_s + k + q.j)}} \tag{2-20}$$

The offset k_m is chosen such that $R_{m,2}(k)$ is a maximum for $k = k_m$ where k is an element of $\{k_{c,1}, k_{c,2}, k_{c,3}, ..., k_{c,M}\}$. By using the multiple candidate re-examination procedure described above, the quality of the output is improved upon over the EM-TSM implementation [Wong '03].

The efficiency of the EM-TSM approach is governed by the number of zero crossings in the overlapping regions of the synthesis frames. In [Wong '03] it is noted that high frequency components and noise introduce many zero crossings, thus increasing the computational requirements of the approach, yet it is the low frequency components that are most important in obtaining a 'good' synthesis overlap[3]. One method used in [Wong '03] of reducing the number of computations within an EM-TSM implementation is to apply the following rule prior to determining $R_m(k)$; if the distance between two adjacent zero-crossing points in A_k or B_k is less than a pre-defined threshold, T_1, the pair is removed from A_k or B_k, respectively.

2.1.4 PSOLA

The pitch synchronous overlap-add (PSOLA) technique, [Moulines '95a], extracts short time segments from the original signal by windowing about analysis pitch marks. For voiced speech, the analysis pitch marks are pitch synchronous and the window length is proportional to an estimate of the local pitch period (typically twice the local pitch period). For unvoiced speech the pitch marks are set at a constant time interval and the window length is fixed.

A set of synthesis pitch marks are then computed based upon the desired time-scale modification and the local pitch period. Referring to Figure 2-6, a set of synthesis pitch marks have been calculated up to $t_{s,3}$. The corresponding analysis time instant, t_{ai}, is joined to $t_{s,3}$ by the dotted line in the diagram. $t_{ai} = \tau^{-1}.t_{s,3}$, where τ is the time-scale factor. The pitch period, P is calculated by taking the mean pitch period of the analysis waveform at $t_{ai.}$ Once P is calculated $t_{s,4}$ can be found from $t_{s,4} = t_{s,3} + P$ and the process can start again until all the synthesis pitch marks have been calculated.

[3] Since time-domain overlap-add algorithms attempt to discard/repeat integer multiples of pitch periods, which tend to be of relatively low frequency.

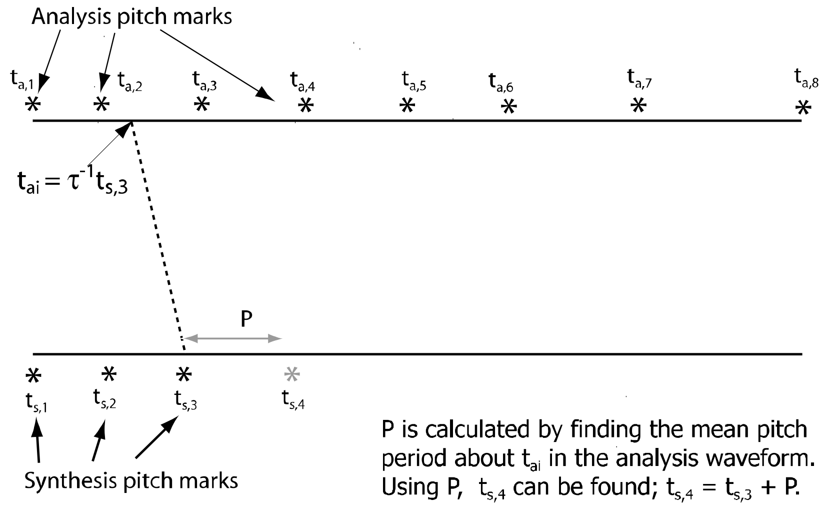

Analysis pitch marks
$t_{a,1}$
$t_{a,2}$
$t_{a,3}$
$t_{a,4}$
$t_{a,5}$
$t_{a,6}$
$t_{a,7}$
$t_{a,8}$
$t_{ai} = \tau^{-1} t_{s,3}$
P
$t_{s,1}$
$t_{s,2}$
$t_{s,3}$
$t_{s,4}$
Synthesis pitch marks
P is calculated by finding the mean pitch period about t_{ai} in the analysis waveform. Using P, $t_{s,4}$ can be found; $t_{s,4} = t_{s,3}$ + P.

Figure 2-6 Calculation of analysis pitch marks within a PSOLA implementation.

Each calculated synthesis pitch mark must have an analysis short-time segment mapped to it. Given a synthesis pitch mark at $t_{s,p}$, the analysis short time segment chosen for mapping is that segment whose centre is closest to $\tau^{-1}t_{s,p}$. Note that the centre of each short-time segment lies on an analysis pitch mark. This process is shown diagrammatically in Figure 2-7. The dotted lines join synthesis pitch marks, $t_{s,p}$, to their corresponding analysis time instants, $\tau^{-1}t_{s,p}$.

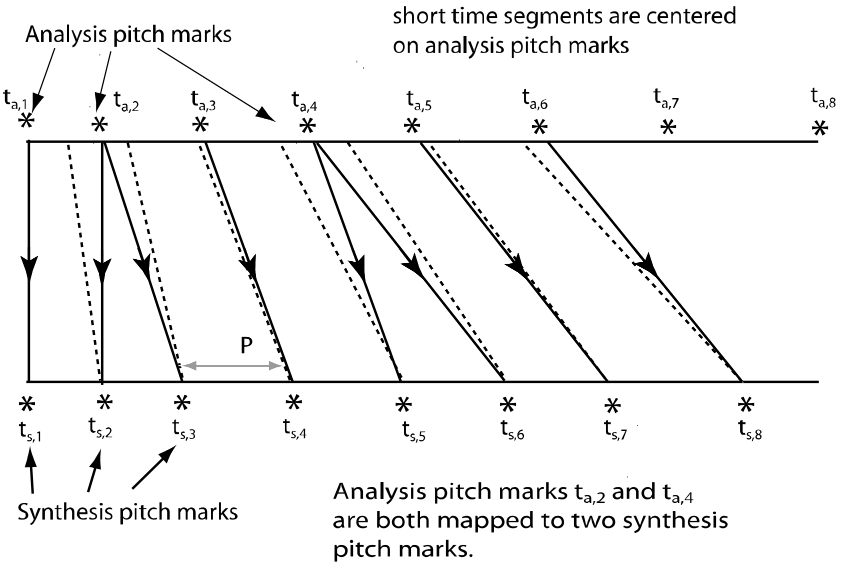


Figure 2-7 Mapping of analysis pitch marks to synthesis pitch marks within a PSOLA implementation.

For time-scale expansion certain short-time segments will be duplicated, and for time-scale compression certain short-time segments will be discarded. In Figure 2-7 the short time segments centred at $t_{a,2}$ and $t_{a,4}$ are repeated to achieve time-scale expansion.

2.1.5 WSOLA

The waveform similarity overlap-add (WSOLA) technique, [Verhelst '93], creates a time-scaled version of the original waveform by continually appending short time segments, typically of 20 ms duration, taken from the original waveform. Referring to the Figure 2-8 below, segment (1), obtained from $x(n)$, is appended to $y(n)$ in an overlap-add fashion during the k-1th iteration of the algorithm, so that segment (1) becomes segment (a). Then a segment is searched for in the region $\tau^{-1}S_k$ in $x(n)$, where τ is the time-scale factor and S_k is the location of the kth synthesis frame i.e. $k.L$, where L is a fixed step typically of 10 ms duration, such that the segment searched for is maximally similar to the segment that follows segment (1) i.e. segment (1'). Having determined the maximally similar frame, i.e. segment (2), it is appended to the output $y(n)$ in an overlap-add manner. The algorithm then continues by searching for a segment that is maximally similar to the segment that follows segment (2) i.e. segment (2'). This process is continued for the complete input waveform.

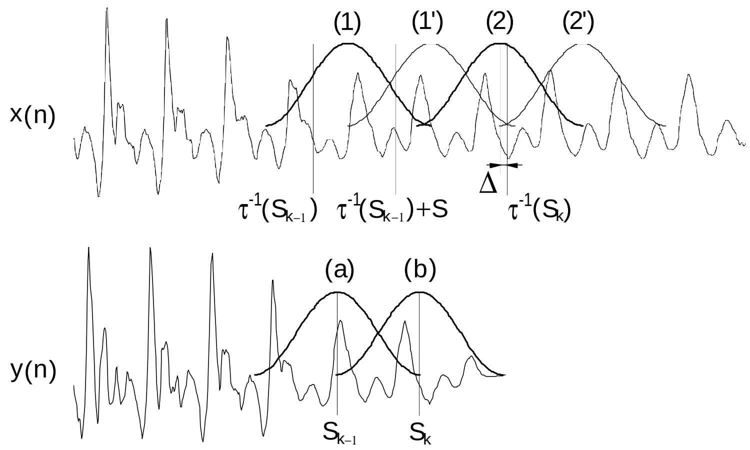


Figure 2-8 Operation of WSOLA from [Verhelst '93].

Three methods of measuring the similarity are presented in [Verhelst '93]. These are:

- A cross correlation coefficient

$$C_c(k,\delta) = \sum_{n=o}^{N-1} x(n + \tau^{-1}((k-1)L) + \Delta_{k-1} + L).x(n + \tau^{-1}(kL) + \delta) \qquad (2\text{-}21)$$

- A normalised cross-correlation coefficient

$$C_n(k,\delta) = \frac{C_c(k,\delta)}{\sqrt{\sum_{n=0}^{N-1} x^2(n + \tau^{-1}(kL) + \delta)}} \qquad (2\text{-}22)$$

- A cross-AMDF coefficient

$$C_A(k,\delta) = \sum_{n=0}^{N-1} \left| x(n + \tau^{-1}((k-1)L) + \Delta_{k-1} + L) - x(n + \tau^{-1}(kL) + \delta) \right| \qquad (2\text{-}23)$$

where

C is the similarity measure that we are attempting to maximise, *x*(*n*) is the input waveform, *k* is the current segment number, *L* is the output segment length, *Δ* is the tolerance/deviation factor, σ is the variable tolerance factor for the k^{th} iteration i.e. $\sigma = \Delta_k$.

Verhelst states: "We found that all tested variants of the algorithm provided similar high quality, from which we concluded that waveform-similarity is a real powerful principle for time-scaling". Verhelst shows, via an illustration, that the choice of similarity measure does not have a significant impact on the location the segment selected; from the illustration, it appears that approximately 375 occurrences from approximately 500 tests, yield the same result when the similarity measures C_A and C_n were applied. Of the results which are not the same, approximately 90% appear to within two samples of each other.

2.1.6 Laroche Implementation

In [Laroche '93a] a method is presented which extends individual frames, as required, in order to achieve the desired time-scale modification of the entire signal.

A frame x, of duration T_a, 40 ms , is extended by first calculating the unbiased autocorrelation function of that frame, using equation 2-24)

$$R(k) = \frac{1}{T_a - k} \sum_{`j=1}^{T_a - k} x(j)x(j+k) \tag{2-24}$$

Equation 2-24) is then determined for k in the range k_{min} to k_{max} , and the value of k for which $R(k)$ is a maximum is used as the offset at which the frame is overlap-added with a copy of itself. For expansion k_{min} and k_{max} are negative; for compression k_{min} and k_{max} are positive. k_{min} and k_{max} are chosen so as to ensure that the minimum segment discard/repeated, L_{min}, is less than 10 ms and that the maximum segment discard/repeated, L_{max}, is 25 ms .

In [Laroche '93a] the input is segmented into non overlapping frames of duration T_a. Each frame is then either appended to the output or expanded/compressed and then appended to the output. The rule to decide whether a frame is to be compressed/expanded can be explained as follows: for an input frame, if the length of the output is above/below a certain tolerance of the expected length of the output then the frame is compressed/expanded. Laroche uses a tolerance of 40 ms .

2.1.7 Suzuki Implementation

In [Suzuki '92a], [Suzuki '92b] two adjacent frames of a fixed length T_a are first extracted from the input x. The biased correlation function of equation 2-25) is then applied for k in the range k_{min} to k_{max}.

$$R(k) = \sum_{m=0}^{T_a - 1} x(i+m+k)x(j+m) \tag{2-25}$$

where i is the start of the first segment and j is the start of the second segment.

The value of k for which $R(k)$ is a maximum, i.e. k_m, is then used as the offset at which the two frames are overlap-added, thus producing a composite signal of length $T_a + k_m$. For time-scaled expansion, a segment, of duration $T_a/(\alpha\text{-}1) + k_m$, that starts at $j + T_a$, is extracted from the input and appended to the overlap-added signal, thereby producing the current time-scaled output. For time-scale compression, a segment of duration (2α-1) T_a /(1- α) – k_m, that starts at j + T_a, is extracted from the input and appended to the overlap-added signal. The input pointers, i and j, are then updated as follows and the process described above is repeated until the entire signal has been time-scale expanded:

For time-scale expansion

$$i \leftarrow i + T_a /(\alpha\text{-}1) \tag{2-26}$$

$$j \leftarrow i + \alpha\ T_a /(\alpha\text{-}1) + k_m \tag{2-27}$$

For time-scale compression

$$i\ \leftarrow j + \alpha\ T_a /(\alpha\text{-}1) - k_m \tag{2-28}$$

$$J \leftarrow j +\ T_a /(\alpha\text{-}1) \tag{2-29}$$

Suzuki determined an ‘optimum’ value for T_a through an objective evaluation, in which the power weighted cepstral distance [Nagabuchi ‘88] is applied for T_a in the range 4 ms to 40 ms . Suzuki found that the power weighted cepstral distance is a minimum when T_a is set to 12 ms . Suzuki notes that k_{min} and k_{max} “are determined by the pitch distribution of the speech signal and by an informal listening test”; Suzuki sets k_{min} and k_{max} to -10 ms and +10 ms , respectively.

2.1.8 AOLA

The adaptive overlap-add (AOLA) algorithm, [Lawlor ‘99b], works in the following manner. A window of length *w* is applied to the input waveform to produce a segment as shown in Figure 2-9 (a). The window/segment length is such that the lowest frequency component of the original waveform will have at least two cycles within the window. A duplicate of the segment is created and shifted to the right so that the two peaks are aligned as in Figure 2-9 (b). A weighted addition is performed on the segment and its shifted duplicate to produce a ‘naturally expanded’ waveform as in Figure 2-9 (c). The length of the expanded waveform is *w.ne*, where *ne* is the natural expansion factor.

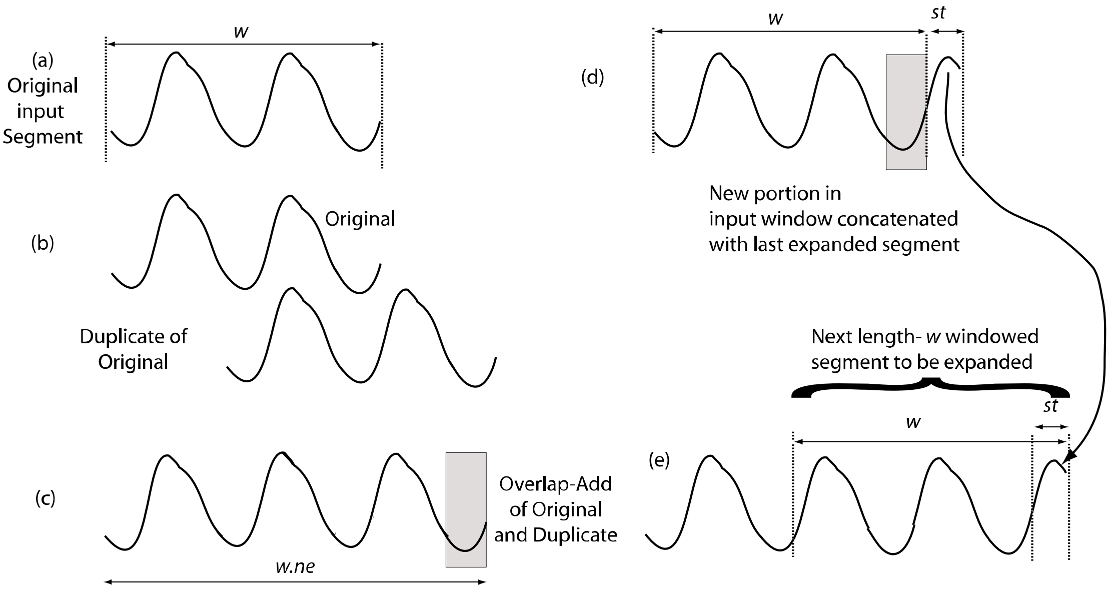


Figure 2-9 Operation of the AOLA algorithm from [Lawlor '99b].

A segment of length *st* is then taken from the original waveform, as illustrated in Figure 2-9 (d). The length of *st* is a function of *w*, *ne* and *de* (the desired expansion factor), and its calculation will be explained later. The segment *st* is then appended to the expanded waveform of Figure 2-9 (c) to produce the waveform in Figure 2-9 (e). The window is then stepped forward by a length *st* to produce a new segment of length *w*, as in Figure 2-9 (e). The same process is then applied to the new segment of length *w*.

It should be noted that the shaded area in Figure 2-9 (c) and the shaded area in Figure 2-9 (d) are the same segment; therefore, the segment *st* appended to the expanded waveform, as in Figure 2-9 (e), will be aligned perfectly i.e. there will be no discontinuities.

To determine the value of *st* another iteration of the above process is performed. The input segment is extracted by a window of length *w*, as shown in Figure 2-9 (e), which includes the appended *st* segment. By performing a weighted addition of this input segment and its shifted duplicate, another naturally expanded waveform is obtained. Assuming that the natural expansion factor is equal to the previous iteration i.e. *ne*, the expanded waveform would again be of length *w.ne.* Part of this waveform contains the *st* segment, which has also been expanded to *st.ne.* Similarly, if a third iteration of the process were applied, one more naturally expanded waveform is obtained, of length *w.ne.* Part of this waveform would also contain an expanded *st* segment, *st.ne.* In addition, this waveform would have an *st* segment that has undergone the iterative process twice, which means that this *st* segment has been multiplied by *ne* twice i.e. $st.ne^2$. If A+1 iterations have been applied, such that the entire input segment is entirely made up of processed *st* segments, the expanded waveform of length *w.ne* would contain an *st* segment expanded *A* times i.e. $st.ne^A$, as in Figure 2-10.

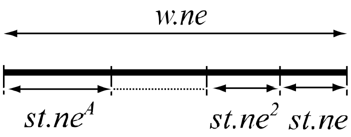


Figure 2-10 AOLA expansion process.

On the next iteration, the $st.ne^A$ segment will leave the expansion process.

Since A+1 is the number of iterations required to ensure the input segment is entirely made up of processed *st* segments, as more iterations are applied a segment of length $st.ne^A$ will continue to leave the expansion process, therefore ne^A should then equal the desired time-scale factor.

From Figure 2-10,

$$st.(ne) + st.(ne)^2 + st.(ne)^3 + st.(ne)^4 + \ldots + st.(ne)^{A-1} + st.(ne)^A = w.ne \qquad (2\text{-}30)$$

Applying a Taylor-series expansion then yields

$$st \approx w(1\text{-}ne)/(1\text{-}\ ne^A) = w(1\text{-}ne)/(1 - de) \qquad (2\text{-}31)$$

where *de* is the desired expansion factor i.e. ne^A.

Since *ne* and *st* are varying continuously, 2-30) and 2-31) are only approximations and both *ne* and *st* must be calculated at each iteration.

To time-scale compress the original waveform the input segment is, again, duplicated and shifted to align peaks (or troughs) but the sections to the left and right of the overlapping region (Figure 2-10 (b)) are discarded to leave a naturally compressed segment. If the input segment has a natural compression factor, *nc*, and the desired compression factor is *dc*, then equation 2-31) becomes

$$st = w(1 - nc)/(1 - dc) \tag{2-32}$$

2.1.9 Time-Domain Conclusion

Time-domain approaches for time-scale modification are appealing due to their low computational requirements and their ability to produce a high quality output for simple monophonic audio, such as speech and monophonic music, for a wide range of time-scaling factors. They are also capable of producing good sound quality when applied to complex polyphonic audio for moderate time-scaling factors (90%-110%). When time-scaling outside the 90%-110% range objectionable artefacts are readily perceived; nevertheless the resulting time-scaled signals are easily recognisable as being a time-scaled version of the original, making them useful for a number of applications outlined in the introduction. The computational efficiency of these approaches makes them particularly suited to devices with limited processing power such as the provision of a speed-up slow-down facility during playback of a dictaphone or a portable music-learning tool.

From the review of time-domain implementations no clear favourite emerges. The majority of recent implementations are based on the SOLA algorithm, however the choice of parameters vary considerably and appear to be chosen on an ad hoc basis; this may be due to SOLA's development having its origins within short time Fourier transform (STFT) theory and its use of typical STFT parameters. Recent 'predictive' variants of SOLA highlight the drawbacks of a poor parameter choice by providing significant computational savings through predictive skipping. Suzuki's implementation determines its analysis parameters though an objective analysis of a number of candidates rather than from an analysis of the algorithm's operation. The AOLA and Laroche implementations are appealing since a thorough explanation of the various parameters employed are given, however their procedures do not provide predictable splice positions, which are required within time-domain/subband implementations for the purpose of subband synchronisation (see sections 2.2 and 6.2); in contrast SOLA based implementations provide predictable splice locations.

In chapter 4 an analysis and explanation as to the motivation behind the choice of SOLA parameters is given within the context of a waveform editing procedure. This analysis results in the derivation of an efficient parameter set for use within SOLA based implementations.

Within time-domain implementations there exist a number of alternative synchronisation procedures; however a comparison of these alternatives in terms of computational requirements and output quality does not exist. Such a comparison is presented in chapter 5.

2.2 Time-Domain/Subband Techniques

Time-domain techniques rely on the existence of a quasi-periodic signal to produce a high quality output. For complex multi-pitched signals such as polyphonic music this requirement is not always satisfied; however, quasi-periodicity does exist at a subband level. [Spleesters '94] recognised this fact and took advantage of it by first partitioning the complex multi-pitched signal into less complex subbands, before applying time-scale modification, using WSOLA [Verhelst '93], to each subband. The resulting time-scaled subbands are then summed to produce a time-scaled version of the original signal. This process is illustrated in Figure 2-11.

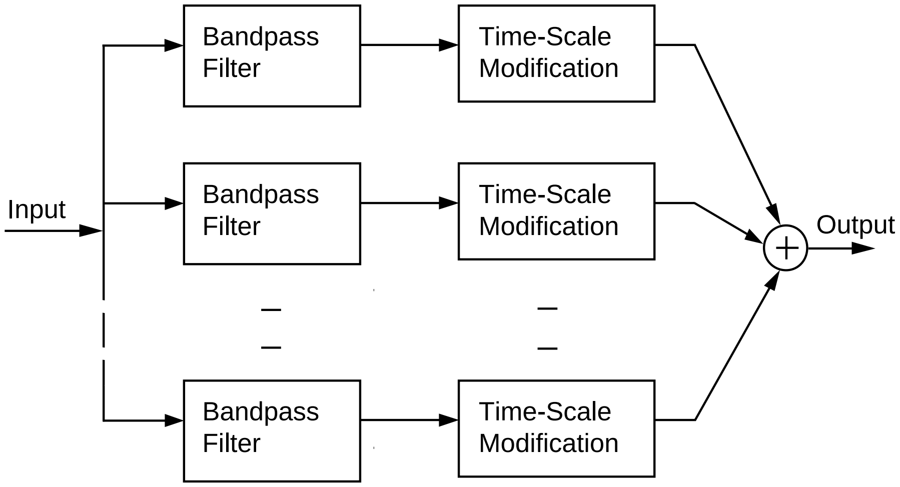


Figure 2-11 Overview of the time-domain/subband approach.

Subband WSOLA partitions audio signals sampled at 10kHz into subbands using a 16-channel, perfect reconstruction, uniform filterbank [Vaidyanathan '92]. The WSOLA algorithm is then applied to each subband using smaller frame lengths for higher frequency subbands. The values of the frame lengths are not given in [Spleesters '94], however in [Verhelst '03], the following recommendations are given:

"For speed-up the use of subbands was not really necessary. A strategy could be to start with 40 ms windows and 20 ms tolerance, and see if results improve if the window length is increased (with the tolerance always 50% of the window length). After this optimisation step, optimise the tolerance (i.e., see if improvements can be obtained by reducing the tolerance). For slow-down, the use of a filterbank is certainly recommended. Start with 16 or 32 channels and 40 ms windows in the low freq channels, 20 ms in the high freq channels. Take a tolerance of 50% of the window length. See if the window lengths need to be increased (first for the low freq channels, then for the high ones). Finally, try to reduce the tolerance in the LF channels and then in the HF channels."

[Spleesters '94] notes that, in regard to the advantages of a subband approach over a time-domain approach (WSOLA), "In general, it appeared that quality improvements could be obtained, but without reaching truly high quality". In [Spleesters '94] an additional experiment is undertaken whereby each subband is delayed by an independent random number of samples before summing all subband signals without time-scaling. It is noted that the distortion introduced is similar to that of the time-scaled signals using a subband implementation and suggests that "the loss of inter-subband synchrony is a probable explanation for the distortions that come with large subband timing tolerances" [Spleesters '94].

The subband analysis synchronised overlap-add algorithm (SASOLA) [Tan '00] differs from [Spleesters '94] in that it is applied to signals sampled at 44.1KHz, applies the SOLA algorithm [Roucos '85] to each subband and uses uniform filterbank of a different passband width. SASOLA partitions broadband audio signals sampled at 44.1 kHz into subbands using a 17-channel cosine-modulated, perfect reconstruction, uniform filterbank [Vaidyanathan '92]. A SOLA based algorithm is then applied to each subband. SASOLA uses a non-standard implementation of SOLA whereby the input is segmented into non-overlapping frames of length N during the analysis stage. The first synthesis frame contains the first αN samples from the first analysis frame (and extends into subsequent input samples in the case of time-scale expansion). The remaining analysis frames are appended to the output by first determining the optimum overlap position (using a cross correlation function) and then applying a linear cross-fade function. The output is then truncated, in the case of time-scale compression, or extended using additional input samples, in the case of time-scale expansion, to $m.\alpha N$, where m represents the m^{th} iteration. In its implementation, SASOLA uses a search range ($k_{max} - k_{min}$) equal to the length of the synthesis frame i.e. αN, and the analysis frame length N is set equal to 40 ms for time-scale factors less than 1, for all subbands, and to 40/α ms for the first subband and 40/(2*α) ms for all other subbands for time-scale factors greater than 1. Tan notes that the analysis frame lengths, and hence the search range, were determined experimentally; details of the experiments undertaken are not provided.

[Covell '01] explains that MPEG layer 2 compression partitions the input into 32 uniform subbands which are subsampled by a factor of 32 and the top two bands are discarded; giving 30 maximally decimated subband streams (MDSS). Covell achieves time-scale scale modification by first applying SOLA to one of the subbands and then using the same offset for all other subbands. Covell notes "For most applications, we simply use the lowest frequency band for this determination. However, if we are working with a band-limited audio input signal (e.g., telephone speech), we instead check to see which MDSS has the maximum energy". Covell also notes that this approach is more efficient than decompressing the MPEG encoded data and time-scaling the uncompressed data; however the quality produced is less than that of SOLA applied to uncompressed audio [Covell '01]. In addition, Covell notes that the output quality is similar to AM radio.

2.2.1 Time-Domain/Subband Conclusion

Time-domain approaches rely on the existence of a strong quasi-periodic signal in order to produce a high quality output for moderate to large time-scale factors, making them generally unsuitable for high fidelity time-scaling of complex polyphonic music in this range. Time-domain/subband approaches address the problems associated with time-domain approaches and operate by first partitioning the complex input into less complex subbands, through the use of an appropriate filterbank. Each subband is then time-scaled using a time-domain approach and the time-scaled subbands are simply summed to produce a time-scaled version of the original signal. Within existing time-domain/subband implementations the approach to partitioning the complex input into less complex subbands appears somewhat ad hoc with little motivation given to the approach taken other than the aim of having subbands of less complexity than the broadband input. Given this objective any partitioning technique would suffice. In chapter 6 the issues involved within a subband implementation are considered and an approach to the partitioning of Western tonal music based on the bark scale is presented. Chapter 6 also presents a further improvement to time-domain/subband implementations by improving synchronisation between the time-scaled subbands.

2.3 Iterative Frequency-Domain Techniques

This section provides a brief overview of the well known discrete Fourier transform (DFT) and short-time Fourier transform (STFT) [Oppenheim '89]. Section 2.3.2 then describes an iterative approach to determine a set of STFT phases for a given magnitude only STFT representation, for the purpose of audio time-scale modification. The DFT and STFT overviews provided in this section are also useful for the remaining sections of this chapter.

2.3.1 The Discrete Fourier Transform

Fourier analysis[4] is based on the fact that any signal can be represented by the sum of properly chosen sinusoidal waveforms [Smith '97]. By performing a Fourier analysis on a time-domain signal the signal's frequency-domain representation, i.e. the magnitude, frequency and phases of the sinusoidal components that sum to create the time-domain signal, can be obtained. In digital signal processing, the version of the Fourier analysis family that is mainly used is known as the discrete Fourier transform (DFT) [Smith '97].

Given a discrete time-domain representation of a signal, $x(n)$, of length N samples, the DFT is given by

$$X(k) = \frac{1}{N}\sum_{n=0}^{N-1} x(n).e^{-j\omega_K n} \tag{2-33}$$

where k is the so-called bin number and ω_k is the frequency of bin k, which is given by $\omega_k = 2\pi k/N$ and $X(k)$ is determined for $0 \leq k \leq N$. It should be noted that the DFT is often written as a function of the bin frequency i.e. $X(\omega)$.

The inverse discrete Fourier transform (IDFT) [Smith '97] is used to recover a time-domain representation of a signal from $X(k)$ and is given by

[4] After the French mathematician and physicist Jean Baptiste Joseph Fourier (1768-1830)

$$x(n) = \sum_{k=0}^{N-1} X(k).e^{j\omega_K n} \tag{2-34}$$

For non-stationary signals, i.e. signals whose frequency content changes over time, a single DFT of the entire signal is not generally sufficient to describe the characteristics of that signal [Oppenhiem '89] and the short-time Fourier transform[5] (STFT) is employed in its place.

Figure 2-12 serves to illustrate the basic operation of an STFT analysis. Waveform (a) is the original audio signal. To analyse the frequency content at the time instant t_1 the audio signal is multiplied by a windowing function, shown in waveform (b), centred at t_1. The result of multiplying the original waveform by the windowing function is shown in (c), i.e. a small portion of the original waveform known as a frame. By applying the DFT to this frame the frequency content of the frame is determined.

If the windowing function is then shifted to the right, to position t_2, and the process outlined above is applied once more, the frequency content of a frame centred at the time instant t_2 would be determined. Similarly, if this process is applied at enough points (normally equally spaced and referred to as the STFT hop size or step size) on the original signal, the result would be a time-frequency representation of the original audio signal i.e. an STFT representation.

Using the notation given in [Griffin '84] the STFT representation of a signal $x(n)$ is formally given by

$$X(mS,\omega) = \sum_{l=-\infty}^{\infty} w(mS-l)x(l).e^{-j\omega l} \tag{2-35}$$

[5] Also referred to as the time-dependent Fourier Transform [Oppenhiem '89]

where S is the analysis step size i.e. separation between analysis time instants, m represents the m^{th} frame at position mS and $\omega = 2\pi k/N$, where N in this case is the length of the window $w(n)$.

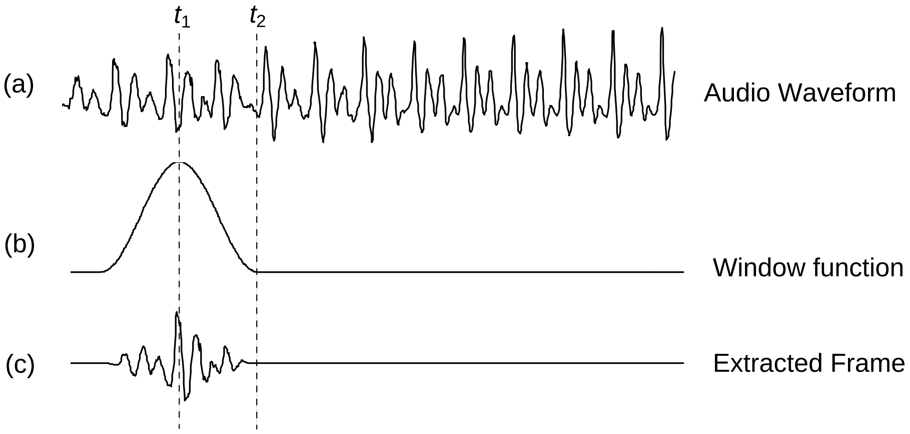


Figure 2-12 The application of a windowing function to an audio waveform.

The length of the window controls both the time and frequency resolutions of the STFT representation. A short window implies good time resolution but poor frequency resolution [Oppenhiem '89]. Conversely, a long window implies good frequency resolution but poor time resolution. Typically, for general audio, the duration of the analysis window is approximately 45-60 ms; shorter windows are typically used for speech.

2.3.2 Signal Estimation from a Modified Short Time Fourier Transform

In [Griffin '84] it is explained that an arbitrary STFT representation, $Y(mS, \omega)$ is not necessarily a valid STFT in the sense that there is not generally a vector sequence whose STFT is given by $Y(mS, \omega)$. In other words, if $x(n)$ is obtained by finding the inverse STFT of $Y(mS, \omega)$, then the STFT of $x(n)$, $X(mS, \omega)$, will not equal $Y(mS, \omega)$ if $Y(mS, \omega)$ is not a valid STFT representation. Conversely, if $Y(mS, \omega)$ is a valid STFT representation $X(mS, \omega) = Y(mS, \omega)$.

The motivation behind the signal estimation techniques described in [Griffin '84] is to determine a method by which a signal, $x(n)$, can be produced from a given $Y(mS,\ \omega)$ such that the resulting STFT representation of $x(n)$, i.e. $X(mS,\ \omega)$, is similar to that of $Y(mS,\ \omega)$ in a squared error sense. In particular [Griffin '84] requires the minimisation of the distance measure given by equation 2-36) to achieve this aim

$$D[x(n), Y(mS,\omega)] = \sum_{m=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} [x_w(mS,l) - y_w(mS,l)]^2 \tag{2-36}$$

where

$$x_w(mS,l) = w(mS-l)x(l), \text{ where } w(n) \text{ is a windowing function.} \tag{2-37}$$

and $y_w(mS,l)$ is the inverse Fourier transform of the m^{th} frame of the STFT representation $Y(mS,\ \omega)$. Griffin notes that the distance measure is given as a function of $x(n)$ and $Y(mS,\ \omega)$ to emphasise that $X(mS,\ \omega)$ is a valid STFT while $Y(mS,\ \omega)$ is not necessarily a valid STFT.

Griffin explains that $D[x(n),\ Y(mS,\ \omega)]$ can be minimised by "setting the gradient with respect to $x(n)$ to zero and solving for $x(n)$" which results in

$$x(n) = \frac{\sum_{m=-\infty}^{\infty} w(mS-n) y_w(mS,n)}{\sum_{m=-\infty}^{\infty} w^2(mS-n)} \tag{2-38}$$

Therefore, by determining $x(n)$ from $Y(mS,\ \omega)$ using equation 2-38) $Y(mS,\ \omega)$ is then most similar to $X(mS,\ \omega)$, in a squared error sense.

Griffin also explains how the result obtained in equation 2-38) provides the basis for iteratively attempting to determine a set of 'valid' phase values for a given magnitude only STFT representation $|Y(mS,\ \omega)|$ of a signal. Given $|Y(mS,\ \omega)|$ and by using any initial phase estimate (for example zeros or random numbers), an initial signal estimate $x^1(n)$ can be found via equation 2-38). In general, after the k^{th} iteration a signal estimate $x^k(n)$ is found. For the k^{th} iteration, the phase values of the resulting STFT representation of $x^k(n)$, i.e $\angle X^k(mS,\ \omega)$, are then used as a new set of estimated phase values for $Y^{k+1}(mS,\ \omega)$ i.e.

$$Y^{k+1}(mS,\omega)=\left|Y(mS,\omega)\right|.e^{j\angle X^k(mS,\omega)} \qquad (2\text{-}39)$$

The inverse of $Y^{k+1}(mS,\ \omega)$ is then found using equation 2-38) giving the k+1th signal estimate, $x^{k+1}(n)$, and the corresponding new phase estimates $\angle X^{k+1}(mS,\ \omega)$. This process is then repeated for as many iterations as is necessary. In [Griffin '84], 25 to 100 iterations are performed to produce a high quality output.

Griffin applies the method outlined above to achieve time-scale modification. In the approach presented, the STFT of the signal to be time-scaled is obtained, with each frame separated by a distance S_a, i.e. S_a is the analysis step size. The STFT magnitudes of the resulting STFT, $Y(mS_a,\ \omega)$, are then used to generate a time-scaled version of the original, however the frames are separated by an amount $S_s = \alpha S_a$, where α is the desired time-scale factor. The update equation used is then given by a modified version of equation 2-38) i.e.

$$x^{k+1}(n)=\frac{\sum_{m=-\infty}^{\infty} w(mS_s-n)\,y_w\left(\left|Y(mS_a,\omega)\right|,\angle X^k(mS_s,\omega)\right)}{\sum_{m=-\infty}^{\infty} w^2(mS_s-n)} \qquad (2\text{-}40)$$

where $y_w\left(\left|Y(mS_a,\omega)\right|,\angle X^k(mS_s,\omega)\right)$ is the inverse Fourier transform of $\left|Y(mS_a,\omega)\right|.e^{j\angle X^k(mS_s,\omega)}$.

In [Irino '92] a similar approach to Griffin's is proposed which makes use of the wavelet transform[6] and in [Abe '89] Griffin's technique is used in conjunction with homomorphic deconvolution [Oppenhiem '89] to achieve high quality pitch-scale modifications for speech signals.

[6] A brief overview of the wavelet transform is given in section 2.4.5.

2.3.3 Iterative Frequency-Domain Conclusion

Iterative frequency-domain approaches attempt to determine a suitable set of phases for a magnitude only STFT representation. The main drawback of the approach is the relatively large computational load required to produce a high quality output. The motivation behind the development of the SOLA algorithm [Roucos '85] was to provide an initial estimate of the phase values so as to reduce the number of iterations required; however, the quality of output produced by using SOLA is very high for the case of speech and no further processing is necessary [Roucos '85]. The possibility of using alternative techniques, such as the phase vocoder, time-domain/subband or sinusoidal modelling, to provide an initial phase estimate for complex signals are topics for further work.

2.4 Phase Vocoder Techniques

The phase vocoder [Flanagan '66] was developed principally as a method for compressing speech prior to transmission. The speech signal is modelled by a set of parameters, i.e. the amplitude and frequencies of the sinusoidal components of the short time segments of the signal, which can be used to reproduce the original signal. This is illustrated in Figure 2-13 below.

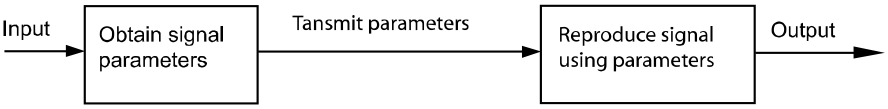


Figure 2-13 Phase vocoder overview.

The input signal is first analysed to determine the signal parameters. The parameters are then transmitted to the receiver, where they are used to reproduce the original signal. Since less information is required to describe the parameters than the signal, compression is achieved.

Flanagan also notes a useful by-product of the phase vocoder, i.e. that of time-scale and/or pitch-scale modification.

The phase vocoder is put into its efficient FFT based form in [Portnoff '76], by making use of the short-time Fourier transform (STFT) [Oppenheim '89]. [Portnoff '81] and [Seneff '82] expanded upon Flanagan's ideas by providing detailed analysis of the phase vocoder for the purpose of time-scale and pitch-scale modifications. A tutorial on the phase vocoder is presented in [Dolson '86]. In addition [Fischman '97] and [De Goetzen '00] provide implementation specific tutorials. In Dolson's tutorial two different views of the phase vocoder are presented; these alternative views are summarised in the following sections (2.4.1 and 2.4.2.).

2.4.1 Filter Bank Interpretation of the Phase Vocoder

As mentioned previously, the phase vocoder operates on the principle that a signal can be modelled by a sum of sinusoids of various amplitudes and frequencies. By determining the time varying amplitudes and frequencies of the sinusoidal components within the signal these parameters can then be used to reproduce the signal [Dolson '86].

Referring to Figure 2-14, the signal is passed through a set of parallel filters. It is assumed that the width of each filter is sufficiently narrow so as to ensure that only one sinusoidal component passes through, and the time varying amplitude and frequency of each sinusoid are determined. The amplitude and frequency parameters are then used to drive a set of oscillators. The sinusoidal outputs of each oscillator are then recombined to produce the original signal.

Dolson notes that at most one sinusoidal component should exist within the passband of any individual filter and that the frequency response of the sum of the filters should be flat.

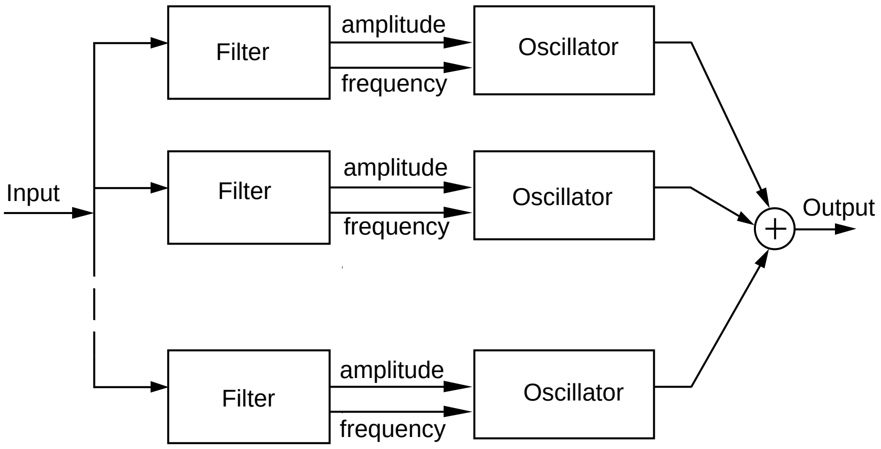


Figure 2-14 Filterbank interpretation of the phase vocoder.

The filter in the diagram has two stages i.e. filtering out a sinusoidal component using a standard bandpass filter and determining the frequency and magnitude of the sinusoidal component. The magnitude and phase are determined with the aid of a process known as heterodyning [Dolson '86]. The time varying frequency parameter is then obtained by subtracting successive values of phase and dividing the result by the duration between calculations of successive phases i.e. frequency = (change in phase)/time. Dolson notes that the calculation of the time varying frequency in this manner must incorporate a technique known as phase unwrapping. Dolson explains this requirement through the use of an example; consider the situation where a sequence of phases such as 180, 225, 270, 315, 0, 45, 90 degrees is obtained. It would appear that the frequency was not always constant, since the difference between successive phase values obtained is not always constant. Unwrapping the phases i.e. adding 360 degrees to the phase value every time the phase has undergone a full cycle resolves this problem. The new sequence of unwrapped phases is then 180, 225, 270, 315, 360, 405, 450 and the phase difference between successive phases is a constant.

Referring to Figure 2-14, Dolson notes that time-scale expansion can be achieved simply by increasing the duration that the parameters are applied to the oscillators. This does not change the local frequency (or amplitude) of the output of the individual oscillators, but does change the duration that particular sine waves exist in the output signal. The result is a time-scale expanded version of the original signal. Time-scale compression can be achieved in a similar manner. In [Flanagan '66] time-scale modification is achieved by first frequency-scaling the output of each filter and then playing the resulting signal at a faster or slower rate. Flanagan achieves frequency-scaling by multiplying the phase output of each filter by the desired frequency-scale factor.

Although the filterbank method is intuitively appealing it is rarely used in practise due its computational overhead. In general the signal's short time Fourier transform (STFT) representation is determined, which is the second view presented in Dolson's tutorial.

2.4.2 Fourier Transform Interpretation of the Phase Vocoder

Dolson notes that the filterbank interpretation is focused on the evolution of the magnitudes and phases of a sinusoidal component that lies within a particular band of frequencies; the Fourier transform interpretation focuses on the magnitudes and phases of all the sinusoidal components at a particular point in time. The magnitudes and phases of the components are obtained by extracting a short time segment from the input though a windowing process and then applying the discrete Fourier transform to the short time segment i.e. by applying the short time Fourier transform (STFT) to the input (see section 2.3.1).

Dolson points out two close relationships between the two alternative viewpoints. Firstly, the number of filters corresponds to the number of frequency bins, the equal spacing between filters corresponds to the equal spacing that exists between the Fourier transform frequency bins. Secondly, the filter shape corresponds to the choice of STFT window. It should also be noted that the precise calculation of the time varying frequency within a bin is also calculated by using successive phase values from that bin. In [Portnoff '76] the mathematical equivalence of the alternative views is proven. The following section outlines the techniques used to alter the time-scale of audio using STFT based implementations of the phase vocoder.

2.4.3 TSM using FFT Phase Vocoder Interpretation

Having obtained the STFT representation of $x(t)$ i.e. $X(t_a^u, \Omega_k)$, using a window of length N and an analysis step size of R_a, where t_a^u is a set of analysis time instants given by uR_a, where u is a sequence of integers that represent the u^{th} analysis frame; Ω_k is the centre frequency of the k^{th} STFT bin/channel and is given by $\Omega_k = 2\pi k / N$. A synthesis STFT, $Y(t_s^u, \Omega_k)$, must then be determined which is used to produce a time-scaled signal, $y(t)$, of the original signal, $x(t)$. t_s^u is a set of synthesis time instants given by uR_s, where u is a sequence of integers that represent the u^{th} synthesis frame [Laroche '99a]. In [Laroche '99a] a synthesis step size of R_s is used, where $R_s = \alpha R_a$, therefore the number of STFT frames is the same during analysis and synthesis but the separation between frames differs.

In [Laroche '99a] the magnitudes of the analysis STFT are used during synthesis i.e. $|Y(t_s^u, \Omega_k)| = |X(t_a^u, \Omega_k)|$. The synthesis STFT phases, $\angle Y(t_s^u, \Omega_k)$, differ from the analysis phases, $\angle X(t_a^u, \Omega_k)$, and are given by

$$\angle Y(t_s^u, \Omega_k) = \angle Y(t_s^{u-1}, \Omega_k) + R_s \hat{\omega}_k(t_a^u) \tag{2-41}$$

where $\hat{\omega}_k(t_a^u)$ is the instantaneous frequency, in radians per sample, of the sinusoidal component within bin *k*.

Equation 2-41) above is known as the phase-propagation formula, and ensures that sinusoidal components within successive synthesis short time segments will overlap coherently i.e. ensures horizontal phase coherence [Laroche '99a]. [Sylvestre '92] offers an intuitively appealing explanation as to the need for a phase propagation formulation; a similar explanation is given here. In Figure 2-15, the sinusoidal waveform of (a) is segmented into two consecutive frames of duration L. Waveform (b) is created by appending shortened versions of the frames in (a). As can be seen a discontinuity is introduced into (b), due to the difference in phase of the consecutive frames. In (c) the phase of the second frame of (b) is updated so that this discontinuity is removed.

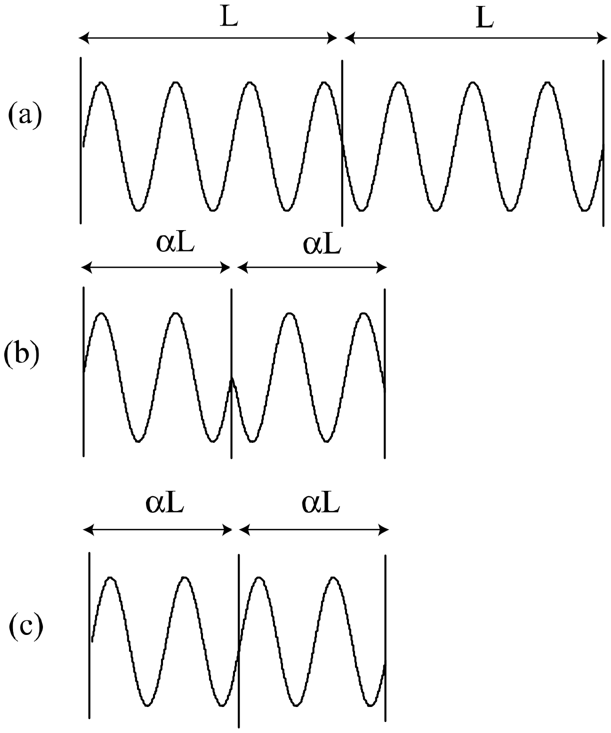


Figure 2-15 Removal of discontinuity using a phase update procedure.

$\hat{\omega}_k(t_a^u)$ in equation 2-41) is calculated from:

$$\hat{\omega}_k(t_a^u) = \Omega_k + \frac{1}{R_a}\Delta_P\Phi_k^u \tag{2-42}$$

where

$\Delta_P\Phi_k^u$ is the principle determination of the phase increment $\Delta\Phi_k^u$ (between ±π)

and

$$\Delta\Phi_k^u = \angle X(t_a^u,\Omega_k) - \angle X(t_a^{u-1},\Omega_k) - R_a\,\Omega_k \tag{2-43}$$

[Laroche ‘99a] develops the phase propagation formula further as follows:

“By iterating ‘the phase propagation formula’ for successive values of *u*, starting at *u* = 0, we obtain

$$\begin{aligned}\angle Y(t_s^u,\Omega_k) &= \angle Y(t_s^0,\Omega_k) + \sum_{i=1}^{u} \mathrm{R_S}\hat{\omega}_k(t_a^i)\\ &= \phi_s(0,k) + \sum_{i=1}^{u} \mathrm{R_S}\hat{\omega}_k(t_a^i)\end{aligned}$$

where $\hat{\omega}_k(t_a^i)$ is the estimated instantaneous frequency at time t_a^i in channel *k*. Now, using ‘ $\hat{\omega}_k(t_a^u) = \Omega_k + \frac{1}{R_a}\Delta_P\Phi_k^u$ ’, we get

$$\angle Y(t_s^u,\Omega_k) = \phi_s(0,k) + \sum_{i=1}^{u}\left[\mathrm{R_S}\Omega_k + \frac{R_s}{R_a}\Delta_p\Phi_k^i\right]$$

and using the definition of $\Delta_P\Phi_k^i$ we get

$$\angle Y(t_s^u,\Omega_k) = \phi_s(0,k) + \alpha\sum_{i=1}^{u}\left[\angle X(t_a^i,\Omega_k) - \angle X(t_a^{i-1},\Omega_k) + 2m_k^i\pi\right]$$

where m_k^i is the unwrapping factor at the analysis time-instant

$t_a^i : 2m_k^i\pi = \Delta_p\Phi_k^i - \Delta\Phi_k^i$. This yields

$$\angle Y(t_s^u,\Omega_k) \quad = \phi_s(0,k) + \alpha\left[\angle X\left(t_a^u,\Omega_k\right) - \angle X\left(0,\Omega_k\right)\right] + \alpha\sum_{i=1}^{u} 2m_k^i\pi \quad \text{”.}$$

Laroche notes that a number of important points can be taken from this final equation:

The initial synthesis phase, $\phi_s(0, k)$, is not a predetermined value, a fact that can be taken advantage of.

For integer time-scale factors, phase unwrapping errors do not have an effect on the synthesis phase. Therefore, for integer time-scale factors the phase unwrapping stage can be skipped, resulting in a significant computational saving.

For non-integer time-scale factors, phase unwrapping errors effect subsequent synthesis frame calculations.

Laroche then considers the case where a signal is being time scaled by an integer value. The synthesis phase at time t_s^u for channel k is given by:

$$\angle Y(t_s^u,\Omega_k) \ = \phi_s(0,k) + \alpha\angle X\left(t_a^u,\Omega_k\right) - \alpha\angle X\left(0,\Omega_k\right) \qquad (2\text{-}44)$$

Laroche then considers the case when a quasi-sinusoidal component whose frequency slowly moves between two adjacent channels k_1 and k_2. The synthesis phases of adjacent channels at particular points in time could vary greatly, depending on the choice of initial synthesis phase. Also, as the sinusoid moves from channel k_1 to k_2 the synthesis phase experiences a phase jump equal to $\theta_{k2} - \theta_{k1}$, where $\theta_k = \phi_s(0, k)$ - $\alpha\angle X\left(0,\Omega_k\right)$.

These two potential problems could be prevented by ensuring that θ_k is a constant i.e. $\theta_k = \phi_s(0, k)$ - $\alpha\angle X\left(0,\Omega_k\right) = C$, where C is a channel independent constant [Laroche ‘99a].

In [Bonada ‘00] the analysis step size is kept equal to the synthesis step size, i.e. R_s =

R_a, and time-scale expansion is achieved by appropriately repeating STFT frames e.g. to time-scale by a factor of 1.5 every second frame is repeated, as illustrated in Figure 2-16; similarly time-scale compression is achieved by omitting frames e.g. to time scale by a factor of 0.9, every tenth analysis frame is omitted. Like traditional implementations of the phase vocoder, the magnitudes of the time-scaled STFT remains unaltered i.e.

$$\left|Y\left(t_s^m, \Omega_k\right)\right| = \left|X\left(t_a^n, \Omega_k\right)\right| \tag{2-45}$$

where n = round(m/α), m is a set of successive integer values starting at 0 and α is the time-scale factor.

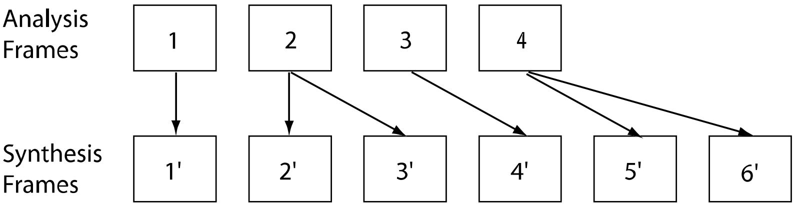


Figure 2-16 Analysis to synthesis frame mapping used in [Bonada '00].

Bonada notes that "since the frame rate is the same for analysis and synthesis then the phase variation between two consecutive frames can be supposed to be the same", which is interpreted to mean that the phase propagation formula becomes

$$\angle Y\left(t_s^m, \Omega_k\right) - \angle Y\left(t_s^{m-1}, \Omega_k\right) = \angle X\left(t_a^n, \Omega_k\right) - \angle X\left(t_a^{n-1}, \Omega_k\right) \tag{2-46}$$

which then becomes

$$\angle Y\left(t_s^m, \Omega_k\right) = \angle Y\left(t_s^{m-1}, \Omega_k\right) + \angle X\left(t_a^n, \Omega_k\right) - \angle X\left(t_a^{n-1}, \Omega_k\right) \tag{2-47}$$

[Puckette '95] uses a similar phase propagation formula as Bonada, but achieves time-scale modification by setting $R_s = \alpha R_a$, as in [Laroche '99a]. Puckette determines an additional set of 'analysis' STFT parameters i.e. at time instants $t'^u_a = t^u_a - R_s$, giving $X(t'^u_a, \Omega_k)$. The following phase propagation formula is then used

$$\angle Y\left(t_s^u, \Omega_k\right) - \angle Y\left(t_s^{u-1}, \Omega_k\right) = \angle X\left(t_a^u, \Omega_k\right) - \angle X\left(t'^u_a, \Omega_k\right) \tag{2-48}$$

which becomes

$$\angle Y\left(t_s^u,\Omega_k\right)=\angle Y\left(t_s^{u-1},\Omega_k\right)+\angle X\left(t_a^u,\Omega_k\right)-\angle X\left(t'^u_a,\Omega_k\right) \tag{2-49}$$

The phase update procedures described above assume that the signal is quasi-stationary; [Bonada ‘00], [Duxbury ‘02] and [Röbel ‘03] note that transients should be handled in a different manner to steady-state regions of the signal, as is the case with time-domain implementations [Lee ‘97]. [Bonada ‘00] suggests that the analysis frames in the region of a transient be neither discarded nor repeated, so that the transient is preserved. Bonada also suggests that the original phases be used for bins above a given cutoff frequency, in the region of a transient, noting that 2500 Hz has proven to be a good selection. For frequencies below the cutoff frequency the phases of bins associated with quasi-stable sinusoidal peaks should be updated using the phase propagation formula and all other bins keep the original phase values. In [Duxbury ‘02] the phases of all bins are locked to the original analysis phases at a transient. Duxbury notes that any artefact introduced by locking phases in this manner is masked by the presence of the transient, rendering them inaudible. As in [Bonada ‘00], Duxbury stretches steady state regions more than transient regions. [Röbel ‘03] uses a measure of the centre of gravity of the instantaneous energy [Cohen ‘95] to detect transients. The measure is applied to each spectral peak and if the result is above a given threshold, the peak is deemed to be associated with a transient and the phase of the bin of the corresponding analysis frame is used.

2.4.4 Phase Locked Vocoder

[Puckette ‘95] explains that when a single sinusoidal component is analysed using the DFT, more than one frequency bin is excited, due to spectral spreading. Puckette goes on to explain that a relationship exists between the phases of the frequency bins that are excited during analysis, and that it is the loss of this phase relationship (referred to as loss of vertical phase coherence in [Laroche ‘99a]), when synthesis STFT parameters are determined, that introduces beating between adjacent channels, which results in a reverberant sound in the time-scaled output.

[Puckette '95] proposes an efficient method of ensuring some level of vertical phase coherence. The phase-propagation formula given by equation 2-41) uses the phase of the previous synthesis frame, $\angle Y(t_s^{u-1}, \Omega_k)$, to calculate the phase of the current frame, $\angle Y(t_s^u, \Omega_k)$, for each frequency bin. Puckette suggests that the phase of the complex number resulting in a combination of the adjacent channels of the previous frame should be used, i.e. $\angle[\,Y(t_s^{u-1}, \Omega_k)$ - $Y(t_s^{u-1}, \Omega_{k-1})$ - $Y(t_s^{u-1}, \Omega_{k+1})\,]$, in place of $\angle Y(t_s^{u-1}, \Omega_k)$, in the phase propagation formula. Puckette notes that the signs of the adjacent channels, *k*-1 and *k*+1, are reversed since the phase of the adjacent channels should be 180 degrees out of phase with the channel centred at *k*.

Puckette's approach has the effect of averaging out the phase. The phase of the channel with the largest magnitude will dominate in the phase propagation formula, essentially locking phases of adjacent channels to that of a peak channel.

In [Di Martino '01] an iterative approach for the time-scale modification of speech is presented that ensures that the phases of the dominant sinusoidal components accumulate coherently at the instants of glottal closure (pitch pulses). Such an approach is also applied in [Quatieri '95a] within the framework of a sinusoidal model.

The phase locking scheme proposed in [Puckett '95] is applied to all channels, resulting in a gradual phase locking process. [Laroche '99a] and [Laroche '99b] describe techniques that use a scheme whereby the channels that contain peaks are used to determine the phase of adjacent channels. In [Laroche '99a], a channel is said to contain a peak if its amplitude is greater than its four nearest neighbours[7]. Channels in the region of a peak are then 'locked' to the phase of the peak channel.

[7] In [Smith '85] it is noted that a hamming window introduces a main lobe width of four bins, which is wider than that of side lobes. Smith notes that any peak narrower than the main lobe width of the analysis window is rejected since it is likely to be introduced by side lobe interference and therefore not a 'true' sinusoidal component. This supports Laroche's choice of using four nearest neighbours.

Laroche notes "In our experiments the upper limit of the region around peak Ω_{kl} was set to the middle frequency between that peak and the next one (Ω_{kl} + Ω_{kl+1})/2. Another reasonable choice would be the channel of the lowest amplitude between the peaks".

Laroche presents two techniques that 'lock' the phases of adjacent channels i.e. identity phase locking and scaled phase locking, which are explained in the following sections.

2.4.4.1 Identity Phase Locking

The identity phase locking technique [Laroche '99a] determines a set of synthesis phases for all channels in the region of a channel containing a peak, k_l, by setting the synthesis phase differences equal to the phase differences between corresponding analysis channels; such an approach is also suggested in [Ferreira '99]. So, if Ω_{kl} is the centre frequency of a dominant peak then

$$\angle Y(t_s^u,\Omega_{kl}) \ - \angle Y(t_s^u,\Omega_k) \ = \angle X(t_a^u,\Omega_{kl}) \ - \angle X(t_a^u,\Omega_k) \qquad (2\text{-}50)$$

for all channels in the region of influence of the channel containing the peak.

Re-written this equation becomes

$$\angle Y(t_s^u,\Omega_k) = \angle Y(t_s^u,\Omega_{kl}) + \angle X(t_a^u,\Omega_k) \ - \angle X(t_a^u,\Omega_{kl}) \qquad (2\text{-}51)$$

Laroche notes that the value of $\angle Y(t_s^u,\Omega_{kl})$- $\angle X(t_a^u,\Omega_{kl})$ need only be calculated once for each peak, and the result is then used for all channels in the region of a peak.

2.4.4.2 Scaled Phase Locking

The phase increment between analysis STFT frames at time t_a^{u-1} and t_a^u for channel k is generally calculated using:

$$\Delta\Phi_k^u = \angle X(t_a^u,\Omega_k) \ - \angle X(t_a^{u-1},\Omega_k) \ - R_a \ \Omega_k \qquad (2\text{-}52)$$

[Laroche '99a] explains that if a peak moves from channel *k₁* to *k₂* at a time t_a^u, the phase increment between analysis STFT frames at times t_a^{u-1} and t_a^u at channel *k₂* should be calculated using:

$$\Delta\Phi_{k_2}^u = \angle X(t_a^u,\Omega_{k_2}) - \angle X(t_a^{u-1},\Omega_{k_1}) - R_a\ \Omega_{k_2} \qquad (2\text{-}53)$$

Also, the phase propagation formula should be given by

$$\angle Y(t_s^u,\Omega_{k_2}) = \angle Y(t_s^{u-1},\Omega_{k_1}) + R_s \hat{\omega}_{k_2}(t_a^u) \qquad (2\text{-}54)$$

To ensure vertical phase coherence, the synthesis phase differences are set equal to the phase differences between corresponding analysis channels scaled by a factor β [Laroche '99a] i.e.

$$\angle Y(t_s^u,\Omega_k) = \angle Y(t_s^u,\Omega_{k_2}) + \beta[\angle X(t_a^u,\Omega_k) - \angle X(t_a^u,\Omega_{k_2})] \qquad (2\text{-}55)$$

β is a phase scaling factor introduced to reduce phasiness. Laroche explains, "Exactly how the phases should be modified upon re-synthesis to ensure vertical phase coherence is not easy to assess. However, it appears that identity phase locking can be further improved by setting β to a value between one and α". Laroche goes on to state "informal listening tests have shown that setting $\beta = 2/3 + \alpha/3$ helps further reduce phasiness".

In the conclusion of [Laroche '99a] it is noted that the quality of output produced by the improved phase vocoder remains inferior to that of time-domain techniques for monophonic pitched signals such as speech. Laroche explains that this is because the magnitudes of the STFT bins should be altered as well as the phases. Laroche supports this statement by way of an example; "In the frequency domain, a chirp sinusoid has a wider centre lobe than a constant-frequency sinusoid. Time-stretching it by a large factor should turn it into a near-constant frequency with a narrower centre lobe." In [Bristow-Johnson '01] the time-scale modification of chirp signals, linearly ramped in amplitude, is explored further and a mathematical description of how the STFT magnitudes (and phases) should be altered is derived.

2.4.5 Multi-resolution Implementations

As noted in section 2.3.1, the STFT frame length controls both the time and frequency resolution of the STFT representation. [Bonada ‘00] notes that if an important low frequency sound is present in an audio signal then a long window is required in order to detect peaks associated with the low frequency component. Bonada also notes that making use of a long window during the time-scale process can add reverberation and smoothness to the sound. To resolve this problem Bonada suggests the use of parallel windowing; “For low frequencies the window should be longer than for high frequencies” [Bonada ‘00]. Figure 2-17 presents the block diagram outlining Bonada’s approach, where Af, SF and Ch are acronyms for analysis frame, synthesis frame and channel, respectively

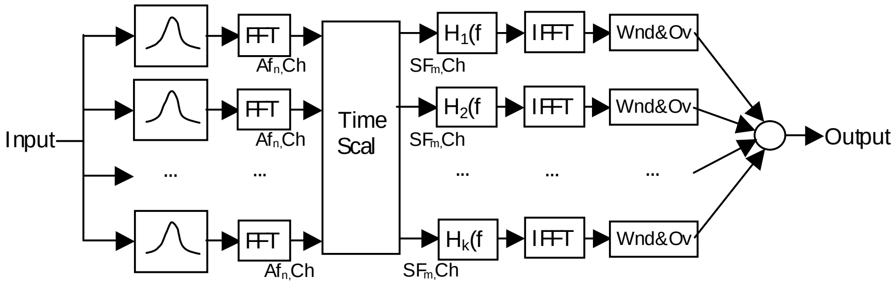


Figure 2-17 Multi-resolution Phase Vocoder block diagram from [Bonada ‘00].

Bonada’s approach equates to partitioning the input into a number of subbands/channels and applying an STFT analysis of a particular resolution to each channel. Each channel can then be time-scaled using a standard phase vocoder approach. Bonada notes that the channel frequency cutoffs should be time-varying to take into account the fact that a spectral peak could occur close to a cutoff frequency; which could result in artefacts if two channels detect the peak. To resolve this potential problem Bonada suggests that the cutoff frequency be updated on a frame by frame basis such that the cutoff frequency lies between two spectral peaks in the proximity of the desired cutoff frequency. Figure 2-18 illustrates this procedure.

In [Bonada ‘00] three channels are used, as a “useful example”, with desired upper cutoff frequencies of 700, 2400 and 22050; with a frame length of 93 ms , 46.5 ms and 34.9 ms , respectively; no justification is given for the choice of three levels of resolution.

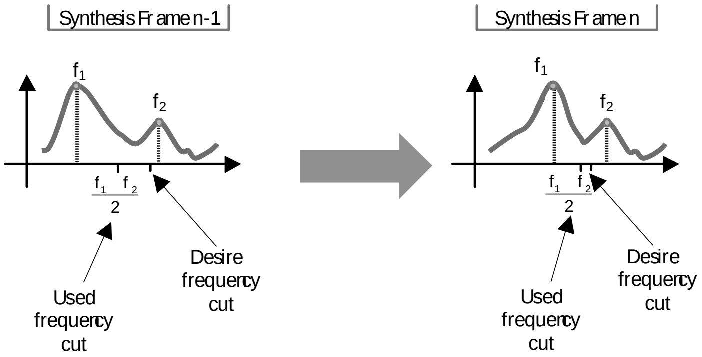


Figure 2-18 Time-varying cutoff frequency within a multi-resolution phase vocoder from [Bonada '00].

[Garas '98] notes that the FFT performs a constant bandwidth spectral analysis on a signal and that it does not take the nonuniform characteristics of the human auditory system into consideration. Garas also notes that [Youngberg '78] reports an improvement in quality when a constant-Q analysis is performed, owing this result to the fact that a constant-Q analysis resembles that of human auditory systems [Zwicker '99]. Garas states that the reason for the unpopularity of the constant-Q phase vocoder is due to its excessive computational complexity.

In [Garas '98] an efficient implementation of a constant-Q phase vocoder is realised by first performing time-warping on the input signal and then calculating the STFT of the time-warped signal.

Garas first explains that a warping of the frequency co-ordinate f to $\gamma(f) = f_0.a^f$ is required, where a controls the bin spacing and the bandwidth on the warped frequency co-ordinate; f_0 is a reference frequency taken to be the smallest frequency of interest. Garas then explains that this frequency warping is equivalent to a time-warping $\gamma(t) = t_0.a^t$ followed by an FFT, where t_0 is an arbitrary time reference and a has the same function as before. Garas demonstrates, through use of an illustration, that "the constant-Q spectrum has a higher resolution at low frequency, while the resolution decreases as the frequency increases".

In [Hoek '01] an approach is presented in which each frequency bin of the analysis STFT is convolved with a filter. Hoek notes that the filter used has the effect of 'blurring' the frequency domain information and "blurring or spreading the frequency domain data corresponds to a narrowing of the equivalent window in the time-domain frame. Therefore each frequency bin of the fast Fourier Transform is effectively calculated as if a time-domain window had been applied before the FFT operation". Hoek uses a frequency variable kernel function which is said to approximate the excitation response of the human cilia located on the basilar membrane [Zwicker '99]. The control function is given by

$$s(f) = 0.4 + 0.26 \arctan(4\ln(0.1f) - 18) \qquad (2\text{-}56)$$

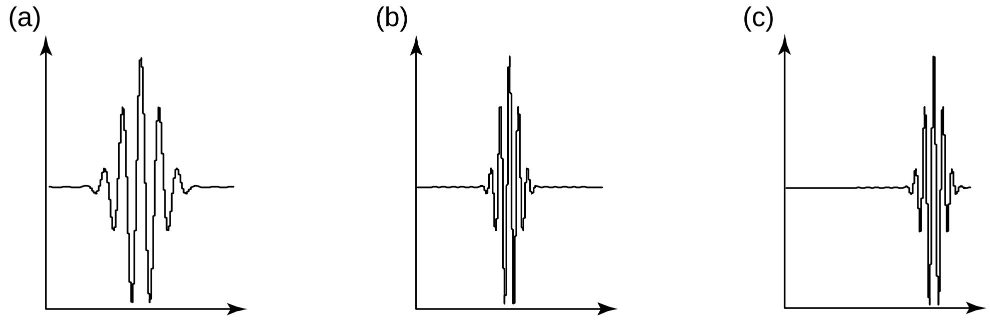
where f is the frequency in hertz.

The kernel function forms part of the filter which is given by

$$y_{out}(f) = [1-s(f)]y_{in}(f) + s(f)y_{out}(f-1) \qquad (2\text{-}57)$$

In [Kronland-Martinet '88], [Gersem '97] and [Pallone '99] the wavelet transform [Vetterli '95] is put forward as a multi-resolution alternative to the standard phase vocoder implementation for time-scaling audio. The following paragraphs present a brief overview of the wavelet transform. A more thorough description can be found in [Vetterli '95] and tutorials are given in [Polikar '05] and [Valens '99].

Fourier analysis essentially measures the similarity of a given signal with a set of sinusoidal basis functions. Wavelet analysis essentially measures the similarity of a given signal with a set of wavelet basis functions (wavelets). The set of wavelets used to analyse the given signal is derived from what is referred to as the mother wavelet. The set of analysis wavelets is obtained from the mother wavelet by dilation (compression) and translation (shifting) of the mother wavelet; see Figure 2-19.



Mother Wavelet

Mother Wavelet compressed

Mother Wavelet compressed and translated

Figure 2-19 Compression and translation of the mother wavelet.

The mother wavelet shown Figure 2-19 is the real part of the Morlet wavelet which, from [Kronland-Martinet ‘88], is given by

$$\psi(t) = e^{j\omega_o t} e^{\frac{-t^2}{2}} \tag{2-58}$$

where ω_o is a frequency parameter which controls the number of cycles in the mother wavelet and t is time.

There are many different wavelets available; however a valid wavelet must adhere to certain constraints and cannot be generated arbitrarily [Kronland-Martinet ‘88].

The set of wavelet basis functions is then given by

$$\psi_{a,b}(t) = \frac{1}{\sqrt{|a|}} \psi\left(\frac{t-b}{a}\right) \tag{2-59}$$

where a (scale) controls the dilation/compression ratio applied to the mother wavelet and b (time) controls the amount of shift/translation.

The continuous wavelet transform (CWT) of a signal $x(t)$ is then given by

$$F(a,b) = \int_{-\infty}^{\infty} x(t) \psi^{*}_{a,b}(t) dt \tag{2-60}$$

The CWT can be interpreted as being the correlation of $x(t)$ with the mother wavelet at an infinite number of compression and translation factors. The original signal can be recovered from the inverse continuous wavelet transform, ICWT, which is given by

$$x(t) = C_{\psi}^{-1} \int_0^{\infty} \frac{da}{a^2} \int_{-\infty}^{\infty} F(a,b) \psi_{a,b}(t) db \tag{2-61}$$

When dealing with discrete signals the smallest usable scale, *a*, is determined by the sampling rate and the largest scale is dependent on the duration of the signal being analysed [Gersem '97]. The *b* parameter is often taken to equal the sampling rate [Gersem '97]. [Valens '99] notes that the when dealing with discrete signals the CWT can be rewritten as

$$\psi_{j,k}(t) = \frac{1}{\sqrt{\left|a_0^{\ j}\right|}} \psi\left(\frac{t - kb_0 a_0^{\ j}}{a_0^{\ j}} \right) \tag{2-62}$$

where *j* and *k* are integers. A common choice for a_0 and b_0 is 2 and 1, respectively [Valens '99]. This scheme is referred to as dyadic sampling [Valens '99].

Parallels have already been drawn between wavelet analysis and Fourier analysis in terms of correlating a set of basis functions with the input signal, and just as a Fourier analysis can be associated with a uniform width filterbank, a wavelet analysis can be associated with a non-uniform filterbank [Valens '99]. The time-domain waveforms of the Morlet wavelet are given in Figure 2-19. Figure 2-20 illustrates the frequency-domain response of a series of wavelets using the dyadic scheme.

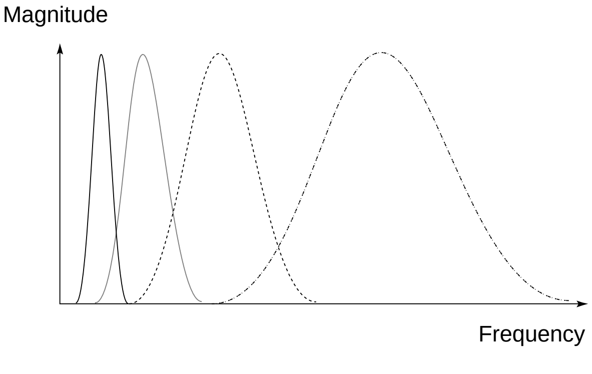


Figure 2-20 Frequency response of Morlet Wavelet at scales 8, 4, 2 and 1 i.e. dyadic scales from [Gersem '97].

The filterbank shown is a constant-Q filterbank and Valens notes that making use of such a filterbank to determine a wavelet representation is an efficient technique to employ.

In [Kronland-Martinet '88] the output of each filter is shown to have a magnitude and phase associated with it, as is the case with the filterbank interpretation of the phase vocoder. Kronland-Martinet notes that the input signal can be frequency-scaled by multiplying the phase value of the output of each filter by the desired frequency-scale factor. As in the case with the filterbank interpretation of the phase vocoder, time-scale modification can then be achieved through resampling.

In [Pallone '99] a non-uniform filterbank based on the bark scale is applied to the input signal, $s(t)$ with the output of each filter being assumed to be a slowly varying sinusoid. Pallone notes that the filter bandwidth is constant at 20 Hz up to 500 Hz and then is proportional to the centre frequency above 500 Hz . Pallone also notes that this analysis is similar to a wavelet analysis but matches the operation of the ear in a closer manner. The output of the i^{th} filter is then given by

$$s_i(t) = M_i(t)e^{j\varphi_i(t)} \tag{2-63}$$

Pallone notes that pitch shifting is achieved by keeping the original magnitude and multiplying the phase by a desired expansion factor. The transposed signal, $s'(t)$, is obtained by summing the real parts of the output of each filter i.e.

$$s'(t) = \sum M_i(t)\cos(\alpha\varphi_i(t)) \tag{2-64}$$

A time-scaled version of the signal can then be synthesised by appropriate resampling of $s'(t)$.

[Pallone '99] and [Kronland-Martinet '88] make use of a filterbank interpretation of the wavelet analysis; In [Gersem '97], as in STFT interpretations of the phase vocoder, the wavelet analysis is performed on a frame by frame basis. Gersem notes "if one wants to raise the pitch of an audio signal by a factor *c*, and therefore divide all the scales in the CWT of the signal by c, and transform the result, one does not get the expected result". Gersem provides the following algorithm steps and notes that "no rigorous mathematical analysis has been done yet, but the procedure delivers acceptable results":

"*coefs* = cwt(*f*, *scales*)

absc = abs(*coefs*)

phac = angle(*coefs*)

phac_unwrap = unwrap(*phac*)

coefs_shifted = *absc*.*exp(i**phac_unwrap***c*)

scales_shifted = *scales/c*

f_shifted = icwt(*coefs_shifted, scales_shifted*)"

As is the case with the STFT interpretation of the phase vocoder a phase unwrapping procedure is employed.

2.4.6 Time-scaling via Spectral Expansion

In [Ferreira '98] and [Ferreira '99] an approach to time-scaling using 'spectral expansion' is presented. The central idea behind the approach is illustrated in Figure 2-21 (A) is the original signal and (B) is obtained from a spectral modification of (A). Ferreira notes that this step represents a delicate operation, since the "frequency and phase relationships among the most relevant components must be preserved in order to maintain the fine temporal structure of the waveform".

The remaining steps, to obtain (C) and (D), are achieved through the use of resampling/interpolation. Ferreira also notes that the interpolation by a factor K, to obtain (D), should be performed before the spectral expansion step in order to avoid reducing the original signal bandwidth.

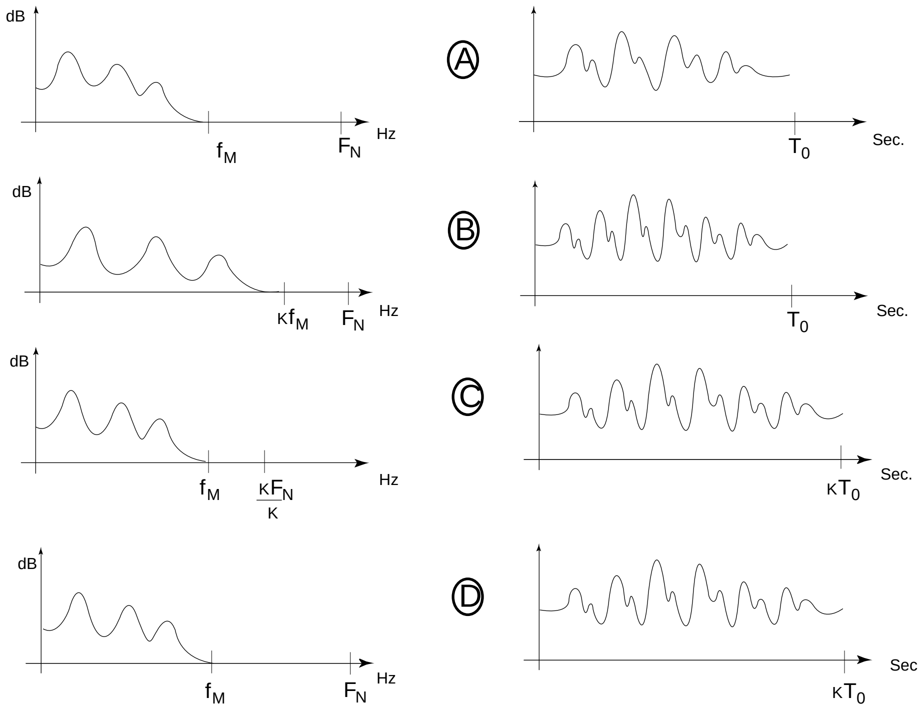


Figure 2-21 Overview of time-scaling by spectral expansion from [Ferreira '99].

Ferreira determines how the original spectrum should be modified by first determining, through mathematical analysis, the 'expected results' when a sinusoid, ramped in amplitude, is synthetically expanded. As in [Laroche '99a] (and sinusoidal modelling, see section 1.1), Ferreira associates spectral peaks within the spectrum with quasi-sinusoidal components and uses the mathematical analysis of a single sinusoidal component to determine the modifications that should be made to each spectral peak and its neighbouring bins. Ferreira shows that the phase difference between neighbouring bins, about a spectral peak, remains unchanged regardless of the phase of the spectral peak, supporting the implementation presented in [Laroche '99a]. Ferreira notes that his approach is limited to time-scaling by integer factors and is designed to be integrated with a perceptual audio coder [Ferreira '96] and is therefore constrained to operate with a filterbank based on the odd discrete Fourier transform [Bellanger '89].

2.4.7 Phase Vocoder Conclusion

Phase vocoder approaches are capable of producing a high quality output for a variety of input signals, including polyphonic music, within a wide range of time-scale factors. Phase vocoder implementations operate by first obtaining an STFT representation of the signal, thus providing the time-varying phase and magnitude values of the 'sinusoidal' components of the signal on a frame-by-frame basis. A time-scaled version of the original signal can be obtained by appropriately modifying the phase, analysis parameters and, for certain implementations, the magnitudes of the STFT. The modified STFT is then inverted to produce a time-scaled version of the original signal.

The phase vocoder time-scaled outputs suffer from an artefact known as phasiness, which sounds similar to a reverberation effect. The cause of this artefact has been attributed to the loss of vertical phase coherence within the modified STFT representation. Recent improvements to the phase vocoder [Laroche '99a] have resulted in a significant reduction in the levels of phasiness introduced into the time-scaled output, however the artefact still remains a problem within phase vocoder implementations. The contributing factors to loss of vertical phase coherence is the inherent time-frequency resolution issues arising from STFT analysis and the difficulty that lies in determining appropriate phase values of STFT components that are 'spread' due to the influence of the windowing function employed. In chapter 7 a phase vocoder implementation is presented that results in a further reduction in phasiness, for moderate time-scale factors, by taking advantage of some flexibility that exists in the choice of phase required so as to maintain horizontal phase coherence between related STFT bins. Furthermore, the approach leads to a reduction in computational load within the range of time-scaling factors for which phasiness is reduced

A number of multi-resolution implementations of the phase vococder exist (see section 2.4.5); a comparison of these approaches is an open issue. In addition an investigation into the use of alternative time-frequency representations, such as the

Wigner distribution [Classen '80] and Cohen class representation [Cohen '95], for the purpose of time-scale modification is also a topic for further work.

The improved phase vocoder [Laroche '99a] can be viewed as a merging of the traditional phase vocoder with sinusoidal modelling techniques (see section 2.5 for an overview of sinusoidal modelling). Improvements to sinusoidal modelling have been suggested in the literature (see section 2.5.1) such as improved peak picking and tracking; the incorporation of these improvements into a phase vocoder based implementation is open to further investigation.

2.5 Sinusoidal Modelling Techniques

Many sounds comprise of a relatively small number of slowly varying dominant sinusoidal components. Sinusoidal modelling techniques [McAulay '86a], [Smith '87] explicitly attempt to identify the dominant sinusoidal components of a signal[8], through a spectral analysis, and use these components to resynthesise a perceptually equivalent version of the analysed signal. The sinusoidal model can also be used to apply modifications to the analysed signal; such modifications include time-scale modification [McAualy '86b]. Formally the sinusoidal model is represented by

$$s(t) = \sum_{i=1}^{L} a_i(t)\cos(\varphi_i(t)) \qquad (2\text{-}65)$$

where a_i and φ_i are the amplitude and phase of the i^{th} dominant sinusoidal component.

It should be noted that the main difference between the approaches of McAulay and Smith is that McAulay's technique is specifically related to speech and separates the excitation source from vocal cavity system when applying modifications[9]; Smith's approach is applicable to more general audio.

In the following sections the sinusoidal model described is based on the more general model of [Smith '87]. It should also be noted that many improvements to the sinusoidal model have been proposed, but the 'classic' model remains the building block upon which modifications are achieved. The focus of this section is to introduce the main features of the sinusoidal model and demonstrate how it can be used to achieve time-scale modification; improvements to the model which are not specifically related to time-scale modification are referred to but not detailed.

[8] This is the key difference between sinusoidal and vocoder models; vocoder models assume the output of each filter is sinusoidal in nature.

[9] [Ninness '00] found that for time-scale modification "no perceivable difference exists between strategies of ignoring the vocal tract response, or accounting for it".

2.5.1 Analysis

2.5.1.1 Peak Identification

The approaches presented in [McAulay '86a] and [Smith '87] both start by obtaining an STFT[10] representation of the signal. Each frame is analysed so as to identify the peaks (local maxima) of the spectrum associated with each STFT frame. The peaks are associated with the dominant sinusoidal components of the signal over the duration of the frame. Some difficulty lies in determining peaks that are related to sinusoidal components as opposed to those spurious peaks that are not [Smith '97]; Smith notes that 'any peak that is substantially narrower than the main-lobe width of the analysis window will be rejected as a local maximum due to side-lobe oscillations'; in [George '92] the influence of side-lobe oscillations is reduced by iteratively removing the peaks identified as being sinusoidal components; [Van Schijndel '03] uses psychoacoustic masking principles to extract perceptually dominant components; [Griffin '85] makes use of a sinusoidal likeness measure which essentially correlates the magnitude frequency response of a synthesised sinusoid with the spectrum to determine if a sinusoidal component is present; In [Levine '98a] a procedure known as the F-Test [Thompson '82], essentially a measure of the ratio of the peak components to the non-harmonic part of the spectrum, is used to determine sinusoidal peaks.

2.5.1.2 Parameter Extraction

Having determined the 'valid sinusoidal' peaks of a frame, the challenge is then to determine the parameters, i.e. the amplitude, frequency and phase, of the sinusoidal component associated with each peak. One technique is to simply use the magnitude, frequency and phase of the frequency bin associated with each peak; however the frequency resolution of the STFT representation is often poor (~10 Hz for a 4096 point DFT applied to a signal sampled at 44.1kHz) [Smith '87]. Smith notes that an

[10] See section 2.3.1 for an overview of the STFT.

improvement in accuracy using this approach could be achieved by increasing the size of the DFT applied in determining the STFT through zero-padding; however, Smith also notes that such an approach is computationally intensive and a more efficient technique employing parabolic interpolation could be employed in its place. Parabolic interpolation can be understood by considering the situation shown in Figure 2-22.

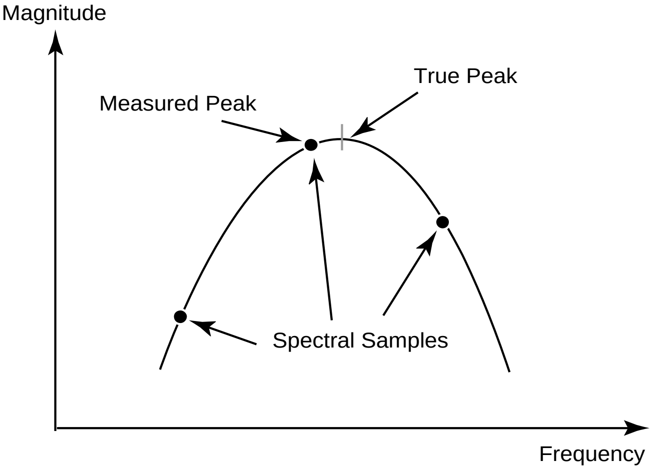


Figure 2-22 Parabolic frequency interpolation to obtain an accurate frequency estimation from a DFT.

Figure 2-22 shows the magnitudes of three spectral samples; the spectral peak between two of its nearest neighbours. The curve passing through each of the samples is a parabola, which Smith shows approximates the actual continuous frequency response of the hamming window in the region of a peak; by finding the maximum of the parabola an accurate estimate of the frequency can be determined. Smith notes that it was empirically found that the frequency estimates tend to be twice as accurate when dB magnitudes are used rather than linear magnitudes. Smith provides the following equation to determine the location of the parabola's maximum along the frequency axis, relative to the position of the measured peak, in bins.

$$p = \frac{1}{2}\frac{\alpha - \gamma}{\alpha - 2\beta + \gamma} \tag{2-66}$$

where β is the magnitude of the bin associated with the spectral peak; α and γ are the magnitudes of the samples to the left and right of spectral peak, respectively.

An estimate of the ‘true’ frequency, in bins, is then obtained from

$$w^* = w_\beta + p \tag{2-67}$$

where w_β is the bin number associated with the peak.

The ‘true’ frequency in radians per sample can then be determined from

$$\omega = w^* 2\pi / N \tag{2-68}$$

The ‘true’ peak amplitude, A, can then be estimated from

$$A = \beta - \frac{1}{4}(\alpha - \gamma)p \tag{2-69}$$

Smith notes that whilst the magnitudes of the three spectral samples are used to determine p, the complex values of the spectral samples can be used in equation 2-69) to determine accurate estimates of the amplitude and phase.

An alternative approach to determining the frequency is to determine the phase derivative from successive analysis frames, as is typically used within phase vocoder implementations [Laroche ‘99a].

2.5.1.3 Peak Continuation

Having determined the amplitude, frequency and phase of the spectral peaks for each STFT frame, the evolution of the sinusoidal tracks is determined by mapping spectral peaks from frame to frame. [Mc Aulay ‘86a] and [Smith ’87] use similar techniques to perform sinusoidal tracking; a summary of Mc Aulay’s approach of matching each frequency track in frame k, ω_n^k, to some frequency in frame k+1, ω_m^{k+1}, is given here:

Step 1:

Having matched peaks of frequencies ω_0^k, ω_1^k,.., ω_{n-1}^k with peaks in frame k+1, a match for ω_n^k is attempted to be found. If no frequency exists in frame k+1 that lies within the 'matching interval' Δ i.e.

$$\left|\omega_n^k - \omega_m^{k+1}\right| \geq \Delta \tag{2-70}$$

for all *m*, then the frequency track associated with ω_n^k is said to be dead. ω_n^k is then matched to itself in frame *k*+1 and is given an amplitude of zero. Step 1 is then repeated for the next frequency ω_{n+1}^k.

However, if a frequency ω_m^{k+1} exists in frame *k*+1 that does lie within the 'matching interval' about ω_n^k and is closest to ω_n^k i.e.

$$\left|\omega_n^k - \omega_m^{k+1}\right| < \left|\omega_n^k - \omega_i^{k+1}\right| < \Delta \tag{2-71}$$

for all $i \neq m$, then ω_m^{k+1} is a tentative match for ω_n^k. It is only a tentative match since there may yet be a better match. A definitive match is found in step 2.

Step 2:

Having made a tentative match, in step 1, to frequency ω_m^{k+1}, then determine if a better match for ω_m^{k+1} exists for all the other unmatched frequencies in frame *k*. If a better match is found then one of two possible cases occur. First, if no other frequency exists in frame *k*+1 that lies within the 'matching interval' about ω_n^k, then the frequency track associated with ω_n^k in frame *k* is declared 'dead' and is matched to itself in frame *k*+1 and is given an amplitude of zero. Second, if a frequency component in frame *k*+1 does exist within the 'matching interval' about ω_n^k, then a definitive match is made.

After either case step 1 is repeated. McAulay notes 'that many other situations are possible in this step, but to keep the tracker alternatives as simple as possible, only the two cases discussed were implemented'.

Step 3:

When all the frequencies in frame *k* are either matched to a frequency in frame *k*+1 or declared 'dead', there may be some frequencies in frame *k*+1 that have no match in frame *k*. If frequency ω_m^{k+1} is such a frequency, it is 'born' in frame *k* with a frequency of ω_m^{k+1} and amplitude zero. The frequency that is 'born' in frame *k* is then matched to the frequency ω_m^{k+1} in frame *k*+1.

[Smith '87] notes that a useful approach is to 'work backwards' in determining sinusoidal tracks i.e. start at the end of the signal and determine sinusoidal tracks moving frame by frame towards the start of the signal. This approach has the advantage of determining the steady-state tracks first and continuing the steady-state tracks into transient regions of the signal, since transients are generally associated with note onsets [Smith '87].

In [Depalle '93a], [Depalle '93b] a statistical approach, based on Hidden Markov Models (HMM), is employed to track the evolution of sinusoidal tracks. In [Rodet '97] it is noted that the tracking approach of [Smith '87] and [Mc Aulay '86a] "work well enough for some categories of sounds (harmonic, voiced, and slow time-varying sounds), but fails in the presence of multiple harmonic structures, inharmonic partials, crossing partials, voiced/unvoiced transitions, and large frequency variations". Rodet notes that the HMM approach copes well with these problems.

In [Lagrange '03] and [Lagrange '04] linear prediction is used to improve partial tracking. In the conclusion of [Lagrange '04] it is noted that the approach is yet to be compared with the HMM technique. In [Raspaud '04], a co-author of [Lagrange '04] notes that the use of linear prediction adds high frequency noise, and HMM may be investigated as an alternative.

In [Virtanen '00] the similarity of spectral amplitudes and continuation of phases are taken into consideration, in addition to the similarity of frequency, when forming sinusoidal tracks, resulting in an improvement in the tracking procedure.

2.5.2 Synthesis

Before proceeding with a description of the synthesis stage, it is worth noting that the analysis stage requires manual adjustment of analysis parameters in order to obtain a useful sinusoidal representation [Serra '90], [Kahrs '01].

[McAulay '86a] notes that since the amplitudes, frequencies and phases are obtained on a frame by frame basis, it might seem reasonable to estimate a segment of the original waveform by generating the synthetic waveform from

$$s(n) = \sum_{l=1}^{L(k)} A_l^k \cos\left(n\omega_l^k + \theta_l^k\right) \tag{2-72}$$

where A_l^k, ω_l^k and θ_l^k are the amplitude, frequency and phase of the l^{th} sinusoidal component in the k^{th} frame; and n is in the range $0 \le n < S$, where S is the length of the frame.

McAualy notes that such an approach generates discontinuities at the frame boundaries and that a method should be employed to smoothly interpolate parameters from one frame to the next. McAulay also notes that the most straightforward technique to resolve this problem is to weight each synthesis frame and perform an overlap and add procedure on successive synthesis frames; however McAulay found that such an approach resulted in a poor quality output.

To overcome this problem McAulay suggests that parameters be explicitly interpolated from frame to frame. Interpolating amplitude values is achieved through linear interpolation i.e.

$$A(n) = A^k + \frac{\left(A^{k+1} - A^k\right)}{R} n \tag{2-73}$$

for n in the range $0 \le n < R$, where R is the synthesis hop size, which is equal to the analysis hop size when no modifications are applied.

McAulay notes that frequency and phase are closely related and a more complicated cubic interpolation must be performed in order to determine the phase values that satisfy both phase and frequency constraints between successive frames. McAulay develops the interpolation function in [McAulay '86a] and the results are reproduced here:

$$\varphi(n) = \theta^k + \omega^k n + \alpha(M^*)n^2 + \beta(M^*)n^3 \tag{2-74}$$

for n in the range $0 \le n < R$, where

$$\alpha(M) = \frac{3}{R^2}\left(\theta^{k+1} - \theta^k - \omega^k R + 2\pi M\right) - \frac{1}{R}\left(\omega^{k+1} - \omega^k\right) \tag{2-75}$$

$$\beta(M) = \frac{-2}{R^3}\left(\theta^{k+1} - \theta^k - \omega^k R + 2\pi M\right) - \frac{1}{R^2}\left(\omega^{k+1} - \omega^k\right) \tag{2-76}$$

McAulay notes that M^* is an integer that ensures that the interpolating phase function is "maximally smooth", and is that integer value that is closest to x, where

$$x = \frac{1}{2\pi}\left[\left(\theta^k + \omega^k R - \theta^{k+1}\right) + \left(\omega^{k+1} - \omega^k\right)\frac{R}{2}\right] \tag{2-77}$$

The synthesis equation then takes on the form

$$s(n) = \sum_{l=1}^{L(k)} A_l^k(n)\cos\left(\varphi_l^k(n)\right) \tag{2-78}$$

where the subscript, l, and the superscript, k, are included to represent the l^{th} sinusoidal track of the k^{th} frame.

In [Smith '87] an implementation is described which does not take the phase values of the sinusoidal peaks into consideration; Smith calls this approach a 'magnitude only' reconstruction. Smith notes that doing this simplifies both analysis and synthesis procedures, since equation 2-69) need only be evaluated for magnitudes and the cubic interpolation function need not be determined during resynthesis; phases can simply be updated so as to maintain phase coherence from sample to sample. Smith notes that discarding the phase information will result in the synthesised signal looking different to the original, but will sound perceptually the same for many applications. [McAulay '86a] also performed 'magnitude only' reconstruction and found that the resulting speech was “very intelligible and free of artefacts” but was perceived as being different from the original speech. McAulay also noted that when ‘magnitude only’ reconstruction was applied to noisy speech, the noise took on a tonal quality that was unnatural and annoying.

In [Rodet ‘92] an efficient synthesis technique is presented, which first generates a set of STFT frames using the parameters obtained during analysis and then determines the inverse STFT efficiently using the FFT. Rodet reports that this technique provides a computational saving of a factor of approximately 15 over the ‘oscillator’ method of resynthesis, as described above. Further work on this approach can be found in [Goodwin ‘94], [Goodwin ‘95] and [Laroche ‘00b].

2.5.3 Time-Scaling using the Sinusoidal Model

Time-scale modification can be achieved using the sinusoidal model from the following equation [Kahrs ’01]

$$s(n) = \sum_{l=1}^{L(k)} A_l^k (n/\alpha)\cos\left(\varphi'^{k}_{l}(n/\alpha)\alpha\right) \qquad (2\text{-}79)$$

where α is the desired time-scale factor, A represents the amplitude and φ' represents the unwrapped time-varying phase of a sinusoidal track; the unwrapped phase can be determined iteratively from frame to frame.

Quatieri notes that whilst the cubic interpolation function used to determine phase values ensures waveform preservation when time-scaling is not applied, a time-scaled synthesis results in a loss of the phase relationship between sinusoidal components; resulting in a reverberant artefact typical of other modification systems[11] as well as ‘magnitude only’ sinusoidal modelling systems. Solutions to this problem are described in section 2.5.5.

[11] It is assumed that Quatieri is referring to phase vocoder based implementations.

For ‘magnitude only’ reconstruction, time-scaling becomes straightforward and simply requires that the synthesis frame separation, *R*, be set equal to the analysis frame separation multiplied by the desired time-scale factor [Smith ‘87].

2.5.4 Sinusoidal plus Stochastic Modelling

[Serra ‘90] addresses the problem of a sum of sinusoids representing noise; the problem being that many sinusoids would be required to accurately model a noisy signal. Serra’s spectral modelling synthesis (SMS) approach assumes that the input signal can be decomposed into a deterministic plus a stochastic component. The deterministic part can be accurately modelled by a sum of sinusoids, whose frequencies and amplitudes vary over time. The stochastic, or noise, component is that part of the signal that cannot be characterised as deterministic. In Serra’s model the input signal, using the notation in [Serra ‘90], is given by

$$s(t) = \sum_{r=1}^{R} A_r(t)\cos[\theta_r(t)] + e(t) \qquad (2\text{-}80)$$

where $A_r(t)$ and $\theta_r(t)$ are the instantaneous amplitude and phase of the r^{th} sinusoidal component. The *R* sinusoidal components are summed to form the deterministic part of the input. The phase $\theta_r(t)$ is given by the integral of the instantaneous frequency plus some fixed initial phase offset

$$\theta_r(t) = \int_0^t \omega_r(\tau)d\tau + \theta_r(0) \qquad (2\text{-}81)$$

where $\omega_r(t)$ is the frequency in radians per second.

The noise component can be modelled as filtered white noise and can be represented as

$$e(t) = \int_0^t h(t,\tau)u(\tau)d\tau \qquad (2\text{-}82)$$

where $u(\tau)$ is white noise and $h(t,\tau)$ is the impulse response of a time varying filter at time *t*.

Figure 2-23 and Figure 2-24 illustrate the analysis and synthesis procedures of the Serra's SMS approach, respectively. During the analysis stage the deterministic component is determined using the same approach as described in previous sections i.e. tracking the spectral peaks of STFT frames. The spectral envelope of Figure 2-23 is obtained by first synthesising the deterministic component. The STFT of the resulting synthesis signal is computed and its magnitude spectrum is then subtracted from the magnitude spectrum of the original signal, resulting in a residual magnitude spectrum. The spectral envelope is obtained by first segmenting the residual magnitude spectrum into segments[12], and then finding the maximum in each segment. Each maximum is then linearly interpolated to produce the final spectral envelope. Serra supports this approach by stating the following: "assuming that the residual is quasi-stochastic, each magnitude spectrum can be approximated by its envelope, since only its shape contributes to the sound characteristics". This is further supported in [Goodwin '96] where it is shown that noise is well modelled by maintaining the original power that existed in Equivalent Rectangular Bandwidth's [Moore '83] (ERB's) of the analysed signal; Serra's approach preserves the power within ERB's[13].

During synthesis, the deterministic and stochastic components are synthesised separately and then summed to synthesise the final output. The deterministic component is synthesised in the same manner as described in section 2.5.2. The stochastic component is synthesised by taking the inverse STFT of a synthesised STFT where the magnitudes of the synthesised STFT are the amplitudes of the spectral envelope of the stochastic component determined during the analysis stage

[12] The length of these segments is not explicitly provided; however, from Serra's illustrations they are equivalent to approximately 100 Hz .

[13] Serra's approach is not the most efficient method of representing noise, in terms of data compression; [Levine '98] takes advantage of Goodwin's representation of noise, together with a sinusoidal model of deterministic components and transient modelling, to develop a data compressed representation of audio that can be readily modified.

and the phases are randomly generated. Serra notes that the result is essentially white noise with its amplitude modified by a time-varying filter. Time-scale modification of the stochastic component is achieved by simply altering the synthesis hop size so that it differs from the analysis hop size [Serra '90].

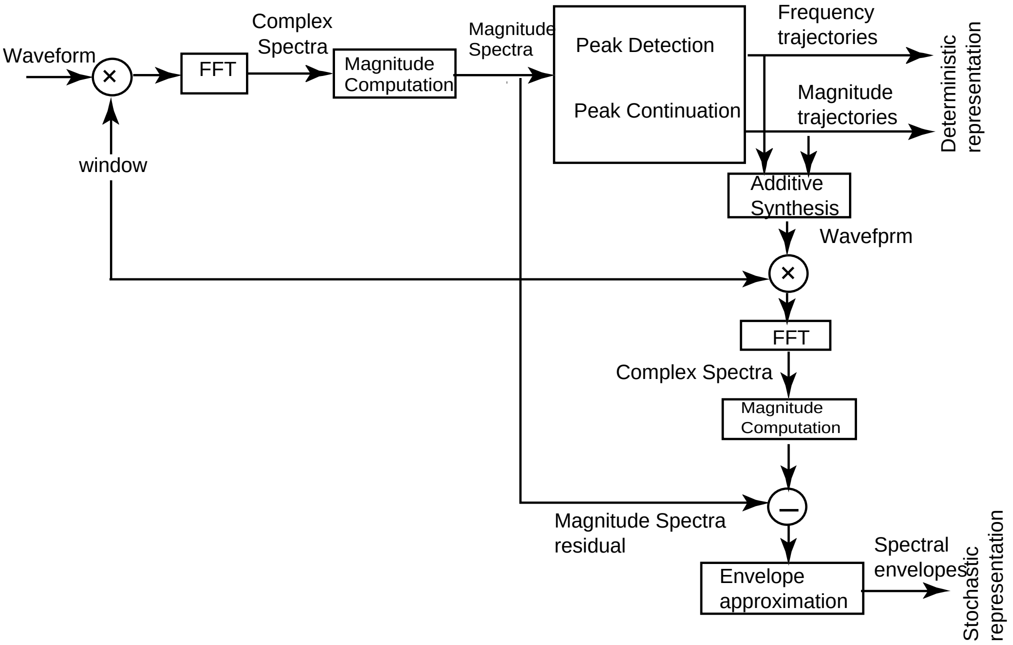


Figure 2-23 Sinusoids plus noise analysis block diagram from [Serra '90].

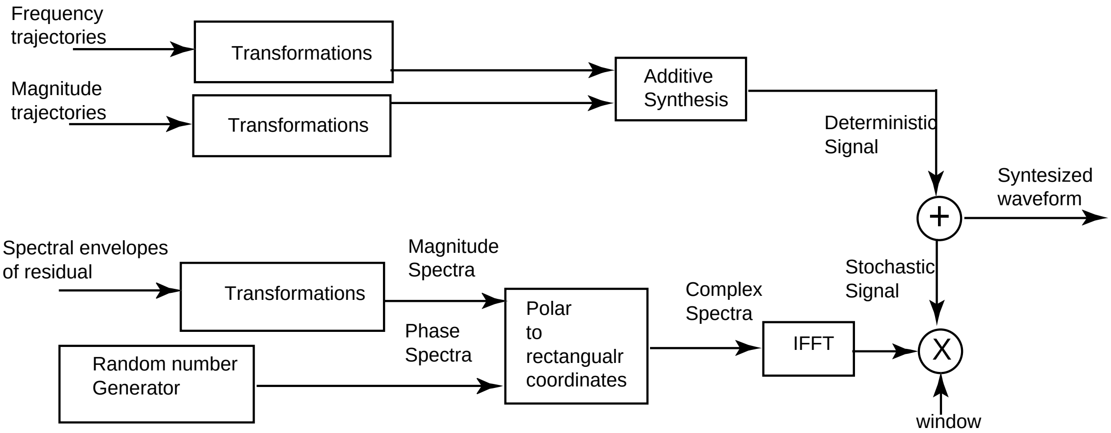


Figure 2-24 Sinusoids plus noise synthesis block diagram from [Serra '90].

[Hanna '03] recognised that Serra's approach to time-scaling the noise-like component of a signal introduces an altering of the variance of the component, thereby introducing audible artefacts into the time-scaled output. Hanna proposes a synthesis method whereby the variance of the time-scaled stochastic component is held equal to that of the original stochastic component in order to remove the artefacts.

In [Goodwin ‘96] it is noted that the sinusoids plus noise representation introduces a ‘muddying’ of sharp transients. In [Masri ‘96], [Verma ’97], [Verma ‘98] and [Hamdy ‘97] this problem is addressed, whereby the audio signal is represented by a transient component in addition to Serra’s sinusoids plus noise representation. For time-scale modifications the transients are simply translated, without modification, to their new positions on the synthesised output, in a similar manner to the time-domain approach of [Lee ‘97].

In [Macon ‘97] it is noted that a tonal character is introduced into the noise-like unvoiced regions of speech when time-scale modification is achieved using the sinusoidal modelling approach. Macon also notes that whilst a sinusoidal plus noise representation offers an improvement in quality “it is more desirable to handle harmonic and stochastic elements within a single unified framework”. To achieve this aim, Macon uses many sinusoidal components to represent the noise-like segments of speech, but during modifications the phases of the sinusoidal components representing the noise-like segments are randomised so as to eliminate the tonal artefact and preserve the noise-like character.

2.5.5 Shape Invariant Techniques

[Quatieri ‘92] notes that time-scaling using sinusoidal modelling techniques introduces a reverberant artefact into the time-scaled output, which Quatieri associates with the inability of the sinusoidal model to preserve the structure of the original waveform due to a loss of phase coherence. Quatieri preserves the waveform structure of speech signals by ensuring that sinusoidal components add coherently at the pitch pulse onset times (instant of glottal closure); one onset per frame is chosen to achieve this aim. Quatieri notes “although any one of the onset times can be used, in the face of computational errors due to discrete Fourier transform (DFT) effects, it may be best to choose the onset time which is nearest the centre of the frame”.

[Pollard '97] presents an approach in which any number of pitch pulse onsets can be used, within a frame, to ensure more accurate shape invariance. In [Quatieri '93a] and [Quatieri '95a] this idea is extended further to incorporate complex acoustic signals; whereby phase relationships are maintained at instants that are associated with distinctive features of the envelope, such as a note onset or transient. This approach is similar to the phase vocoder based approach which preserves transients which is presented in [Duxbury '02].

In [Laroche 93b] and [Laroche '93c] a harmonic + noise model for speech is introduced. Laroche's model is described by

$$s(t) = \sum_{k=-K(t)}^{K(t)} A_k(t)\exp(jkt\omega_0(t)) + e(t) \tag{2-83}$$

where $A_k(t)$ is the complex amplitude (which Laroche notes represents both the amplitude and phase) of the k^{th} harmonic at time t, $w_0(t)$ is the fundamental frequency, $e(t)$ is the stochastic component and $K(t)$ is the time varying number of harmonics. The local pitch, and hence the local fundamental frequency, is obtained through use of a correlation function. The time varying complex amplitude values are then determined via a weighted least squares method [Laroche '93b]; thus yielding the deterministic component of the model. The stochastic component is obtained by simply subtracting the deterministic component from the original signal. Laroche notes that these parameters are determined at pitch synchronous time-instants during voiced regions and every 10 ms during unvoiced regions. Laroche achieves time-scale modification in a manner similar to PSOLA i.e. analysis time instants are mapped to a set of synthesis time instants. Whilst PSOLA extracts short time segments from the input, the method presented in [Laroche '93b] and [Laroche '93c] synthesises short time segments using equation 2-83) together with the parameters determined at each analysis time instant. The result is then multiplied by a hamming window which is twice the duration of the local pitch period and centred at the synthesis time instant. The advantage of the harmonic + noise model over the

PSOLA method is that it is a more flexible representation[14] of the signal [Laroche '93b].

The technique described in [Di Federico '98], [O' Brien '99] [Laroche '03] also makes use of the underlying principle that voiced regions of a speech signal comprise of harmonically related partials. The approach time-scales the fundamental harmonic using the standard sinusoidal modelling approach; however, all other harmonics are updated by maintaining the same 'relative phase delay' (between the fundamental and the harmonic being updated) that existed during analysis. Di Federico describes a phase delay as

$$\tau_{i,k} = \frac{\theta_{i,k}}{\omega_{i,k}} \tag{2-84}$$

where $\omega_{i,k}$ and $\theta_{i,k}$ are the frequency and phase of the i^{th}, harmonically related, sinusoidal track of the k^{th} frame, respectively.

Di Federico then defines the 'relative phase delay' as

$$\Delta\tau_{i,k} = \tau_{i,k} - \tau_{1,k} \tag{2-85}$$

where $\tau_{1,k}$ is the phase delay associated with the fundamental harmonic of the k^{th} frame.

The approach described in [O' Brien '99] operates in a similar manner to that of [Di Federico '98].

2.5.6 Multi Resolution Implementations

[Rodriguez-Hernandez '94] makes use of a dyadic wavelet based analysis, i.e. an octave spaced filterbank, together with sinusoidal modeling, for the purpose of time-scale modification. Sinusoidal modeling is first applied to each subband, with a

[14] Laroche notes that the model is intended to be used for text-to-speech synthesis and high quality timbre modification.

different window being applied to each band[15]; the desired modifications are applied to each sinusoidal model representation before synthesising a time-scaled version of the original signal. [Levine '98a] notes that a wavelet based analysis introduces aliasing between subbands, resulting in a less than high quality output; Levine uses a filterbank which is designed to be alias-free. In addition Levine applies an analysis window of different length to each subband during the sinusoidal model analysis stage. Levine notes that the multi resolution approach reduces the requirement for parameter tracking, resulting in a more robust system. In [Beltran '03] the continuous wavelet transform [Vetterli '95] is obtained and analysed to determine the sinusoidal parameters; as in the case of the classical sinusoidal model, magnitude and phase values of the dominant components are used to determine these parameters.

2.5.7 Sinusoidal Modelling Conclusion

The fundamental operation of sinusoidal modelling relies on the assumption that most audio signals of interest are predominantly composed of slowing varying quasi-sinusoidal components. Sinusoidal modelling techniques use various rules to determine sinusoidal parameters from an STFT representation of the input. Oftentimes these rules may have to be adjusted in order to produce an accurate and *useful* sinusoidal representation of the signal[16].

Sinusoidal modelling has recently been used for modifications in the compressed audio domain, an area of particular interest given the current growth in audio downloads via the Internet and the resulting need to reduce download times through audio compression.

[15] It should be noted that Rodriguez-Hernandez does not explicitly state how the analysis windows differ. It is assumed that shorter windows are used for higher frequencies since Rodriguez-Hernandez states that using different windows provides a better method for handling transients.

[16] the word *useful* is highlighted since the use of a residual in the model can ensure an accurate representation of the original signal, however the representation may not be useful if the dominant sinusoidal components reside in the residual.

A potential improvement to the model is to group sinusoids which are perceptually close to each other; whilst this is effectively achieved through a multi-resolution analysis, further investigation of techniques that focus on psychoacoustic properties on a local frequency basis is required. The benefits of such an approach are two-fold; firstly, further data reduction is achieved, and secondly the phase relationship between perceptually close sinusoids is inherently maintained during modifications.

3 Review Summary and Conclusions

The previous chapter presents a review of approaches used to achieve audio time-scale modification, which is sub-sectioned into time-domain, time-domain/subband, iterative frequency-domain, phase vocoder and sinusoidal modelling approaches. At the end of each sub-section a conclusion is given which provides a brief summary of each approach and suggests some areas for future research. This section provides a brief overview of the advantages and disadvantages of each broad category; this is followed by a discussion which outlines a framework for future work, based on an analysis of the review chapter.

Time-Domain

Advantages

- Efficient in comparison with other approaches.
- Produces a high quality output for simple quasi-periodic signals, such as speech and soloist monophonic music, within a wide range of time-scaling factors.
- Produces a good quality output for complex multi-pitched signals, such as polyphonic music, for small to moderate time-scale factors (90% to 110%).

Disadvantages

- Requires a quasi-periodic input in order to produce a high quality time-scaled output for moderate to large amounts (>110% or < 90%).
- Objectionable artefacts are introduced into the time-scaled output for complex signals time-scaled by moderate to large amounts.

Time-domain/Subband

Advantages

- Produces a high quality output for a variety of signals within a wide range of time-scale factors.

- Can be used to efficiently time-scale MPEG encoded audio.

Disadvantages

- Less efficient than time-domain approaches.
- Introduces a reverberant artefact into the time-scaled output.

Iterative Frequency-Domain

Advantages

- Produces a high quality output for a variety of signals within a wide range of time-scale factors.

Disadvantages

- Computationally intensive compared to other approaches.
- Introduces a reverberant artefact into the time-scaled output.

Phase Vocoder

Advantages

- Produces a high quality output for a variety of signals within a wide range of time-scale factors.
- Computationally efficient in comparison with sinusoidal modelling and iterative frequency-domain approaches.

Disadvantages

- Less efficient than time-domain approaches.
- Introduces a phasiness/reverberant artefact into the time-scaled output.

Sinusoidal Modelling

Advantages

- Capable of producing a high quality output for a variety of signals within a wide range of time-scale factors, when the signal is well modelled.
- Provides a flexible representation that is useful for other modifications e.g. sound source separation.
- Provides a data compressed representation of the signal suitable for transmission and storage on devices with limited memory.

Disadvantages

- Requires accurate modelling of input to produce a high quality output, which, oftentimes, requires parameter adjustment.
- Computationally demanding in comparison with other approaches, with the exception of iterative frequency-domain approaches.
- Introduces a phasiness/reverberant artefact into the time-scaled output.

Based on the overall review of existing approaches it is concluded that time-domain approaches offer the best trade-off in terms of computational load versus output quality for the time-scale modification of single pitched sounds such as speech and monophonic music. For more complex sounds, such as polyphonic music, more sophisticated alternatives are required. Of the suitable alternative approaches, phase vocoder and time-domain/subband approaches are most appealing, due to the lack of robustness of sinusoidal modelling approaches and the computational demands of iterative frequency-domain approaches.

Phase vocoder and time-domain subband approaches are similar; both partition the input into subbands before further processing. Subbands are assumed to be sinusoidal in nature within phase vocoder implementations, whereas time-domain/subband implementations assume the subbands are periodic in nature. Despite their similarities, they operate in a significantly different manner and it is not clear which approach is capable of providing the highest quality output; suggesting the need for further investigation.

Within time-domain/subband implementations the choice of filterbank in the literature appears to be chosen on an ad-hoc basis, suggesting that more intelligent methods are required. Furthermore no attempt is made to synchronise time-scaled subbands; an issue that should be addressed.

Recent improvements to phase vocoder implementations have incorporated features of sinusoidal modelling. Traditional phase vocoder implementations assume that each subband is sinusoidal in nature and the same processing is applied to each subband. However, a significant number of subbands are comprised of interference terms introduced by the filtering process and cannot be considered true sinusoidal components. The improved phase vocoder determines those subbands that contain true sinusoidal components, in a similar manner to sinusoidal modelling approaches, and updates these components in the traditional way; interference components are updated in an attempt to maintain subband synchronisation. This approach results in an improvement in the quality of output produced by the phase vocoder; however the reverberant artefact is still present. This suggests that further analysis of the choice of phase so as to maintain subband synchronisation is required.

Based on the above analysis further work on the phase vocoder and time-domain/subband implementations is proposed, with an emphasis on incorporating aspects of other approaches for improvements. In addition, since a reverberant artefact affects both approaches and time-domain approaches do not suffer from such an artefact, additional focus is directed on incorporating elements of time-domain techniques. Owing to the variations in time-domain analysis and synthesis parameters, chapter 4 presents a unified view of time-domain approaches within a waveform editing procedure. Chapter 5 presents a comparison of a number of synchronisation procedures commonly used within time-domain algorithms. Chapter 6 supports the use of a time-domain/subband implementation based on the bark scale for the time-scale modification of music; in addition, a number of subband synchronisation procedures are presented.

In chapter 7 a method of time-scaling is presented that results in a reduction in phasiness; the technique is a hybrid of time-domain and phase vocoder implementations, which takes advantage of the benefits of both approaches.

4 Analysis of SOLA Parameters

An overview of SOLA is provided in section 2.1.1. An understanding of the effects of the various parameters used in the algorithm is obtained through an examination of some particular situations.

First consider the case where a perfectly periodic signal, of period *P*, is being time-scaled and two frames of the input are being overlapped. In Figure 4-1(a) if the synthesis frames overlap is allowed vary from *P* to 1 samples, the correlation function, graphed to the right of the overlapping frames, produces a single maximum, corresponding to the overlapping region of maximal similarity. Allowing frames overlap in the range *P* to 1 ensures that a maximum correlation will occur, however, consider the case of Figure 4-1 (b); if these frames are allowed overlap from *P* to 1 an unsuitable overlap may be returned due to 'ambiguous' maxima being returned by the correlation function, as shown in the figure. In general, for perfectly periodic signals, to remove the risk of ambiguous results being returned the overlap should be allowed vary from 2*P* to *P*, as demonstrated in Figure 4-1(c). It should be noted, however, that *k* is still in the range $0 \leq k \leq P$.

Typically the voiced/quasi-periodic regions of a speech signal have a waveform similar to that of Figure 4-1 (a) and this constraint can be somewhat relaxed since potential ambiguities generally arise in the 'lower amplitude' section of a period of a typical speech waveform. It should also be noted that the pitch of an audio signal changes frequently and that *P* should be chosen so as to equate to the longest likely pitch period of the signal being analysed (typically 10-12 ms). For these reasons, allowing the synthesis overlap to vary from 3*P*/2 to *P*/2 will produce adequate results for speech signals.

In order to allow the synthesis overlaps vary from 2*P* to *P* (or 3*P*/2 to *P*/2) the difference between k_{max} and k_{min} should be set equal to *P* i.e.

$$k_{max} - k_{min} = P \tag{4-1}$$

If k_{min} is set to zero then k_{max} becomes P.

The next constraint is to ensure the initial synthesis overlap is $2P$ (or $3P/2$ for a more efficient and generally adequate implementation). For convenience the initial synthesis overlap is labelled $L_{max.}$ The length of L_{max} is also constrained by equation 2-2) when k is set to its minimum i.e. 0, therefore

$$L_{max} = N - S_s + k_{m-1} \tag{4-2}$$

If the output, y, is truncated to $mS_s + N$ samples after each iteration then L_{max} becomes independent of the previous synthesis offset and is given by

$$L_{max} = N - S_s \tag{4-3}$$

Truncating the output also has the effect of altering the length of overlap between synthesis frames, which from equation 2-2) then becomes

$$L_k = N - S_s - k \tag{4-4}$$

Since S_s is constrained to be αS_a, there is only one 'unknown' parameter, i.e. S_a, and an analysis similar to that given in [Dorran '03b] is now performed in order to determine a suitable setting for S_a. From the description of SOLA given at the start of section 2.1.1, the distance between successive synthesis frames is $S_s+k_m–k_{m-1}$. The length of the segment discarded/repeated during an iteration of the algorithm is then given by $|S_a - (S_s+k_m–k_{m-1})|$. Consider the case where $k_{m-1} = k_{max} = P$ and $k_m = k_{min} = 0$ i.e. maximum overlap; then a segment of length $|S_a - (S_s - P)|$ is discarded/repeated during the overlap-add process.

For high quality time-scale modification the discarded/repeated segment should be short enough to ensure quasi-stationarity during voiced regions, so

$$|S_a - (S_s - P)| \leq L_{stat} \tag{4-5}$$

where L_{stat} is the duration over which the input is quasi-stationary.

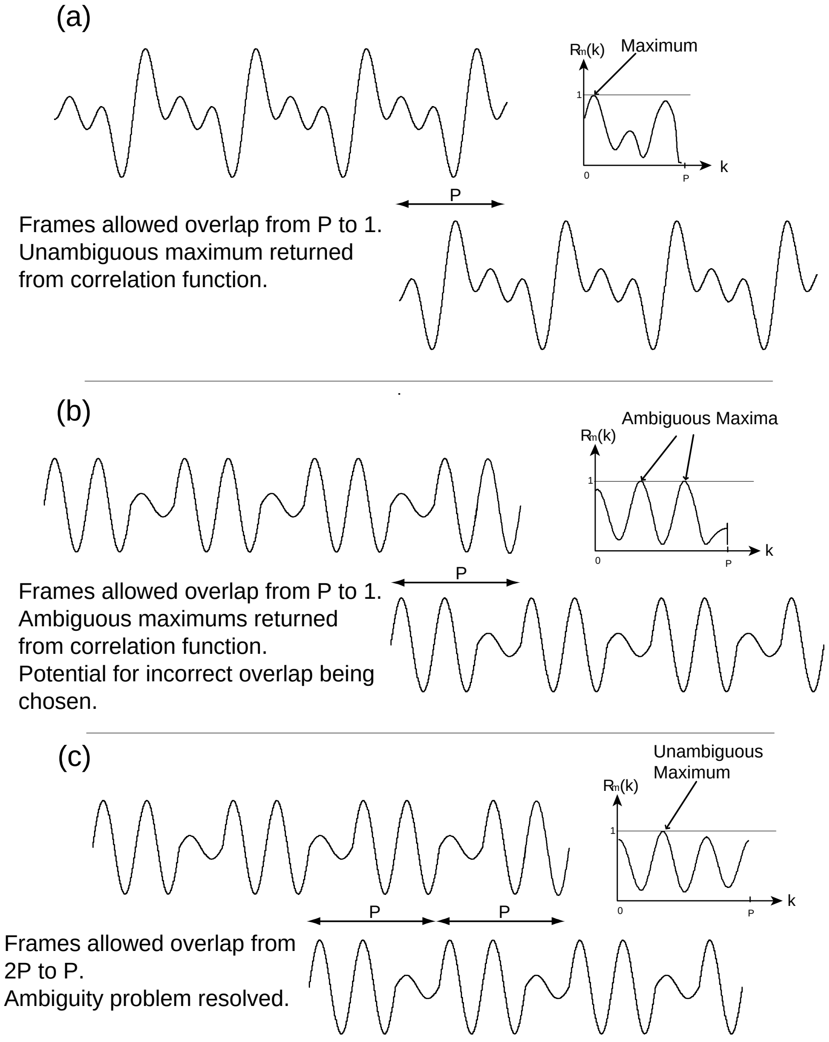

(a)
Rn(k)
Maximum
1
0
P
k
P
Frames allowed overlap from P to 1.
Unambiguous maximum returned
from correlation function.
(b)
Ambiguous Maxima
Rn(k)
1
0
P
k
P
Frames allowed overlap from P to 1.
Ambiguous maximums returned
from correlation function.
Potential for incorrect overlap being
chosen.
(c)
Unambiguous
Maximum
Rn(k)
1
0
P
k
P
P
Frames allowed overlap from
2P to P.
Ambiguity problem resolved.

Figure 4-1 Scenarios involving overlapping synthesis frames.

Since $S_s = \alpha S_a$

$$|(1 - \alpha)S_a + P| \leq L_{stat} \qquad (4\text{-}6)$$

Also, $|(1 - \alpha)S_a + P|$ is a maximum when $\alpha < 1$, therefore equation 4-6) should be satisfied when $\alpha < 1$. Since

$$|(1 - \alpha)S_a + P| = (1 - \alpha)S_a + P \text{ when } \alpha < 1 \qquad (4\text{-}7)$$

then

$$S_a \leq \frac{L_{stat} - P}{1 - \alpha} \quad \text{for } \alpha < 1 \qquad (4\text{-}8)$$

Now consider the case when $k_{m-1} = k_{min} = 0$ and $k_m = k_{max} = P$ i.e. minimum overlap; then a segment of length $|(1 - \alpha)S_a - P|$ is discarded/repeated. As in the case described above, the discarded/repeated segment should be short enough to ensure quasi-stationarity during voiced regions, so

$$|(1 - \alpha)S_a - P| \le L_{stat} \qquad (4\text{-}9)$$

$|(1 - \alpha)S_a - P|$ is a maximum when $\alpha > 1$, therefore equation 4-9) should be satisfied when $\alpha > 1$. Since

$$|(1 - \alpha)S_a - P| = P - (1 - \alpha)S_a \quad \text{when } \alpha > 1 \qquad (4\text{-}10)$$

then

$$S_a \le \frac{L_{stat} - P}{\alpha - 1} \quad \text{for } \alpha > 1 \qquad (4\text{-}11)$$

To achieve high quality time-scale modification equations 4-8) and 4-11) must simply be satisfied, however, the number of iterations that are executed is inversely proportional to S_a, therefore S_a should be maximised for the purpose of computational efficiency giving

$$S_a = \frac{L_{stat} - P}{|1 - \alpha|} \quad \text{for all } \alpha \qquad (4\text{-}12)$$

And since $N = L_{max} + \alpha S_a$, from 4-3),

$$N = L_{\max} + \alpha\left(\frac{L_{stat} - P}{|1 - \alpha|}\right) \quad \text{for all } \alpha \qquad (4\text{-}13)$$

A quasi-stationary segment is a segment in which the frequency content is approximately constant throughout the entire duration of the segment. The duration of stationary segments within an audio input is constantly changing for most naturally occurring sounds. The choice of L_{stat} within the SOLA based implementation described above has a significant effect on both the quality of output and the number of computations required; choosing too small a value results in possibly too many iterations of the synchronisation procedure; choosing too large a value results in inappropriate segments being discarded/repeated. For the general case, where the same parameters are being applied along the entire duration of an input signal, the maximum segment discarded/repeated is L_{stat} and the minimum is $L_{stat} - P$. To ensure the algorithm operates as expected L_{stat} cannot be less than P, since this would suggest that a single period of the lowest likely dominant frequency component could not be discarded or repeated. For typical speech it is found, through informal listening tests by the author[17], that setting L_{stat} to approximately 25 ms results in a high quality output, although varying this parameter for a specific input can yield an improvement in quality.

The above analysis has been performed on pitched signals, such as voiced regions of speech, leaving the question of how unvoiced or noise-like regions are time-scaled somewhat unanswered. However, the unvoiced/noisy regions of speech can also be viewed as being 'quasi-periodic' in the sense that the perceptually important characteristics of noise, i.e. the power within bark bands [Goodwin '96], are effectively constant, and therefore repetitive, over short time segments of the input. Furthermore, the choice of overlap position for noisy signals is not as critical as for periodic segments since discontinuities introduced through a 'poor' choice of overlap will not generally be perceived. For the case where noise energy does not extend over a sufficient duration, i.e. transients, the method described above may result in the undesired repeating or discarding of these short time energy segments; however this problem can be resolved if these segments of audio are detected [Lee '97].

The final important consideration within a time-domain implementation is the duration over which a cross-fade is applied between overlapping synthesis frames. The purpose of the cross-fade is to smooth out discontinuities between synthesis frames [Lee '72]. Typically, a cross-fade is applied along the entire duration of the 'optimal' synthesis overlap determined during the search stage; however a cross-fade

[17] Author testing was undertaken in a quite office environment on a PC using headphones. Test signals comprised of approximately 15 speech signals from the TIMIT database [TIMIT '05], and time-scale factors in the range 0.5 – 3 were applied.

of this duration may be unnecessary and can result in redundant computations. Consider the case of a perfectly periodic signal being time-scaled and synthesis frames are perfectly synchronised, if no cross-fade is applied and synthesis frames are simply appended to each other at an appropriate point within the 'optimal' overlap, then no artefacts will be perceived since discontinuities are not introduced and the period of the signal remains unaltered. However, audio signals are typically not perfectly periodic and some level of cross-fading is required in order to ensure that a smooth transition occurs between synthesis frames. The duration of the cross-fade required to ensure that a smooth transition occurs is dependent on the level of similarity of the synthesis frames and an 'intelligent' method of determining the cross-fade duration for each iteration of the SOLA algorithm could take the value of $R_m(k_m)$ returned by the correlation function into consideration. However a method such as this would require the introduction of a threshold which can introduce problems of its own if the threshold is not adequately set (or if an alternative means for determining the optimal overlap position is used). It has been found experimentally that fixing the duration of the cross-fade to between 2 ms - 5 ms generally provides an adequate solution, although longer cross-fades will reduce the effects of discontinuities by a greater degree.

The analysis given above yields a new parameter set, given by equations 4-12) and 4-13), for use within SOLA based implementations. This parameter set can also take advantage of offset prediction, which is described in section 2.1.1. In [Wong '03] it is shown that offset prediction occurs if

$$k_{min} \le k_{m-1} + S_a - S_s \le k_{max} \qquad (4\text{-}14)$$

To determine the benefits of 'predictive skipping' the probability of equation 4-14) occurring is determined. For this purpose, the values derived above are substituted into 4-14)

$$0 \le k_{m-1} + (1\text{-}\alpha)\ \frac{L_{stat} - P}{|1-\alpha|} \le P \qquad (4\text{-}15)$$

For time scale compression the probability is given by:

Probability $(0 \leq k_{m-1} + L_{stat} - P)$ i.e. 1, since k_{m-1} is in the range 0 to P and $L_{stat} > P$,

and

Probability $(P \geq k_{m-1} + L_{stat} - P)$ i.e. $(2P - L_{stat})/P$ if k_{m-1} is equally likely to be any value in the range 0 to P.

For time scale expansion the probability is given by:

Probability $(0 \leq k_{m-1} - L_{stat} + P)$ i.e. $(2P - L_{stat})/P$, if k_m is equally likely to be any value in the range 0 to P,

and

Probability $(P \geq k_{m-1} - L_{stat} + P)$ i.e. 1, since k_m is in the range 0 to P and $L_{stat} > P$.

From above, the probability of equation 4-15) being satisfied is $(2P - L_{stat})/\mathrm{P}$ for all time-scale modifications. Therefore, the benefit of prediction is dependent on the maximum length of segment that can be discarded/repeated during a single iteration of the algorithm i.e. L_{stat}; if L_{stat} is greater than or equal to $2P$ then prediction provides no additional advantage.

5 A Comparison of Synchronisation Procedures

A number of synchronisation procedures have been employed within time-domain time-scale modification algorithms. This chapter provides a comparison of those synchronisation procedures in terms of computational requirements and quality of output. Section 5.1 provides a computational comparison based on the number of basic arithmetic, compare and shift operations required by each synchronisation procedure within the SOLA based framework described in chapter 4; section 5.2 summarises the computational load comparison. Section 5.3 compares the output quality of the synchronisation techniques through the use of a number of objective quality measures.

5.1 Comparison of Computational Requirements

5.1.1 Basic Unbiased Correlation

The normalising denominator of equation 2-1) has the effect of reducing the magnitude of the correlation function when a high energy noise burst (transient), or any other high energy segment, exists within one of the synthesis frames. This normalising process comes at a relatively high computational expense and the less complex unbiased correlation function has been used in [Laroche '93a], and is given by

$$R_m(k) = \frac{\sum_{j=0}^{L_k-1} y(mS_s + k + j)x(mS_a + j)}{L_k} \tag{5-1}$$

where the parameters are the same as those given in equation 2-1) and described in section 2.1 (This is also the case for all other sections in this chapter).

5.1.1.1 Computational Requirements

As noted in [Laroche '93a] correlation can be efficiently determined through the use of an FFT-based convolution technique [Mitra '93]. From [Lawlor '99] this approach requires the following steps to be taken:

Calculate the FFT of the overlapping synthesis segments. This involves two L_{max} point real input FFT's.

Multiply the resulting FFT's. This involves L_{max} complex multiplications i.e. $4L_{max}$ multiplies and $2L_{max}$ additions.

Calculate the inverse FFT of the result of step 2. This involves one L_{max} complex input inverse FFT.

An N-point radix-2 FFT of a complex input requires approximately $2N\text{Log}_2N$ real multiplications and $3N\text{Log}_2N$ real additions [Mitra '93]. It can also be shown that two N-point FFT's of two real inputs can be efficiently determined using one N-point complex FFT and $2N - 4$ real additions [Mitra '93]. An N-point inverse FFT of a complex input requires approximately the same number of operations as an N-point FFT. In addition, the unbiasing denominator term of equation 5-1) requires one division per overlap i.e. P divisions.

Having determined the correlation function, the maximum value of $R_m(k)$ must be found. This requires P comparisons.

5.1.2 Normalised Correlation

The normalised correlation function is given by:

$$R_m(k) = \frac{\sum_{j=0}^{L_k-1} y(mS_s + k + j)x(mS_a + j)}{\sqrt{\sum_{j=0}^{L_k-1} x^2(mS_a + j)\sum_{j=0}^{L_k-1} y^2(mS_s + k + j)}} \qquad (5\text{-}2)$$

5.1.2.1 Computational Requirements

The numerator of equation 5-2) can be determined in the same manner as described in section 5.1.1. Within the SOLA based implementation resulting from the analysis provided in chapter 4, the correlation function is determined for a minimum synthesis overlap of $L_{max} - P$ to a maximum of $L_{max.}$ If the denominator is first determined for the minimum synthesis overlap, it can then be computed efficiently through an iterative process for successive, increasing, synthesis overlaps. Each summation term of the denominator initially requires $L_{max} - P$ multiplies and $L_{max} - P$ additions for the minimum overlap. For subsequent synthesis overlaps one addition and one multiplication per summation term is required, followed by one multiplication of the summation terms and the application of a square root to the resulting product. Also, for each of the P possible synthesis overlap positions one division of the denominator into the numerator is required.

There are numerous methods for determining the square root. One common approach is the iterative Newton-Raphson algorithm [Patterson '96]. This approach requires one shift (to determine a division by 2), one addition and one division per iteration with an adequate result, for this application, generally being returned after 10 iterations.

P comparisons are then required to determine the maximum of the correlation function.

5.1.3 Simplified Normalised Correlation

The simplified normalised correlation function is suggested for use with a time-scale modification algorithm in [Lawlor '99] and is given by

$$R_m(k) = \frac{\sum_{j=0}^{L_k-1} y(mS_s + k + j)x(mS_a + j)}{\sum_{j=0}^{L_k-1} |x(mS_a + j)| \sum_{j=0}^{L_k-1} |y(mS_s + k + j)|} \qquad (5\text{-}3)$$

5.1.3.1 Computational Requirements

The simplified normalisation function of equation 5-3) is calculated in a similar manner to equation 5-2), however, each summation term of the denominator initially requires only $L_{max} - P$ additions for the minimum overlap. For subsequent synthesis overlaps only one addition per summation term is required together with one multiplication of the summation terms.

P comparisons are required to determine the maximum.

5.1.4 Average Magnitude Difference Function

The average magnitude difference function AMDF is suggested for use in [Verhelst '93] and is given by

$$R_m(k) = \frac{\sum_{j=0}^{L_k-1} \left| y(mS_s + k + j) - x(mS_a + j) \right|}{L_k} \tag{5-4}$$

5.1.4.1 Computational Requirements

This function requires, on average, $L_{max} - P/2$ subtractions, $L_{max} - P/2$ additions and 1 division for each synthesis overlap/offset.

P comparisons are required to determine the minimum.

5.1.5 Mean Square Difference

Similar to the AMDF, the mean square difference provides additional emphasis on large differences in magnitudes and is given by

$$R_m(k) = \frac{\sum_{j=0}^{L_k-1} \left(y(mS_s + k + j) - x(mS_a + j) \right)^2}{L_k} \tag{5-5}$$

5.1.5.1 Computational Requirements

The mean square distance measure requires, on average, $L_{max} - P/2$ subtractions $L_{max} - P/2$ additions, $L_{max} - P/2$ multiplications and 1 division for each synthesis overlap/offset.

P comparisons are required to determine the minimum.

5.1.6 Envelope Matching-TSM

The envelope matching technique is summarised in section 2.1.3; the following analysis uses the notation defined in that section.

5.1.6.1 Computational Requirements

The steps involved in the implementation of the EM-TSM algorithm can be summarised as follows:

1. Determine the envelope function of both of the synthesis frames.

2. Determine the set K_0 that is defined in [Wong '03] and described as being:

   a) 'any lag k such that there is at least one common crossing point between $x_{2,k}$ and $y_{2,k}$'.

   b) 'any k such that $y_{2,k-1}$ has a zero-crossing point which disappears in $y_{2,k}$'.

   c) 'any k such that $x_{2,k-1}$ has a zero-crossing point that disappears in $x_{2,k}$'.

   d) k_{min} and k_{max}.

3. For the first k in the set K_0 i.e. k_1, determine $R(k)$ using equation 2-15)

4. For the remaining k in K_0 determine $R(k)$ using equation 2-19).

5. Find the value of k for which $R(k)$ is a maximum.

The computations required for each step are now expanded:

Step 1:

L_{max} comparisons for each frame. A_k and B_k can also be found in parallel.

Step2:

a) Approximately $ZC_{avg}(L_{max} - P/2)$ comparisons, where ZC_{avg} is the average number of zero crossings per sample. C_k and $g(k,j)$ can also be determined in parallel.

b) 1 comparison for each overlap to determine if $b_{k-1,1} = 1$ i.e. P comparisons.

c) 1 comparison for each overlap to see if $a_{k-1,M_{k-1}} = L_{k-1}$

d) No operations are required to find k_{min} and $k_{max.}$

Step 3:

Equation 2-14) requires the following operations:

1 addition and 1 subtraction to determine M_k+N_k and $M_k+N_k - 2r_{k.}$.

1 comparison to determine each $(-1)^{M_k+N_k}$.

$M_k+N_k - 2r_k$ comparisons to determine $(-1)^{j+1}$. Each comparison is followed by an addition or subtraction.

1 addition of the terms within brackets.

1 shift to calculate the multiply by 2.

1 comparison to determine $\beta_{z,k} = x_{2,k}[0]y_{2,k}[0]$.

1 subtraction to determine L_k, since L_k is given by $L_{max} - k$.

1 division by L_k.

Step 4:

Equation 2-16) requires the following operations to be performed:

1 multiplication to determine $L_{ki}R(k_i)$.

1 comparison to determine β_{z,k_i} .

1 shift is required for the multiply by 2.

2 additions to determine $M_{k_i} + N_{k_i} + 1$.

1 compare is required to determine $(-1)^{M_{k_i}+N_{k_i}+1}$.

2 subtractions to determine $L_{k_i} - (k_{i+1} - k_i)$.

1 multiplication to determine $(k_{i+1} - k_i)\left(2\beta_{z,k_i}\xi_{k_i} + \beta_{z,k_i}(-1)^{M_{k_i}+N_{k_i}+1}\right)$.

1 addition to sum LHS and RHS.

1 division by $L_{k_i} - (k_{i+1} - k_i)$.

$g(k,j)$ can be calculated during step 2 (a) since the location of $b_{k,j}$ in the set C_k can be determined at that time.

One iteration in the calculation of ξ_{k_i} requires one comparison to determine whether an addition or subtraction is required, followed by an addition or subtraction. On average ZC_{avg} $(L_{max} – P/2)$ iterations are required to determine each ξ_{k_i} .

Step 5:

Requires $Q+1$ comparisons, where $Q+1$ is the cardinality of $K_{o.}$

On average M_k and N_k are approximately $ZC_{avg}(L_{max} - P/2)$ and r_k is approximately $ZC_{common,avg}(L_{max} - P/2)$, where $ZC_{common,avg}$ is the average number of common zero crossings per sample.

5.1.7 Modified Envelope Matching - TSM

The modified envelope matching technique is summarised in section 2.1.3; the following analysis uses the notation defined in that section.

5.1.7.1 Computational Requirements

The function of equation 2-20) is a ‘decimated’ version of the normalised correlation function. Since the decimated correlation function is only determined for a relatively small number of synthesis overlaps, it is no longer computationally efficient to employ an FFT in determining the numerator. Calculating the numerator, on average, requires $(L_{max} - P/2)/q$ multiplies and $(L_{max} - P/2)/q$ additions for each candidate offset that is being re-examined.

The denominator of the decimated function is found in a similar manner to that described in section 5.1.3, however the number of operations required to determine each summation term is proportional to the maximum overlap associated with each of the candidate offsets i.e. the overlap associated with the minimum candidate offset. An estimate of the number of operations required to determine the summation terms for all candidate offsets is then $L_{c,max}/q$ multiplies and $L_{c,max}/q$ additions, where $L_{c,max}$ is the overlap associated with the smallest candidate offset. Assuming that the occurrence of a candidate offset is equally likely to occur anywhere in the range 0 to P, and assuming $M << L_{max}$, where M is the number of candidates being re-examined, it can be statistically shown that $L_{c,max}$ is approximately given by $P(M/(M+1))+ L_{max} - P$. Having determined both summation terms for each candidate offset, the summation terms are multiplied and the square root of their product is determined. The square root can be determined via the Newton-Raphson method as in subsection 5.1.2.1. Finally the denominator is divided into the numerator. M comparisons are then required to determine the maximum.

In order to reduce the number of zero-crossings the distance between consecutive pairs is first determined and then compared with the defined threshold. This operation requires one subtraction and one comparison for each pair being evaluated in each frame.

5.1.8 GLS-TSM

The global and local search approach is summarised in section 2.1.2; the following analysis uses the notation defined in that section.

5.1.8.1 Computational Requirements

The steps involved within the GLS-TSM algorithm can be summarised as follows:

Step 1: For each overlap position determine the number of zero-crossings in each of the synthesis frames.

The most efficient way to determine this is to first find the zero-crossings in the minimum overlap region and iteratively determine the zero-crossings for the remaining overlap positions. This initially requires $L_{max} - P$ compares and one addition for each zero crossing in the minimum overlap region. Then for each of the remaining P possible overlaps a comparison is required with an addition required if a zero-crossing is detected.

Step 2: Determine the overlap position that provides the minimum difference between the number of zero crossings in the analysis frame and the number of zero crossings in the synthesis frames. This provides the global search overlap.

Using the data determined in Step 1, this procedure then requires P comparisons.

Step 3: Find the slope at each zero crossing of the analysis frame within the global search overlap.

For each zero crossing a subtraction is required followed by a division, to determine the slope. Assuming that the average global search overlap is $P/2$, then $(P/2).ZC_{avg}$ subtractions and $(P/2)$. ZC_{avg} divisions are required.

Step 4: Find the zero crossing that corresponds to the maximum of all the slopes calculated. This zero crossing then becomes the reference zero crossing point.

Finding the maximum slope requires $(P/2).ZC_{avg}$ comparisons.

Step 5: Compute the feature vector for the reference zero crossing.

Calculating an 11 point feature vector requires:

5 subtractions, 2 shifts (for two divide by 2 operations) and 2 divisions.

Step 6: Compute the synthesis zero-crossings in the neighbourhood of the reference zero crossings. Use the U nearest neighbours.

This requires U x step 5 operations.

Step 7: Find the ‘distance measure’ between the reference feature vector and the candidate synthesis feature vectors.

Calculating one distance measure requires:

11 subtractions, 11 additions, 1 division.

Step 8: Find the minimum 'distance measure' and use the corresponding synthesis zero crossing point and reference to determine the final overlap position.

This requires U comparisons.

5.1.9 Peak Alignment

A peak alignment approach has been described in [Lawlor '99] and [Dorran '03b]. Here a method is briefly outlined that allows a peak alignment approach be applied to the overlap-add procedure described in chapter 4 .

The first step is to determine the maximum, i.e. a peak, in $x(mS_a + j)$ for $1 \leq j \leq P$. Given that a maximum occurs at $j = j_{max,x}$, the next step is to determine the maximum in $y(mS_s - k_{m-1} + j_{max,x} + j)$ for $1 \leq j \leq P$. Given that a maximum occurs at $j = j_{max,y}$

$$k_m = j_{max,y} - j_{max,x} \qquad (5\text{-}6)$$

It should be noted that L_{max} must be $2P$ for the peak alignment process to operate as expected.

5.1.9.1 Computational Requirements

This approach requires P comparisons to determine the peak/maximum in each frame. Calculating $mS_s - k_{m-1} + j_{max,x}$ and k_m requires one addition and two subtractions.

5.2 Computational Comparison Summary

	Compares	**Additions/ Subtractions**	**Shifts**	**Multiplies/divides**	**TOTAL**
Peak Alignment	$2.P$	3	0	0	1.00
GLS-TSM	$2.L_{max} + \mathrm{ZC}_{\mathrm{avg}}.(\mathrm{P}/2) + U+P$	$2.L_{max}.ZC_{avg} + (P/2). ZC_{avg} + U.27 + 5$	$(1+ U).2$	$(1+ U).3 + (P/2). ZC_{avg}$	4.04
EM-TSM	$3.Q + 2.P + 3 + 2.L_{max} + 2.(ZC_{avg} - ZC_{common,avg})(L_{max} - P/2) + ZC_{avg}.(L_{max} - P/2)(Q+1)$	$4 + 5.Q + Q.ZC_{avg}.(L_{max} - P/2)+ 2.(\mathrm{ZC}_{\mathrm{avg}} - ZC_{common,avg})(L_{max} - P/2)$	$Q+1$	$3.Q+1$	43.58
MEM-TSM **Reduced zero crossings**	$\mathrm{ZC}_{\mathrm{avg}}.\mathrm{L}_{\mathrm{max}}$	$\mathrm{ZC}_{\mathrm{avg}}.\mathrm{L}_{\mathrm{max}}$	0	0	18.12
MEM-TSM Candidate re-examination	M	$(P(M/(M+1))+ L_{max} - P)/q + M(L_{max} - P/2)/q + 10.M$	10.M	$11.M + (P(M/(M+1))+ L_{max} - P)/q + M.(L_{max} - P/2)/q$	21.67
Unbiased Correlation	P	$2.L_{max} (3.\mathrm{Log}_2(2.L_{max}) -1) - 4$	0	$4.L_{max}.\mathrm{Log}_2(2.L_{max}) +P$	91.63
Simplified Normalised Correlation	P	$2.L_{max}(3.\mathrm{Log}_2(2.L_{max}) -1) - 4+ 2.L_{max}$	0	$4.L_{max}.\mathrm{Log}_2 (2.L_{max}) +2.P$	94.11
Normalised Correlation	P	$2.L_{max}(3.\mathrm{Log}_2(2.L_{max}) -1) -4+ 2.L_{max} + 10.P$	$10. P$	$4.L_{max}.\mathrm{Log}_2 (2.L_{max}) +12.P+2.L_{max}$	111.01
AMDF	P	$2.P. (L_{max} - P/2)$	0	P	239.50
Mean Square Difference	P	$2.P. (L_{max} - P/2)$	0	$P.(L_{max} - P/2) + P$	358.75

Table 5-1 Computation comparison of time-domain synchronisation procedures.

The column furthermost to the right of Table 5-1 shows a normalised comparison of the number of operations each approach requires. The comparison assumes that each operation requires the same duration to process and uses the parameter values given below. The totals are normalised by dividing the total by the number of operations required by a peak alignment approach. The totals for the two MEM-TSM rows (shown shaded) also take the number of operations required by EM-TSM, after the zero crossing reduction has been applied, into consideration.

For a sampling rate of 16kHz the following values typically apply:

Maximum period, P = 160 samples; corresponding to 10 ms.

The initial (and maximum) synthesis overlap, L_{max} = 320; corresponding to 20 ms.

Average zero crossings per sample, ZC_{avg} = 0.19[18].

Average common zero crossings, between synthesis frames, per sample, $ZC_{common,avg}$ = 0.068[18].

The number of re-examined candidates in GLS-TSM, U = 10.

The number of re-examined candidates in EM-TSM, M = 8.

Average number of elements in K_0, Q = 128.

The MEM-TSM correlation decimation factor, q = 5.

The figures shown for the two MEM-TSM rows were obtained when the T_1 parameter is set to 6 samples, for the application of the zero crossing reduction procedure. After the zero crossing reduction procedure is applied ZC_{avg} becomes 0.072, $ZC_{common,avg}$ becomes 0.0066 and Q becomes 99.2[18].

[18] The parameters Q, ZC_{avg} and $ZC_{common,avg}$ were determined from the examination of 250 test signals obtained from the TIMIT speech corpus [TIMIT '05].

It should be noted that a further reduction in the computational complexity of synchronisation procedures which employ the FFT could also be achieved through the use of techniques such as FFT pruning [Markel '71] or the Goertzel technique [Oppenheim '75]. In addition, as noted in [Wong '03], synchronising high frequency content of a signal is not as important as the low frequency content, therefore all of the synchronisation procedures are likely to produce high quality results when applied to down sampled data. Such an approach is also suggested in [Laroche '93a] whereby the input (sampled at 48 kHz) was first down sampled by a factor of 6, thus providing significant computational reduction.

5.3 Objective Output Quality Evaluation

Whilst the previous section provides a method of comparing the computational cost of each synchronization procedure, the quality of output that each produces remains unclear. A statistically relevant subjective comparison between each approach is a considerable undertaking; a more pragmatic approach, within the confines of this study, involves an analysis of the behaviour of each procedure for some particular instances, such as that given in section 4, which considers particular cases of the application of a normalized correlation function to a perfectly periodic signal (see Figure 4-1).

Firstly, it should be noted that all synchronisation procedures are likely to produce a similar quality of output when dealing with noisy signals, since any overlap selected will result in a perceptually noisy signal. One exception to this statement is if there is a significant change in energy during the overlap region of the noisy signal; for this case, the GLS-TSM and EM-TSM algorithms are likely to perform worse than other techniques since they do not explicitly consider amplitude values. It should be noted, however, that such an energy change could be regarded as a transient and should, therefore, be simply translated to its time-scaled position [Lee '97].

The synchronisation procedures presented in this chapter can be classified into two groups i.e. similarity measures (correlation, AMDF, MSD, EM-TSM, MEM-TSM) and feature matching processes (GLS-TSM and peak alignment); similarity measure processes can be viewed as multiple feature matching processes. The feature matching processes are considerable less computationally intensive than their similarity measure counterparts, due to the fact that they consider a relatively smaller number of features. However, using a small feature set to identify a suitable overlap can result in some difficulties. Consider the case where either feature matching process is employed in the situation illustrated in Figure 4-1; there exists a possibility of choosing an unsuitable overlap regardless of the search range used. Also consider the case shown in Figure 5-1 in which the overlapping segments of two overlapping frames from a sinusoid injected with noise are shown; the detected peaks are also shown. If these peaks were aligned a discontinuity would be introduced into the resulting signal. Similar situations can be foreseen within the GLS-TSM algorithm, whereby the feature vectors associated with zero crossings could be incorrectly chosen due to the influence of noise in an otherwise periodic signal.

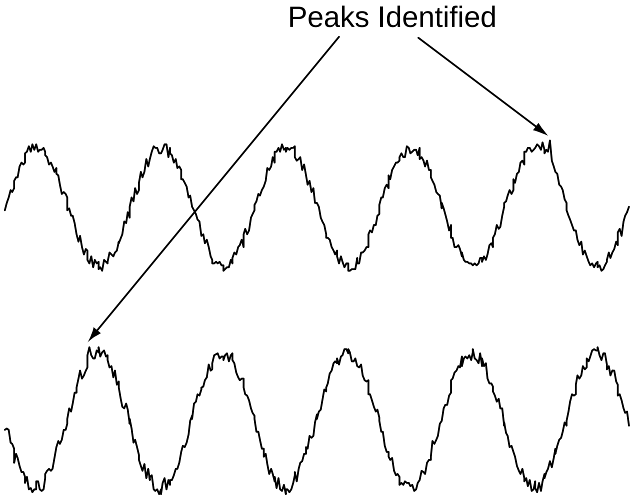


Figure 5-1 Case where peak alignment procedure would result in poor synchronisation.

It should be noted that the EM-TSM algorithm would also have difficulty with the situation shown in Figure 4-1, since amplitude values are not considered in the one-bit correlation function employed. This problem is recognised in [Wong '03] and results in the development of a refinement to the EM-TSM algorithm i.e. MEM-TSM. It is also possible that the refined MEM-TSM algorithm could have difficulty in identifying a suitable overlap given a situation similar to that illustrated in Figure 4-1 (b) and in Figure 4-1 (c), for the case where there is a greater number of peaks in a period of the waveform, with the likelihood of some difficulty arising reducing as the number of re-examined candidates increases.

The normalized correlation function is not influenced by the power that exists in the overlapping region to the same degree as the unbiased correlation function. Therefore, given the situation where a perfectly periodic overlap exists and a less than perfect, but higher energy, overlap also exists the unbiased correlation function may return the offset associated with the less than perfect overlap (depending on the energy present and level of periodicity); for this case the normalized correlation function will always return the offset associated with the perfect overlap. This suggests that the normalized correlation function is, in general, more robust.

For the case where the overlapping regions are perfectly periodic throughout, the synchronisation function should return the same value for each perfectly synchronised overlap position since each overlap is equally valid; this is not the case for the simplified normalized correlation function, which is biased toward shorter overlaps. The cause of the bias can be understood by first defining two functions $f_{sum\_abs}(x)$ and $f_{sum\_square}(x)$, which are defined to be the sum of the absolute values and the sum of the square of the values of the same periodic sequence x over one period, respectively. For the case where perfectly synchronised overlaps occur at overlapping lengths of kP, where k is any integer and P is the period of the periodic sequence x, the simplified normalized correlation function at positions of perfectly synchronised overlap is given by

$$\frac{f_{sum\_square}k}{f_{sum\_abs}k.f_{sum\_abs}k} \tag{5-7}$$

From equation 5-7) it can be seen that the value returned is dependent on length of the overlapping region, since f_{sum_square} and f_{sum_abs} are constants; for larger values of k a smaller value is returned. A solution to this problem is to multiply the simplified normalized correlation function of equation by the length of the overlapping region

L_k.

Given the search range suggested in section 4, it is expected that the AMDF, MSD and normalized correlation synchronization procedures would return a similar offset ([Verhelst '93] found this to be the case in comparisons between a normalized correlation function and an AMDF within a WSOLA implementation). It should be noted that the MSD places additional emphasis on magnitude differences in comparison with the AMDF function (the AMDF function therefore places relatively more emphasis on the 'amplitude normalized' waveform shape); it is unclear if this will have a significant impact in the perceived quality of output.

In a further effort to objectively identify the 'best' synchronisation procedure a method of statistically comparing the techniques is given in section 5.3.1.

5.3.1 An Objective Quality Measure

The output quality of a time-domain time-scale modification algorithm is primarily dependent on how similar the overlapping segments of the synthesis frames are; hence the reason for making use of similarity measures in finding the optimum overlap. The same similarity measures can be (and have been in [Wong '03]) used to provide an objective evaluation of the quality of each synchronisation procedure; however some difficulty lies in determining which similarity measure is perceptually 'best'. Here, a method of comparing synchronisation procedures is used which makes use of a subset of the similarity measures employed for time-scaling purposes. Since the GLS-TSM and peak alignment procedures are not similarity measures they are not used in the proposed objective measure; also since the EM-TSM, MEM-TSM, and simplified normalized correlation functions are derivatives of the normalized correlation function, they will not be used. The similarity measures that are used in the objective measure are the normalized correlation function, the unbiased correlation function, the MSD, and the AMDF.

In an attempt to determine which synchronisation procedure is 'best', the results obtained from each similarity measure are statistically normalised and the 'best' synchronisation procedure is deemed to be that procedure that is associated with the maximum of the sum of the normalised measures, as described below.

250 test signals obtained from the TIMIT speech corpus were used during the objective evaluation. Speech signals were chosen since they are quasi-periodic and due to the availability of a suitable corpus. Similar results are expected for monophonic musical recordings since the important feature of the signal when time-scaling in the time-domain is quasi-periodicity and both speech and monophonic music have similar levels of periodicity. Each test signal is time-scaled by a factor of 2 using each of the synchronisation procedures described in chapter 5. It should be noted that any time-scale factor could be used; however a time-scale factor close to one requires a relatively small number of iterations, on the other hand a very large time-scale factor would return very similar results from successive iterations of the algorithm. For each iteration of each algorithm, the similarity measures given by equations 5-8) to 5-11) are applied; in addition the similarity measures are applied to a random offset, to provide a point of reference.

$$measure_1 = \frac{\sum_{j=0}^{L_{k_m}-1} y(mS_s + k_m + j)x(mS_a + j)}{\sqrt{\sum_{j=0}^{L_{k_m}-1} x^2(mS_a + j) \sum_{j=0}^{L_{k_m}-1} y^2(mS_s + k_m + j)}} \tag{5-8}$$

$$measure_2 = \frac{\sum_{j=0}^{L_{k_m}-1} y(mS_s + k_m + j)x(mS_a + j)}{L_{k_m}} \tag{5-9}$$

$$measure_3 = \frac{\sum_{j=0}^{L_{k_m}-1} \left| y(mS_s + k_m + j) - x(mS_a + j) \right|}{L_{k\,m}} \tag{5-10}$$

$$measure_4 = \frac{\sum_{j=0}^{L_{k_m}-1} \left(y(mS_s + k_m + j) - x(mS_a + j) \right)^2}{L_{k_m}} \tag{5-11}$$

Having accumulated the measures for each synchronisation procedure, the measures are then normalised using a standard score approach [Coolidge ‘00] e.g. given that the accumulation of *measure*$_1$ is given by the set *measure*$_{1,acc}$ = {m_1, m_2, …, m_{10}}, where m_w is the accumulation of *measure*$_1$ when applied to synchronisation procedure number *w*, then the normalised set is given by

$$\left\{\frac{m_1 - mean\left(measure_{1,acc}\right)}{stdDev\left(measure_{1,acc}\right)}, \frac{m_2 - mean\left(measure_{1,acc}\right)}{stdDev\left(measure_{1,acc}\right)}, \ldots., \frac{m_9 - mean\left(measure_{1,acc}\right)}{stdDev\left(measure_{1,acc}\right)}\right\} \tag{5-12}$$

where *stdDev* is the standard deviation.

In addition, the sign of the normalised data set is inverted for *measure*$_3$ and *measure*$_4$ to take account of the fact that a minimisation of these functions is desired. Table 5-2 shows the results of the objective output quality assessment, with the furthermost right column showing the sum of the normalised measures for each of the synchronisation procedures.

In two separate tests the objective measures were applied to the test signals with additional noise injected into the test signals and only those frames considered voiced (since appropriate overlapping of voiced regions of speech is, in general, perceptually more important than that of unvoiced or silent regions). The results of both additional tests are closely approximated by those presented in Table 5-2.

The results given in Table 5-2 indicate that the normalized correlation function is the ‘best’ synchronisation procedure to employ. The MSD and AMDF functions achieve similar results; however the computational benefits of the normalised correlation also outweighs these alternatives. As expected the MEM-TSM provides an improvement in quality over the EM-TSM approach.

The two feature matching algorithms (GLS-TSM and peak alignment) perform worse than their similarity measure counterparts; from the discussion presented at the start of section 5.3, this is in keeping with expectations. It should be noted that the peak alignment approach performs better than GLS-TSM; this can be attributed to the fact that, in general, there are more zero-crossings (the feature considered by the GLS-TSM algorithm) present in a speech signal than ‘dominant’ peaks; it therefore follows that the probability of obtaining an 'incorrect' feature match is greater when zero-crossings are being considered.

	Measure_1	Measure_2	Measure_3	Measure_4	TOTAL
Normalised Correlation	0.83	0.57	0.55	0.59	2.55
Mean Square Difference	0.55	0.52	0.73	0.70	2.50
AMDF	0.44	0.47	0.80	0.63	2.35
Simplified Normalised Correlation	0.71	0.42	0.35	0.45	1.93
Unbiased Correlation	0.36	0.79	0.25	0.35	1.76
MEM-TSM (M=20, q = 5)	0.45	0.36	0.39	0.37	1.58
EM-TSM	0.26	0.22	0.29	0.20	0.98
Peak alignment	-0.33	-0.32	-0.27	-0.20	-1.12
GLS-TSM	-0.83	-0.42	-0.50	-0.46	-2.22
Random Offset	-2.46	-2.62	-2.60	-2.64	-10.33

Table 5-2 An objective output quality comparison of time-domain synchronisation procedures.

The results supplied by the objective measure provide a quantifiable method of comparing various synchronisation procedures and support the statements made at the start of section 5.3; however a large number of subjective listening tests are required to verify the accuracy of the measure with statistically relevant accuracy.

6 Time-Domain/Subband Improvements

Time-domain/subband approaches, reviewed in section 2.2, partition a complex input into less complex subbands; each subband is then time-scaled using a time-domain technique and the time-scaled subbands are summed to produce a time-scaled version of the original complex signal.

The major issues concerning a subband approach are the choice of filterbank used to partition the waveform into subbands and the recombination of the time-scaled subbands in a synchronous manner [Spleesters '94]. The solutions to these issues are diametrically opposite since partitioning a complex waveform into many subbands reduces the complexity of each subband but increases potential synchronisation problems and vice versa.

Prior work in the area made use of uniform width filterbanks to partition the input into subbands. In this section a time-domain/subband implementation based upon the bark scale is proposed as an effective partitioning technique for the time-scale modification of Western tonal music that offers a suitable compromise to the issues outlined above [Dorran '03d]. In addition, a number of subband synchronisation procedures are presented in section 6.2 which attempt to reduce the loss of synchronisation between time-scaled subbands [Dorran '04a].

6.1 Bark Subband Approach to TSM of Music

6.1.1 The Relationship between Bark Bands and Music

The concept of consonance is somewhat vague due to its subjective nature, but in general consonant sounds are those sounds that are perceived as being pleasing or harmonious to the ear [Howard '01]. In [Plomp '65] the relationship between tonal consonance and critical bandwidth is investigated; findings show that two pure tones of different frequency are perceived as being maximally consonant when they are separated in frequency by, or more than, their associated critical bandwidth [Zwicker '99]. Figure 6-1 illustrates this relationship. The plot, from [Plomp '65] shows that consonance is at a minimum when two tones are separated by approximately one quarter of their associated critical bandwidth.

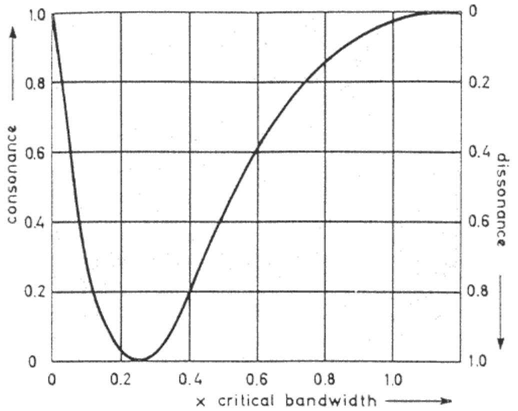


Figure 6-1 Tonal consonance as a function of critical bandwidth separation from [Plomp '65].

The critical band scale [Zwicker '99] is set by the upper and lower limit of the critical bands if they are aligned in such a way that the upper cut-off frequency of the lower critical band is identical to the lower cut-off frequency of the next higher critical band. Formulation of the critical band scale in this way led to the introduction of a new frequency scale called the bark scale[19]. Table 6-1 shows the corresponding lower cutoff (LC) and upper cutoff (UC) frequency values of the bark scale in hertz. Defining the bark scale in this manner also provides an assurance that a perfectly consonant sound will have only one frequency component within each bark band.

In [Plomp '65] a close relationship between tonal consonance and the frequency ratios on which Western music is developed was identified. Figure 6-2 plots the consonance/dissonance (inverse of consonance) levels of two complex tones, both consisting of a fundamental and five harmonics, when one complex tone's fundamental frequency is held at 250 Hz and the other's fundamental frequency is allowed vary from 250 Hz to 500 Hz. As can be seen from the plot, typical music

[19] In honour of the German physicist Heinrich Georg Barkhausen (1881-1956).

frequency ratios are shown to correspond to peaks in the consonance/dissonance curve. It should be noted that the frequency ratios shown do not correspond to an equal tempered tuning system.

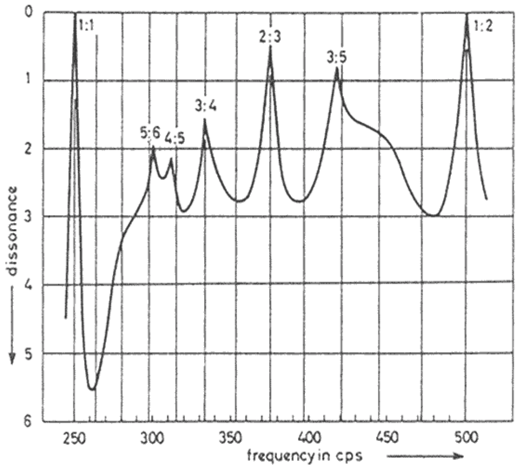


Figure 6-2 Relationship between tonal consonance and western musical frequency ratios from [Plomp '65].

Bark	LC	UC	Bark	LC	UC
1	0	100	13	1720	2000
2	100	200	14	2000	2320
3	200	300	15	2320	2700
4	300	400	16	2700	3150
5	400	510	17	3150	3700
6	510	630	18	3700	4400
7	630	770	19	4400	5300
8	770	920	20	5300	6400
9	920	1080	21	6400	7700
10	1080	1270	22	7700	9500
11	1270	1480	23	9500	12000
12	1480	1720	24	12000	15500

Table 6-1. Bark band upper cutoff (UC) and lower cutoff (LC) frequencies in hertz.

As mentioned above, if a sound is perfectly consonant, at most one dominant sinusoidal component will exist within each bark band. Whilst music cannot be generally described as being perfectly consonant, the above analysis supports the notion that the 'rules' upon which Western tonal music is formulated are closely related to consonance, as also noted in [Howard '01]. It should be noted that as Western music evolves the rules governing its development have relaxed with increasingly dissonant intervals becoming acceptable [Howard '01]. Since Western tonal music is formulated on the basis of maintaining a certain level of consonance, it therefore follows that the distribution of sinusoidal components within a Western tonal music signal is more evenly spread along a bark based frequency scale than that of a linear frequency scale. It follows that partitioning of a music signal into subbands using a filterbank based upon the critical band/bark scale is more appropriate than the fixed-width filterbank used in [Tan '00] and [Spleesters '94] since the complexity of each subband will be reduced to a greater degree. The following analysis is presented in support of this statement:

Chords, in particular the major and minor triads, are the basic building blocks of Western tonal music harmony [Howard '01]. In this analysis, the frequencies of the components that are present, for a given chord, are determined; for example, if a two note chord with fundamentals of 330 Hz and 440 Hz which have three significant harmonics is played the following components are present {330, 440, 660, 880, 990, 1320, 1760}. A 'complexity' measure for each subband is then defined to be the number of components present within a given subband. The purpose of the analysis is to compare the 'complexity' of subbands when bark and uniform based partitioning schemes are applied to the music signal. In the comparison the same number of subbands is used in each case i.e. 25 bark subbands and 25 uniformly spaced subbands across a frequency range of 0-22050 Hz .

During analysis a complex tone, comprising of the fundamental and a given number of harmonics, were used. The number of harmonics used in the experiment is in the range of four to thirty nine[20]. The fundamentals of the complex tones correspond to the root of the chord and the associated intervals of the chord structure; the following triad and seventh chords are used during the analysis; the ratio of each note in the chord to the root is given in brackets:

Major triad: root, major 3rd, perfect 5th (1 5/4 3/2)

Minor triad: root, minor 3rd, perfect 5th (1 6/5 3/2)

Augmented triad: root, major 3rd, augmented 5th (1 5/4 8/5)

Diminished triad: root, minor 3rd, diminished 5th (1 6/5 7/5)

Major 7th: root, major 3rd, perfect 5th, major 7th (1 5/4 3/2 17/9)

Minor 7th: root, minor 3rd, perfect 5th, minor 7th (1 6/5 3/2 9/5)

Dominant 7th: root, major 3rd, perfect 5th, major 7th (1 5/4 8/5 9/5)

Minor/Major 7th: root, minor 3rd, perfect 5th, major 7th (1 6/5 7/5 17/9)

Half diminished 7th: root, minor 3rd, diminished 5th, minor 7th (1 6/5 7/5 9/5)

Full diminished 7th: root, minor 3rd, diminished 5th, diminished 7th (1 6/5 7/5 24/17)

Each note[21] in the range 27.5 Hz to 4186 Hz [22] is used as the root for each chord.

[20] This range is used since it corresponds to the maximum and minimum number of harmonics observed by the author in recordings of the following instruments playing the note c4: bassoon, cello, clarinet, flute, guitar, harmonica, marimba, oboe, organ, piano, saxophone, trombone, trumpet, tuba, viola, violin and xylophone. It should be noted that harmonics with amplitudes which were less than 1/50th of the amplitude of the maximum harmonic were not considered.

[21] Based on Even Tempered tuning [Howard ‘01].

[22] This range corresponds to the lowest and highest note of an 88-note piano keyboard.

The analysis procedure has three variables i.e. root note, chord structure and number of harmonics. For both of the partitioning schemes and for each combination of the three variables an average and maximum subband complexity is determined; the average complexity is calculated by dividing the total number of components present by the number of subbands that contains at least one sinusoidal component; the maximum complexity is defined to be the number of components within the subband with the most components.

Figure 6-3 and Figure 6-4 below plot the mean 'maximum complexity' and mean 'average complexity', respectively, for each root note.

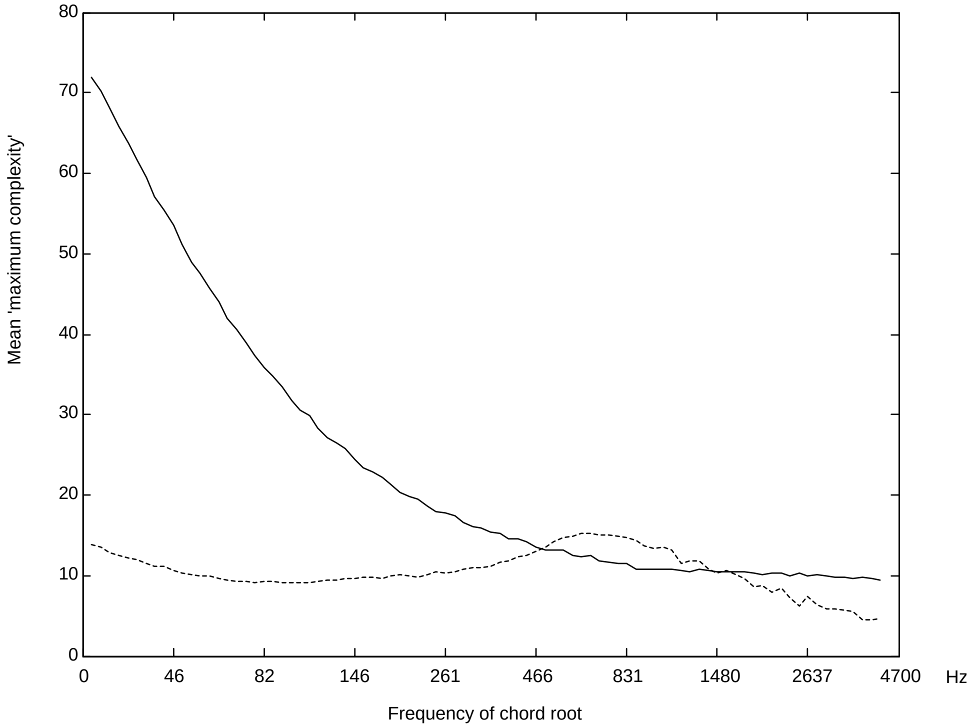


Figure 6-3 Mean 'maximum complexity' of subbands for uniform (continuous line) and bark based (dotted line) partitioning schemes.

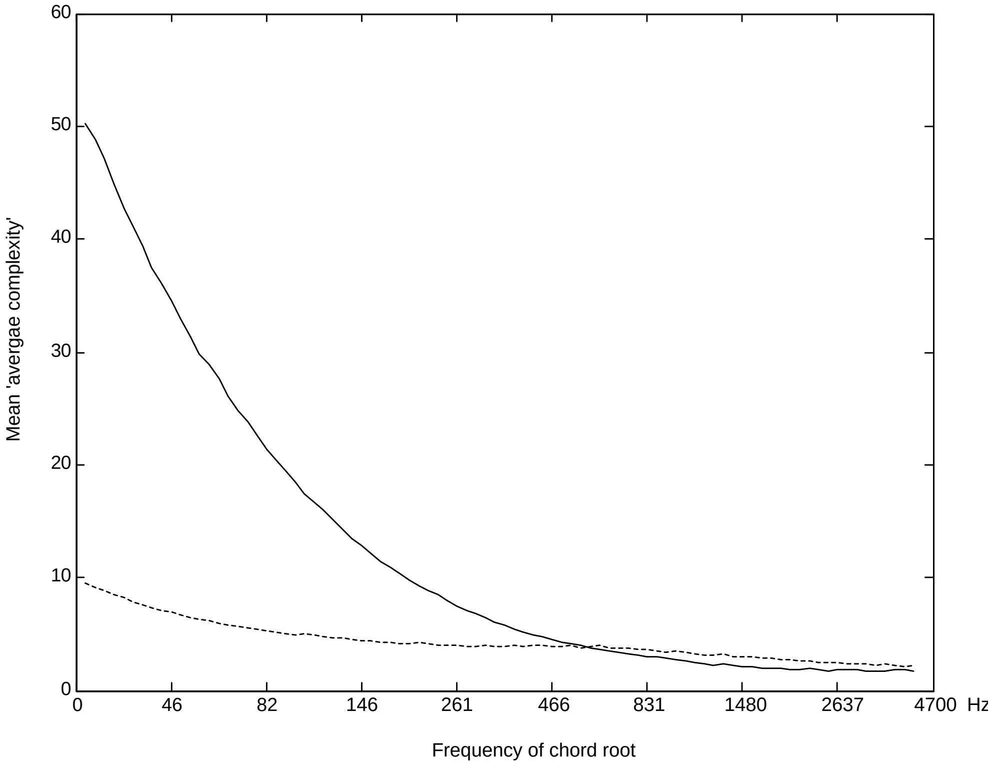


Figure 6-4 Mean ‘average complexity’ of subbands for uniform (continuous line) and bark based (dotted line) partitioning schemes.

From the figures it can be seen that the ‘maximum’ and ‘average’ complexities are approximately constant over the entire range of chord root frequencies for the case of the Bark based partitioning scheme. In contrast, the complexity associated with the uniform partitioning scheme is heavily dependent upon the frequency of the chord root for chord root frequencies less than 450 Hz . For the Bark based partitioning scheme there is a noticeable rise in both maximum and average complexities for chord root frequencies less than 80 Hz ; however, from [Benade ‘76], this range of frequencies falls outside the operating range of most instruments. In addition, there is a noticeable increase in the maximum complexity in the 450 Hz to 1500 Hz region; this is mainly due to the accumulation of components in the highest frequency bark band. For chord root frequencies below 450 Hz the bark based partitioning scheme offers an increasing improvement in both ‘average’ and ‘maximum’ subband complexity over the uniform scheme. Above the chord root frequency of 450 Hz the ‘average’ and ‘maximum’ complexities of both bark based and uniform partitioning schemes are approximately equal; however fewer bark subbands are required, thereby providing a better subband synchronisation-complexity trade-off.

It should be noted that similar results to those of Figure 6-3 and Figure 6-4 are obtained for any chord and/or number of harmonics chosen from the sets given above. In addition, partitioning schemes similar to the bark based scheme, such as Equivalent Rectangular Bandwidth (ERB) [Moore '83] and constant-Q filterbank, provide similar results, as illustrated in Figure 6-5 below.

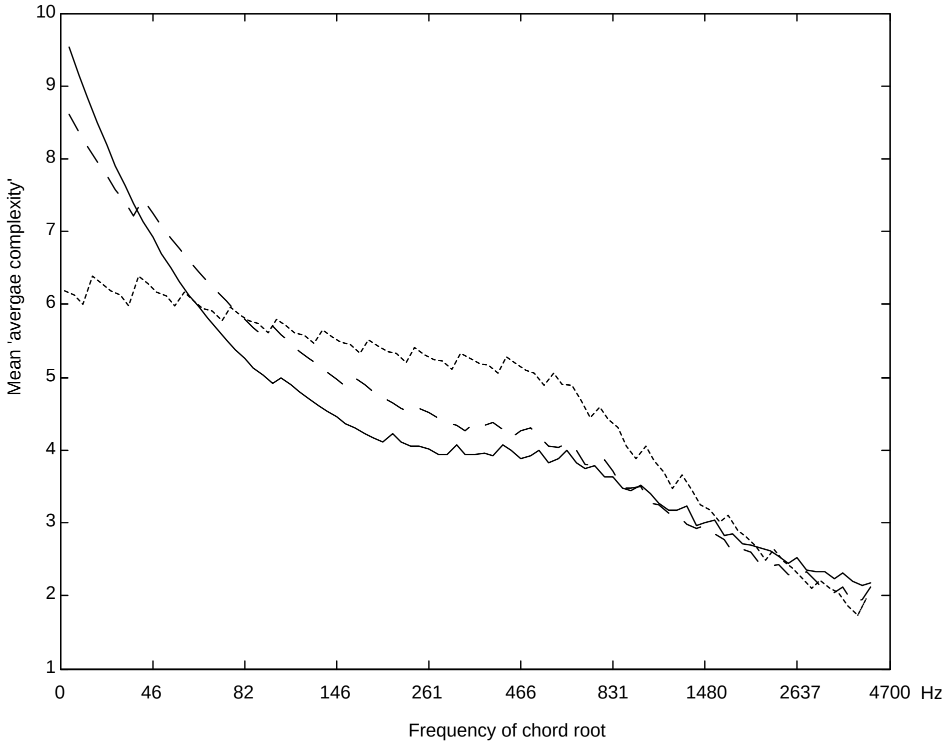


Figure 6-5 Mean 'average complexity' of subbands for bark (continuous line), constant-Q (dotted line) and ERB (dashed line) partitioning schemes.

The above analysis provides justification for bark based partitioning of subbands; however, the question of how many subbands are necessary in order to provide the optimum subband complexity-synchronisation trade-off remains unanswered. From informal experimentation, in the form of listening tests by the author[23], it is found that the number of subbands required is dependent on the nature of the input; for example, for speech or single instrument monophonic music just one subband produces the best results i.e. no partitioning of the input, since synchronisation issues do not arise and the broadband input is suitably periodic. In general it is found, from informal listening tests by the author, that partitioning the input into 12 bark based subbands produces good results. The cutoff frequencies of the filterbank, in hertz, for music signals sampled at 44.1kHz, are then {0, 200, 400, 630, 920, 1270, 1720, 2320, 3150, 4400, 6400, 9500, 15500, 22050}.

Additional considerations can be taken for high frequency components of the input. In [Levine '98b] a sinusoidal modelling based approach is used to time-scale audio; the proposed model assumes that the content of music signals above 5kHz is noisy in nature since 'for must music (but not all), there exists very few isolated tonal elements above 5000 Hz '. In addition [Levine '98b] states 'We could have included an additional octave of sinusoids, but this would have added a considerable amount to the total bit rate, and would only benefit a very small percentage of sound examples'. Following from Levines observations, frequencies above 5(or10) kHz could be grouped into a single subband, as there is no need to partition a noisy signal further, since the resulting subbands would still be noisy. Such an approach results in an improvement in quality for the case where transients occur in the signal, since synchronisation is maintained by a greater degree over the perceptually important high frequencies components of the transient; loss in synchronisation during transients results in transients sounding harsh and metallic, as observed by the author. However, for music in which high frequency notes are played i.e. those for which significant harmonics reside in the >5kHz range, the introduction of additional distortion is likely. An alternative approach to maintaining synchronisation at transients is given

[23] Author testing was undertaken in a quite office environment on a PC using headphones. Test signals comprised of approximately 10 music signals covering a range of genres, and time-scale factors in the range 0.5 – 2 were applied.

in section 6.2.1.2, whereby synchronisation is forced at transients, in a similar manner to that of [Lee '97] , [Bonada '00], [Duxbury '02] and [Röbel '03].

6.1.2 Choice of SOLA Parameters

Chapter 4 provides an analysis of the SOLA algorithm, which resulted in the derivation of the following equations for SOLA based parameters

$$S_a = \frac{L_{stat} - P}{|1-\alpha|} \tag{6-1}$$

$$N = L_{\max} + \alpha\left(\frac{L_{stat} - P}{|1-\alpha|}\right) \tag{6-2}$$

Since the maximum segment discarded/repeated is L_{stat} and the minimum is $L_{stat} - P$, the difference between the maximum and minimum discarded segments is P; therefore the choice of P controls the level of synchronisation between subbands i.e. if P is small then the loss of synchronisation between subbands is also small. However, P must be sufficiently large so as to be greater than or equal to the longest likely period of the subband being analysed. Determining the longest likely period within a subband is difficult, due to the fact that a number of components may exist within a subband and that the period of the subband is given by the highest common denominator of the differences between their frequencies. [Verhelst '03b] also recognised the difficulty in determining optimum parameters and suggests a manual iterative approach to determine them, starting with a search range (which equates to the P parameter) of 20 ms for low frequency subbands and 10 ms for high frequency subbands. [Tan '00] suggests a 20 ms search range for the lowest subband and uses 10 ms for all other subbands. Informal experimentation, in the form of listening tests by the author[24], found that the most significant variations in output quality arose when P is altered in high frequency subbands when transients occur in the signal being

[24] Author testing was undertaken in a quite office environment on a PC using headphones. Test signals comprised of approximately 10 music signals covering a range of genres, and time-scale factors in the range 0.5 – 2 were applied.

tested; the choice of a small value for P resulted in the transients sounding less metallic (as also observed when high frequency subbands are grouped into a single subband). In general it is found that setting P to 5 ms , 10 ms , 15 ms and 20 ms for subbands with lower cutoff frequencies greater than 6400 Hz , 1720 Hz , 630 Hz and 0 Hz , respectively, provides a suitable trade-off in terms of providing an adequate search region versus the reduction of potential subband synchronisation problems. Small variations in the choice of L_{stat} and L_{max} (+/- 3 ms) did not have a significant impact in the quality of output; in general fixing L_{stat} and L_{max} to $3/2P$ results in a good quality output.

6.1.3 Subjective Output Quality Comparison

Ten evaluation subjects carried out an informal listening test (see Appendix A). The test comprised of ten comparisons between a music track time-scaled using a bark subband approach (12 bark based subbands) and the same track time-scaled using a SASOLA subband approach (17 uniform subbands), applying the same time-scale factor. The tracks covered rock, pop, country and classical genres. The subjects were not informed which track was a SASOLA time-scaled track or which was a bark subband time-scaled track. The tests made use of time-scale factors of 1.5 and 2. These relatively large time-scale factors were chosen so that artefacts could be clearly heard and identified by untrained listeners. For all tests the sampling rate was 44.1kHz. Test signals are available in the Electronic Appendix on the CD which accompanies this dissertation.

Test subjects indication	**% of total comparisons**
Bark based approach much better than SASOLA	22 %
Bark based approach slightly better than SASOLA	38 %
Bark based approach equal to SASOLA	22 %
Bark based approach slightly worse than SASOLA	15 %
Bark based approach much worse than SASOLA	3 %

Table 6-2 Summary of listening test results comparing the use of bark based and uniform subbands within a time-domain/subband implementation for time-scale modification of polyphonic music.

Referring to appendix C, the mean ‘average score’ is 2.39 and the standard deviation is 0.48. This provides a greater than 95% T-test probability that there is an improvement in the perceived quality of output produced by the bark-based algorithm.

The results of the listening tests are summarised in Table 6-2.

In a separate test, a set of speech signals[25] were time-scaled using the bark based and uniform partitioning schemes; the test procedure was as set out above. Once again referring to appendix C, the mean 'average score' is 3.5 and the standard deviation is 0.25. This provides a greater than 95% T-test probability that there is an reduction in the perceived quality of output produced by the bark-based algorithm. The results of the test indicate a preference for the uniform partitioning approach, see Table 6-3. This is likely to be due to the fact that there are less synchronisation issues between the most perceptually relevant frequency components of the signal i.e. frequencies less than 4kHz[26], and that narrower subbands are not required in this frequency range in order to produce a high quality output. Better synchronisation between subbands results in less reverberation being introduced into the time-scaled output.

Test subjects indication	**% of total comparisons**
Bark based approach much better than SASOLA	1%
Bark based approach slightly better than SASOLA	15.2%
Bark based approach equal to SASOLA	45.5 %
Bark based approach slightly worse than SASOLA	34.3%
Bark based approach much worse than SASOLA	4%

Table 6-3 Summary of listening test results comparing the use of bark based and uniform subbands within a time-domain/subband implementation for time-scale modification of speech.

[25] Speech signals from the TIMIT speech corpus [TIMIT '04] were resampled from 16kHz to 44.1kHz.

[26] In telephony systems speech is bandlimited to 300 Hz -3400 Hz .

6.2 Improved Subband Synchronisation

6.2.1 Choice of Subband Offsets

When dealing with quasi-periodic signals the normalised correlation function of the SOLA algorithm, $R_m(k)$ generally returns a periodic signal with prominent peaks corresponding to the pitch period of the input; a fact that has been exploited by a number of pitch detection algorithms e.g. [Rabiner '77]. The SOLA algorithm chooses a synthesis offset, k_m, related to the most prominent or maximum peak of the correlation function. In general, however, any offset that is related to any of the prominent peaks of the correlation function could be used and result in a high quality output. Whilst choosing the offset that corresponds to the maximum peak in the correlation function is an obvious choice when SOLA is directly applied to a broadband input signal, for a subband implementation the offset for each subband should be chosen so as to minimise the delay differences between subbands in order to reduce the amount of reverberation/phasiness introduced into the output.

In attempting to determine the 'best' offset for each subband synthesis frame a set of suitable offsets must be established. One method of determining prominent peaks, and hence suitable offsets, is to first locate all peaks in the subband correlation function, $R_{m,i}(k)$, where a peak is defined as a sample that is greater than its two nearest neighbours and the i subscript represents the i^{th} subband. Then, any peak of the normalised correlation function that is within a certain percentage tolerance of the maximum peak's magnitude is considered a candidate peak, from which corresponding candidate offsets can be found; a tolerance of 10% has been found, through subjective listening tests by the author[27], to produce good quality results. These set of subband candidate offsets are denoted $\{k_{c1,i}, k_{c2,i}, k_{c3,i}, \ldots, k_{cp,i}\}$. One approach to determine the 'best' offset from this set of candidates is to provide a

[27] Author testing was undertaken in a quite office environment on a PC using headphones. Test signals comprised of approximately 10 music signals covering a range of genres and 10 speech signals from the TIMIT speech corpus, and time-scale factors in the range 0.5 – 2 were applied.

global target offset, k_{target}, to which all subband offsets should be focused i.e. for each set of subband candidate offsets the offset that is closest to the global target is used. It is proposed that k_{target} be chosen such that

$$R_{m,sum}(k) = \sum_{i=1}^{J} R_{m,i}(k).E_i.W_i \tag{6-3}$$

is a maximum for $k = k_{target}$, where J is the number of subbands, E_i is the energy in subband i in the region of the maximum overlap and W_i is a subband perceptual weighting factor. $R_{m,sum}(k)$ is then most influenced by subbands with the greatest energy and W_i provides an additional weight towards those subbands that are perceptually louder. The standard 'A' loudness-weighting curve is used in calculating W_i for each subband, where the centre frequency for each of the J subbands is used to determine the relevant weighting factor from the 'A' weighting curve.

Then, as described above, the offset for the i^{th} subband, $k_{m,i}$, is chosen such that

$$D_{m,i}(k_c) = \left|k_{target} - k_c\right| \tag{6-4}$$

is a minimum for $k_c = k_{m,i}$ with k_c being every element in the set of candidate offsets in the i^{th} subband i.e. $\{k_{c1,i}, k_{c2,i}, \ldots, k_{cp,i}\}$.

It should be noted that the approach for determining the 'best' offset for each subband described above requires that the same analysis parameters, S_a and N, be applied to all subbands.

6.2.1.1 Synchronisation during silent/masked regions

Masked regions and regions of silence within subbands can be utilised for subband synchronisation purposes. Consider the case where, for an iteration of the SOLA algorithm, within one of the subbands, the energy in the overlapping region of the overlapping synthesis frames falls below the threshold of hearing or the masking threshold [Zwicker '99]; then any offset could be used to overlap the frames without the introduction of audible distortion (once an adequate overlap for cross-fading is provided so as to eliminate the possibility of clicking). When this situation occurs within a subband, the offset for that subband is set to the global target offset, k_{target}, described above, thereby improving synchronisation between subbands.

For the case where a region of silence or masked region occurs in some position along a synthesis frame other than within the overlapping region, as shown in Figure 6-6, some level of synchronisation can once again be established. Improved synchronisation is achieved by altering the length of the silent/masked region from L_r to $L_r + (k_{target} - k_{m,i})$, where $k_{m,i}$ is the offset used by the subband in the overlapping region, thereby ensuring that all portions of the subband after the silent/masked region are synchronised to the global target. The expansion/compression of the silent/masked region r of length, L_r, assuming $L_r \geq SR$, is achieved by replacing r in the frame with $r_{replacement}$, of length $L_r + k_{target} - k_{m,i}$, where

$$r_{replacement}(j) = r(j), \text{ if } j \leq k_{target} - k_{m,i} \qquad (6\text{-}5)$$

$$r_{replacement}(j) = (1-f(j))r(j) + r(k_{m,i} - k_{target} + j)f(j),$$
$$\text{if } k_{target} - k_{m,i} < j < L_r$$
$$r_{replacement}(j) = r(k_{m,i} - k_{target} + j), \text{ if } j \geq L_r \qquad (6\text{-}6)$$

for $1 \leq j \leq L_r + k_{target} - k_{m,i}$,

where

$$f(j) = (j - \max(k_{target} - k_{m,i}, 1)) / (L_r - |k_{target} - k_{m,i}| - 1) \qquad (6\text{-}7)$$

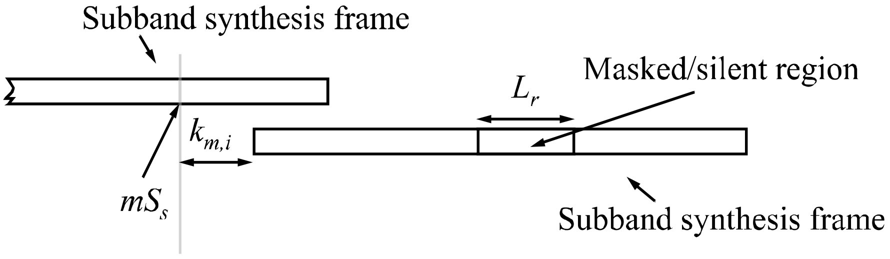


Figure 6-6 A masked/silent region within a synthesis frame.

6.2.1.2 Synchronisation of transients

Transients have posed a problem for all time-scale modification algorithms, both time-domain and frequency-domain, and must be treated differently to other portions of the signal in order to produce a high quality output. Typical artefacts related to time-domain handling of transients are the repetition or skipping of transients and, for a subband approach, the introduction of a harsh metallic effect within the transient portion. In [Lee '97] a solution to the transient handling problem within (non-subband) SOLA is proposed in which transient portions of the input are translated to their new time-scaled positions without modification, therefore keeping them in a synchronised state, while non-transient portions are time-scaled to a greater degree to ensure that the overall signal is time-scaled to the desired duration. Handling transients in this manner has an added advantage for a subband implementation since, as well as removing any harshness from time-scaled transients, it brings subsequent subband frames back into a synchronised state.

6.2.2 Subjective Output Quality Comparison

12 evaluation subjects of various age and gender carried out an informal blind listening test (see Appendix A). The test comprised of 12 comparisons between a variety of music and speech (6 speech and 6 music) tracks time-scaled using a subband approach both with and without the synchronisation schemes described previously. The test used time-scale factors ranging from 0.66 to 2 and all tracks were sampled at 44.1 kHz[28]. The 'non-synchronised' tracks were time-scaled using the same parameters given in section 6.1. The 'synchronised' tracks were partitioned into subbands using the same cutoff frequencies as in section 6.1, with *P* set to 20 ms , for all subbands. Test signals are available in the Electronic Appendix on the CD which accompanies this dissertation.

Referring to appendix C, the mean 'average score' is 2.44 and the standard deviation is 0.39. This provides a greater than 99.5% T-test probability that there is an improvement in the perceived quality of output produced by the algorithm. The

[28] Speech signals from the TIMIT database were resampled from 16Khz to 44.1kHz.

results of the listening tests indicate a preference for both speech and music time-scaled using the synchronisation schemes. The results of the listening tests are summarised in Table 6-4 and Table 6-5. The results for speech show a stronger improvement in quality in comparison with the music results. This can be attributed to the fact that music typically contains more reverberation than speech, therefore a reduction, or introduction, of reverberation will be more difficult to perceive. It should be noted that in section 6.1.3 it was found that test subjects preferred a uniform filterbank SASOLA implementation over a non-synchronised bark based filterbank approach for the time-scale modification of speech; however, the introduction of subband synchronisation results in a similar quality output being produced for both uniform and bark based approaches, when applied to speech, as found through subjective listening tests by the author[29].

Test subjects indication	% of total comparisons
Synchronisation (sync) much better than no sync.	11.1 %
Sync slightly better than no sync.	29.2 %
Sync approach equal to no sync.	50.0 %
Sync slightly worse than no sync.	9.7 %
Sync much worse than no sync.	0.0 %

Table 6-4 Summary of listening test results comparing the use of subband synchronisation techniques within a time-domain/subband implementation for the time-scale modification of polyphonic music.

[29] Author testing was undertaken in a quite office environment on a PC using headphones. Test signals comprised of approximately 10 speech signals from the TIMIT speech corpus, and time-scale factors in the range 0.5 – 2 were applied.

Figure 6-7 illustrates the effect of synchronisation upon a small excerpt from a time-scaled oboe signal. It can be seen that the temporal structure of the original waveform is maintained to a greater degree when synchronisation techniques are employed; the preservation of the waveform structure (shape invariance) is used in [Quatieri '92], [Pollard '97], [Di Federico '98], [O' Brien '99] and [Laroche '03] as an indication of a reduction in reverberation/phasiness. Figure 6-8 illustrates the effect of synchronisation upon an excerpt from a time-scaled signal composed of a guitar and castanets; the transient portion is preserved and is not subject to spreading. In addition, an objective test was also undertaken, whereby the average 'delay difference' between subbands was calculated for the subband approach outlined above both with and without synchronisation schemes being applied. The average 'delay difference' measure, D_{avg}, is given by

$$D_{avg} = \sum_{w=1}^{J} \left(\sum_{m=1}^{M_w} \sum_{i=1}^{J} \left| k_{m,w} - k_{n,i} \right| \Big/ M_w J \right) \Big/ J \qquad \text{(6-8)}$$

where M_w is the number of overlapping synthesis frames in the w^{th} subband, J is the number of subbands, $k_{m,w}$ is the offset of the w^{th} subband, for the m^{th} synthesis frame. $k_{n,i}$ is the offset of the i^{th} subband, for the n^{th} synthesis frame, where n is chosen such that $|mS_{s,w} - nS_{s,i}|$ is minimised for $1 \le n \le M_i$. When synchronisation schemes are employed $k_{n,\mathrm{i}} = k_{m,\mathrm{i}}$, since the same analysis parameters are applied to all subbands.

In one comparison 250 test signals from the TIMIT database [TIMIT '05] were time-scaled by factors ranging from 0.4 to 2. The results of the comparison show a reduction in relative delay differences between subbands of 33%. The comparison also shows that the reduction in relative delay differences is greater for lower frequencies; for the case where only frequencies up to 1720 Hz are considered the reduction becomes 51%.

In a second comparison randomly selected ten second snippets were extracted from 100 recordings the RWC music genre database [RWC '05] time-scaled by factors ranging from 0.4 to 2. The results of the comparison show a reduction in relative delay differences between subbands of 22%. As in the case of speech, the comparison also shows that the reduction in relative delay differences is greater for lower frequencies; for the case where only frequencies up to 1720 Hz are considered the reduction becomes 38%.

The greater reductions in relative delay differences of speech over music can be attributed to the fact that speech has significantly less power in the higher frequency range, with the upper frequency subbands often containing what is deemed to be silence; These ‘silent’ subbands are then readily synchronised.

Test subjects indication	% of total comparisons
Synchronisation (sync) much better than no sync.	22.2%
Sync slightly better than no sync.	36.1 %
Sync approach equal to no sync.	31.9 %
Sync slightly worse than no sync.	8.3 %
Sync much worse than no sync.	1.5 %

Table 6-5 Summary of listening test results comparing the use of subband synchronisation techniques within a time-domain/subband implementation for the time-scale modification of speech.

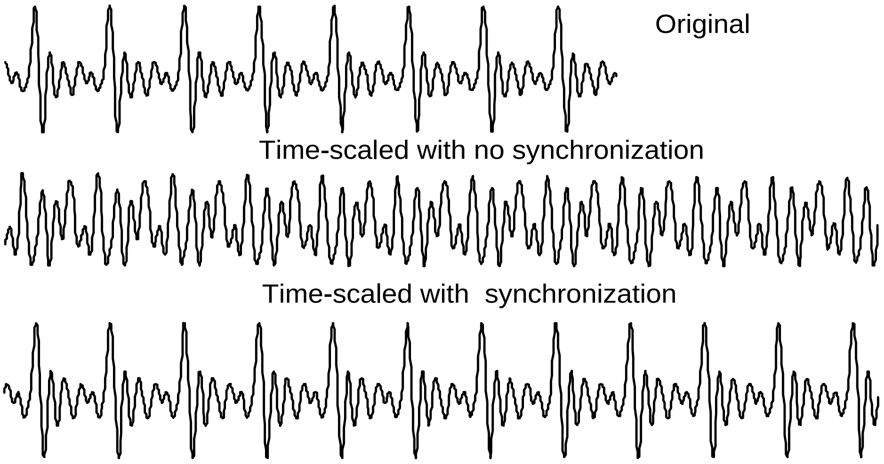


Figure 6-7 The effects of subband synchronisation on an oboe signal.

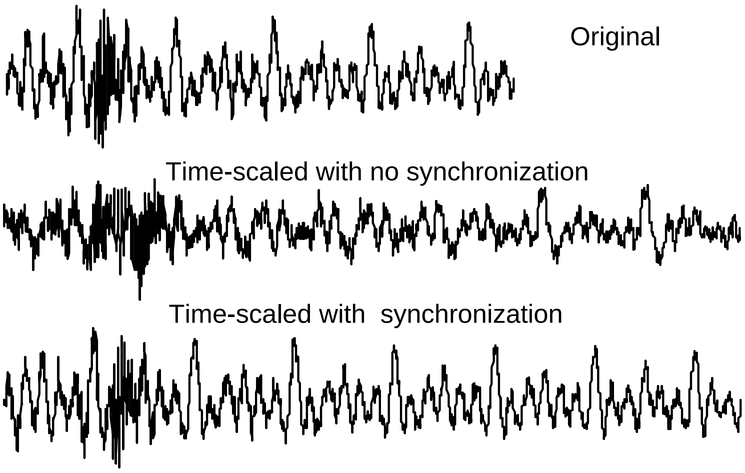


Figure 6-8 The effects of synchronisation on a guitar and castanets signal.

Whilst the synchronisation procedures described in this section generally result in an improvement in the quality of output produced by time-domain/subband approaches, the results are of lesser quality to those produced by the improved phase vocoder, as found through informal subjective listening tests by the author[30]. In the following section subband synchronisation is revisited, but within the context of a phase vocoder implementation.

[30] Author testing was undertaken in a quite office environment on a PC using headphones. Test signals comprised of approximately 10 music signals covering a range of genres and ten speech signals from the TIMIT speech corpus, and time-scale factors in the range 0.5 – 2 were applied.

7 Phase Vocoder Improvements

Phase vocoder approaches to time-scale modification of audio introduce a reverberant/phasy artefact into the time-scaled output due to a loss in phase coherence between short-time Fourier transform (STFT) bins. Recent improvements to the phase vocoder [Laroche '99a] have reduced the presence of this artefact; however, it remains a problem of note. A method of time-scaling is presented in this chapter that results in a further reduction in phasiness, for moderate time-scale factors, by taking advantage of some flexibility that exists in the choice of phase required so as to maintain horizontal phase coherence between related STFT bins. Furthermore, the approach leads to a reduction in computational load within the range of time-scaling factors for which phasiness is reduced.

A number of variations of the phase vocoder exist; the technique used in this chapter is based upon that used in [Bonada '00], which is described in section 2.4.3 and for convenience is summarised in this section as follows:

The first step is to obtain an STFT representation, $X(t_u,\Omega_k)$, of the input

$$X(t_u,\Omega_k)=\sum_{n=-\infty}^{\infty}h(n)x(t_u+n)e^{-j\Omega_k n} \qquad (7\text{-}1)$$

where x is the input signal, $h(n)$ is the analysis window, Ω_k is the centre frequency of the k^{th} vocoder channel and t_u is the u^{th} analysis time instant and $t_u = uR$, where R is the analysis (and synthesis) hop size and u is a set of successive integer values, starting at 0.

Time-scale modification is achieved by appropriately discarding or repeating STFT frames. The magnitudes of the modified, time-scaled, STFT remain unaltered i.e.

$$\left|Y(t_m,\Omega_k)\right|=\left|X(t_n,\Omega_k)\right| \text{ for all } k \qquad (7\text{-}2)$$

where n = round(m/α), m is a set of successive integer values starting at 0, t_n and t_m are a set of analysis and synthesis time instants, respectively, and α is the time-scale factor.

The phases of the modified STFT, $\angle Y(t_m,\Omega_k)$, are determined so as to attempt to maintain both horizontal and vertical phase coherence. To achieve phase coherence, first the peaks, representing the dominant components of each frame are identified. A peak is defined as any bin whose magnitude is greater than its four nearest neighbours, as given in [Laroche '99a]. In the simplest, most efficient, implementation phases of peaks are updated by maintaining the same phase difference between consecutive synthesis frames that exists between corresponding analysis frames i.e.

$$\angle Y\left(t_m,\Omega_{k_p}\right)-\angle Y\left(t_{m-1},\Omega_{k_p}\right)=\angle X\left(t_n,\Omega_{k_p}\right)-\angle X\left(t_{n-1},\Omega_{k_p}\right) \text{ for all } k_p \qquad (7\text{-}3)$$

which becomes

$$\angle Y\left(t_m,\Omega_{k_p}\right)=\angle Y\left(t_{m-1},\Omega_{k_p}\right)+\angle X\left(t_n,\Omega_{k_p}\right)-\angle X\left(t_{n-1},\Omega_{k_p}\right) \text{ for all } k_p \qquad (7\text{-}4)$$

where k_p are the bins of the detected peaks.

More complex techniques require the use of peak tracking techniques, as used in sinusoidal modelling (section 2.5).

Having determined the phases of the synthesis peaks, the phases of bins in each peak's region of influence are updated by maintaining the same phase difference between peaks and the bins in their region of influence that exists in the mapped analysis frame. The upper limit of the region of influence of a peak is set to the middle frequency between that peak and the next one [Laroche '99a]. The lower limit is set equal to the next bin above the previous upper limit. Then

$$\angle Y\left(t_m,\Omega_k\right)=\angle Y\left(t_m,\Omega_{k_p}\right)+\angle X\left(t_n,\Omega_k\right)-\angle X\left(t_n,\Omega_{k_p}\right) \qquad (7\text{-}5)$$

for all k in each peak's region of influence.

A time-scaled version of the original signal is then obtained by calculating the inverse STFT of $Y(t_m,\Omega_k)$.

7.1 Flexibility of Horizontal Phase Coherence

The inverse STFT of a given STFT is found by calculating the inverse discrete Fourier transform (IDFT) of each STFT frame. Successive inverse STFT frames are then overlapped and added together to produce the time-domain signal. A single iteration of the overlap and add process is illustrated in the upper three waveforms of Figure 7-1, where the first and second halves of two frames of a sinusoidal signal are overlapped and summed together to reproduce a perfect sinusoid. Now consider the case where the overlapping frames are no longer perfectly synchronised i.e. they are slightly out of 'horizontal' phase, as illustrated by the lower three waveforms of Figure 7-1. When the 'out of horizontal phase' sinusoids are summed together the resulting signal is no longer a perfect sinusoid but is a quasi-sinusoidal signal modulated in both amplitude and frequency. As expected intuitively, the greater the relative phase difference between the sinusoidal frames the greater the modulation that is introduced. From [Zwicker '99], human hearing is insensitive to certain amounts of frequency and amplitude modulations, and in an effort to determine the maximum phase difference that can be introduced without introducing audible distortion a set of equations representing the situation described above is derived.

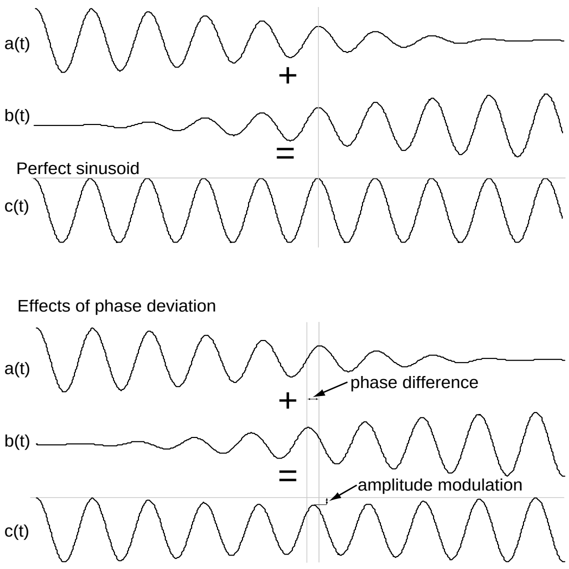


Figure 7-1 The effect of the loss of horizontal phase coherence upon a pure sinusoid within a phase vocoder frame.

The first step in achieving this aim is to describe the above situation through the use of a vector representation. From Figure 7-2, the ramped sinusoidal components are represented by the vectors ***a*(*t*)** and ***b*(*t*)**, which vary with time, according to the ramping function, but are constantly separated in phase by θ, and which sum to produce vector ***c*(*t*)**.

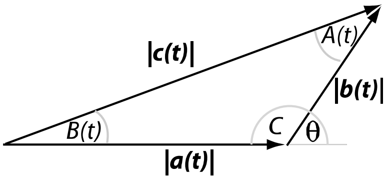


Figure 7-2 Vector representation of Figure 7-1.

From the well known cosine-rule, the magnitude of ***c*(*t*)** is given by

$$|\boldsymbol{c}(\boldsymbol{t})| = \sqrt{|\boldsymbol{a}(\boldsymbol{t})|^2 + |\boldsymbol{b}(\boldsymbol{t})|^2 - 2|\boldsymbol{a}(\boldsymbol{t})||\boldsymbol{b}(\boldsymbol{t})|\cos C} \tag{7-6}$$

where $C = \pi - \theta$ radians.

Typically, a hanning window is used within a phase vocoder implementation, therefore, if the magnitude of the original sinusoid is normalised to one, |***a*(*t*)**| is given by

$$|\boldsymbol{a}(\boldsymbol{t})| = 0.5(\cos(\pi t / L) + 1) \tag{7-7}$$

where L is the duration of the overlap and $0 \leq t \leq L$.

The sum of |***b*(*t*)**| and |***a*(*t*)**| must be one for perfect reconstruction, therefore

$$|\boldsymbol{b}(\boldsymbol{t})| = 1 - |\boldsymbol{a}(\boldsymbol{t})| \tag{7-8}$$

To determine the maximum variation in |***c*(*t*)**| the derivative of |***c*(*t*)**| with respect to *t* is found, then set to zero and solved for *t*. It should be noted that value of |***c*(*t*)**| at the boundary conditions, i.e. t=0 and t=L, is always 1 and that it is the variation from this value that is important in this analysis. It can be shown[31] that when

$$\frac{d|\boldsymbol{c}(\boldsymbol{t})|}{dt} = 0 \tag{7-9}$$

t = L/2 provides the only non trivial solution. Therefore, the maximum amplitude variation is given by

$$1 - |\boldsymbol{c(L/2)}| = 1 - \sqrt{0.5^2 + 0.5^2 - 2(0.5)(0.5)\cos C} \qquad \text{(7-10}$$

$$= 1 - \sqrt{0.5 + 0.5\cos\theta} \tag{7-11}$$

since the magnitude of the original sinusoid has been normalised to one, $C = \pi - \theta$ radians and $|\boldsymbol{a}(\boldsymbol{L}/2)| = 0.5$.

From [Zwicker '99], the human ear is insensitive to amplitude variations of tones, introduced by sinusoidal amplitude modulation, for degrees of modulation that are less than 2% for tones that are less than 80dB. It is important to note that the total variation in amplitude from a maximum to a minimum is twice the degree of modulation. This value varies significantly with pressure levels, for example for a pure tone of pressure level 40dB the degree of modulation increases to 4% while at 100dB it decreases to 1%. These values are independent of the frequency of the tone. It should also be noted that, from [Zwicker '99], these values are dependent on the frequency of modulation, but the values given above are based on the modulating frequency at which human hearing is most sensitive. It can be shown that the amplitude modulation of $\boldsymbol{c(t)}$ is quasi-sinusoidal in nature, with the degree of modulation, D_m, given by, from equation 7-11),

[31] MAPLE^{TM} software was used extensively thoughout this chapter to solve mathematical problems. The MAPLE code employed to determine solutions is included in the Electronic Appendix on the CD which accompanies this dissertation.

$$D_m = \left(1 - \sqrt{0.5 + 0.5\cos\theta}\right)/2 \qquad (7\text{-}12)$$

where the divisor of 2 is required since the degree of modulation is half the total variation in amplitude [Zwicker '99].

By making the assumption that maximum pressure levels of tonal components of the signals being analysed are below 80dB, the degree of modulation of $|\boldsymbol{c(t)}|$ must then be kept below 2%. So, from equation 7-12)

$$\left(1 - \sqrt{0.5 + 0.5\cos\theta}\right)/2 \le |0.02| \qquad (7\text{-}13)$$

Therefore

$$\theta \le |0.5676| \text{ radians} \qquad (7\text{-}14)$$

to ensure no perceivable amplitude modulations are introduced.

It should be noted that the amplitude modulation introduced results in an average decrease in signal amplitude level, however, the decrease is within the just noticeable amplitude level difference, as given in [Zwicker '99], if equation 7-14) is satisfied.

$B(t)$ represents the time-varying phase variation between $\boldsymbol{a(t)}$ and $\boldsymbol{c(t)}$ and, from the well known sine-rule, is given by

$$B(t) = \sin^{-1}\left(\frac{|\boldsymbol{b(t)}|\sin C}{|\boldsymbol{c(t)}|}\right) \qquad (7\text{-}15)$$

then

$$\frac{dB(t)}{dt} = \frac{\sin C\left(|\boldsymbol{c(t)}|\dfrac{d|\boldsymbol{b(t)}|}{dt} - |\boldsymbol{b(t)}|\dfrac{d|\boldsymbol{c(t)}|}{dt}\right)}{|\boldsymbol{c(t)}|^2 \cos B(t)} \qquad (7\text{-}16)$$

The frequency f_c of the quasi-sinusoidal component $c(t)$ is given by

$$f_c = f_a + \frac{dB(t)}{dt} \text{ rads/second} \tag{7-17}$$

where f_a is the frequency of the sinusoidal component ***a*(*t*)**.

Since f_a is constant, the derivative of the $B(t)$ with respect to t represents the frequency modulating component of f_c. The maximum frequency modulation is determined by first finding the derivative of f_c with respect to t, setting it to zero and solving for t. Then

$$\frac{df_c}{dt} = \frac{d^2B(t)}{dt^2} \tag{7-18}$$

and when 7-18) is set to zero it can, once again, be shown that $t = L/2$ provides the only non trivial solution. Therefore, it can be shown that the maximum frequency deviation is given by

$$\frac{dB(L/2)}{dt} = \frac{\pi}{L}\tan\left(\frac{\theta}{2}\right) \tag{7-19}$$

From [Zwicker '99], the human ear is insensitive to frequency variations introduced by frequency modulation; for tones greater than 500 Hz , modulations less than 0.7% are not perceived and for tones less than 500 Hz , a fixed modulation of 3.6 Hz is tolerated. Once again, these values are dependent on the frequency of modulation, however the values given above are based on the modulating frequency at which the human ear is most sensitive. Therefore, in order to ensure the ear does not perceive distortion for any frequency, the variation of f_c must be kept below 3.6 Hz or 22.62 radians/second. So, from equation 7-19) and setting L = 30 ms

$$\frac{\pi}{0.03}\tan\left(\frac{\theta}{2}\right) \leq |22.62| \qquad \text{radians} \tag{7-20}$$

Then

$$\theta \leq |0.4255| \qquad \text{radians} \tag{7-21}$$

From 7-14) and 7-21) the maximum phase deviation, Ψ_{max}, that can be introduced without introducing audible modulations is

$$\Psi_{max} = 0.\ 4255 \ \text{radians} \tag{7-22}$$

This value only strictly applies to frequencies less than 500 Hz , if the dependence of modulations on frequency is considered then Ψ_{max} could be increased to 0.5676 radians for frequencies greater than

$$\left(\frac{\pi}{.03}\tan\left(\frac{0.5676}{2}\right) \Big/ 2\pi \right) \frac{100}{0.7} = 694.46\text{Hz} \tag{7-23}$$

and varied accordingly between 0. 4255 and 0.5767 radians for all other frequencies.

The above analysis is carried out based on a single pure sinusoidal tone, however, most audio signals of interest are, for the most part, a sum of quasi-sinusoidal components, a feature exploited by sinusoidal modelling techniques (see section 2.5) and is the underlying assumption of the phase vocoder. It is assumed that the sum of sinusoids that have been amplitude and frequency modulated to the maximum limit, such that they are perceptually equivalent to the original individual sinusoids, results in a signal that is perceptually equivalent to the sum of the non-modulated sinusoids. Informal listening tests by the author in a quiet office environment support this assumption.

The above analysis is also based on an ‘ideal’ horizontal phase shift i.e. vertical phase coherence is maintained. Such a phase shift can be achieved in a straightforward manner with synthesised pure sinusoids but is difficult with real audio signals; this difficulty is, of course, the reason for the existence of the phasiness artefact in the first place. However, the above analysis does suggest that a certain amount of flexibility exists in the choice of phase in order to maintain horizontal phase coherence of dominant sinusoidal components. This is further supported by the fact that phase vocoder implementations are capable of producing high quality time-scale modifications even though frequency estimates, used in [Laroche ‘99a] to determine synthesis phases, are prone to inaccuracies [Puckett ‘98], [Abeysekera ‘01].

The derivation of amplitude and frequency modulations introduced due to phase deviation is based on a hop size of half the analysis window length. A similar, albeit more tedious, approach can be used to determine modulations introduced for the case of different hop sizes; a hop size of half the analysis window length is used in this section for its intuitive appeal and mathematical simplicity. Another commonly used hop size is one quarter of the analysis frame length, for which it can be shown that $\Psi_{max} \approx 0.27$ radians for analysis window lengths of 60 ms , see Appendix B.

It should be noted that the maximum phase deviations determined are based on an analysis of the maximum amplitude and frequency modulations perceptually tolerated within sinusoidal components in somewhat idealised conditions [Zwicker 99]. From informal listening tests by the author[32] the maximum phase deviations can be increased by as much as 20% for the case when complex signals are being time-scaled in a quiet office environment, without any perceived loss in quality. The inclusion of this additional tolerance without a perceived loss in quality can be attributed to the fact that an increase in the tolerance allows for a faster transition to perfect phase coherence, which offsets the distortion caused by increase in amplitude and frequency modulations.

[32] Author testing was undertaken in a quite office environment on a PC using headphones. Test signals comprised of approximately 10 music signals covering a range of genres and 10 speech signals from the TIMIT speech corpus, and time-scale factors in the range 0.5 – 2 were applied.

7.2 Reduction in Phasiness and Computations

In the previous section it is shown that a certain amount of flexibility exists in the choice of phase required to achieve horizontal phase coherence within a phase vocoder implementation. This flexibility can be used to 'push' or 'pull' modified STFT frames into a phase coherent state; however a set of coherent target phases for each frame are first required. One set of target phases that would ensure vertical phase coherence are the phases of the original frames that are mapped to each synthesis frame. So, having determined an estimate of the synthesis phases using the procedure described at the start of this chapter, the synthesis phases are updated further using the following rules:

If

$$\left|princ\_arg\left(\angle Y\left(t_m,\Omega_k\right)-\angle X\left(t_n,\Omega_k\right)\right)\right| \le \Psi_{\max} \tag{7-24}$$

then

$$\angle Y\left(t_m,\Omega_k\right)=\angle X\left(t_n,\Omega_k\right) \tag{7-25}$$

else

$$\begin{aligned}\angle Y\left(t_m,\Omega_k\right) &= \angle Y\left(t_m,\Omega_k\right)\\ &\quad + sign\left(princ\_arg\left(\angle Y\left(t_m,\Omega_k\right)-\angle X\left(t_n,\Omega_k\right)\right)\right)\Psi_{\max}\end{aligned} \tag{7-26}$$

where Ψ_{max}, is the maximum deviation in frequency, *sign* is a function that returns the sign of the submitted value i.e. 1 or –1 and *princ_arg* returns the principle argument of the submitted value between ±π.

For the following paragraphs it is important to be aware of two situations; the first situation is where consecutive analysis frames are mapped to consecutive synthesis frames e.g. in Figure 7-3 the consecutive analysis frames 2, 3 and 4 are mapped to three consecutive synthesis frames 3', 4' and 5', this case can be described more generally as the situation when $t_m \rightarrow t_n$ and $t_{m-1} \rightarrow t_{n-1}$; the second situation covers all other cases.

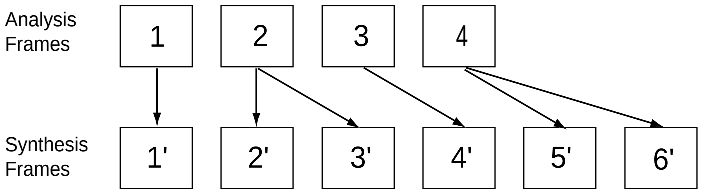


Figure 7-3 Frame mapping required to achieve the desired time-scaling.

It should be noted that for the case where consecutive analysis frames are not mapped to consecutive synthesis frames, Ψ_{max} should be reduced to take the likelihood of increased inaccuracies of phase estimates into consideration when using equation 7-4). Phase estimates of consecutive analysis frames that are mapped to consecutive synthesis frames are likely to be accurate, at least for peaks, since the same phase differences are kept between consecutive analysis frames as consecutive synthesis frames. The same cannot be said for the case where consecutive analysis frames are not mapped to consecutive synthesis frames; consequently it is difficult to determine a value for the maximum phase deviation that can be introduced for this case. From experimentation it was found that reducing Ψ_{max} to $\Psi_{max}/2$ is an adequate choice.

It should also be noted that, for the case where multiple consecutive analysis frames are mapped to multiple consecutive synthesis frames, a reduction in phase differences between one synthesis frame and its corresponding, mapped, analysis frame results in the same phase reduction for all consecutive synthesis frames that follow; since from equation 7-4) the phase modifications are propagated through the remaining synthesis frames. Following from this observation, it can be noted that if $(\pi-\Psi_{max}/2)/\Psi_{max}$ consecutive analysis frames are mapped to $(\pi-\Psi_{max}/2\ )/\Psi_{max}$ consecutive synthesis frames the phase coherence is guaranteed to be recovered for at least one of the consecutive synthesis frames (the $\Psi_{max}/2$ value represents the phase deviation introduced for non-consecutive synthesis frames). Therefore, the closer the time-scale factor is to one the greater the opportunity to recover phase coherence, since the number of consecutive analysis frames mapped to consecutive synthesis frames, w, is given by

$$w = 1/|1-\alpha| \tag{7-27}$$

It then follows that phase coherence is guaranteed to be recovered at least once every *w* frames if

$$\alpha > (\pi - 3\Psi_{max}/2)/(\Psi_{max}/2 - \pi) \text{ for } \alpha < 1 \tag{7-28}$$

or

$$\alpha < (\pi + \Psi_{max}/2)/ (\pi - \Psi_{max}/2) \text{ for } \alpha > 1 \tag{7-29}$$

Since phase coherence is ensured for some sections of the time-scaled output if equation 7-28) or 7-29) is satisfied, it follows that these sections are copies of sections of the input. Therefore, these 'copied' sections do not have to be processed in the frequency domain and can be simply overlapped and added to the time-scaled output; resulting in a reduction in the computational requirements of the approach. This process is illustrated in Figure 7-4, where the analysis frame marked B would achieve phase coherence and the synthesis frame marked A' is almost phase coherent i.e. all STFT bins of frame A' are within Ψ_{max} radians of the phase of the mapped analysis frame marked A.

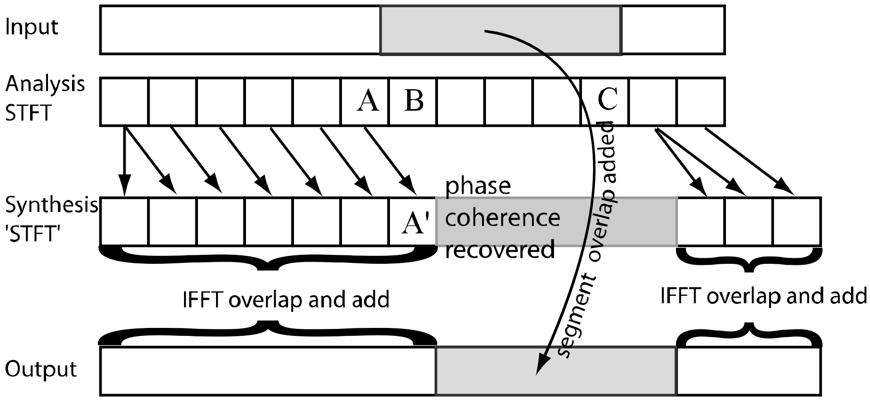


Figure 7-4 Copying a time-domain segment to the output.

The phases of the analysis frame marked C are required to calculate equation $_{7\text{-}4}$), therefore, given a set of analysis time instants $t_u = uR$, where u is a set of consecutive integer values starting at 0, the STFT needs only be calculated, at most, for the cases when

$$floor(u|1\text{-}\alpha|)/|1\text{-}\alpha| -1 \leq u \leq floor(u|1\text{-}\alpha|)/|1\text{-}\alpha|+ceil((\pi\text{-} \Psi_{max}/2)/\Psi_{max}) \tag{7-30}$$

where *ceil* and *floor* are functions that return the nearest integer greater than and less than the value submitted, respectively.

Figure 7-5 illustrates the computational advantage of the phasiness reduction technique; the vertical axis shows the ratio of computations of the standard phase vocoder to the computations of the phase vocoder that utilises the phasiness reduction technique presented in this chapter. The solid line is plotted for $\Psi_{max} = 0.4255$ radians and the dashed line is plotted for $\Psi_{max} = 0.27$ radians.

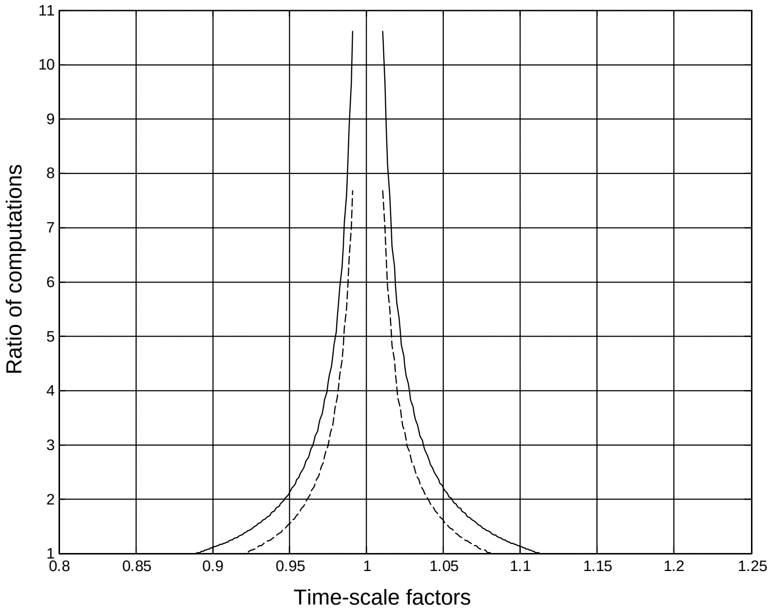


Figure 7-5 Ratio of computations required for the improved phase vocoder implementation to the number of computations required using the phasiness reduction technique.

7.3 Improving Initial Phase Estimates

In the previous section it is shown how the phase tolerance established can be used to push or pull synthesis STFT phases into a phase coherent state. So, given a set of phase coherent target phases the synthesis phases are gradually moved toward them. If the initial phase estimates are close to the target phases then the transition to perfect phase coherence would be reduced and therefore result in a further reduction in the level of reverberation present. The following paragraphs detail a hybrid implementation which draws upon time-domain synchronised overlap-add theory to provide a ‘good’ set of initial phase estimates to the phase vocoder based implementation described in the previous section.

The original motivation behind the SOLA algorithm (section 2.1.1) was to provide an initial set of phase estimates for the reconstruction of a magnitude only STFT representation of a signal [Griffen ‘84]. The same principle is used here to provide a set of phase estimates for use within the procedure outlined above.

Consider the situation shown in Figure 7-6, in which a frame extracted from the input is shown overlapping with the current output. As with the standard SOLA implementation the overlap shown is determined through the use of a correlation function. For the m^{th} iteration of the algorithm the offset τ_m is chosen such that the correlation function $R_m(\tau)$, given by

$$R_m(\tau)=\frac{\sum_{j=0}^{L_m-1} y(mS_s+\tau+j)x(mS_a+j)}{\sqrt{\sum_{j=0}^{L_m-1} x^2(mS_a+j)\sum_{j=0}^{L_m-1} y^2(mS_s+\tau+j)}} \tag{7-31}$$

is a maximum for $\tau = \tau_m$, where x is the input signal, y is the time-scaled output, L_m is the length of the overlapping region and τ is in the range $0 < \tau < \tau_{max}$, where τ_{max} is typically the number of samples which equates to approximately 20 ms. S_a is the analysis hop size of the SOLA component of the algorithm, and, from section 4, is given by

$$S_a = \frac{L_{stat} - SR}{|1-\alpha|} \tag{7-32}$$

where L_{stat} is the stationary length (approx 25-30 ms), SR is the search range over which τ_m is determined (approx 20 ms), and α is the desired time-scale factor. S_s is the synthesis step size, and is related to S_a by $S_s = \alpha S_a$.

The optimum frame overlap L_{ov} shown in Figure 7-6 is then given by

$$L_{ov} = N\text{-}\ S_s - \tau_m \tag{7-33}$$

where N is the frame length which, from section 4, is given by

$$N = SR + \alpha\left(\frac{L_{stat} - SR}{|1-\alpha|}\right) \qquad (7\text{-}34)$$

Also shown in Figure 7-6, below the input frame, are the synthesis windows and the synthesis frame. In this section frames refer to the frames of length N, associated with the SOLA component of the algorithm; the term window is reserved to refer to windows typical of a STFT analysis/synthesis. It is this synthesis frame which is appended to the current output within the hybrid approach and not the input frame, as is the case in SOLA. The following details the generation of the synthesis frame.

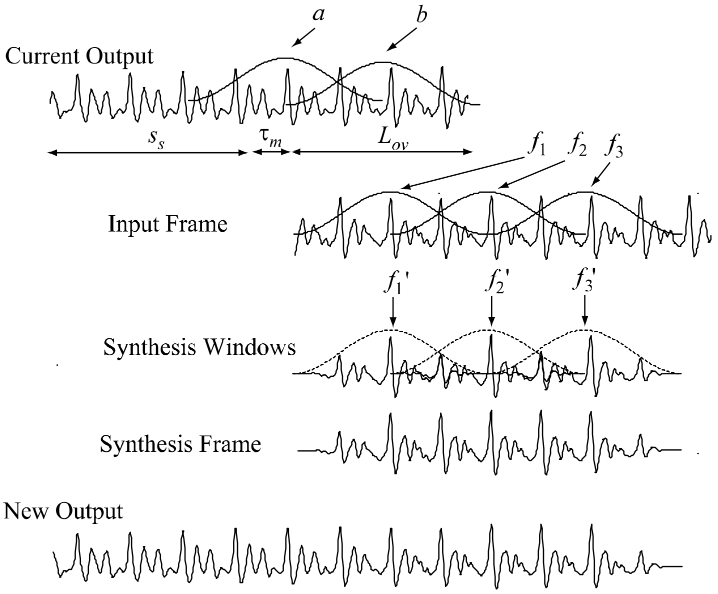


Figure 7-6 An iteration of the hybrid algorithm.

Window b is first extracted from the output y and is positioned such that it has its centre at the centre of the 'optimum' overlap, as shown in the diagram. More specifically, for the m^{th} iteration of the algorithm, frame b is given by

$$b(j) = y(mS_s + \tau_m + L_{ov}/2 - L/2 + j).w(j) \text{ for } 0 < j \leq L \qquad (7\text{-}35)$$

where w is the STFT analysis window, typically hanning, L is the STFT window length, typically the number of samples which equates to approximately 60 ms .

The window f_1 is extracted from the input x and is positioned such that it is aligned with frame b. Subsequent windows are sequentially spaced by the STFT hop size H. More specifically, for the m^{th} iteration of the algorithm window f_n is given by

$$f_n(j) = x(mS_a + L_{ov}/2 + H.(n-1) - L/2 + j).w(j) \text{ for } 0 < j \leq L \qquad (7\text{-}36)$$

$F_1^{'}$ the DFT representation of $f_1^{'}$, is then derived using the magnitudes of F_1 and the phase values B, where F_n and B are the DFT representations of f_n and b, respectively; then

$$F_1^{'}(k) = |F_1(k)| \exp(i\angle B(k)) \text{ for all } k \text{ in the set } P_1 \qquad (7\text{-}37)$$

where P_1 is the set of peak bins found in $|F_1|$. All other bins are updated so as to maintain the original phase difference between a peak and bins in its region of influence, as described in [Laroche '99a] (see section 2.4.4).The phase values of STFT window B are chosen since they provide a set of phase values that naturally follow the window labelled a in Figure 7-6 and therefore maintain horizontal phase coherence. Subsequent synthesis windows are derived from

$$F_n^{'}(k) = |\angle F_n(k)| \exp\left(i\left(\angle F_{n-1}^{'}(k) + \angle F_n(k) - \angle F_{n-1}(k) + D(k)\right)\right) \qquad (7\text{-}38)$$

for all k in the set P_n, where P_n is the set of peak bins found in $|F_n|$. As above, all other bins are updated so as to maintain the original phase difference between a peak and bins in its region of influence. For the hybrid case perfect phase coherence is achieved when synthesis STFT window $F_n^{'}$ has the magnitude and phase values of window F_n. D is the phase deviation which is used to push or pull the frames into a phase coherent state. D is dependent on the bin number denoted by k and is given by

$$D(k) = \angle F_{n-1}(k) - \angle F_{n-1}^{'}(k) \qquad \text{if princarg } \left(\angle F_{n-1}(k) - \angle F_{n-1}^{'}(k)\right) \leq \Psi_{\max} \qquad (7\text{-}39)$$

or

$$D(k) = sign\left(\angle F_{n-1}(k) - \angle F_{n-1}^{'}(k)\right)\Psi_{\max} \qquad \text{if princarg } \left(\angle F_{n-1}(k) - \angle F_{n-1}^{'}(k)\right) > \Psi_{\max}$$

(7-40)

where $\Psi_{\max}$ is the maximum phase tolerance (see section 7.1).

The number of synthesis STFT windows required is such that an inverse STFT on these windows results in a synthesis frame of duration N+3L/2. This is to ensure that window b is available for the next iteration of the algorithm. It should be noted that the number of synthesis windows also controls the ability of the algorithm to recover phase coherence; if N is large (which is the case when is α is close to one, see equation 4-13)) phase coherence is recovered more easily. The synthesis frame x_m is obtained through the application of an inverse STFT on windows $F_1^{'}$, $F_2^{'}$, $F_3^{'}$,…. The output y is then updated by

$$y(mS_s+\tau_m + L_{ov}/2–L/2 +j) := E(j).y(mS_s + \tau_m + L_{ov}/2 – L/2 +j) + x_m(j) \text{ for } 0<j\leq L–H \tag{7-41}$$

$$y(mS_s + \tau_m + L_{ov}/2 – L/2 +j) = x_m(j) \text{ for } L\text{-}H < j \leq N +3L/2 \tag{7-42}$$

where := in equation 7-41) means ‘becomes equal to’ and E is an envelope function which ensures that the output y sums to a constant during the overlap-add procedure.

E is dependent on the STFT hop size H and whether a synthesis window is employed during the inverse STFT procedure. For the case where a synthesis window is employed, which is equal to the analysis hanning window w, and $H = L/4$

$$E(j) = w^2(H + j) + w^2 (2H + j) + w^2 (3H + j) \text{ for } 0<j \leq L–H \tag{7-43}$$

It should be noted that for the case where the input is perfectly periodic the initial phase estimates provided by STFT window B are assured to be equal to the target phase values of window F_1 and the time-scaled output is always perfectly phase coherent. For quasi-periodic signals, such as speech, the initial phase estimates are generally close to the target phase, and the transition period to perfect phase coherence is generally short.

For the case where more complex audio is being time-scaled, the transition to perfect phase coherence is relatively long; nevertheless, the reverberant artefact introduced, due to the loss of perfect phase coherence, is perceptually less objectionable in these types of signals, due to the reverberation level generally already present.

7.4 Considerations for Stereo Recordings

In [Bonada '00] the implications of the application of a phase vocoder based time-scale modification algorithm to stereo recordings are outlined. Bonada maintains the stereo image by ensuring that both magnitude and phase differences between related channel components are preserved. Bonada notes that the magnitude differences are maintained within standard phase vocoder implementations if the same parameters are used to time-scale each channel. Bonada then preserves phase differences between related sinusoidal components.

Within the hybrid implementation, segments of different duration could be discarded/repeated from each channel if the channels are time-scaled separately; even if the same algorithm parameters are applied to each channel. This could result in an alteration of the stereo image, since magnitude differences between channels are unlikely to be maintained. The solution to this potential problem is to sum channels before applying the correlation function of equation 7-31). The offset identified, by finding the maximum of the correlation function, is then applied to both channels for each iteration of the algorithm.

Phase differences are preserved between peaks, at the same bin location, between channels, by first updating the peak with the greater magnitude in the manner described earlier; the peak with the lesser magnitude is updated so as to preserve the original phase relationship. Bins in the region of influence of a peak are updated in the usual manner.

7.5 Subjective Output Quality Comparison

Thirteen test subjects undertook eight subjective listening tests (see Appendix A) to compare the quality of time-scaled speech produced by the hybrid algorithm against a phase vocoder implementation. For both cases a 60ms STFT analysis and synthesis window were employed, whilst the hybrid algorithm used a search range, *SR*, equal to 20 ms, and the stationary length, L_{stat}, was set to 30 ms. The first set of tests were limited to relatively small time-scale factors, in the range of 0.8 – 1.25. Test signals are available in the Electronic Appendix on the CD which accompanies this dissertation. A summary of the test results are given in Table 7-1.

Referring to appendix C, the mean 'average score' is 1.72 and the standard deviation is 0.39. This provides a greater than 99.5% T-test probability that there is an improvement in the perceived quality of output produced by the algorithm. There is also an indication that the improvements are more noticeable within male speech; when only male speakers are considered 58.9% of test subjects results indicate that the hybrid approach is much better than the phase vocoder; when only female speakers are considered 25% of test subjects results indicate that the hybrid approach is much better than the phase vocoder. [Barry '05] offers an intuitively appealing explanation as to the reasons for this finding; Barry suggests that since the harmonic components of female speech are separated by a greater distance in frequency than in male speech, and since bins in the region of influence of a peak are divided evenly between harmonic components, there will be significantly more phase locking between peaks and nearby bins in female speech. Barry also noted that there will generally be fewer dominant sinusoidal components in female speech than male and, therefore, that phase coherence will play a more important role in male speech. This explanation is in keeping with the author's finding that improvements are also more noticeable within gravelly or rough speech; since, in this type of speech a number of additional subharmonic components are typically present, as found in [Loscos '04]. From [Loscos '04] it is also shown that the phase values associated with the subharmonic components do not adhere to the sinusoidal phase propagation formula, which is used within the standard phase vocoder algorithm; it therefore follows that standard phase vocoder will produce erroneous results when applied to these subharmonic components, whereas the hybrid approach will cater for these types of signal more readily, once the search range employed in determining the correlation function is of sufficient duration to encompass the 'growl macro period', as defined in

[Loscos '04].

Test subjects indication	% of total comparisons
Hybrid much better than phase vocoder	42.0%
Hybrid slightly better than phase vocoder	44.6 %
Hybrid equal to phase vocoder.	8.0 %
Hybrid slightly worse than phase vocoder.	5.4 %
Hybrid much worse than phase vocoder.	0.0 %

Table 7-1 Summary of listening test results comparing the use of the hybrid implementation against a phase vocoder implementation for the time-scale modification of speech for factors in the range 0.8-1.25.

In a second set of subjective listening tests, using the same algorithm parameters and format as outlined above, subjects were requested to compare the quality of time-scaled speech produced by the hybrid algorithm against a phase vocoder implementation for time-scale factors in the range 0.6 to 0.8 and 1.25 to 1.75. A summary of the test results are given in Table 7-2.

As for the case of smaller time-scale factors, there is a preference for the hybrid approach over the standard phase vocoder implementation; however, results suggest that the preference is less significant. This finding is in keeping with expectations, since the hybrid implementation is more likely to recover 'perfect phase coherence' for time-scale factors close to one. Referring to appendix C, the mean 'average score' is 2.12 and the standard deviation is 0.47. This still provides a greater than 99.5% T-test probability that there is an improvement in the perceived quality of output produced by the algorithm.

Test subjects indication	% of total comparisons
Hybrid much better than phase vocoder	23.8%

Hybrid slightly better than phase vocoder	41.3 %
Hybrid equal to phase vocoder.	30.0 %
Hybrid slightly worse than phase vocoder.	5.0 %
Hybrid much worse than phase vocoder.	0.0 %

Table 7-2 Summary of listening test results comparing the use of the hybrid implementation against a phase vocoder implementation for the time-scale modification of speech for factors in the range 0.6-0.8 and 1.25-1.75.

In a final set of subjective listening tests, using the same algorithm parameters and format as outlined above, subjects were requested to compare the quality of time-scaled music produced by the hybrid algorithm against a phase vocoder implementation. Time-scale factors in the range 0.6 to 1.75 were employed. A summary of the test results are given in Table 7-3.

Referring to appendix C, the mean ‘average score’ is 2.9 and the standard deviation is 0.44. This provides a less than 1% T-test probability that there is an improvement in the perceived quality of output produced by the algorithm, suggesting that results of the subjective test indicate no significant preference for either approach; this is attributed to the fact that there is generally a significant level of reverberation present in music, and that relatively small reduction, or introduction, of reverberation will be difficult to perceive.

Test subjects indication	% of total comparisons
Hybrid much better than phase vocoder	7.5%
Hybrid slightly better than phase vocoder	25.0 %
Hybrid equal to phase vocoder.	42.5 %

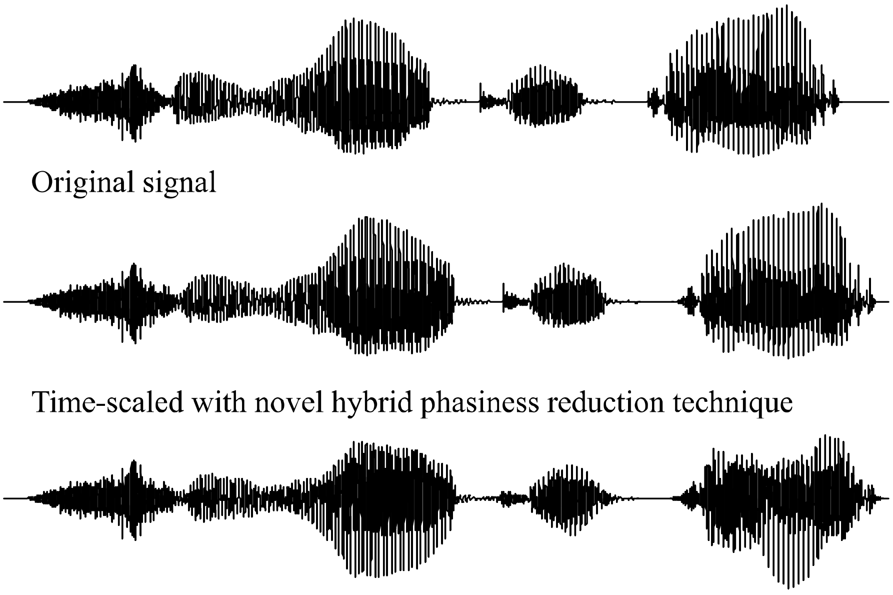
Hybrid slightly worse than phase vocoder.	20.0 %
Hybrid much worse than phase vocoder.	5.0 %

Table 7-3 Summary of listening test results comparing the use of the hybrid implementation against a phase vocoder implementation for the time-scale modification of music for factors in the range 0.6-1.75.

7.6 Discussion

The phasiness reduction technique described in this chapter has similarities with time-domain approaches, in that, for moderate time-scaling, certain segments of the time-scaled signal are a copy of the original, as is the case in time-domain approaches; the phase vocoder, however, has the advantage of producing better results for complex polyphonic audio. The technique also has similarities to the synchronised time-domain/subband approach described in section 6.2, where individual subbands are ‘pulled’ or ‘pushed’ into a synchronised state by taking advantage of psychoacoustic properties.

Figure 7-7 illustrates the effects of the phasiness reduction technique on a speech signal. It should be noted that while the preservation of the waveform shape, i.e. shape invariance, does not ensure phase coherence, the loss of shape invariance can be attributed to a loss of phase coherence; shape invariance is used in [Quatieri ‘92], [Pollard ‘97], [Di Federico ‘98], [O’ Brien ‘99] and [Laroche ‘03] as an indication of a reduction in reverberation/phasiness.



Original signal

Time-scaled with novel hybrid phasiness reduction technique

Time-scaled without phasiness reduction

Figure 7-7 The effects of the reduction of phasiness using the hybrid implementation on the utterance ‘She had your dark’.

8 Summary and Future Work

A review of existing time-scale modification techniques was undertaken, from which it was concluded that phasiness, a reverberant type artefact, is a notable issue within frequency-domain implementations. Time-domain techniques do not suffer from this artefact, however they rely on the existence of a quasi-periodic signal to produce a high quality output; a feature not always present in complex signals such as polyphonic music.

It was concluded that further investigation into the development of frequency-domain techniques that incorporate aspects of time-domain implementations, with a view to reducing the reverberant artefacts associated with frequency-domain implementations, was warranted.

Owing to the emphasis placed on the use of time-domain techniques as a means to reduce phasiness, it was concluded that a further analysis of time-domain implementations should be undertaken. The subsequent analysis resulted in the derivation of an efficient and flexible parameter set for use within time-domain, and time-domain/subband, implementations. Furthermore, the review highlighted the fact that a number of synchronisation/pitch-determination procedures are used within time-domain implementations; however, it was unclear which procedure is best, from a computational and output quality viewpoint. Following from this observation a comparison of these procedures was undertaken. The results of this comparison found that peak alignment procedures are most efficient at the expense of some reduction in the quality of output; normalised correlation and mean square difference techniques provide the highest quality output, with normalised correlation having the advantage of being less computationally demanding.

Time-domain/subband approaches were identified as suitable candidates for further investigation due to their unique blend of time-domain and frequency-domain features. They operate by partitioning a complex multi-pitched input into less complex subbands; the subbands are then suitable for time-scale modification using time-domain implementations. A review highlighted the fact that existing approaches partitioned the input in an ad-hoc manner and that reverberant artefacts are introduced due to a loss of synchronisation between subbands. To improve upon the quality achieved by existing approaches a novel subband partitioning scheme based on the Bark scale is employed; such an approach is supported by psychoacoustic and music theory. In addition, a number of subband synchronisation techniques are presented, which result in a further improvement in the quality of output by reducing the presence of reverberation. Whilst improvements were realised within time-domain/subband implementations, the quality of the time-scaled output using these techniques were inferior to those obtained using phase vocoder implementations.

Phase vocoder implementations operate in a similar manner to that of time-domain/subband; whilst time-domain/subband approaches partition a signal into subband components which are quasi-periodic in nature, phase vocoder implementations operate on subbands which are quasi-sinusoidal. As in the case of time-domain/subband, phase vocoder implementations suffer from a loss of synchronisation between subbands, thereby introducing a phasy artefact into the time-scaled output. Improvements to the phase vocoder, which incorporate aspects of sinusoidal modelling, have resulted in a significant reduction in the presence of this artefact; nevertheless it remains a problem. The improved phase vocoder takes steps to preserve the phase relationship that exists between dominant sinusoidal components and their associated cross-terms/side-lobes, but does not preserve phase coherence between dominant sinusoidal components.

By undertaking an investigation into the amount of flexibility that exists in the choice of phase with the phase vocoder, and using the results of this investigation to maintain the phase relationship between dominant sinusoidal components, a further reduction in the phasiness artefact was realised. In addition, the technique employed to achieve the improvement in the quality of output results in a reduction in the number of computations required. In a further exploitation of the phase flexibility tolerated within the phase vocoder, a ‘good’ set of initial phase estimates were determined through the use of features typically incorporated within time-domain implementations. By providing a better initial set of phase estimates the transition to 'perfect' phase coherence is decreased, resulting in a further improvement in the quality of output produced. From subjective testing the phasiness/reverberation reduction improvements offered by the novel 'hybrid' algorithm are very noticeable for speech, whilst not noticeable for music; this is attributed to the fact that reverberation is more present in music than speech, therefore a reduction will not be as readily perceived.

8.1 Future Work

A number of suggestions for future work can be found in the conclusions of each of the general approaches to time-scale modification; see sections 2.1.9, 2.2.1, 2.3.3, 2.4.7 and 2.5.7. In addition, further investigation into time-scale modification in the compressed domain is warranted; [Covell '01] makes use of a time-domain/subband based implementation for the time-scaling of MPEG encoded audio; it is worth investigating the effects of incorporating the subband synchronisation schemes presented in section 6.2 into such a system.

A number of authors have suggested that different levels of time-scaling be applied to different portions of a speech signal; however, whilst general rules for such an implementation have been provided in the literature, for example, that consonants be compressed more than vowels, more detailed rules have not been established; the development of a robust rule set warrants further investigation. Similarly, the non-uniform time-scaling of music instruments requires further exploration. In addition, whilst applying different scaling factors to different speech segments improves the naturalness of the time-scaled output, it should be noted that existing time-scaling techniques will nevertheless expand or compress co articulation effects, resulting in a unnatural sounding output, particularly for the case of time-scale expansion; a further improvement in the quality of an expanded output is likely to be obtained by identifying which phonemes are present during co articulation and synthesising the isolated phonemes, since people pronounce individual phonemes more clearly when they speak slowly.

Within the novel hybrid implementation further improvements in output quality may be realised by incorporating the synchronisation schemes, employed within the time-domain/subband implementations, proposed in section 6.2, i.e. forcing synchronisation of masked or silent bins. It is also worth investigating the effect of increasing the phase deviation allowed further, particularly for the case where such an increase would help maintain vertical phase coherence between harmonically related bins.

9 Appendix A

Informal listening tests were undertaken to compare the perceived differences between approaches. For each evaluation, each test subject was presented with three audio files; the original (for reference), a time-scaled version of the original which was time-scaled using method (1), and a time-scaled version of the original which was time-scaled using method (2). The test subject was not aware which track was time-scaled by which method.

The subject was required to listen to the tracks and select one of five possible ratings, i.e., track1 definitely worse than track2; track1 slightly worse than track2, track1 equal to track2, track1 slightly better than track2, track1 definitely better than track2. The test subjects were required to mark the appropriate box to indicate their rating; see Table 9-1. Tests were undertaken on a PC, and for each evaluation the test files were stored in folders labelled test01, test 02, etc.

1<<2	**1<2**	**1=2**	**1>2**	**1>>2**
X				

1 much worse than 2

1<<2	**1<2**	**1=2**	**1>2**	**1>>2**
	X			

1 slightly worse than 2

1<<2	**1<2**	**1=2**	**1>2**	**1>>2**
		X		

No difference

1<<2	**1<2**	**1=2**	**1>2**	**1>>2**
			X	

1 slightly better than 2

1<<2	**1<2**	**1=2**	**1>2**	**1>>2**
				X

1 much better than 2

Table 9-1 Format of listening test subjects response form.

10 Appendix B

In section 7.1 an analysis of the phase flexibility available within a 50% overlap of an STFT is presented. This appendix provides a similar analysis is presented for another commonly used STFT set up i.e. a 75% overlap. Figure 10-1 is a vector representation of such a situation. Vectors ***a***, ***b***, ***c*** and ***d*** represent the overlapping components and angles θ_1, θ_2 and θ_3 represent the phase differences between the vectors.

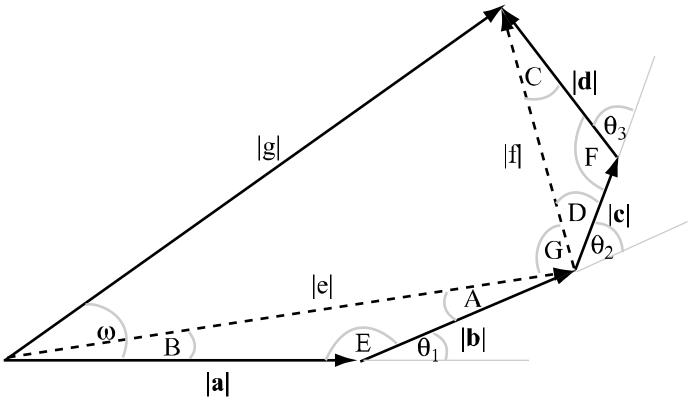


Figure 10-1 Vector representation of 75% STFT analysis overlap.

From Figure 10-1 and using the well-known sine and cosine rules the following equations can be derived, with *L* being the length of the overlapping region i.e. ¼ of the duration of the hamming window:

$$|\boldsymbol{c}(\boldsymbol{t})| = \sqrt{|\boldsymbol{a}(\boldsymbol{t})|^2 + |\boldsymbol{b}(\boldsymbol{t})|^2 - 2|\boldsymbol{a}(\boldsymbol{t})||\boldsymbol{b}(\boldsymbol{t})|\cos C} \qquad (10\text{-}1)$$

$$|\boldsymbol{d}(\boldsymbol{t})| = 0.25(\cos(\pi t/(2L) + \pi/2) + 1) \qquad (10\text{-}2)$$

$$|\boldsymbol{c}(\boldsymbol{t})| = 0.25(\cos(\pi t/(2L)) + 1) \qquad (10\text{-}3)$$

$$|\boldsymbol{a}(\boldsymbol{t})| = 0.5 - |\boldsymbol{c}(\boldsymbol{t})| \qquad (10\text{-}4)$$

$$|\boldsymbol{b}(\boldsymbol{t})| = 0.5 - |\boldsymbol{d(t)}| \qquad (10\text{-}5)$$

$$E = \pi - \theta_1 \qquad (10\text{-}6)$$

$$F = \pi - \theta_3 \qquad (10\text{-}7)$$

$$|\boldsymbol{e}(\boldsymbol{t})| = \sqrt{|\boldsymbol{a}(\boldsymbol{t})|^2 + |\boldsymbol{b}(\boldsymbol{t})|^2 - 2|\boldsymbol{a}(\boldsymbol{t})||\boldsymbol{b}(\boldsymbol{t})|\cos E} \tag{10-8}$$

$$|\boldsymbol{f}(\boldsymbol{t})| = \sqrt{|\boldsymbol{c}(\boldsymbol{t})|^2 + |\boldsymbol{d}(\boldsymbol{t})|^2 - 2|\boldsymbol{c}(\boldsymbol{t})||\boldsymbol{d}(\boldsymbol{t})|\cos F} \tag{10-9}$$

$$|e(\boldsymbol{t})| = \sqrt{|\boldsymbol{a}(\boldsymbol{t})|^2 + |\boldsymbol{b}(\boldsymbol{t})|^2 - 2|\boldsymbol{a}(\boldsymbol{t})||\boldsymbol{b}(\boldsymbol{t})|\cos E} \tag{10-10}$$

$$D(t) = \sin^{-1}\left(|\boldsymbol{d}(\boldsymbol{t})|\sin(F)/|\boldsymbol{f}(\boldsymbol{t})|\right) \tag{10-11}$$

$$A(t) = \sin^{-1}\left(|\boldsymbol{a}(\boldsymbol{t})|\sin(E)/|\boldsymbol{e}(\boldsymbol{t})|\right) \tag{10-12}$$

$$B(t) = \sin^{-1}\left(|\boldsymbol{b}(\boldsymbol{t})|\sin(E)/|\boldsymbol{e}(\boldsymbol{t})|\right) \tag{10-13}$$

$$G(t) = \pi - \left(A(t) + D(t) + \theta_2\right) \tag{10-14}$$

$$|\boldsymbol{g}(\boldsymbol{t})| = \sqrt{|\boldsymbol{e}(\boldsymbol{t})|^2 + |\boldsymbol{f}(\boldsymbol{t})|^2 - 2|\boldsymbol{e}(\boldsymbol{t})||\boldsymbol{f}(\boldsymbol{t})|\cos G(t)} \tag{10-15}$$

$$\omega(t) = \sin^{-1}\left(|\boldsymbol{f}(\boldsymbol{t})|\sin(G(t))/|\boldsymbol{g}(\boldsymbol{t})|\right) + B(t) \tag{10-16}$$

The vector ***g*** represents the sum of the vectors ***a, b, c*** and ***d*** and, therefore, represents the amplitude modulation component. As in section 7.1 the modulation effects must be kept below perceptually tolerable amounts. The maximum modulation effects are found by setting the derivative of ***g***, with respect to *t*, to zero i.e.

$$\frac{d|\boldsymbol{g}(\boldsymbol{t})|}{dt} = 0 \tag{10-17 }$$

It can be shown[33] that the maximum modulation of **g(*t*)** occurs when $\theta_1 = \theta_2 = \theta_3$. Furthermore, it can be shown that the only non-trivial solution to equation 10-17), when this condition arises, occurs at $t = L/2$, and that

[33] This was found to be the case from simulation in Matlab; verification within Maple was attempted; however it was not realised owing to the excessive computational load on the system employed.

$$\left|\boldsymbol{g(L/2)}\right| = \frac{\sqrt{2}}{4}\sqrt{\left(1+\cos\left(-2\sin^{-1}\left(\frac{\left(\sqrt{2}-2\right)\sin(\theta)}{2\sqrt{3+\cos(\theta)}}\right)+\theta\right)\right)\left(3+\cos(\theta)\right)} \qquad (10\text{-}18)$$

where θ is substituted for θ_1, θ_2 and θ_3.

From section 7.1 the maximum degree of modulation must be less than 2% for tones that are less than 80db. The degree of modulation is given by

$$D_m = \left(1 - \boldsymbol{g(L/2)}\right)/2 \qquad (10\text{-}19)$$

So, in order to ensure that amplitude modulation effects will not be perceived

$$D_m \leq |0.02| \qquad (10\text{-}20)$$

Solving equation 10-20) for θ gives

$$\theta \leq |\pm 0.27| \text{ radians} \qquad (10\text{-}21)$$

The derivative of ω(*t*) represents the frequency modulation component of the sum of vectors ***a**, **b**, **c*** and ***d***. From section 7.1 the frequency modulation component must be less than 0.7% of the frequency of the tone for frequencies greater than 500 Hz and 3.6 Hz for tones less than 500 Hz . Therefore, in general, the frequency modulations must be less than 22.62 radians per second. The maximum frequency modulation introduced can be determined by solving

$$\frac{d^2\omega(t)}{dt^2} = 0 \qquad (10\text{-}22)$$

It can be shown that the only non-trivial solution to equation 10-22) occurs at $t = L/2$. The next step is to determine the phase derivative at $t = L/2$. The general solution to this problem is somewhat complicated and sizeable; however solving the problem within the constraints imposed by equation 10-21) results in a more manageable solution, which is given below

$$\frac{d\omega(L/2)}{dt}=\frac{\left(\left(\begin{array}{l}\sin(\theta)\cos(\theta)+\sin(\theta)\cos(\theta)\cos\left(-2\sin^{-1}\left(\frac{\left(\sqrt{2}-2\right)\sin(\theta)}{2\sqrt{3+\cos(\theta)}}\right)+\theta\right)+\\ \sqrt{2}\sin\left(-2\sin^{-1}\left(\frac{\left(\sqrt{2}-2\right)\sin(\theta)}{2\sqrt{3+\cos(\theta)}}\right)+\theta\right)\sqrt{3+2\cos(\theta)+3\cos^{2}(\theta)+2\sqrt{2}}\cos(\theta)\\ +\sin(\theta)\sqrt{2}\cos(\theta)+\sin(\theta)\sqrt{2}\cos(\theta)\cos\left(-2\sin^{-1}\left(\frac{\left(\sqrt{2}-2\right)\sin(\theta)}{2\sqrt{3+\cos(\theta)}}\right)+\theta\right)\\ +\sqrt{2}\sin\left(-2\sin^{-1}\left(\frac{\left(\sqrt{2}-2\right)\sin(\theta)}{2\sqrt{3+\cos(\theta)}}\right)+\theta\right)\sqrt{3+2\cos(\theta)+3\cos^{2}(\theta)+2\sqrt{2}\cos^{2}(\theta)-2\sqrt{2}}\\ -\sin(\theta)-\sin(\theta)\cos\left(-2\sin^{-1}\left(\frac{\left(\sqrt{2}-2\right)\sin(\theta)}{2\sqrt{3+\cos(\theta)}}\right)+\theta\right)+\sin(\theta)\sqrt{2}\\ +\sin(\theta)\sqrt{2}\cos\left(-2\sin^{-1}\left(\frac{\left(\sqrt{2}-2\right)\sin(\theta)}{2\sqrt{3+\cos(\theta)}}\right)+\theta\right)\end{array}\right)\pi\right)}{\left(\begin{array}{l}2\left(3+\cos(\theta)\right)\sqrt{3+2\cos(\theta)+3\cos^{2}(\theta)+2\sqrt{2}\cos^{2}(\theta)-2\sqrt{2}}\\ \left(1+\cos\left(-2\sin^{-1}\left(\frac{\left(\sqrt{2}-2\right)\sin(\theta)}{2\sqrt{3+\cos(\theta)}}\right)+\theta\right)\right)L\end{array}\right)}$$

(10-23)

For the purpose of further simplification equation 10-23) can be closely approximated (within 0.2%, over the range imposed by equation 10-21)) by

$$\frac{d\omega(L/2)}{dt}=\sqrt{5}\,\frac{\sin\left(\frac{\theta}{2}\right)}{L} \qquad (10\text{-}24)$$

In order to ensure that the frequency modulation introduced is less than a perceptual amount

$$\frac{\sqrt{5}\sin\left(\frac{\theta}{2}\right)}{L}\leq 22.62 \qquad (10\text{-}25)$$

For hop size of 15 ms overlap, then

$$\theta\leq\left|\pm 0.304\right| \qquad (10\text{-}26)$$

Considering equations 10-26) and 10-21) then to satisfy all cases equation 10-21) must be satisfied, for this situation.

11 Appendix C

A number of subjective listening tests have been undertaken to support techniques developed in this dissertation. This appendix sets out the method used to statistically analyse the results of those listening tests.

The statistical test employed is the One Sample T-test [Spiegel '98], which is used to measure to determine if the mean of set of data has changed significantly. In the tests undertaken in this dissertation the change in the mean is used to measure whether modifications to an existing algorithm result in a significant perceived improvement in the quality of output. From appendix A, the test subjects are asked to choose from one of five options in deciding which algorithm produced the best results. The following scoring system is then employed to measure the mean:

- If the subject indicates that the modified algorithm is much better than the original the score is 1
- If the subject indicates that the modified algorithm is slightly better than the original the score is 2
- If the subject indicates that the algorithms are the same the score is 3
- If the subject indicates that the original algorithm slightly worse than the modified algorithm the score is 4
- If the subject indicates that the original algorithm much worse than the modified algorithm the score is 5

It then follows that if the average score is 3 there is no perceived difference between the algorithms; if the average score is less than 3 there is a perceived preference for the modified algorithm; if the score is more than 3 there is a perceived preference for the original algorithm.

The One Sample T-test is used to measure the significance of the findings based on the number of test subjects, the measured mean and the standard deviation. The T-test score is given by [Spiegel '98]

$$t = \frac{\mu - \overline{X}}{\left(\frac{\sigma_x}{\sqrt{N}} \right)} \qquad (11\text{-}1)$$

where μ is the original mean, which in this case will be 3, $\overline{X}$ is the measured mean 'average score' of all test subjects, σ_x is the standard deviation of the 'average score' between test subjects and N is the number of test subjects. The 'average score' refers to the average score of a particular test subject over the number of comparisons he/she is required to undertake.

The T-test score is then compared against a T-test table [Spiegel '98] to determine the probability that the new measured mean is an accurate measure of the actual mean. For example, say the measured mean is 2.6, the standard deviation is 0.7 and the number of test subjects is 13, then the T-test score is 2.06. From the table below, and looking at the row labelled 12 on the leftmost column (indicating 12 degrees of freedom = N -1) , it can be appreciated that the T-test score lies between 50% and 75% on the topmost row. This indicates that there is between a 50% and 75% probability that the measured mean of 2.6 is accurate. From [Spiegel '98] such a result is not significantly relevant and it should be concluded that there is no significant improvement provided by the modified algorithm over the original algorithm. From [Spiegel '98] a T-test score indicating a 95% probability of a difference between the original mean and the measured mean is required for statistical relevance.

T-Test Table

Probabilities

	1%	50%	75%	90%	95%	99%	99.5%	
	0.10	0.05	0.025	0.01	0.005	0.001	0.0005	
1	3.078	6.314	12.71	31.82	63.66	318.3	637	1
2	1.886	2.920	4.303	6.965	9.925	22.330	31.6	2
3	1.638	2.353	3.182	4.541	5.841	10.210	12.92	3
4	1.533	2.132	2.776	3.747	4.604	7.173	8.610	4
5	1.476	2.015	2.571	3.365	4.032	5.893	6.869	5
6	1.440	1.943	2.447	3.143	3.707	5.208	5.959	6
7	1.415	1.895	2.365	2.998	3.499	4.785	5.408	7
8	1.397	1.860	2.306	2.896	3.355	4.501	5.041	8
9	1.383	1.833	2.262	2.821	3.250	4.297	4.781	9
10	1.372	1.812	2.228	2.764	3.169	4.144	4.587	10
11	1.363	1.796	2.201	2.718	3.106	4.025	4.437	11
12	1.356	1.782	2.179	2.681	3.055	3.930	4.318	12
13	1.350	1.771	2.160	2.650	3.012	3.852	4.221	13
14	1.345	1.761	2.145	2.624	2.977	3.787	4.140	14
15	1.341	1.753	2.131	2.602	2.947	3.733	4.073	15
16	1.337	1.746	2.120	2.583	2.921	3.686	4.015	16
17	1.333	1.740	2.110	2.567	2.898	3.646	3.965	17

12 References

[Abe ‘89] Abe, M.; Tamura, S.; Kuwabara, H., “A new speech modification method by signal reconstruction”, IEEE International Conference on Acoustics, Speech, and Signal Processing, vol. 1, pp. 592 – 595, May 1989.

[Abeysekera ‘01] Abeysekera, S.S.; Padhi, K.P.; Absar, J.; George, S., “Investigation of different frequency estimation techniques using the phase vocoder,” IEEE International Symposium on Circuits and Systems, vol. 2, pp. 265-268, May 2001.

[Acero ‘98] Acero, A., “Source-filter models for time-scale pitch-scale modification of speech”, IEEE International Conference on Acoustics, Speech and Signal Processing, vol. 2, pp. 881 – 884, May 1998.

[Amir ‘00] Amir, A.; Cohen, G.; Ponceleon, D.; Blanchard, B.; Petkovic, D.; Srinivasan, S., “Using audio time scale modification for video browsing”, Proceedings of the 33rd Annual Hawaii International Conference on System Sciences, pp. 1117 - 1126, January 2000

[Ansari ‘97] Ansari R., “Inverse filter approach to pitch modification: Application to concatenative synthesis of female speech”, Proceedings of IEEE International Conference on Acoustics, Speech, Signal Processing, Munich, Germany, pp. 1623-1626, April 1997.

[Ansari ‘98] Ansari, R.; Kahn, D.; Macchi, M.J., “Pitch modification of speech using a low-sensitivity inverse filter approach”, IEEE Signal Processing Letters, vol. 5(3), pp. 60 – 62, March 1998.

[Arons ‘92] Arons, B., “Techniques, perception, and applications of time-compressed speech”, In Proceedings of 1992 Conference, American Voice I/O Society, pp. 169-177, September 1992.

[Arons ‘97] Arons, B., “SpeechSkimmer: A system for interactively skimming recorded speech”, ACM Transactions on Computer-Human Interaction (TOCHI) archive, vol. 4(1) pp. 3 - 38, March 1997.

[Barry ‘04] Barry, D., Private Communication.

[Beltran ‘03] Beltran, J.R.; Beltran, F., “Additive synthesis based on the continuous wavelet transform”, Proceedings of the 6th International Conference on Digital Audio Effects, London, September 2003.

[Bellanger ‘89] Bellanger, M., *Digital Processing of Signals*, New York: Wiley, 1989.

[Benade ‘76] Benade, A. H., *Fundamentals of Musical Acoustics*, Oxford University Press, 1976.

[Bialick ‘89] Bialick, United States Patent No. 4,864,620.

[Bonada ‘00] Bonada, J., “Automatic technique in frequency domain for near-lossless time-scale modification of audio”, 'Proceedings of International Computer Music Conference, Berlin, Germany 2000.

[Bonada ‘02] Bonada, J., “Time-scale modification of audio in the context of professional audio post-production”, Doctoral Pre-thesis Work, UPF, Barcelona, 2002.

[Blauert ‘78] Blauert, J.; Laws, P., “Group Delay Distortions in Electroacoustical Systems”, Journal of the Acoustical Society of America, vol. 63(5), pp. 1478-1483, May 1978.

[Bristow-Johnson ‘95] Bristow-Johnson, R., “A detailed analysis of a time-domain formant-corrected pitch-shifting algorithm”, Journal of the Audio Engineering Society, vol. 43(5), pp. 340-352, May 1995.

[Bristow-Johnson ‘01] Bristow-Johnson, R.; Bogdanowicz, K., “Intraframe time-scaling of nonstationary sinusoids within the phase vocoder”, IEEE Workshop on the Applications of Signal Processing to Audio and Acoustics, pp. 215 - 218, October 2001.

[Charpentier ‘90] Charpentier, F.; Moulines, E., “Pitch-synchronous waveform processing techniques for text-to-speech synthesis using diphones”, Speech Communication, vol. 9(5/6), pp 13-19, 1990.

[Claasen ’80] Claasen, T.A.C.M.; Mecklenbrauker, W.F.G, “The Wigner Distribution - A tool for time-frequency signal analysis, part 1: Continuous-time signals,” Phillips Journal of Research, vol. 35(3), pp. 217-250, 1980.

[Cohen ‘95] Cohen, L., *Time-Frequency Analysi*s, Prentice Hall, 1995.

[Coolidge ‘00] Coolidge F.L., *Statistics: A Gentle Introduction*, Sage Publications, 2000.

[Covell ‘98] Covell, M.; Withgott, M.; Slaney, M., “MACH1: nonuniform time-scale modification of speech”, IEEE International Conference on Acoustics, Speech and Signal Processing, vol. 1, pp. 349 - 352 , May 1998.

[Covell ‘01] Covell, M.; Slaney, M.; Rothstein, A., “FastMPEG: time-scale modification of bit-compressed audio information” IEEE International Conference on Acoustics, Speech, and Signal Processing, vol. 5, 3261 - 3264 , May 2001.

[Cox ‘83] Cox, R.; Crochiere, R.; Johnston, J., “Real-time implementation of time domain harmonic scaling of speech for rate modification and coding”, IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. 31(1), pp. 258 – 272, February 1983.

[Crokett ‘03] Crokett B.G., “High quality multi-channel time-scaling and pitch-shifting using auditory scene analysis”, Audio Engineering Society Convention, preprint no. 5948, New York, October 2003.

[Di Federico ‘98] Di Federico, R., “Waveform preserving time stretching and pitch shifting for sinusoidal models of sound”, Proceedings of the 6th International Conference on Digital Audio Effects, pp. 44–48, Barcelona, 1998.

[Di Martino ‘97] Di Martino, J., “Speech synthesis using phase vocoder techniques”, EuroSpeech, Rhodes, Greece 1997.

[Di Martino '01] Di Martino, J.; Laprie, Y., "Suppression of phasiness for time-scale modifications of speech signals based on a shape invariance property", IEEE International Conference on Acoustics, Speech, and Signal Processing, vol. 2, pages:853 – 856, May 2001.

[Depalle '93a] Depalle, P.; Garcia, G.; Rodet, X., "Tracking of partials for additive sound synthesis using hidden Markov models", IEEE International Conference on Acoustics, Speech and Signal Processing, vol. 1, pp. 225 - 228, April 1993.

[Depalle '93b] Depalle, P.; García, G.; Rodet, X. "Analysis of sound for additive synthesis: Tracking of partials using hidden markov models", Proceedings of International Computer Music Conference, October 1993.

[De Goetzen '00] De Goetzen, A; Bernardini, N.; Arfib, D., "Traditional implementations of a phase vocoder: the tricks of the trade," Proceedings of the International Conference on Digital Audio Effects, pp. 37-43, Verona, Italy, December 2000.

[Deller '93] Deller, J.R., Proakis, J.; Hansen, J. G., *Discrete-time processing of speech signals*, Macmillan publishing company, 1993.

[Demol '04] Demol, M.; Struyve, K.; Verhelst, W.; Paulussen, H.; Desmet P.; Verhoeve, P., "Efficient non-uniform time-scaling of speech with WSOLA for CALL applications", Proceedings of InSTIL/ICALL Symposium, paper 007, Venice, Italy, June 2004.

[Dolson '86] Dolson, M., "The phase vocoder: A tutorial", Computer Music Journal, vol. 10, pp. 14-27, 1986.

[Donnellan '03] Donnellan, O.; Elmar Jung; Coyle, E , "Speech-adaptive time-scale modification for computer assisted language-learning", The 3rd IEEE International Conference on Advanced Learning Technologies, pp. 165 – 169, July 2003.

[Dorran '03a] Dorran D.; Lawlor, R.; Coyle E., "Time-scale modification of speech using a synchronised and adaptive overlap-add (SAOLA) algorithm", Audio Engineering Society 114th Convention 2003, Amsterdam, The Netherlands, preprint no. 5834, March 2003.

[Dorran '03b] Dorran D.; Lawlor, R.; Coyle E., "High quality time-scale modification of speech using a peak alignment overlap-add algorithm (PAOLA)", IEEE International Conference on Acoustics, Speech and Signal Processing, Hong Kong, vol. 1, pp. I-700 - I-703, April 2003.

[Dorran '03c] Dorran D.; Lawlor, R., "An efficient time-scale modification algorithm for use within a subband implementation", Proceedngs of the International Conference on Digital Audio Effects, London, pp. 339-343, Sept. 2003.

[Dorran '03d] Dorran D.; Lawlor, R., "Time-scale modification of music using a subband approach based on the bark scale", IEEE Workshop on Applications of Signal Processing to Audio and Acoustics New Paltz, New York, pages 173-176, October 2003.

[Dorran '04a] Dorran D.; Lawlor, R., "Time-scale modification of music using a synchronised subband/time-domain approach", IEEE International Conference on Acoustics, Speech and Signal Processing, pp. IV 225 – IV 228, Montreal, May 2004.

[Dorran '04b] Dorran D.; Lawlor, R.; Coyle E., "An efficient phasiness reduction technique for moderate audio time-scale modification", International Conference on Digital Audio Effects, pages 83-88, Naples, Italy, October 2004.

[Dudley '38] Dudley, H., U.S. Patent No. 2,352,023.

[Dudley '39] Dudley, H., "Remaking speech", Journal of the Acoustic Society of America, Vol. 11, No. 2, pp. 169-175, 1939.

[Duxbury '02] Duxbury, C.; Davies, M.E.; Sandler, M.B., "Improved time-scaling of musical audio using phase locking of transients", 112th Convention of the Audio Engineering Society, Munich 2002.

[Duxbury '03] Duxbury C.; Davies M.; Sandler M., "Temporal segmentation and pre-analysis for non-linear time-scaling of audio", 114th Convention of the Audio Engineering Society, Preprint no. 5812, Amsterdam, 2003.

[Ellis '92] Ellis, D.P.W., "Timescale modifications and wavelet representations", Proceedings of the 1992 International Computer Music Conference, San Francisco, California, 1992.

[Erogul '98] Erogul, O.; Karagoz, I., "Time-scale modification of speech signals for language-learning impaired children", Proceedings of the 2nd International Conference Biomedical Engineering Days, pp. 33 – 35, May 1998.

[Fairbanks '54] Fairbanks, G.; Everitt, W. L.; Jaeger, R. P., "Method for time or frequency compression-expansion of speech", Transactions of the Institute of Radio Engineers professional group on audio, pp. 7 – 12, 1954.

[Fant '60] Fant, G., *Acoustic Theory of Speech Production*, Mouton and Co., The Hague, The Netherlands, 1960.

[Ferreira '96] Ferreira, A.J.S., "Convolutional effects in transform coding with TDAC:an optimal window," IEEE Transactions on Speech Audio Processing, vol. 4, pp. 104–114, March 1996.

[Ferreira '98] Ferreira, A.J.S., "A new frequency domain approach to time-scale expansion of audio signals", IEEE International Conference on Acoustics, Speech, and Signal Processing, vol. 6, pp. 3577 – 3580, May 1998.

[Ferreira '99] Ferreira, A.J.S., "An odd-DFT based approach to time-scale expansion of audio signals", IEEE Transactions on Speech and Audio Processing, vol. 7(4), pp. 441 – 453, July 1999.

[Flanagan '65] Flanagan, J. L., *Speech Analysis, Synthesis and Perception*, Academic Press, New York, 1965.

[Flanagan '66] Flanagan J.L.; Golden R.M., "Phase vocoder", Bell System Technical Journal, vol. 45, pp. 1493-1509, 1966.

[Fischman '97] Fischman R., "The phase vocoder: Theory and practice", Organised Sound, vol. 2(2)., pp 127-145, 1997.

[Fitzgerald '04] Fitzgerald, D., "Automatic drum transcription and source separation", PhD Thesis, Dublin Institute of Technology, 2004.

[Gabor '44] Gabor, D., British patent application No.24624/44.

[Gabor '46] Gabor, D., "Theory of communication", Journal IEE, vol. 93, pp. 429-457, 1946.

[Garas '98] Garas, J.; Sommen, P.C.W., "Time/pitch scaling using the constant-Q phase vocoder", IEEE Workshop on Circuits, Systems and Signal Processing, pp. 173-176, Mierlo, Netherlands, November 1998.

[Garvey '53] Garvey, W.D., "The intelligibility of abbreviated speech patterns", Quarterly Journal of speech, vol. 39, pp. 296 – 306, 1953.

[George '92] George, E.B.; Smith, M.J.T., "Analysis-by-synthesis/overlap-add sinusoidal modelling applied to the analysis and synthesis of musical tones", Journal of the Audio Engineering Society, vol. 40(6) pp. 497-516, June 1992.

[Gersem '97] Gersem, P.; Moor,B.; Moonen, M., "Applications of the continuous wavelet transform in the processing of musical signals," 13th International Conference on Digital Signal Processing, Santorini, Greece, 1997.

[Goodwin '94] Goodwin, M.; Rodet, X., "Efficient Fourier synthesis of nonstationary sinusoids", Proceedings of the International Computer Music Conference, San Francisco, 1994.

[Goodwin '95] Goodwin, M.; Kogon, A., "Overlap-add synthesis of nonstationary sinusoids", Proceedings of the International Computer Music Conference, 1995.

[Goodwin '96] Goodwin, M.; Vetterli, M., "Time-frequency signal models for music analysis, transformation, and synthesis", IEEE International Symposium on Time-Frequency and Time-Scale Analysis, pp. 133 – 136, June 1996.

[Griffin '84] Griffen, D. W.; Lim, J. S., "Signal estimation from modified short-time Fourier transform", IEEE Transactions on Acoustics, Speech and Signal Processing, vol. ASSP-32(2), pp.236-243, April 1984.

[Griffin '85] Griffen, D. W.; Lim, J. S., "A new model based speech analysis/ synthesis system", IEEE International Conference on Acoustics, Speech and Signal Processing, pp.513-516, 1985.

[Hamdy '97] Hamdy, K.N.; Tewfik, A.H.; Ting Chen; Takagi, S., "Time-scale modification of audio signals with combined harmonic and wavelet representations", IEEE International Conference on Acoustics, Speech, and Signal Processing, vol. 1, pp. 439 – 442, April 1997.

[Hanna '03] Hanna, P.; Desainte-Catherine, M., "Time scale modification of noises using a spectral and statistical model", IEEE International Conference on Acoustics, Speech, and Signal Processing, vol. 6, pp. VI_181 - VI_184, April 2003.

[Hardam '90] Hardam, E., "High quality time scale modification of speech signals using fast synchronised-overlap-add algorithms", Proceedings of the IEEE International Conference on Acoustics, Speech and Signal processing, pp. 409 -412, 1990.

[Harris '78] Harris, F.J., "On the use of windows for harmonic analysis with the discrete Fourier transform," Proceedings of IEEE, vol. 66, pp. 51-83, 1978.

[Howard '01] Howard, D.; Angus, J., *Acoustics and Psychoacoustics*, Focal Press, Music Technology Series, 2nd Edition, 2001.

[Hejna '90] Hejna, D. J., "Real-time time-scale modification of speech via the synchronized overlap-add algorithm", M.I.T. Masters Thesis, Department of Electrical Engineering and Computer Science, February 1990.

[Hoek '01] Hoek, S., U.S. Patent No. 6,226,003.

[Irino '92] Irino, T.; Kawahara, H., "Signal reconstruction from modified wavelet transform-An application to auditory signal processing", IEEE International

Conference on Acoustics, Speech, and Signal Processing, vol. 1, pp. 85 – 88, March 1992.

[Jiang '01] Jiang, Y.; Murphy, P., "Voice source analysis for pitch-scale modification of speech signals" , Proceedings of the COST G-6 Conference on Digital Audio Effects, Limerick, Ireland, December, 2001.

[Kahrs '98] Karhs, M., Brandenburg, K., *Applications of Digital Signal Processing to Audio and Acoustics*, Kluwer Academic Publishers, 1998.

[Kim '03] Ki-Hong, K.; In-Ho, H., "A multi-resolution ABS/OLA sinusoidal model using wavelet transform", Proceedings Seventh International Symposium on Signal Processing and Its Applications, vol. 1 , pp. 377 – 380, July 2003.

[Kapilow '99] Kapilow, D.; Stylianou, Y.; Schroeter, J., "Detection of non-stationarity in speech signals and its application to time-scaling", Proceedings of Eurospeech, Budapest, Hungary, 1999.

[Kronland-Martinet '88] Kronland-Martinet, R., "The use of the wavelet transform for the analysis, synthesis and processing of speech and music sounds", Computer Music Journal, MIT Press, vol. 12(4), pp. 11-20, December '88.

[Lagrange '03] Lagrange, M.; Marchand, S.; Raspaud, M.; Rault, J.B., "Enhanced partial tracking using linear prediction", Proceedings of the Digital Audio Effects Conference, London, September 2003.

[Lagrange '04] Lagrange, M.; Marchand, S.;Rault, J.B., "Using linear prediction to enhance the tracking of partials", Proceedings of the IEEE International Conference on Speech and Signal Processing, Montreal, Quebec, Canada, May 2004.

[Laroche '93a] Laroche, J., "Autocorrelation method for high-quality time/pitch-scaling", IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, pp. 131 – 134, October 1993.

[Laroche '93b] Laroche, J.; Stylianou, Y.; Moulines, E., "HNM: a simple, efficient harmonic+noise model for speech", IEEE Workshop on Applications of Signal

Processing to Audio and Acoustics, pp. 169 – 172, October 1993.

[Laroche '93c] Laroche, J.; Stylianou, Y.; Moulines, E., "HNS: Speech modification based on a harmonic+noise model", IEEE International Conference on Acoustics, Speech, and Signal Processing, vol. 2, pp. 550 - 553, April 1993

[Laroche '97] Laroche, J.; Dolson, M., "Phase-vocoder: about this phasiness business", IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, October 1997 .

[Laroche '99a] Laroche, J.; Dolson, M., "Improved phase vocoder", Speech and Audio Processing, IEEE Transactions on Speech and Audio processing, vol. 7(3), pp. 323 –332, May 1999.

[Laroche '99b] Laroche, J.; Dolson, M., "New phase-vocoder techniques for pitch-shifting, harmonizing and other exotic effects", IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, pp. 91 - 94, October 1999.

[Laroche '00a] Laroche, J., U.S. Patent no. 6,049,766.

[Laroche '00b] Laroche J., "Synthesis of sinusoids via non-overlapping inverse Fourier transform", IEEE Transactions on Speech and Signal Processing, vol. 8(4), July 2000.

[Laroche '03] Laroche J., "Frequency-domain techniques for high quality voice modification", Proceedings of the 6th International Conference on Digital Audio Effects, London, September 2003.

[Lawlor '99] Lawlor, R., "Audio time-scale and frequency-scale modification", Department of Electrical and Electronic Engineering, University College, Dublin. November 1999.

[Lawlor '99b] Lawlor, B.; Fagan, A.D., "A novel efficient algorithm for audio time-scale modification", Irish Signals and Systems Conference, National University of Ireland, Galway, 1999.

[Lawlor ‘99c] Lawlor, B.; Fagan, A.D., “A novel high quality efficient algorithm for time-scale modification of speech”, Eurospeech, 6th Conference on Speech Communication and Technology, Budapest, Hungary, 1999.

[Lee ‘72] Lee, F., “Time compression and expansion of speech by the sampling method”, Journal of the audio engineering society, pp. 738 –742, May 1972.

[Lee ‘97] Lee, S., “Variable time-scale modification of speech using transient information”, IEEE International Conference on Acoustics, Speech, and Signal Processing, vol. 2, pp. 1319 -1322, 1997.

[Lee ‘02] Lee, S.J.; Kim, H. S., “Computationally efficient time-scale modification of speech using 3 level clipping”, 7th International Conference on Spoken Language Processing, Denver, vol.4, pp. 2385-2388, September 2002.

[Levine ‘97] Levine, S.; Verma, T.; Smith J.O., “Alias-free, multiresolution sinusoidal modeling for polyphonic, wideband audio”, IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, Mohonk, 1997.

[Levine ‘98a] Levine, S.N.; Verma, T.S.; Smith, J.O., “Multiresolution sinusoidal modeling for wideband audio with modifications”, Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing, vol. 6, pp. 3585 – 3588, May 1998.

[Levine ‘98b] Levine, S.; Smith J.O, “A Sines+transients+noise audio representation for data compression and time/pitch-scale modifications", 105th Audio Engineering Society Convention, San Francisco 1998.

[Lin ‘95] Lin G.L.; Chen S.G,; Wu, T., “High quality and low complexity pitch modification of acoustic signals”, International Conference on Acoustics, Speech, and Signal Processing, vol. 5 , pp. 2987 - 2990, May 1995.

[Loscos ‘04] Loscos, A.; Bonada, J., “Emulating rough and growl voice in spectral domain”, Proceedings of the International Conference on Digital Audio Effects, pp. 49-52, October 2004.

[Makhoul ‘86] Makhoul, J.; El-Jaroudi, A., “Time-scale modification in medium to low rate speech coding”, IEEE International Conference on Acoustics, Speech, and Signal Processing., vol. 11 , pp. 1705 – 1708, April 1986.

[Macon ‘96] Macon, M.W.; Clements, M.A., “Speech concatenation and synthesis using an overlap-add sinusoidal model”, IEEE International Conference on Acoustics, Speech, and Signal Processing, vol. 1, pp. 361 – 364, May 1996.

[Macon ‘97] Macon, M.W.; Clements, M.A., “Sinusoidal modeling and modification of unvoiced speech”, IEEE Transactions on Speech and Audio Processing, vol. 5(6), pp. 557 – 560, November 1997.

[Mansour ‘01] Mansour, M.F.; Tewfik, A.H., “Audio watermarking by time-scale modification”, IEEE International Conference on Acoustics, Speech, and Signal Processing, vol. 3, pp. 7-11, May 2001.

[Masri ‘96] Masri, P.; Bateman, A., “Improved modelling of attack transients in music analysis-resynthesis”, International Computer Music Conference, pp. 100-103, 1996.

[Malah ‘79] Malah, D., “Time-domain algorithms for harmonic bandwidth reduction and time scaling of speech signals”, IEEE Transactions on Acoustics, Speech and Signal Processing, vol. ASSP-27(2), pp. 121 – 133, April 1979.

[Malah ‘81] Malah, D.; Crochiere, R.E.; Cox, R.V., “Performance of transform and subband coding systems combined with harmonic scaling of speech”, IEEE Transactions on Acoustics Speech, Signal Processing, vol. ASSP-29(2), pp. 273-283, April 1981.

[Markel ‘71] Markel, J., “FFT pruning”, IEEE Transactions on Audio and Electroacoustics, vol. 19(4), pp:305 – 311, December 1971

[McAulay ‘86a] McAulay, R. J.; Quatieri, T. F., “Speech analysis/synthesis based on a sinusoidal representation”, IEEE Transactions on Acoustics, Speech and Signal Processing, vol. ASSP-34(4), pp. 744 – 754, August 1986.

[McAulay ‘86b] McAulay, R. J.; Quatieri, T. F., “Speech transformation based on a sinusoidal representation”, IEEE Transactions on Acoustics, Speech and Signal Processing, vol. ASSP-34(6), pp. 1449 – 1464, December 1986.

[Mitra ‘93] Mitra, S. K.; Kaiser, J. F., *Handbook for Digital Signal Processing*, Wiley Interscience, 1993.

[Moore ‘83] Moore, B.; Glasberg, B., “Suggested formulae for calculating auditory-filter bandwidths and excitation patterns”, Journal of the Acoustical Society of America, Vol. 3(74), pp. 750-753, September 1983.

[Moulines ‘88] Moulines, E.; Charpentier, F., “Diphone synthesis using multipulse LPC technique”, Proceedings 7th FASE symposium of Speech ’88, Edinburgh, pp. 47 – 53, 1988.

[Moulines ‘95a] Moulines, E.; Laroche, J., “Non-parametric techniques for pitch-scale and time-scale modification of speech”, Speech Communication vol. 16, pp. 175-205, 1995.

[Moulines ‘95b] Moulines, E.; Verhelst W., Chapter entitled “Time-domain and frequency-domain techniques for prosodic modification of speech”, *Speech Coding and Synthesis*, Edited By W.B. Kleijn and K.K. Paliwal, 1995.

[Nagabuchi ‘88] Nagabuchi, H., “Objective measure for coded speech quality evaluation considering talker-dependency of quality”, IEICE Transactions, No. 8, pp. 1523-1531, August 1988.

[Ninness ‘00] Ninness, B.; Henriksen, S., “Time and frequency scale modification of speech signals”, IEEE International Conference on Acoustics, Speech, and Signal Processing, vol. 3, pp.1295 – 1298, June 2000.

[Neuberg ‘78] Neuberg E., “Simple pitch-dependent algorithm for high quality speech rate changing”, Journal of the Acoustical Society of America, vol. 63(2), pp.624-625, 1978.

[O’ Brien ‘99] O’Brien, D.; Monaghan, A., “Shape invariant time-scale modification of speech using a harmonic model," in Proceedings IEEE International Conference on Acoustics, Speech, Signal Processing, pp. 381–384., Phoenix, Arizona, 1999.

[Omoigui '99] Omoigui, N.; He, L.; Gupta A.; Grudin, J.; Sanocki, E., "Time-compression: Systems concerns, usage, and benefits", CHI Conference Proceedings, 136-143, 1999.

[Oppenheim '75] Oppenheim A.V.; Shafer R.W., *Digital Signal Processing*, Eaglewood Cliffs, Prentice-Hall, 1975.

[Pallone '99] Pallone G.; Boussard P.; Daudet L.; Guillemain P.; Kronland-Martinet R., "A wavelet based method for audio-video synchronization in broadcasting applications", Proceedings of the International Conference on Digital Audio Effects, Norway, December 1999.

[Patterson '96] Patterson, D.A.; Hennessey, J.L., *Computer Architecture: A Quantitative Approach*, Morgan Kaufmann Publishers, Inc., San Francisco, California, 2nd edition, 1996.

[Plomp '65] Plomp, R.; Levelt, W. J., "Tonal consonance and critical bandwidth", Journal of the Acoustical Society of America, vol. 37, pp.548-560, 1965.

[Polikar '05] http://users.rowan.edu/~polikar/WAVELETS/WTtutorial.html

[Pollard '96] Pollard, M.P.; Cheetham, B.M.G.; Goodyear, C.C.; Edgington, M.D.; Lowry, A., "Enhanced shape-invariant pitch and time-scale modification for concatenative speech synthesis", Fourth International Conference on Spoken Language, vol. 3, pp. 1433 – 1436, October 1996.

[Pollard '97] Pollard, M.P.; Cheetham, B.M.G.; Goodyear, C.C.; Edgington, M.D., "Shape-invariant pitch and time-scale modification of speech by variable order phase interpolation", IEEE International Conference on Acoustics, Speech, and Signal Processing, vol. 2, pp. 919 – 922, April 1997.

[Portnoff '76] Portnoff, M.R., "Implementation of the digital phase vocoder using the fast Fourier transform", IEEE Transactions on Acoustics, Speech and Signal

Processing, vol. ASSP-24, pp. 243-248, 1976.

[Portnoff '81] Portnoff, M. R., "Time-scale modifications of speech based on short-time Fourier analysis", IEEE Transactions on Acoustics, Speech and Signal Processing, vol. ASSP-29(3), pp. 374 –390, June 1981.

[Puckett '95] Puckett, M.S., "Phase locked vocoder", IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, pp: 222 –225, 1995.

[Puckett '98] Puckett, M.S., "Accuracy of frequency estimates using the phase vocoder", IEEE Transactions on Speech and Audio Processing, vol. 6(2), pp. 166-176, March 1998.

[Quatieri '85] Quatieri, T. F.; McAulay, R. J., "Speech transformation based on a sinusoidal representation", Proceedings of the IEEE International conference on Acoustics, Speech and Signal processing, pp. 489 – 492, 1985.

[Quatieri '92] Quatieri, T. F.; McAulay, R. J., "Shape invariant time-scale and pitch modification of speech", IEEE Transactions on Signal Processing, vol. 40(3), pp. 497 – 510, March 1992.

[Quatieri '93a] Quatieri, T.F.; Dunn, R.B.; Hanna, T.E., "Time-scale modification of complex acoustic signals", IEEE International Conference on Acoustics, Speech, and Signal Processing, vol. 1, pp. 213 – 216, April 1993.

[Quatieri '93b] Quatieri, T.F.; Dunn, R.B.; Hanna, T.E., "Time-scale modification with temporal envelope invariance", IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, pp. 127 – 130, October 1993

[Quatieri '95a] Quatieri, T.F.; Dunn, R.B.; Hanna, T.E., "A subband approach to time-scale expansion of complex acoustic signals", IEEE Transactions on Speech and Audio Processing vol. 3(6), pp. 515 – 519, November 1995.

[Quatieri '95a] Quatieri, T.F.; Hanna, T.E., "Time-scale modification with inconsistent constraints", IEEE ASSP Workshop on Applications of Signal Processing to Audio and Acoustics, pp. 263 – 266, October 1995.

[Rabiner '77] Rabiner, L. R., "On the use of autocorrelation analysis for pitch detection", IEEE Transactions on Speech and Audio Processing vol. 25(1), November 1977.

[Röbel '03] Röbel A., "Transient detection and preservation in the phase vocoder", Proceedings of International Computer Music Conference, pp.247-250, Singapore, 2003.

[Raspaud '04] Raspaud, M.; Marchand, S., "Enhanced time-stretching using order-2 sinusoidal modeling", Proceedings of the International Conference on Digital Audio Effects, pp. 76-82, October 2004.

[Rodet '92] Rodet, X.; Depalle P., "Spectral envelopes and inverse FFT synthesis", Proceedings of AES Convention, 1992.

[Rodet '97] Rodet, X. "Musical sound signals analysis/synthesis: Sinusoidal+residual and elementary waveform models", Proceedings of the IEEE Time-Frequency and Time-Scale Workshop, 1997.

[Rodriguez-Hernandez '94] Rodriguez-Hernandez, M.; Casajus-Quiros, F., "Improving time-scale modification of audio signals using wavelets", IC-SPAT, vol. 2, pp. 1573–1577, 1994.

[Roucos '85] Roucos, S.; Wilgus, A.M., "High quality time-scale modification for speech", IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 493-496, March 1985.

[Scott '67] Scott, R.J., "Time adjustment in speech synthesis", Journal of the Acoustical Society of America, Vol. 41(1), pp. 60-65, 1967.

[Seneff '82] Seneff, S., "System to independently modify excitation and/or spectrum of speech waveform without explicit pitch extraction", IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. 30(4), pp. 566 – 578, August 1982

[Serra ‘90] Serra, X.; Smith, J. O., “Spectral modeling synthesis: A sound analysis /synthesis system based on a deterministic plus stochastic decomposition”, Computer Music Journal, vol. 14(4), pp. 12 – 24, 1990.

[Serra ‘98] Serra, X.; Bonada, J., “Sound transformations based on the SMS high level attributes”, in Proceedings Workshop on Digital Audio Effects, Barcelona, Spain, pp. 188-191, 1998.

[Smith ‘87] Smith, J. O.; Serra, X., “PARSHL: An analysis/synthesis program for non-harmonic sounds based on a sinusoidal representation”, Proceedings of the International Computer Music Conference, pp. 290 – 297, 1987.

[Smith ‘97] Smith, S.W., *The Scientist and Engineer's Guide to Digital Signal Processing*, California Technical Publishing, 1997.

[Spiegel ‘98] Speigel, M.; Stephens, L., *Schuam’s Outline of Statistics,* Schaum, 1998.

[Spleesters ‘94] Spleesters, G.; Verhelst, W.; Wahl, A., “On the application of automatic waveform editing for time warping digital and analog recordings”, Proceedings of the 96th Audio Engineering Society Convention, Amsterdam, preprint 3843, 1994.

[Suzuki ‘92a] Suzuki, R.; Misaki, M., “Time-scale modification of speech signals using cross-correlation”, IEEE International Conference on Consumer Electronics, pp: 166 – 167, June 1992.

[Suzuki ‘92b] Suzuki, R.; Misaki, M., “Time-scale modification of speech signals using cross-correlation functions”, IEEE Transactions on Consumer Electronics, vol. 38(3) , pp. 357 – 363, August 1992.

[Sylvestre ‘92] Sylvestre, B.; Kabal, P., “Time-scale modification of speech using an incremental time-frequency approach with waveform structure compensation”, Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, pp. I:81-84,1992.

[Tan '00] Tan, R.K.C.; Lin, A.H.J., "A time-scale modification algorithm based on the subband time-domain technique for broad-band signal applications", Journal of the Audio Engineering Society, vol. 48(5), pp. 437-449, May 2000.

[Tang '97] Tang, M.; Wang, C.; Seneff, S., "Voice transformations: From speech synthesis to mammalian vocalizations", Proceedings of the 7th European Conference on Speech Communication and Technology, Denmark, 2001.

[Telatar '03] Teletar, Z., Erogul, O., "Heart sounds modification for the diagnosis of cardiac disorders", IJCI Proceedings of International Conference on Signal Processing, vol.1(2), pp. 101-105, September 2003.

[Terhardt '84] Terhardt, E., "The concept of musical consonance: A link between music and psychoacoustics", Music Perception, vol. 1, pp. 276-295, 1984.

[Thomson '82] Thomson, D.J. "Spectrum estimation and harmonic analysis", Proceedings of the IEEE, vol. 70(9), 1982.

[TIMIT '05] http://www.ldc.upenn.edu/readme_files/timit.readme.html

[Vaidyanathan '92] Vaidyanathan P.P.; Koilpillai R. D., "Cosine-modulated FIR filter banks satisfying perfect reconstruction," IEEE Transactions on Signal Processing, vol. 40, pp. 770-783, April 1992

[Valens '99] www.cs.unm.edu/~williams/cs530/arfgtw.pdf

[Van Schijndel '03] Van Schijndel, N.H.; Gomez, M.; Heusdens, R., "Towards a better balance in sinusoidal plus stochastic representation", IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, pp. 197 -200, October 2003.

[Vergin '97] Vergin, R.; O'Shaughnessy, D.; Farhat, A., "Time domain technique for pitch modification and robust voice transformation", IEEE International Conference on Acoustics, Speech, and Signal Processing, vol. 2 , pp:947 – 950, April 1997

[Verhelst '93] Verhelst, W.; Roelands, M., "An overlap-add technique based on waveform similarity (WSOLA) for high quality time-scale modification of speech", IEEE International Conference on Acoustics, Speech, and Signal Processing, vol. 2 , pp. 554 -557 vol.2, 1993.

[Verhelst '00a] Verhelst, W.; Van Compernolle, D.; Wambacq, P., "A unified view on synchronized overlap-add methods for prosodic modification of speech", Proceedings of the International Conference on Spoken Language Processing, vol. 2, pp. 63-66, Beijing, China, October 2000.

[Verhelst '00b] Verhelst, W**.,** "Overlap-add methods for time-scaling of speech", Speech Communication, vol. 30(4), pp. 207-221, 2000.

[Verhelst '03] Verhelst, W.; Brouckxon, H., "Rejection phenomena in inter signal voice transplantations", IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, pp. 165-168, New Paltz, 2003.

[Verhelst '03] Private Communication.

[Verma '98] Verma, T.; Meng, T., "Time scale modification using a sines+ transients+Noise signal model", Proceedings of the Digital Audio Effects Workshop, pp. 49-52, Barcelona, 1998.

[Vetterli '95] Vetterli, M.; Kovacevic, J., *Wavelets and Subband Coding*, Prentice Hall, 1995.

[Virtanen '00] Virtanene, T., "Audio Signal Modelling with Sinusoids plus Noise", Masters Thesis, Tampere University of Technology, Finland, 2000.

[Wayman '88] Wayman, J.L.; Wilson, D.L., "Some improvements on the synchronized-overlap-add method of time scale modification for use in real-time speech compression and noise filtering",], IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. 36(1), pp. 139 - 140 , January 1988

[Wong '97] Wong, P.H.W.; Au, O.C.; Wong, J.W.C.; Lau, W.H.B., "On improving the intelligibility of synchronized over-lap-and-add (SOLA) at low TSM factor", IEEE Region 10 Annual Conference on Speech and Image Technologies for

Computing and Telecommunications, vol. 2, pp. 487 – 490, December 1997.

[Wong ‘98] Wong, J.W.C.; Au, O.C.; Wong, P.H.W., “Fast time scale modification using envelope-matching technique (EM-TSM)”, Proceedings of the IEEE International Symposium on Circuits and Systems, vol. 5 , pp. 550 -553, 1998.

[Wong ‘02] Wong, P.H.W.; Au, O.C., “Fast SOLA-based time scale modification using modified envelope matching”, IEEE International Conference on Acoustics, Speech and Signal Processing, vol. 3, pp. III-3188 - III-3191, May 2002

[Wong ‘03] Wong, P.H.W.; Au, O.C., “Fast SOLA-based time scale modification using envelope matching”, KAP Journal of VLSI Signal Processing-Systems for Signal, Image, and Video Technology, vol. 35(1), pp. 75-90, August 2003.

[Yim ‘96] Yim, S.; Pawate, B.I., “Computationally efficient algorithm for time scale modification (GLS-TSM)”, IEEE International Conference on Acoustics, Speech, and Signal Processing , vol. 2, pp. 1009 -1012, 1996.

[Youngberg ‘78] Youngberg, J.E., “Rate/pitch modification using the constant-Q transform” Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 748-751, 1978.

[Zwicker ‘99] Zwicker, E.; Fastl, H., *Psychoacoustics: Facts and Models*, Springer Verlag, 2nd edition, May 1999.

View publication stats