PhaVoRIT: A Phase Vocoder for Real-Time Interactive Time-Stretching

Thorsten Karrer, Eric Lee, and Jan Borchers
Media Computing Group
RWTH Aachen University
52056 Aachen, Germany
{karrer, eric, borchers}@cs.rwth-aachen.de

Abstract

Time-based, interactive systems for digital audio and video often employ algorithms to control the speed of the media whilst maintaining the integrity of the content. The phase vocoder is a popular algorithm for time-stretching audio without changing the pitch; its characteristic "transient smearing" and "reverberation" artifacts, however, limit its application to "simple" audio signals such as instrumental or vocal music. We propose three new methods to further improve the audio quality of phase vocoder-based time-stretching: multiresolution peak-picking *accounts for the non-uniform frequency resolution of the human auditory system;* sinusoidal trajectory heuristics *constrain the set of spectral peaks that are candidates for sinusoidal trajectory continuation, thereby reducing phase propagation along incorrectly detected trajectories; and* silent passage phase reset *re-establishes the vertical phase coherence at certain intervals. These techniques have been implemented as modules of the* PhaVoRIT *time-stretching software, available as a plug-in to Apple's Core Audio framework, and the* Semantic Time Framework*, a multimedia framework we developed for time-based interactive systems. We obtained favorable results when comparing* PhaVoRIT *to existing, commonly available audio time-stretching software tools in a formal user study.* PhaVoRIT *is also deployed as the audio time-stretching engine for* Maestro!*, an interactive conducting exhibit installed in the Betty Brinn Children's Museum in Milwaukee, USA.*

1 Introduction

Development, production, and use of digital media has been steadily increasing in recent years, and while the technology to store and process this content has become quite advanced, the types of interaction that they support have remained relatively stagnant. VCR metaphors, such as *play, stop, fast forward, rewind* remain the standard for interaction with most types of time-based media such as music, movies, and audio books.

Our previous research in new interaction techniques with time-based media resulted in a series of interactive museum exhibits for digital audio and video recordings, including *Personal Orchestra* (Borchers et al. 2004), *You're the Conductor* (Lee, Nakra, and Borchers 2004), and *Maestro!* (Lee et al. 2006). These systems all allow the user to directly control the time evolution of an audio and video recording using conducting gestures performed with a baton.

It is clear that the time-based interaction afforded by the above systems requires an ability to freely control the playback speed. Unfortunately, in the case of digital PCM audio, the naïve method of changing playback speed by resampling has the unfortunate side effect of changing the pitch as well. There has been an abundance of research in pitch-preserving time-stretching, and existing algorithms can be broadly categorized as "time domain" or "frequency domain" algorithms. Time domain algorithms, while adequate for structurally simple signals such as speech, prove unacceptable for more complex signals such as polyphonic music. Moreover, degradations in quality are often tolerated for time-stretched speech as long as the resulting output remains understandable; subtle degradations in quality for time-stretched music, however, often destroy the user's overall experience. Much of the recent research in time-stretching has thus been based on the phase vocoder, a frequency-domain algorithm for time-stretching. Audio that is time-stretched using a phase vocoder, however, exhibits characteristic "transient smearing" and "reverberation" artifacts.

The development of *PhaVoRIT*, a **Pha**se **Vo**coder for **R**eal-Time **I**nteractive **T**ime-Stretching, was motivated by a need for an improved audio time-stretching engine for *Maestro!*, our latest interactive conducting exhibit installed as a permanent exhibit at the Betty Brinn Children's Museum in Milwaukee, USA (Lee et al. 2006). We required an algorithm that performs time-stretching of complex polyphonic audio signals such as orchestral music, over a wide range of stretching factors, in real-time and with low latency. We also wanted *PhaVoRIT* to be modular to support both slow and fast machines, and to more easily integrate into existing and upcoming frameworks for time-based media.

We begin with an overview of recent work in audio time-stretching, followed by a brief description of the basic phase vocoder algorithm and existing techniques to address transient smearing and reverberation artifacts. Then, we describe in detail the techniques we developed to further improve the quality of the phase vocoder algorithm. The first technique, multiresolution peak-picking, was introduced in (Lee, Karrer, and Borchers 2006) together with previous work; however, the remaining two techniques introduced in this paper, sinusoidal trajectory heuristics and silent passage phase reset are original. We conclude with the results of a formal user study comparing our implementation of *PhaVoRIT* with existing work.

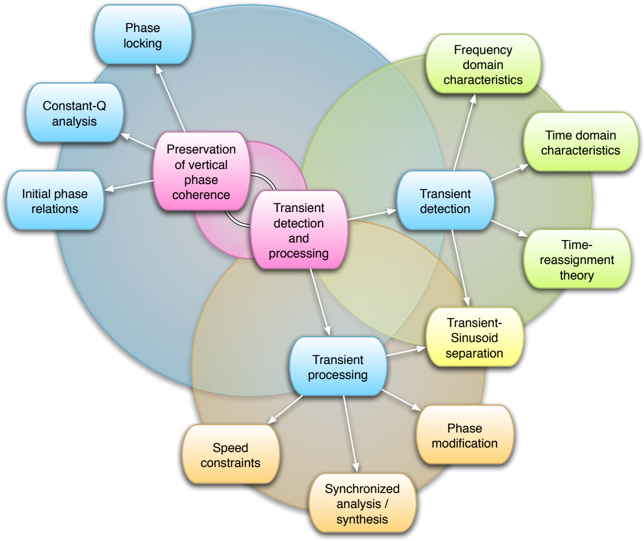

Phase locking
Constant-Q analysis
Initial phase relations
Preservation of vertical phase coherence
Transient detection and processing
Frequency domain characteristics
Time domain characteristics
Transient detection
Time-reassignment theory
Transient-Sinusoid separation
Transient processing
Speed constraints
Phase modification
Synchronized analysis / synthesis

Figure 1: Relationships between some of the time-stretching methods reviewed for the development of *PhaVoRIT*.

2 Related Work

A large body of research has focused on improving the phase vocoder's audio quality, which exhibits characteristic "reverberation" and "transient smearing" artifacts. As it would not be possible to give a detailed review of all existing work here, we provide a brief overview of those that are most relevant here, and refer the reader to (Karrer 2005) for a more detailed review. Figure 1 shows a graphical classification of existing techniques to address these problems.

Laroche and Dolson (1997) discovered that the loss of vertical phase coherence in the synthesized output signal was a primary cause of reverberation artifacts. In (Laroche and Dolson 1999), they extended Puckette's *phase-locked vocoder* (Puckette 1995) and proposed the *identity phase-locking* and *scaled phase-locking* techniques to preserve vertical phase coherence in the time-stretched audio. *PhaVoRIT* builds on this latter technique and introduces some further enhancements. Approaches to solve the peak-picking and peak-matching problems that arise in this class of phase vocoders have long been addressed in the context of sinusoidal modeling (McAulay and Quatieri 1986). The Constant-Q phase vocoder (Garas and Sommen 1998) attempts to incorporate psychoacoustics into the time-stretching algorithm, although the validity of this particular technique has also been debated (Bernsee 2005). In the specific case of integer stretching factors, Laroche and Dolson (1997) developed a formula to calculate a set of initial synthesis phases for the first time-frame that preserve vertical phase coherence.

The process of reducing transient smearing can be divided into two steps: detecting transient events in the input signal and processing these events to preserve the transient characteristics in the time-stretched output signal.

Transient detection techniques include algorithms that analyze the energy signal's first derivative (Levine and Smith III 1998), measure the spectral dissimilarity (Masri 1996), use Mel-Cepstrum coefficients (Bonada 2000), or detect transients by observing the signal's high frequency content (HFC) (Masri and Bateman 1996).

The detected transients can then be preserved, for example, by setting the time-stretching factor to one for the duration of the event and catch up later (Hammer 2001), or by synchronizing the position of the analysis windows to the position of the transient events (Masri and Bateman 1996).

Finally, a transient detection and preservation scheme based on time-reassignment theory (Auger and Flandrin 1995) has been developed by Röbel (2003).

While we incorporated techniques to reduce transient smearing into final implementation of *PhaVoRIT*, the techniques we introduce in this paper focus on the problem of reducing reverberation artifacts.

3 Phase Vocoder Basics

The basic phase vocoder algorithm divides the signal $x(t)$ into overlapping windows of length N that start every R_a samples. By overlap-adding these windows using a different spacing R_s, the duration of the output signal $y(t)$ is changed by a factor $\alpha = \frac{R_s}{R_a}$, where R_a and R_s are the *analysis hop* *factor* and the *synthesis hop factor*, respectively. The phases of the signal's short time Fourier transform (STFT) must be adjusted prior to this respacing to avoid phase jumps between the re-spaced windows.

Let $t_a^u = uR_a$ and $t_s^u = uR_s$ ($u \in \mathbb{N}$) be the corresponding analysis and synthesis time points, $\Omega_k = \frac{2\pi k}{N}$ denotes the center frequency of the kth FFT-bin, $X(t_a^u, \Omega_k)$ denotes the STFT of the input signal $x(t)$ at time t using a window function $h(t)$ of length N, and $Y(t_s^u, \Omega_k)$ denotes the modified STFT from which the output signal $y(t)$ will be synthesized. To calculate the correct synthesis phases, the instantaneous frequency $\hat{\omega}_k(t_a^u)$ has to be determined for each FFT-bin k:

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

$$\hat{\omega}_k(t_a^u) = \Omega_k + \frac{\Delta_p\Phi_k^u}{R_a} \qquad (2)$$

where $\Delta_p\Phi_k^u$ is the principal determination ($\in [-\pi, \pi]$) of the phase deviation $\Delta\Phi_k^u$. Now, the synthesis phases $\angle Y(t_s^u, \Omega_k)$ can be found by advancing from the synthesis phases in the last timeframe $\angle Y(t_s^{u-1}, \Omega_k)$ for the duration of one synthesis hop R_s at the rate of the FFT-bin's instantaneous frequency $\hat{\omega}_k(t_a^u)$:

$$\angle Y(t_s^u, \Omega_k) = \angle Y(t_s^{u-1}, \Omega_k) + R_s\hat{\omega}_k(t_a^u) \qquad (3)$$

The magnitudes of the chunk to be synthesized are set to the original values, as the time-evolution of the sinusoids' amplitudes is automatically time-stretched by reconstructing the chunks using a different synthesis hop factor:

$$|Y(t_s^u, \Omega_k)| = |X(t_a^u, \Omega_k)| \qquad (4)$$

$$Y(t_s^u, \Omega_k) = |Y(t_s^u, \Omega_k)| \cdot e^j \angle Y(t_s^u, \Omega_k) \qquad (5)$$

After that, the synthesis spectrum $Y(t_s^u, \Omega_k)$ is transformed back to the time domain by means of an iFFT and overlap-added with an offset of R_s samples to the already synthesized part of the output signal $y(t)$. Due to the phase modification, the windows will overlap smoothly and the composite signal will not exhibit any phase discontinuities. A more detailed explanation of the phase vocoder algorithm is given in (Flanagan and Golden 1966) and (Dolson 1986).

3.1 Scaled Phase-Locking

PhaVoRIT builds upon the *scaled phase-locking* phase vocoder developed by Laroche and Dolson (1999). We briefly summarize their contribution here, and refer the reader to their original paper (Laroche and Dolson 1999) for a more detailed discussion.

Scaled phase-locking starts by identifying bins that contain peaks in the amplitudes of the STFT spectrum, assuming that those bins correspond to the most important sinusoids in the signal. The synthesis phases of these bins are then computed using a refined version of the phase vocoder algorithm that takes into account that a sinusoid might switch from channel k_0 at frame $u-1$ to channel k_1 at frame u, and uses the appropriate analysis phases when calculating the instantaneous frequency. In particular, equation (1) becomes:

$$\Delta\Phi_{k_1}^u = \angle X(t_a^u, \Omega_{k_1}) - \angle X(t_a^{u-1}, \Omega_{k_0}) - R_a\Omega_{k_1} \qquad (6)$$

Since the synthesis phase calculation for channel k_1 should be based on the last synthesis phase on the sinusoid's trajectory through the spectrogram, equation (3) is changed into:

$$\angle Y(t_s^u, \Omega_{k_1}) = \angle Y(t_s^{u-1}, \Omega_{k_0}) + R_s\hat{\omega}_{k_1}(t_a^u) \qquad (7)$$

To determine which peak in frame $u-1$ corresponds to the peak at Ω_{k_1} in frame u, the peak in frame $u-1$ that is closest to channel Ω_{k_1} is identified. The non-peak bins are said to belong to the closest peak's region of influence, and are then phase-locked to that peak using the phase-locking equation:

$$\begin{aligned}\angle Y(t_s^u, \Omega_k) = {} & \angle Y(t_s^u, \Omega_{k_1}) + \\ & \beta\left[\angle X(t_a^u, \Omega_k) - \angle X(t_a^u, \Omega_{k_1})\right]\end{aligned} \qquad (8)$$

where β is a phase scaling factor. In this way the phase relations between each peak channel and its neighboring non-peak channels are carried over from the original signal to the time-stretched signal which helps to preserve vertical phase coherence.

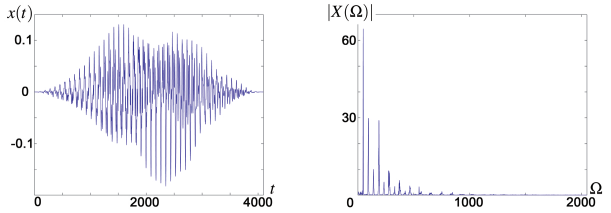
4 Design

4.1 Multiresolution peak-picking

Laroche and Dolson's scaled phase-locking technique identifies the peaks of the signal that are to be phase-locked using a simple local maxima search over a fixed interval in the frequency spectrum. Unfortunately, such a scheme, while simple, also introduces artifacts in the resulting audio signal, that can be heard as a shallow bass and musical overtones, similar to a badly compressed MP3 file. We refer to this peak-picking technique as "constant resolution peak-picking", and propose a multiresolution technique where the peak detection function is made frequency dependent, in order to address these shallow bass and musical overtone artifacts.

Two arguments can be given for this modification of the peak-picking stage: the non-uniform characteristics of the human auditory system, and the peak distribution in the spectra of audio signals.

The human ear processes sound by performing a non-linear frequency-to-location transform at the basilar membrane inside the inner ear. In particular, the critical bands,



(a) Windowed chunk of an audio signal taken from the "Kleine Nachtmusik" performed by the Vienna Philharmonic.

(b) Spectrum of the waveform shown in Figure 2(a). The peak distribution is visibly non-uniform.

Figure 2: Sound chunk of orchestral music and its amplitude spectrum.

the frequency regions inside which different sounds are perceived as one, are distributed logarithmically. Thus, the phase vocoder should take into account this nonlinearity and not put effort into modeling independently different sinusoids in frequency regions where the human auditory system perceives them as one sinusoid anyway.

We further considered the distribution of an audio signal's partials over the frequency spectrum. Most "natural" audio signals—especially music and speech, which are the ones most important for our purposes—exhibit a non-uniform distribution of their partials. Apart from transient events, most of the spectral peaks are concentrated in the low-frequency bands and become fewer at the high frequency bands, roughly following a logarithmic scale (see Figures 2(a), 2(b)). This fact is neglected by the constant resolution peak-picking algorithm used in the scaled phase-locking phase vocoder. Its peak-picking criterion is independent of frequency and takes into account spectral information only locally in a very small area around each bin. The result is that in the very low frequency band, some channels are phase-locked to neighboring ones, even though they contain full partials, and not sidelobes; similarly, in the high frequency band, peaks are detected that do not stem from sinusoids inhabiting that bin, but are merely caused by the noisy or the transient components of the signal. Phase-locking parts of the high frequency spectrum to a spurious peak that is caused by noise in the signal results in a phase value that is completely uncorrelated with the rest of the time-frequency evolution of the signal. The end result is an undesired change in the sound characteristics of the output signal, and should thus be avoided.

Both arguments support that the peak-picking strategy should be modified to use a frequency dependent peak detection function. Consequently, we have developed a multiresolution peak-picking algorithm that implements our proposed peak-picking strategy as part of *PhaVoRIT*. In the lower frequency bands the detection function should have a high resolution, detecting every peak, whereas in the high frequency bands only very salient peaks should be detected, thus a lower resolution is desired (see Figures 3 and 4). In *PhaVoRIT*, we approximate the human ear's response to frequency as a logarithmic function and divide the spectrum into appropriately sized regions (see Figure 4) rather than uniformly applying the local maxima search over the entire spectrum. For a STFT window size of 4096 samples, we assumed the lowest 16 frequency bins contain a peak corresponding to a sinusoid audible to the human ear. For the next 16 bins, a bin is considered to contain a peak if its amplitude is larger than both of its neighboring bins. For the next 32 bins, the amplitude must be larger than its two neighboring bins to either side, and so on.

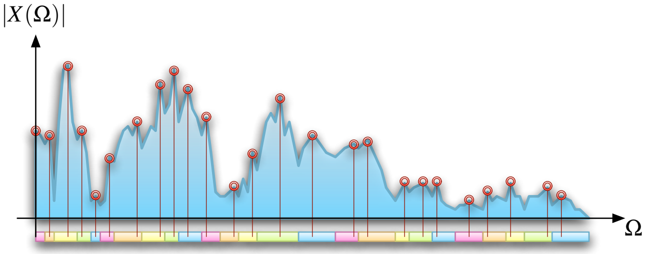


Figure 3: Peak-picking as it is performed by the scaled phase-locking phase vocoder. The peak-picking resolution is constant over the whole spectrum.

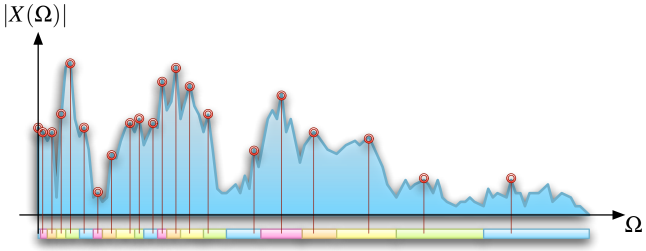


Figure 4: Multiresolution peak-picking. The peak-picking resolution is distributed logarithmically, resulting in more peaks at lower frequencies and fewer peaks at the higher ones.

Our listening tests with users have found that this refined peak-picking algorithm produces an audible improvement in the bass and reduces the musical overtones in the time stretched signal.

4.2 Sinusoidal trajectory heuristics

One advantage of scaled phase-locking is that it is able to preserve the phase coherence of a sinusoid that moves be-

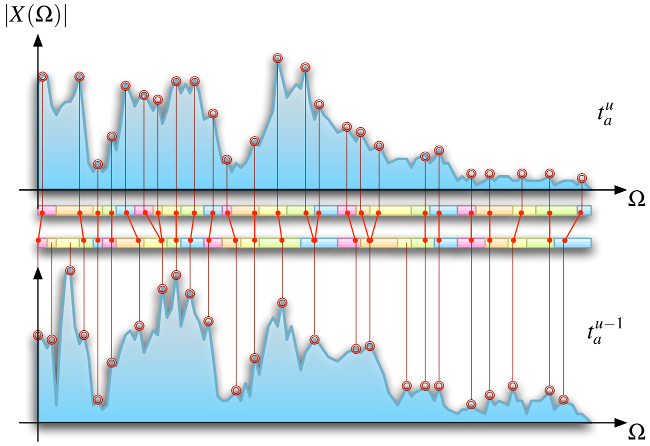


Figure 5: Each peak is linked to its believed predecessor. For this, the peaks are projected to the last timeframe's regions of influence, then linked to those regions' peaks.

tween frequency bins across sample windows. However, we discovered that at note onsets and in frequency areas that are sparsely populated with peaks, the algorithm often incorrectly correlates two unrelated sinusoids, causing the estimated sinusoidal trajectory to jump across multiple frequency channels. This behavior results in blurred note onsets and high frequency 'warbling' artifacts in the output audio signal. The problem can be mitigated, however, by imposing additional constraints that follow a simple heuristic on the continuation process.

The scaled phase-locking algorithm performs sinusoidal trajectory continuation as follows: If a peak is identified at time t_a^u in channel Ω_{k_1} it is automatically assumed that this peak is part of a partial of time-varying frequency. Consequently, the predecessor of the peak is believed to be the peak at channel Ω_{k_0} at time t_a^{u-1}, where Ω_{k_1} was located inside the region of influence of Ω_{k_0} at that time (see Figure 5).

As described earlier, the phase propagation formula (7) for calculating the peak's synthesis phase $\angle Y(t_s^u, \Omega_{k_1})$ is then based on the synthesis phase $\angle Y(t_s^{u-1}, \Omega_{k_0})$. Problems arise when the distance between the bins at Ω_{k_1} and at Ω_{k_0} is large. In that case the peaks will almost certainly not belong to the same partial, but are completely unrelated, and thus should not be forced to form a sinusoidal trajectory. This often happens in the sparsely populated high frequency bands where peaks caused by noise or transient events suddenly appear, or at note onsets, where a new sinusoidal trajectory starts that has little to do with the existing partials of the signal (see Figure 6). Sinusoidal trajectory heuristics decide if the peak should be regarded as a continuation of an existing trajectory, or if it should be the start of a new trajectory.

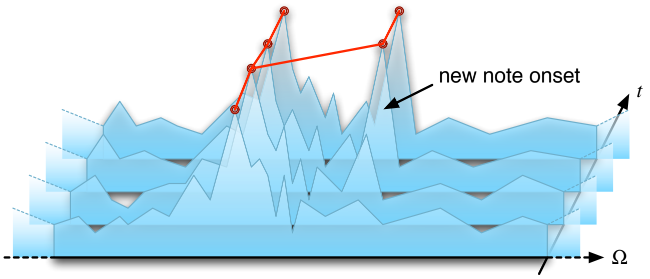


Figure 6: Part of the frequency spectrum with estimated sinusoidal trajectory as it is formed by the scaled phase-locking phase vocoder algorithm. The trajectory forks at new note onsets calculating the synthesis phase of the new note's peak based on that of a different and unrelated note.

A similar approach has been suggested in (McAulay and Quatieri 1986) but in contrast to their peak-matching algorithm that is purely based on the frequency distance between the peak and its assumed predecessor we use a heuristic that additionally depends on the frequency of the peak's channel. In the low frequency bands, only peaks with very short distance in frequency are linked by the phase propagation formula to form a sinusoidal trajectory. In the high frequency bands, the "allowed" distance in frequency is larger, although still limited compared to basic scaled phase-locking. Every peak that has no predecessor inside the allowed frequency distance is considered to be the start of a new trajectory (see Figure 7) and has its phase value calculated by the standard phase propagation equation (3).

In our current implementation the 4096 bins of the FFT are divided into seven sub-bands (see Table 1). The maximum distance in frequency for searching peak predecessors in those sub-bands is equal to the index number of each subband. Thus, if a peak was present in bin 142 at time u it would continue the sinusoidal trajectory of the closest peak at time $u-1$ in the region between bins 129–256. If there were no peaks in this region at time $u-1$ the peak in bin 142 would be considered to start a new sinusoidal trajectory.

4.3 Silent passage phase reset

A further problem with the phase vocoder algorithm is the cumulative nature of phase unwrapping errors, which can never be eliminated completely, especially if the input signal is a very long piece of audio. As has been shown in (Laroche and Dolson 1997) those errors are partly responsible for the characteristic reverberation artifacts of the phase vocoder. We introduced a very simple but effective method to completely restore vertical phase coherence at certain points in the sig-

Sub-band Index	FFT Bin Range
1	Bins 0 – 16 $\simeq$ 0 Hz – 172 Hz
2	Bins 17 – 32 $\simeq$ 183 Hz – 345 Hz
3	Bins 33 – 64 $\simeq$ 355 Hz – 689 Hz
4	Bins 65 – 128 $\simeq$ 700 Hz – 1378 Hz
5	Bins 129 – 256 $\simeq$ 1389 Hz – 2756 Hz
6	Bins 257 – 512 $\simeq$ 2767 Hz – 5513 Hz
7	All remaining bins

Table 1: The spectrum is divided into sub-bands with frequency ranges growing towards the high frequencies.

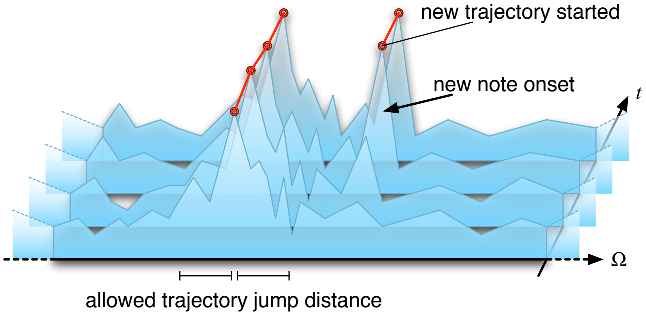


Figure 7: Part of the frequency spectrum with estimated sinusoidal trajectory as it is formed by our sinusoidal trajectory heuristics algorithm. The heuristic prohibits some of the unwanted "trajectory forks", thus allowing new notes to develop their phase independent of other already existing notes.

nal. If the current signal chunk's energy drops below a fixed threshold—in our current implementation[1] the energy in the short time spectrum has to drop below a value of -21 dB—, we assume that the signal does not contain any meaningful audible events at that time point, but is mostly background noise. As soon as the signal energy rises again, exceeding another threshold (-19 dB), all synthesis phases of the time-stretched signal are reset to their analysis phases, giving the phase vocoder a "clean start". The resulting "click" caused by this phase jump is barely audible since the signal energy is still very low, and is furthermore masked by the rise in the signal energy.

5 Implementation

Our system has been implemented both as an Audio Unit for Apple's Core Audio framework, and as a plug-in for the Semantic Time Framework (Lee, Karrer, and Borchers 2006). Due to its internal modular structure, it is possible to switch at run-time between different time-stretching algorithms, such as the basic phase vocoder, the scaled phase-locking algorithm, and *PhaVoRIT*. We have also incorporated a variant of Röbel's transient detection and preservation scheme (Röbel 2003) into *PhaVoRIT*.

The hardware requirements for real-time operation are modest: a PowerPC G4 iBook at 800MHz is sufficient to run *PhaVoRIT*.

6 Evaluation

While there have been measures being proposed to evaluate the audio quality of time-stretching algorithms (Laroche and Dolson 1999), these are not clear indicators for the existence of phase vocoder artifacts in the output signal. Moreover, since the ultimate consumer of the output is a human being, we chose to use the results of formal listening tests to assess the audio quality of *PhaVoRIT*.

6.1 User study

We had a group of 60 users compare the quality of six different time-stretching algorithms—the scaled phase-locking phase vocoder with constant resolution peak-picking (SPL), the scaled phase-locking phase vocoder with multiresolution peak-picking (MRPP), *PhaVoRIT*, Native Instruments' *Traktor DJ Studio*, Serato's *Pitch'n Time*, and Prosoniq's *MPEX-2*. The latter three are commercially available, high quality time-stretching tools.

For the experiment we selected a set of six snippets of music[2], each posing a special problem to the time-stretching algorithms (see Table 2). The sound snippets were time-stretched by all algorithms using six different stretching factors: 50%, 85%, 115%, 146%, 189%, and 200%. Since the Pitch'n Time and MPEX-2 time-stretchers are not operating in real-time and all algorithms have different ways to handle stereo sound, we time-stretched the sound snippets—real-time if the algorithm was able to, or offline otherwise—, mixed the channels down to mono, and normalized the volume. This resulted in a testing set of $6 \times 6 \times 6 = 216$ uncompressed audio files plus the 6 original snippets for reference. [3].

To avoid learning effects, we divided the test so that each participant would only rate the set of audio files belonging to one song. The subjects were asked to produce only one rating that would reflect their judgment on the audio quality of the algorithms averaged over all six different stretching factors, where each algorithm could be given a grade between

[1] Our current implementation processes audio sampled at 44.1 kHz, with a STFT and Hanning window size of 4096 samples.

[3] Three of the six sets of sound samples can be downloaded at *http://media.informatik.rwth-aachen.de/karrerICMC2006.html*

one and ten points, ten being the best. Consequently, each participant had to listen to $6 \times 6 + 1$ audio files: one song at six different stretching factors each produced by six different algorithms plus the original song snippet for reference. To achieve a reliable assessment, we decided upon 10 ratings per song, resulting in a total of 60 participants.

6.2 Results

When comparing mean ranking values across the whole set, *PhaVoRIT* scored second-best (see Table 2); these results confirm that *PhaVoRIT* is a significant improvement over Laroche's scaled phase-locking scheme, and is moreover comparable to algorithms used in existing audio software, some of which do not operate in real-time.

A more detailed analysis for each musical piece revealed more subtle findings. While the other algorithms' performance was strongly dependent on the kind of music, *PhaVoRIT* received very steady score rankings (3× second best, 3× third best of six). Interestingly, the commercially available time-stretchers got their best results when transients were numerous but clearly defined (Lovesong, SmoothSailing), whereas the more basic algorithms were preferred by the test subjects for classical music (the dense and transient-heavy Radetzky march, and the almost monophonic Kleine Nachtmusik which does not contain any salient transients). There are two explanations for why this might be the case:

- The classical pieces were recorded in the same setting (artists, technical crew, equipment, location) on the same day. Some characteristics of these special recordings might be better reproduced by the SPL and MRPP algorithms.

- The test subjects might have subconsciously preferred a "warmer" and "softer" sound when listening to classical music; they would then be more forgiving to the weaker transient reproduction of the SPL and MRPP algorithms. It might have even been a positive side-effect if the note onsets of certain instruments (e.g., strings) were rendered especially soft. On the other hand, clear and crisp transients are more common for contemporary music, and thus *PhaVoRIT*, Traktor, and Pitch'n Time were ranked the highest.

The remaining two pieces of music (Narcotic, Poison) have their spectra dominated by noise; the audio quality of all time-stretching algorithms seems comparable with a slight bias in favor of Pitch'n Time and *PhaVoRIT* which were ranked best respectively second best in both experiments.

7 Future Work

During our user test, we found the level at which the stereo image is preserved in time-stretched audio varies depending on the algorithm. We believe there is more work to be done on techniques for preserving the phase relations between the different channels of audio in a multichannel recording (e.g., stereo, surround sound).

We believe the transient detection currently employed in *PhaVoRIT* can be made more robust by analyzing the signals' high frequency content (Masri and Bateman 1996) as an additional detection measure. This measure could also be part of a future signal separation stage that splits the signal into its sinusoidal, transient, and noise components before time-stretching, enabling us to use more specific time-stretching algorithms optimized for the appropriate sound models. Previous work in this area including (Duxbury, Davies, and Sandler 2001), (Hammer 2001), and (Levine and Smith III 1998) could be used as a starting point.

We propose some further experiments with our newly developed techniques to further improve audio quality:

- Modify the division of the spectrum into sub-bands that is performed for both the multiresolution peak-picking and the sinusoidal trajectory heuristics such that the sub-bands are equivalent to Bark bands.

- When detecting a newly started sinusoidal trajectory with our sinusoidal trajectory heuristics technique, reset the trajectory's initial synthesis phase.

- Constrain each spectral peak's region of influence to the size of the analysis window's spectral footprint.

8 Conclusion

In this paper we have presented three novel techniques to enhance the audio quality of phase vocoder based time-stretching systems. Taking the nonlinear frequency resolution of the human auditory system into account, *multiresolution peak-picking* helps to yield a better bass reproduction and reduces musical overtones in the time-stretched signal. *Sinusoidal trajectory heuristics* reduce warbling in the high frequency bands and improve the quality of note onset reproduction. Phase coherence in the phase vocoder can be re-established at irregular intervals by performing a *silent passage phase reset*. *Multiresolution peak-picking* and *sinusoidal trajectory heuristics* are extensions to the scaled phase-locking algorithm by Laroche and Dolson (1999), whereas the *silent passage phase reset* technique can be included in any phase vocoder. All described techniques have been implemented in *PhaVoRIT*, the time-stretching engine of the

	SPL	MRPP	*PhaVoRIT*	Traktor	Pitch'n Time	MPEX-2	
mean score	5,5573	5,7704	6,4262	6,2622	6,4590	2,3278	
std. deviation	3,0029	2,6101	1,8926	2,5684	2,5597	2,1191	
per-song mean values							comments
Lovesong	1,7	2	7,7	8,2	7	5,1	rapid transients
Smooth Sailing	4,3	4,8	4,8	9	8,1	1,6	sharp transients
Radetzky March	8,4545	8,3636	6,1818	4,7272	4,1818	1,7272	dense arrangement
Narcotic	6	6,1	6,7	6	7,2	1,3	duet vocals
Poison	4,2	5,2	6,6	6,2	7,4	3	noisy characteristic
Kl. Nachtmusik	8,4	7,9	6,6	3,6	5,1	1,3	no transients

Table 2: List of the mean grades for each algorithm. Grades are from 1 (worst) to 10 (best).

Maestro! interactive conducting exhibit. The audio quality of *PhaVoRIT* was assessed in a formal user listening test and has been found to be comparable to some of the best commercially available time-stretching systems.

References

Auger, F. and P. Flandrin (1995, May). Improving the readablility of time-frequency and time-scale representations by the reassignment method. In *IEEE Transactions on Signal Processing*, Volume 43, pp. 1068–89.

Bernsee, S. M. (2005, June). The DSP dimension.

Bonada, J. (2000). Automatic technique in frequency domain for near-lossless time-scale modification of audio. In *Proceedings of International Computer Music Conference*.

Borchers, J., E. Lee, W. Samminger, and M. Mühlhäuser (2004, March). Personal orchestra: A real-time audio/video system for interactive conducting. *ACM Multimedia Systems Journal Special Issue on Multimedia Software Engineering 9*(5), 458–465.

Dolson, M. (1986). The phase vocoder: A tutorial. *Computer Music Journal 1*(4), 14–27.

Duxbury, C., M. Davies, and M. Sandler (2001, December). Separation of transient information in musical audio using multiresolution analysis techniques. In *Proceedings of the COST G-6 Conference on Digital Audio Effects (DAFX-01)*, Limerick, Ireland.

Flanagan, J. L. and R. M. Golden (1966, November). Phase vocoder. *The Bell System Technical Journal 45*, 1493–1509.

Garas, J. and P. C. Sommen (1998). Time/pitch scaling using the constant-Q phase vocoder. In *Proceedings of STW's 1998 workshops CSSP98 and SAFE98*, pp. 173–176.

Hammer, F. (2001). Time-scale modification using the phase vocoder. Master's thesis, Institute for Electronic Music and Acoustics (IEM), Graz University of Music and Dramatic Arts.

Karrer, T. (2005). Phavorit - a phase vocoder for real-time interactive time-stretching. Master's thesis, RWTH Aachen University.

Laroche, J. and M. Dolson (1997). Phase vocoder: About this phasiness business. In *Proceedings of IEEE ASSP Workshop on application of signal processing to audio and acoustics*, New Paltz, NY.

Laroche, J. and M. Dolson (1999, May). Improved phase vocoder time-scale modification of audio. In *IEEE Transactions on Speech and Audio Processing*, Volume 7, pp. 323–332.

Lee, E., T. Karrer, and J. Borchers (2006). Towards a framework for interactive systems to conduct digital audio and video streams. *Computer Music Journal 30*(1). To appear.

Lee, E., H. Kiel, S. Dedenbach, I. Gruell, T. Karrer, M. Wolf, and J. Borchers (2006, April). iSymphony: An adaptive interactive orchestral conducting system for conducting digital audio and video streams. In *Extended Abstracts of CHI 2006 Conference on Human Factors in Computing Systems*, Montréal, Canada. ACM Press.

Lee, E., T. M. Nakra, and J. Borchers (2004, June). You're the conductor: A realistic interactive conducting system for children. In *NIME 2004 International Conference on New Interfaces for Musical Expression*, Hamamatsu, Japan, pp. 68–73.

Levine, S. N. and J. O. Smith III (1998). A sines+transients+noise audio representation for data compression and time/pitch scale modications. In *105th Audio Engineering Society Convention*, San Francisco.

Masri, P. (1996). *Computer Modelling of Sound for Transformation and Synthesis of Musical Signals*. Ph. D. thesis, University of Bristol.

Masri, P. and A. Bateman (1996). Improved modelling of attack transients in music analysis-resynthesis. In *Proceedings of the International Computer Music Conference*.

McAulay, R. J. and T. F. Quatieri (1986, August). Speech analysis/synthesis based on a sinusoidal representation. In *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Volume 34, pp. 744–754.

Puckette, M. (1995). Phase-locked vocoder. In *IEEE ASSP Conference on Applications of Signal Processing to Audio and Acoustics*, Mohonk, New York.

Röbel, A. (2003). Transient detection and preservation in the phase vocoder. In *Proceedings of the Int. Computer Music Conference (ICMC'03)*, Singapore, pp. 247–250.