Time Stretching And Pitch Shifting of Audio Signals
An Overview

by Stephan M. Bernsee, http://www.dspdimension.com

1. Introduction - Pitch Shifting

As opposed to the process of pitch transposition achieved using a simple sample rate conversion, Pitch Shifting is a way to change the pitch of a signal without changing its length. In practical applications, this is usually achieved by changing the length of a sound using one of the below methods and then performing a sample rate conversion to change the pitch.

There exists a certain confusion in terminology, as Pitch Shifting is often also incorrectly named 'Frequency Shifting'. A true Frequency Shift (as obtainable by modulating an analytic signal by a complex exponential) will shift the spectrum of a sound, while Pitch Shifting will dilate it, upholding the harmonic relationship of the sound. Frequency Shifting yields a metallic, inharmonic sound which may well be an interesting special effect but which is a totally inadequate process for changing the pitch of any harmonic sound except a single sine wave.

1.1 Audio Examples:

original sound (WAVE, 106k)	pitch shifted (WAVE, 106k)	frequency shifted (WAVE, 106k)

(Read my Audio Example Notes page for more information on how to use the above examples on your computer)

1.2 Time Compression/Expansion

Time Compression/Expansion, also known as "Time Stretching" is the reciprocal process to Pitch Shifting. It leaves the pitch of the signal intact while changing its speed (tempo). This is a useful application when you wish to change the speed of a voiceover without messing with the timbre of the voice.

There are several fairly good methods to do time compression/expansion and pitch shifting but most of them will not perform well on all different kinds of signals and for any desired amount of shift/stretch ratio. Typically, good algorithms allow pitch shifting up to 5 semitones on average or stretching the length by 130%. When time stretching and pitch shifting single instrument recordings you might even be able to achieve a 200% time stretch, or a one-octave pitch shift with no audible loss in quality.

2. Techniques Used for Time Compression/Expansion and Pitch Shifting

Currently, there are two different principal time compression/expansion and pitch shifting schemes employed in most of today's applications:

2.1 Phase Vocoder. This method was introduced by Flanagan and Golden in 1966 and digitally implemented by

Portnoff ten years later. It uses a Short Time Fourier Transform (which we will abbreviate as STFT from here on) to convert the audio signal to the complex Fourier representation. Since the STFT returns the frequency domain representation of the signal at a fixed frequency grid, the actual frequencies of the partial bins have to be found by converting the relative phase change between two STFT outputs to actual frequency changes. Note the term 'partial' has nothing to do with the signal harmonics. In fact, a STFT will never readily give you any information about true harmonics if you are not matching the STFT length the fundamental frequency of the signal - and even then is the frequency domain resolution quite different to what our ear and auditory system perceives. The timebase of the signal is changed by calculating the frequency changes in the Fourier domain on a different time basis, and then an iSTFT is done to regain the time domain representation of the signal.

Table 1: Fourier Transform Pointers:
Jean Baptiste Joseph Fourier bio
Discrete Time FT Basics
Dave Hales FFT Laboratory (requires Java capable browser)
S.M.Bernsee's DFT à Pied article (with C code)
Chris Bores' Online DSP Courses

Phase vocoder algorithms are used mainly in scientific and educational software products (to show the use and limitations of the Fourier Transform) but have gained in popularity over the past few years due to improvements that made it possible to greatly reduce the artifacts of the "original" phase vocoder algorithm. The basic phase vocoder suffers from a severe drawback because it introduces a considerable amount of artifacts audible as 'smearing' and 'reverberation' (even at low expansion ratios) due to the non-synchronized vertical coherence of the sine and cosine basis functions that are used to change the timebase. Puckette, Laroche and Dolson have shown that the phasiness can be greatly reduced by picking peaks in the Fourier spectrum and keeping the relative phases around the peaks unchanged. Even though this improves the quality considerably it still renders the result somewhat phasey and diffuse when compared to time domain methods.Current research focuses on improving the phase vocoder by applying intra-frame sinusoidal sweep and ramp rate correction (Bristow-Johnson and Bogdanowicz) and multi-resolution phase vocoder concepts (Bonada).

2.1.1 Related topics

There often is a certain confusion between a 'regular' (channel) and the phase vocoder. Both of them are, aside from technical details, obviously different in that they are used to achieve different effects. The channel vocoder uses two input signals to produce a single output channel while the phase vocoder has a one-in, one-out signal path. In the channel vocoder as applied to music processing, the modulator input signal is split into different filter bands whose amplitudes are modulating the (usually) corresponding filter bands splitting the carrier signal. More sophisticated (and expensive) approaches also separate voiced and unvoiced components in the modulator (or, for historical reasons 'speech') input, i.e. vowels and sibilancies, for independent processing.The channel vocoder can not be successfully applied to the time/pitch scaling problem, in the musical context it mainly is a device for analyzing and imposing formant frequencies from one sound on another. Both are similar in that they use filter banks (the STFT can be seen as a filter bank consisting of steep and slightly overlapping constant bandwidth filters) but a maximum of 22 are typical for channel vocoders while a phase vocoder usually employs a minimum of 512 or 1024 filter bands. The term Voice Coder (Vocoder) refers to the original application of the two processes in speech coding for military purposes.

2.1.2 Why Phase?

The term 'phase' in phase vocoder refers to the fact that the temporal development of a sound is contained in its phase information - while the amplitudes just denote that a component is present in a sound, phase contains the structural information. The phase relationship between the different bins will reconstruct time-limited events when the time domain representation is resynthesized. The phase difference of each bin between two successive analysis frames is used to determine that bin's frequencies deviation from its mid frequency, thus providing information about the bin's true frequency (if it is not a multiple of the STFT frame in its period) and thus making a reconstruction on a different time basis possible.

Table 2: Pointers, Phase Vocoder:
The MIT Lab Phase Vocoder
WaveMasher - GPL/Open Source Phase Vocoder by Kenneth Sturgis
Sculptor: A Real Time Phase Vocoder by Nick Bailey
A Phase Vocoder implementation using Matlab
More reading on the Phase Vocoder
The IRCAM "Super Phase Vocoder"
S.M.Bernsee's Pitch Shifting Using The Fourier Transform article (with C code)

Table 3: Pointers, sinusoidal modelling (Phase Vocoder-related technique):
SMS sound processing package (incl. executables for several platforms)

Lemur (Mac program along with references and documentation)

Table 4: Pointers, other interesting spectral manipulation tools
Macintosh programs
Windows programs

2.2 Time Domain Harmonic Scaling (TDHS). This is based on a method proposed by Rabiner and Schafer in 1978. It is heavily based on a correct estimate of the fundamental frequency of the sound processed. In one of the numerous possible implementations, the Short Time Autocorrelation of the signal is taken and the fundamental frequency is found by picking the maximum (alternatively, one can use the Short Time Average Magnitude Difference function and find the minimum, which is faster on an average CISC based computer systems). The timebase is changed by copying the input to the output in an overlap-and-add manner (therefore it's also sometimes referred to as '(P)SOLA' - (pitch) synchronized overlap-add method) while simultaneously incrementing the input pointer by the overlap-size minus a multiple of the fundamental frequency. This results in the input being traversed at a different speed than the original data was recorded at while aligning to the basic period estimated by the above method. This algorithm works well with signals having a prominent basic frequency and can be used with all kinds of signals consisting of a single signal source. When it comes to mixed-source signals, this method will produce satisfactory results only if the size of the overlapping segments is increased to include a multiple of cycles thus averaging the phase error over a longer segment making it less audible. For Time Domain Harmonic Scaling the basic problem is estimating the basic pitch period of the signal, especially in cases where the actual fundamental frequency is missing. Numerous pitch estimation algorithms have been proposed and some of them can be found in the following references:

Table 4: Pointers and References, TDHS/Pitch estimation
'C Algorithms for Realtime DSP' by Paul M. Embree, Prentice Hall, 1995 *(incl. source code diskette)*
'Numerical Recipes in C' by W. Press, S. Teukolsky, W. Vetterling, B. Flannery, Cambridge University Press, 1988/92 *(incl. source code examples, click title to read it online)*
'Digital Processing of Speech Signals' by L.R. Rabiner and R.W.Schafer, Prentice Hall, 1978 *(no source code, covers TDHS basics)*
'An Edge Detection Method for Time Scale Modification of Acoustic Signals', Rui Ren, Computer Science Department, Hong Kong University of Science and Technology.
'Dichotic time compression and spatialization' by Barry Arons, MIT Media Laboratory
Other papers related to Time Compression/Expansion by Barry Arons, MIT Media Lab

2.3 More recent approaches. Due to the amount of objectionable artifacts produced by both of the above methods, there have been a number of more advanced approaches to the problem of time stretching and pitch shifting in the past years. One particular problem of both the TDHS and Phase Vocoder approaches is the high localization of the basis functions (where this term is applicable) in one domain with no localization in the other. The sines and cosines used in the Phase Vocoder have no localization in the Time Domain, which without further treatment contributes to the inherent signal smearing. The sample snippets used in the TDHS approach can be seen as having no localization in the frequency domain, thus causing multi-pitched signals to produce distortion.

Improving existing techniques: Scientific research currently focuses on improving both time and frequency domain methods by investigating and eliminating the possible causes of the artifacts in both domains. For example, there have been numerous improvements to the phase vocoder that were implemented in commercial products recently due to the availability of fast CPU speeds on desktop computers. Among these there is the idea to vertically synchronize phases across a phase vocoder analysis frame which was an idea originally conceived by Miller Puckette in 1995. This assumes tracking and identifying individual harmonics py peak-picking and peak-tracking, which in itself poses new problems, but the result is much more agreeable than that of a "crude" phase vocoder. One commercial product utilizing this improved phase vocoder is Serato's Pitch'n Time whose algorithm is explained in detail here.

Adaptive basis transform algorithms: Aside from this, several entirely new methods have been devised. A method which was developed by Prosoniq uses an approach of representing the signal in terms of more complex basis functions that have a good localization in both the time and frequency domain (like certain types of wavelets have). The signal is transformed on the basis of the proprietary **MCFE** (Multiple Component Feature Extraction), for which the details are shrouded in trade secret but some information is available at the MPEX web site.

Wavelet and multiresolution techniques: The "Dirac" technology comes in a free cross-platform C/C++ object library that exploits the good localization of wavelets in both time and frequency to build an algorithm for time and pitch manipulation that uses an arbitrary time-frequency tiling depending on the underlying signal.

Additionally, the time and frequency localization parameter of the basis can be user-defined, making the algorithm smoothly scalable to provide either the phase coherence properties of a time domain process or the good frequency resolution of the phase vocoder.

Goofs: It is also worth mentioning that there have been some approaches that are flawed or nonsensical. Table 5a lists the most obvious one. The method proposed by Garas and Sommen for example will not work at all in the form he proposed, it is curious (yet understandable) that noone has noticed this. The sound files he has initially provided were flawed, too, and were silently buried when I began asking questions. In the light of recent developments in the area of improving the phase vocoder, they might be still some day prove to be interesting.

Table 5: Pointers, More recent approaches
The free Dirac Cross-Platform Library
The Prosoniq MPEX Time/Pitch manipulation technology (licensing of binary object code)
Scott Levine, Tony Verma, Julius O. Smith III. Alias-Free, Multiresolution Sinusoidal Modeling for Polyphonic, Wideband Audio. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, Mohnonk, NY, 1997.
Scott Levine, Julius O. Smith III. A Sines+Transients+Noise Audio Representation for Data Compression and Time/Pitch-Scale Modifications. 105th Audio Engineering Society Convention, San Francisco 1998.
Aaron Master. Peak-adaptive Phase Vocoder. ICASSP-02.

Table 5a: Pointers, nonsensical approaches
Time/Pitch Scaling Using The Constant-Q Phase Vocoder, J. Garas, P. Sommen, Eindhoven University of Technology, The Netherlands

3. Comparison

It is very difficult to objectively rate or compare various time compression/expansion and pitch shifting algorithms with regard to quality due to their nonlinearity and signal dependency. It is highly difficult to establish a solid measure to estimate their overall performance from simple test signals, because most of them tend to do very well with test signals due to their simple structure. There have been some proposals by Laroche and Dolson to estimate the "phase coherence" from a set of variables obtained from analyzing the sound via the STFT. This is a good approach worth further studies, but still far from providing the type of judgements you can get from extensive listening tests, which I believe is still the method of choice for estimating the quality of a time compression/expansion algorithm.

It is safe to say that none of the algorithms available today is free from flaws and problems across an arbitrary range of stretch ratios, even though many of them come very close to achieving a good quality. As is to be expected, the phase vocoder-based algorithms have to fight residual smearing which renders the results less "punchy" and direct. The time domain methods have to cope with residual distortion, most notably when processing sounds that have critical harmonic relationships in their harmonics.

And even though I realize that this might be a futile attempt to provide a comprehensive overview, I have produced a small number of excerpt audio examples of the various methods as well as some screen shots of impulse responses to show the performance in quality and coherence of each method in comparison.

3.1 Which Method To Use. Principally, this is dependent on the constraints imposed on the actual task, which may be one of the following:

Speed. If you plan on using the method in a realtime application that has many parallel audio tracks or needs many pitch shifted voices, TDHS is probably the best option unless you have a STFT representation of the signal already at hand. Using different optimization techniques, the performance of this approach can be fine tuned to run on any of today's computer in realtime.

Material. If you have a prior knowledge about the signal the algorithm is supposed to work well with, you can further choose and optimize your algorithm accordingly (see below).

Quality. If the ultimate goal of your application is to provide the highest possible quality without performance restrictions, you should decide with the following two important factors in mind: 1) TDHS gives better results for small timebase and pitch changes, but will not work well with most polyphonic material. 2) Phase Vocoder gives smoother results for larger changes and will also work well with polyphonic material but introduces signal smearing with impulsive signals if this is not being dealt with. Even though some methods might indicate that the CPU power can be reduced by preventing the phasiness, ultimately reducing it can cost significantly more CPU cycles than the "regular" phase vocoder.

3.2 Pitch Shifting Considerations: If your goal is to alter the pitch, not the timebase, bear in mind that when upscaling the pitch, echoes and the repetituous behaviour of TDHS are less obvious since the pitch change moves

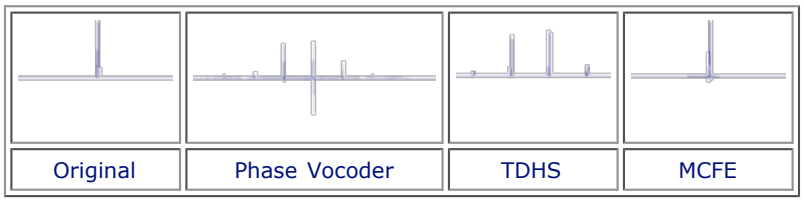
adjacent peaks (echoes) closer to each other in time, thus masking them to the ear. The (pre)smearing behaviour of the Phase Vocoder will be more disturbing in this case, since it occurs before the transient sounds and will easily be recognized by the listener.

3.3 Audio Examples:

Example 1: original sound (WAVE, 106k)	200% time stretched, Phase Vocoder (WAVE, 209k)	200% time stretched, TDHS (WAVE, 209k)	200% time stretched, MCFE (WAVE, 209k)
--	block size: 2048 samples, STFT size: 8192 samples, frame overlap: 1024 samples	block size: 2048 samples, frame overlap: 1536 samples	block size: 1806 samples, frame overlap: 903 samples
Example 2: original sound (WAVE, 230k)	200% time stretched, Phase Vocoder (WAVE, 432k)	200% time stretched, TDHS (WAVE, 451k)	200% time stretched, MCFE (WAVE, 451k)
--	block size: 2048 samples, STFT size: 8192 samples, frame overlap: 1024 samples	block size: 2048 samples, frame overlap: 1536 samples	block size: 1806 samples, frame overlap: 903 samples

(Read my Audio Example Notes page for more information on how to use the above examples on your computer)

Impulse Response Diagrams (achieved using the same settings as for the above audio examples, click to view in detail):



4. Timbre and Formants

Since timbre (formant) manipulation is actually a pitch shifting related topic, it will also be discussed here. Formants are prominent frequency regions produced by the resonances in the instrument's body that very much determine the timbre of a sound. For human voice, they come from the resonances and cancellations of the vocal tract, contributing to the specific characteristics of a speaker's and singer's voice.

If the pitch of a recording is shifted, formants will be moved thus producing the well known 'Mickey-Mouse' effect audible when shifting the pitch. This is usually an unwanted side effect since the formants of a human singing at a higher pitch do not change their position.

To compensate for this, there exist formant correction algorithms that restore the position of the formant frequencies after or during the pitch shifting process. They also allow changing the gender of a singer by scaling formants without changing pitch.

For each of the above pitch shifting methods there exists a corresponding method for changing the formants to compensate for the side effects of the transposition.

4.1 Phase Vocoder and Formants. Formant manipulation in the STFT representation can be done by first normalizing the spectral amplitude envelope and then multiplying it by a non-pitch shifted copy of it. This removes the new formant information generated through the pitch shifting and superimposes the original formant information thus yielding a sound similar to the original voice. This is an amplitude-only operation in the frequency domain and therefore does not involve great additional computational complexity. However, the quality may not be optimal in all cases due to STFT resolution issues.

4.2 Time Domain Harmonic Scaling and Formants. Changing the formants in the time domain is simple, however, efficient implementation is tricky. TDHS in essence can be implemented and regarded as a granular

synthesis using grains of one cycle of the fundamental in length being output at the destination new fundamental frequency rate. Simply put: if each grain is 1 cycle in length and since [cycles/sec] is the definition of fundamental pitch in this case, the output rate of these grains determines the new pitch of the sample. In order to not lengthen the sample, some grains have to be discarded in the process. Since no transposition takes place, the formants will not move. On the other hand, applying a sample rate change to the individual grains results in a change of formants without affecting the pitch. Thus, pitch and formants can be independently moved. The obvious disadvantage of the process is its dependency on the fundamental frequency of the signal, making it unsuited for the application to polyphonic material. See also: 'A Detailed Analysis of a Time-Domain Formant-Corrected Pitch-Shifting Algorithm', by Robert Bristow-Johnson, Journal of the Audio Engineering Society, May 1995. This paper discusses an algorithm previously proposed by Keith Lent in the Computer Music Journal.

Table 6: Pointers, Formant Manipulation
The DSP Dimension Formant Correction page.
An LPC Approach: 'Voice Gender Transformation with a Modified Vocoder' (May 1996), Yoon Kim at CCRMA

The following newsgroups can be acessed for more information and help on the time compression/expansion and pitch shifting topic.

Table 7: News Groups
comp.dsp comp.music.research

If you're seeking general information on DSP, browse to the DSPguru homepage.
