PVOC KIT: NEW APPLICATIONS OF THE PHASE Vocoder

Tom Erbe
UC San Diego
Department of Music
9500 Gilman Drive 0099
La Jolla, CA 92037
http://www.soundhack.com

ABSTRACT

This paper describes applications of the phase vocoder algorithm as developed for the plugin bundle Pvoc Kit. The plugins in this collection explore pitch shifting, time stretching and phase modulation of live signals. Several new techniques using these algorithms have been developed and will be described.

1. INTRODUCTION AND MOTIVATION

The phase vocoder has been in common use in computer music for the past 30 years. With tools like AudioSculpt [1], pfft-, FFTease[7], and many others available, it is not difficult to access techniques for time stretching, pitch shifting or more esoteric spectral transformations. Most of the tools available run as stand-alone applications, as externals or abstractions for Max/MSP and Pure Data, or as patches under other computer music languages such as Kyma, Supercollider, chuck or JSyn. Consequently there is a very rich tool kit for the exploration of phase vocoder techniques for those working in computer music languages. However, there is much less available to composers, computer musicians or sound designers who use sound-file centered applications such as Ableton Live, ProTools or Logic as their main compositional tool. The motivation for developing Pvoc Kit is to bring the same richness of tools to this environment, and at the same time to expand on the set of known techniques.

Earlier the author developed the Spectral Shapers plugins, a set of phase vocoder based filters which for the most part only modulate the amplitude of each spectral band. This technique is useful for timbre modification, detailed filtering and spectral dynamics. In contrast, the current work explores the phase vocoder’s ability to modulate time, pitch and phase and by extension harmonic content, sustain and ambience. These processes are implemented as a set of four audio plugins in the VST, AU and RTAS formats called +phasemash, +pitchshift, +spirastretch and +loopool. This paper will describe the function of the first three plugins and detail the various sound processing algorithms developed within each one.

2. PVOC KIT: COMMON FEATURES

All of the plugins use the phase vocoder as described by Mark Dolson [2]. The input samples are shifted in and rotated to provide proper phase alignment in preparation for further pitch or time processing [5]. Blackman and Kaiser windows are used on the input and output for their good sideband rejection. The phase vocoder is using an overlap factor of 8 to minimize the error in instantaneous frequency estimation.

All of the plugins have a user control for the number of bands. This is a key control when one works with the phase vocoder, as the tradeoff between frequency resolution and time resolution is a constant concern. This control ranges from 8 to 8192 bands (16 to 16384 points in the STFT), in power of two increments. This covers the sweet spot from 512 to 2048 bands where time resolution and frequency resolution are both pretty good. However, it also goes far beyond, where the phase vocoder is providing extremely smeared time or extremely distorted frequencies. These ranges will not be useful to all composers, but as many enjoy pushing the limits, it seemed appropriate to broaden them. Aside from this, all the parameters have been scaled to provide a useful continuity of transformation, and almost all are internally interpolated.

3. PITCH SHIFTING AND SIFTING

Figure 1. +pitchshift user interface.

The plugin +pitchshift is the classic phase vocoder based pitch shifter with a number of enhancements. Pitch shifting is provided with a large range, 4 octaves up and 4 octaves down. At the far extremes of pitch shifting, the resultant sound bears little resemblance to the original. There are controls for octave, cents, and pitchshift, with pitchshift reading either in semitones or in commonly used just ratios. Internally the plugin has two synthesis engines which can be switched. One is based on a bank of sine wave oscillators, and the other is based on an inverse FFT. In the sine bank oscillator synthesis method, amplitude and frequency are linearly interpolated for each block of samples. With this method, the amount of computation needed naturally goes up as the number of bands is increased, and for this reason the partialgate...
Control is provided. This eliminates all harmonics below the level of the gate, and at extreme settings can reduce any sound to a small number of sine waves. This harmonic gating is a nice transformation alone, or when combined with pitch shifting.

![Figure 2. +pitchshift block diagram.](image)

The inverse FFT synthesis method works by first measuring the instantaneous frequency for each band. The bands are then processed by amplitude order; loudest first. Frequencies are reassigned to the band closest to the shifted instantaneous frequency. If a louder harmonic is already in the band, the amplitudes are simply added, with the loudest harmonic for a given band determining the recalculated phase. This method has a lower fidelity than the interpolated sine bank, but is much more computationally efficient.

Aside from pitch shifting and harmonic gating, this plugin can work as a pitch sieve, similar to Eric Lyon’s pvtuner~ Max/MSP external [3]. When midivocoder is selected, any played MIDI note will create a list of harmonics, the number of which is selected by the midiharmonics control (odd harmonics can be selected by moving this control to the left). Multiple MIDI notes add to the harmonic list. All of the pitch shifted instantaneous frequencies are then compared to the list of harmonics. The frequencies that lie in between the highest and lowest harmonics in the list are then shifted to the nearest harmonic of the MIDI notes being played. Those outside the range are discarded.

This pitch sieve performs a type of harmonic quantization. It sounds similar to a harmonizer when only a few MIDI notes are held down, and becomes more like a harmonic jittering autotune effect when the MIDI harmonic list is denser. This is because the larger number of harmonics in the list allows the frequencies to jump more often to new quantized frequencies.

Setting the phase vocoder to a large number of bands may cause poor time resolution, but this can be an advantage when using the pitch sieve as this phase smearing will produce greater sustained harmonics. Conversely a smaller number of bands is more appropriate for MIDI vocoded percussion as the short block size will more closely follow transients.

4. Live Time Stretching and Layering

![Figure 3. +spiralstretch user interface.](image)

The main motivation for this plugin is to provide real time phase vocoder time stretching on live input signals. The obvious problem is that a time stretch algorithm will create more samples than it consumes. To get around this, the input sound is periodically segmented and fed round robin into several parallel phase vocoders. The resultant stretched sound is layered for a real-time time stretch (see figure 4). In an early version of this effect an envelope follower was used to trigger each new layer. The idea was that each note or phrase could be stretched individually, and the result would have notes with the same starting time, but longer sustain. This effect was somewhat difficult to achieve in legato passages, as notes could not be reliably separated (though the note identification could probably be refined and made workable).

![Figure 4. Segmented input layered stretching.](image)

In the current implementation new phase vocoder time stretch layers are started periodically. This happens at a rate from 0.25 to 4 Hz., corresponding to tempi from 15 to 240 BPM. The plugin can use an external clock to track tempo change. With new layers started on tempo, the relationship between the stretched material and the live material is predictable, and much more usable for live processing or post production.

As there are multiple layers, all of the input material can be used in re-synthesis. For a time stretch of 4, 4 layers are required. If there are not enough layers to cover all of the input material, the input becomes
discontinuous and is only sampled at the beginning of each beat clock (length sampled = voices/stretch x beat length). Each input segment is enveloped with a stretched raised cosine function to minimize glitches. With this method, even large time stretch values can be used, and the characteristic sound of the phase vocoder time stretch can be applied to live input signals.

The +spiralstretch plugin also implements a granular time stretch algorithm, which can be used as an alternate to the phase vocoder method, but this is not within the scope of this article.

5. PHASE MODULATION TRICKS

The plugin +phasemash is a collection of simple transformations to phase difference and band assignment. Most of these filters in this plugin have been developed over the years as class demonstrations on the effect of phase modulation in the phase vocoder. None would be difficult to recreate in Max or PD. The number of bands has a dramatic difference on all of the +phasemash processes. Large number of bands narrow the bandwidth and also increase the number of samples processed in each block, which increases the possible effect on phase.

Band shifting is the simple reassignment of phase and amplitude from one band to another. An increment is added to each band number with bands wrapping around the top and bottom. This provides a positive or negative frequency shift that is somewhat complicated by the wrapped bands. Phase shifting adds a static offset from $-\pi$ to $\pi$ to the phase change between blocks. This also gives a linear frequency shift, but with phase cancellation resulting when adding phase offsets near to $\pi$ or $-\pi$. Both of these methods are nice alternatives to the typical single sideband ring modulation frequency shifting.

Phase nulling decreases the phase change between blocks, and when at 100% completely removes all phase change information, locking all harmonics to the lowest band of the phase vocoder (N x bands/Nyquist rate). This results in a typically harmonized or severely comb filtered sound.

The phase noise control adds a random number to the phase change, in effect treating the entire sound as a residual signal. This effect is very effective at either removing pitch from a sound at a low band setting, or adding a scrambled ambiance when the number of bands is large. As phase is cumulative, a phase reset button is provided to reset the phase to the source signal.

All of these transformations are mathematically simple, and have little to do with natural acoustics, but together in one interface give a wide variety of frequency and phase distortion effects.

6. FUTURE DIRECTIONS

The plugins in the P vocoder Kit contain some novel and musically useful applications of the phase vocoder algorithm. However, there are obvious directions for further development of other new processes.

In pitch shifting more could be done with other types of pitch sieves. Also, a compositional algorithm could intelligently control the sifting process. Finally, a cross-synthesis technique could be developed which analyzes two live signals and uses one sound’s harmonics as a pitch sieve for the other.

In time stretching, as mentioned earlier, having new notes initiate new layers or some other performer control of the layer initiation (as simple as a pedal or switch) could be very useful. In addition, as one of the main uses of this layered time stretching will probably be to create large masses of ambiance, a feedback or regeneration path would add greatly to this [9].

Finally, as this work was focused on developing new applications of the phase vocoder, not as much attention was given to the underlying phase vocoder engine. Enhancements such as phase-locking [4][8] and multiresolution peak-picking [3] would make obvious improvements to the overall quality of sound.

7. REFERENCES


