soundhack

pvoc kit

www.soundhack.com

contents

license agreement

License Grant

"You" means the person or company who is being licensed to use the Software or Documentation. "We," "us" and "our" means Tom Erbe, d.b.a. *SoundHack*. “Software” means *SoundHack Pvoc Kit* for Windows or for Macintosh. We hereby grant you a nonexclusive license to use one copy of the Software on any single computer, provided the Software is in use on only one computer at any time. The Software is "in use" on a computer when it is loaded into temporary memory (RAM) or installed into the permanent memory of a computer—for example, a hard disk, CD-ROM or other storage device.

If the Software is permanently installed on the hard disk or other storage device of a computer (other than a network server) and one person uses that computer more than 80% of the time, then that person may also use the Software on a portable or home computer.

Title We remain the owner of all right, title and interest in the Software and related explanatory written materials ("Documentation").

Archival or Backup Copies You may copy the Software for back up and archival purposes, provided that the original and each copy is kept in your possession and that your installation and use of the Software does not exceed that allowed in the "License Grant" section above.

Things You May Not Do

The Software and Documentation are protected by United States copyright laws and international treaties. You must treat the Software and Documentation like any other copyrighted material—for example, a book. You may not:

- copy the Documentation,
- copy the Software except to make archival or backup copies as provided above,
- modify or adapt the Software,
- reverse engineer, disassemble, decompile or make any attempt to discover the source code of the Software,
- place the Software onto a server so that it is accessible via a public network such as the Internet, or
- sublicense, rent, lease or lend any portion of the Software or Documentation.

Limited Warranty

We warrant that for a period of 90 days after delivery of this copy of the Software to you:

- the media on which this copy of the Software is provided to you will be free from defects in materials and workmanship under normal use, and
- the Software will perform in substantial accordance with the Documentation. To the extent permitted by applicable law, THE FOREGOING LIMITED WARRANTY IS IN LIEU OF ALL OTHER WARRANTIES OR CONDITIONS, EXPRESS OR IMPLIED, AND WE DISCLAIM ANY AND ALL IMPLIED WARRANTIES OR CONDITIONS, INCLUDING ANY IMPLIED WARRANTY OF TITLE, NONINFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE, regardless of whether we know or had reason to know of your particular needs. No employee, agent, dealer or distributor of ours is authorized to modify this limited warranty, nor to make any additional warranties. SOME STATES DO NOT ALLOW THE LIMITATION OR EXCLUSION OF LIABILITY FOR INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THE ABOVE LIMITATION MAY NOT APPLY TO YOU.

Limited Remedy

Our entire liability and your exclusive remedy for breach of the foregoing warranty shall be, at our option, to either:

- return the price you paid, or
- repair or replace the Software or media that does not meet the foregoing warranty if it is returned to us with a copy of your receipt.

IN NO EVENT WILL WE BE LIABLE TO YOU FOR ANY DAMAGES, INCLUDING ANY LOST PROFITS, LOST SAVINGS, OR OTHER INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING FROM THE USE OR THE INABILITY TO USE THE SOFTWARE (EVEN IF WE OR AN AUTHORIZED DEALER OR DISTRIBUTOR HAS BEEN ADVISED OF THE POSSIBILITY OF THESE DAMAGES), OR FOR ANY CLAIM BY ANY OTHER PARTY.

SOME STATES DO NOT ALLOW THE LIMITATION OR EXCLUSION OF LIABILITY FOR INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THE ABOVE LIMITATION MAY NOT APPLY TO YOU.

Term and Termination

This license agreement takes effect upon your use of the software and remains effective until terminated. You may terminate it at any time by destroying all copies of the Software and Documentation in your possession. It will also automatically terminate if you fail to comply with any term or condition of this license agreement. You agree on termination of this license to destroy all copies of the Software and Documentation in your possession.

Confidentiality

The Software contains trade secrets and proprietary know-how that belong to us and it is being made available to you in strict confidence. ANY USE OR DISCLOSURE OF THE SOFTWARE, OR OF ITS ALGORITHMS, PROTOCOLS OR INTERFACES, OTHER THAN IN STRICT ACCORDANCE WITH THIS LICENSE AGREEMENT, MAY BE ACTIONABLE AS A VIOLATION OF OUR TRADE SECRET RIGHTS.

General Provisions

1. This written license agreement is the exclusive agreement between you and us concerning the Software and Documentation and supersedes any prior purchase order, communication, advertising or representation concerning the Software.
2. This license agreement may be modified only by a writing signed by you and us.
3. In the event of litigation between you and us concerning the Software or Documentation, the prevailing party in the litigation will be entitled to recover attorney fees and expenses from the other party.
4. This license agreement is governed by the laws of the State of California.
5. You agree that the Software will not be shipped, transferred or exported into any country or used in any manner prohibited by the United States Export Administration Act or any other export laws, restrictions or regulations.

overview

SoundHack, a program that I wrote in 1991, has been used by many composers and sound designers for music, theatre and film. It is a collection of spectral and other filters which emphasize the creative manipulation of sound and techniques that are somewhat experimental or non-commercial. **SoundHack** has been used on many films – from “The Matrix” to “The City of Lost Children” – and by many recording artists – from Nine Inch Nails to Ry Cooder to J Lesser.

Since 2000, I have been releasing my sound processing software in the form of a series of plugins for VST, AU and RTAS. At that point **SoundHack** became the company name for these plugins. I have been releasing the plugins in bundles, some free and some commercial - all available from http://www.soundhack.com. They are:

SoundHack Spectral Shapers (2003). The bundle includes **+morphfilter**, **+spectralgate**, **+spectralcompand** and **+binaural**. These are all filters which emphasize the reshaping of the timbre of sound.

SoundHack Freesound (2006). The bundle includes **+chebyshev**, **+compand**, **+decimate**, **+matrix** and **+phasescope**. This is a free collection of miscellaneous studio and distortion filters.

SoundHack Delay Trio (2008). The bundle includes **+delay**, **+pitchdelay** and **+bubbler**. I created this free collection to broaden the palette of delay based effects. Dual head, pitch-shifting, granular, lo-fi and conventional delay techniques are covered.

SoundHack Pvoc Kit (2011). This bundle includes **+phasemash**, **+pitchsift**, **+spiralstretch** and **+pvocloop**. These plugins use the phase vocoder and granular techniques to stretch time, shift pitch and distort phase.

The emphasis of all of these plugins is cutting-edge computer music and classic experimental electronic music techniques. I always try to include a full spectrum of quality of sound from smooth, detailed and beautiful to distorted chaotic and aggressive. Future releases will include a re-focus on cross synthesis and synthesis using convolution, filterbank and modal techniques.

Tom Erbe
SoundHack

installation

SoundHack Pvoc Kit is fairly simple to install. After downloading the plugins from http://www.soundhack.com, unpack the archive and double-click the installer. Follow all of the default selections for the installer.

On OS X, the installer will place the plugins in the standard locations for VST, Audio Units and Digidesign RTAS. These folders are:

```
/Library/Audio/Plug-Ins/VST
/Library/Audio/Plug-Ins/Components
/Library/Application\ Support/Digidesign/Plug-Ins
```

If successfully installed, the latest versions of these plugins will be in these folders.

On Windows, the installer will place the VST plugins in:

```
C:\Program Files\SoundHack\Pvoc Kit\
```

You may instruct your host program to look in this directory for the plugins, or copy the plugins to another directory if it is more convenient for the host.

On both OS X and Windows, all of the plugins will run for 2 or 3 minutes in demo mode prior to being registered. This allows for full functionality of the plugins for evaluation purposes. This demo mode will be indicated in the title bar of each plugin.

When you purchase the SoundHack Pvoc Kit, you will receive an unlock code and unlock program within 24 hours. This unlock program will allow you to permanently register the plugins on your computer. The terms of the license allows you to register the plugins on two of your computers as long as they are not being used at the same time.

If you have any trouble with installation or registration, please contact us at support@soundhack.com

computer requirements

SoundHack Pvoc Kit contains plugins that are very CPU intensive. Most of the plugins allow you to vary the quality, number of bands or number of voices, which will take up more or less of the CPU. Nevertheless, they will run best on a fast processor with lots of memory. Increasing the latency or blocksize is required to run the plugins with more CPU intensive settings.

On OS X, the VST and AU plugins require at least OS X version 10.4 or later. The RTAS plugins require ProTools 8 or later. They will run on either PPC or Intel processors.

On Windows, XP or newer is required. The installer will require the appropriate version of .NET to run the plugins.

SoundHack Pvoc Kit has been tested on the following host programs:

OS X

Ableton Live
Apple Logic
Apple GarageBand
Avid ProTools
BIAS Peak
cockos Reaper
Cycling74 Max/MSP
Plogue Bidule
Renoise
Steinberg Nuendo

Windows

Ableton Live
cockos Reaper
Cycling74 Max/MSP

pvoc kit - common features

All of the plugins in **SoundHack Pvoc Kit** are based on the phase vocoder and the granular synthesis algorithms. Because of this, they all share a common characteristic of sound quality. The phase vocoder operates by sampling the incoming sound block by block, and by dividing each block into many frequency bands. The number of frequency bands and number of samples in each block are tied together, so that if one wants high frequency resolution, the block size needs to be increased and the time resolution is decreased. Because of this you will find that there is a constant trade-off between frequency response and good transient (time) response in these plugins. The granular synthesis algorithm also divides the sound into small segments (grains), but does not divide the spectrum into individual frequency bands.

All of the plugins have a **bands** control to set the frequency resolution and transient response when using the phase vocoder algorithm. You will find a sweet spot around 1024 to 2048 bands (at 44100 sample rate), which gives you a frequency response between 10 -

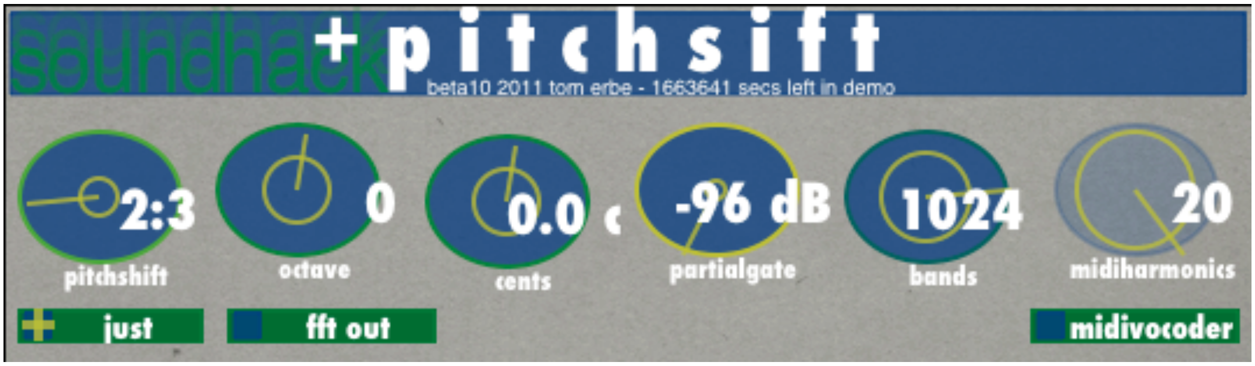
20 Hz per band, and a transient response between 0.05 and 0.10 seconds. This resolution insures that most bands will contain no more than one harmonic and that transient response is still fairly good. For better transient response you may want to drop down to 512 bands. At higher sample rates, you will want to use a correspondingly higher band count.

Setting the **bands** control above 2048 will give you even finer pitch resolution, but will start smearing the transients very noticeably. This can be very useful when making backgrounds, drones or ambient washes. Setting **bands** below 512 will allow multiple harmonics to enter each band and will typically cause frequency distortions and cross modulations when the sound is processed. I included both of these ranges for their creative uses.

For those interested: Internally the phase vocoder is running at an 8 time overlap with a Blackman window. Also: Adjusting the **bands** control changes the latency, but since hosts do not yet support a plugin with variable latency, 1024 samples is always reported.

Finally, these plugins are very CPU hungry. Running at a high number of bands will not be possible on all computers.

+pitchsift - pitch shifting and vocoding



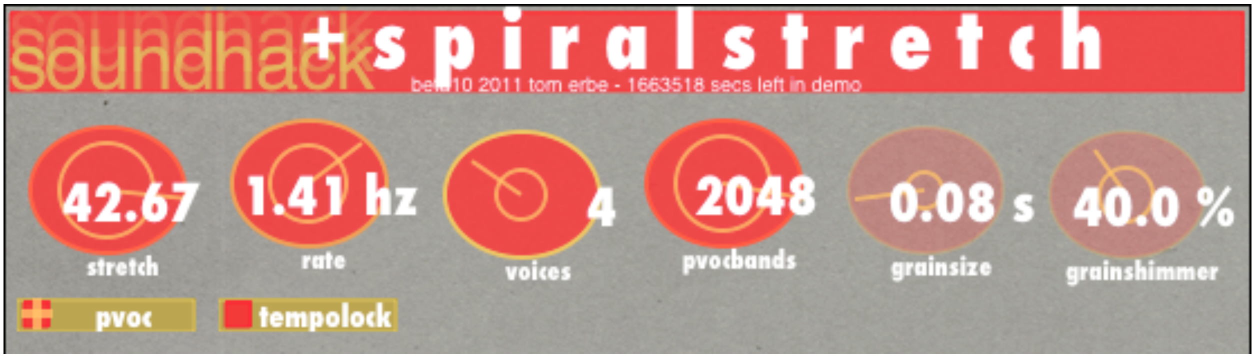
The plugin **+pitchsift** is the classic phase vocoder based pitch shifter with a number of enhancements. Pitch shifting is provided with a large range, 4 octaves up and 4 octaves down. At the far extremes of pitch shifting, the resultant sound bears little resemblance to the original. The highest quality pitch shifting comes when shifting no more than an octave, and with 1024 or 2048 bands.

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 (**fft out**/**sinebank out**). One is based on a bank of sine wave oscillators, and the other is based on an inverse FFT. These methods have a slightly different sound quality and greatly different CPU load. 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.

Aside from pitch shifting and harmonic gating, this plugin can work as a type of vocoder or pitch sieve. 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 input 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.

+spiralstretch - layered real time time stretching



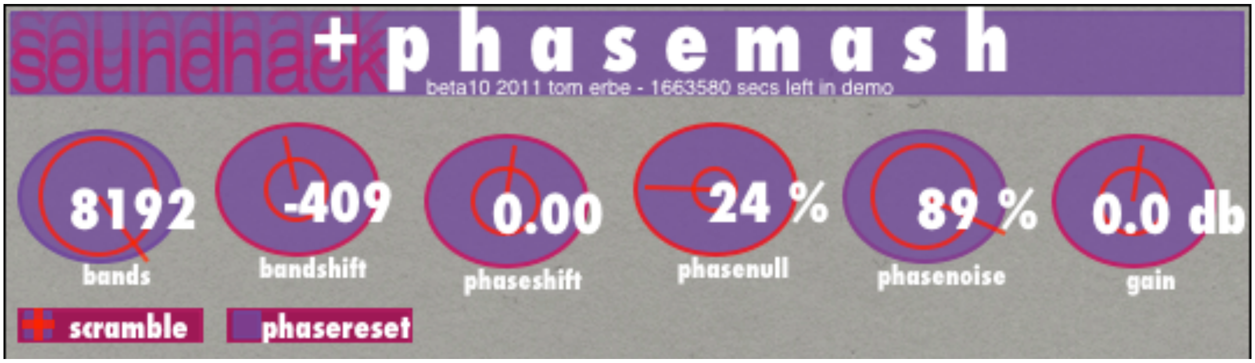
42.67 stretch
1.41 hz rate
4 voices
2048 pvocbands
0.08 s grainsize
40.0 % grainshimmer

The plugin **+spiralstretch** is a multilayered time stretcher which processes real time signals (in contrast to the typical offline time stretcher). It is most useful in a processing sidechain, providing delayed ambience, drones or texture. **+spiralstretch** works by dividing the input into segments, and layering the time stretched segments into the output. Because of this segmenting and layering, the time stretched material always stays relatively close to the original material.

The stretch value is set by the **stretch** control, and ranges between 1.0x and 100.0x. The **voices** control sets the number of layers, or voices. It should be noted that the CPU usage goes up as the number of voices are increased. The **rate** control sets the speed of input segmentation, with each new input segment assigned round-robin to one of the voices. This control ranges between 0.25 Hz and 4.0 Hz which corresponds to a tempo from 15 to 240 BPM. If **tempolock** is selected, the segmentation rate will lock to the host's tempo, segmenting and stretching each beat separately.

+spiralstretch contains two stretch algorithms, one based on the phase vocoder and the other based on granular synthesis. The algorithm is selected with the **pvoc/granular** button. The **pvocbands** knob controls the number of bands used in the phase vocoder algorithm with best results at around 1024 - 2048 bands, time smeared sounds at values above that, and distorted sounds at values below that. The **grainsize** and **grainshimmer** controls allow one to fine tune the granular synthesis algorithm. The **grainshimmer** control controls the random distribution of each grain. At 0% the grains become synchronous and can become resonant. The **grainsize** control obviously controls the size of each grain, with stuttering sounds available at the larger grain sizes.

+phasemash - phase modulation tricks



8192 bands
-409 bandshift
0.00 phaseshift
24 % phasenull
89 % phasenoise
0.0 db gain

The plugin **+phasemash** is a collection of simple transformations to phase difference and band assignment. 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.

The number of bands makes 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.

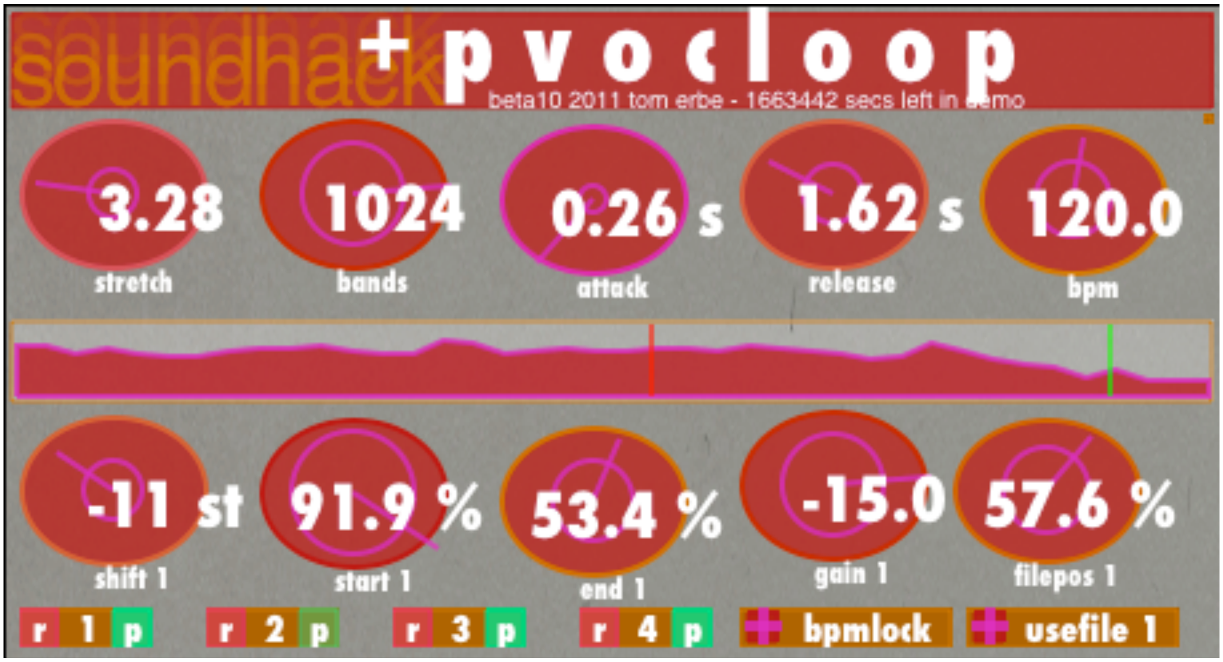
The **bandshift** control adjusts the reassignment of phase and amplitude from one band to another. An increment is added to each band number with bands wrapping around at the top and bottom. This provides a positive or negative frequency shift that is somewhat complicated by the wrapped bands. The **phaseshift** control adds a static offset from -π to π to the phase change between blocks. This also gives a linear frequency shift, but with phase cancellation affecting steady tones. Both of these methods are nice alternatives to the typical single sideband ring modulation frequency shifting.

The **phasenull** control 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. This results in a typically harmonized or severely comb filtered sound (the robot voice effect).

The **phasenoise** control adds a random number to the phase change. 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 **phasereset** button is provided to reset the phase to the source signal.

The **scramble** button randomly reassigns bands, and does so in a non-repeatable way.

+pvocloop - a phase vocoding looper



stretch 3.28
bands 1024
attack 0.26 s
release 1.62 s
bpm 120.0
shift 1 -11 st
start 1 91.9 %
end 1 53.4 %
gain 1 -15.0
filepos 1 57.6 %

The plugin **+pvocloop** combines the time stretching and pitch shifting into a four-voice looper. Four samples with a maximum length of 11.8 seconds at 44.1k or 5.4 seconds at 96k can be loaded into this looper either by sampling the input or by loading WAVE or AIFF soundfiles. Each voice can be controlled independently by either MIDI control or the parameter knobs. There is a small waveform display with loop start and end points that shows any of the four samples.

Controls above the waveform display control all four voices, controls below the display control each voice individually. To select which voice is controlled, click on the voice number button on the bottom row between the **r** and **p** (record and play) buttons. You will see the name of each knob in the bottom row change to the indicate the selected voice.

The **stretch** control affects all voices, and changes the time stretch value from 1.0 to 100.0. The time it takes to go from the loop start to the loop end is automatically quantized to the nearest beat depending on the BPM. If the BPM changes during the performance, the **stretch** value is changed along with it.

Attack and **release** control the volume envelopes for each voice. Both controls are exponential and can be varied from 0.01 to 5.00 seconds. The **bpm** control allows one to set the tempo. There is a pixel LED in the upper right corner which flashes with the beat. When **bpmlock** is selected, the **bpm** control locks to the host application’s tempo.

To load sound into each voice, you either need to sample the input, or load a soundfile. To sample the input simply click the **r** button for the voice. Once the MIDI clock reaches beat one, it will start recording and will stop recording on the beat before the voice fills.

To load a soundfile, click the **usefile** button. You will have to click a voice button to select a specific voice before loading. **+pvocloop** can open WAVE (.wav), AIFF (.aiff/.aif) and AIFF-C (.aifc/.aic) files in 16, 24 and 32 bit, mono and stereo. If successful the button will remain on, and the waveform will be displayed. When a soundfile is loaded, you can scrub through the soundfile with the **filepos** control. Because the phase vocoder is continually smoothing the audio, this scrubbing can provide some interesting effects. When sessions are saved, **+pvocloop** retains the name of the loaded soundfiles. If the files aren’t moved, it will load them again when the patch is opened.

Once sound is loaded into **+pvocloop**, the **p** (play) buttons can be clicked to start and stop each voice. The attack and release envelope is applied with release starting when the **p** is turned off. Incoming MIDI notes will also control playback of each voice with MIDI channel 1 mapping to voice 1, channel 2 to voice 2, etc. The pitch of the MIDI note will control pitchshifting, with pitch 60 (middle C) causing no pitchshift, pitch 72 shifting one octave up, and pitch 48 one octave down.

The rest of the controls are fairly obvious. **Shift** controls the pitchshifting of the specified voice. **Start** and **end** control the start and end looping points in the sample buffer. Note that the **end** can be placed before the **start** of the loop for reverse looping. The **start** and **end** values will be quantized to the nearest beat. **Gain** controls the gain of the voice.

MIDI CC Mapping

The VST and AU versions of the plugins have fixed MIDI CC assignments. They are as follows:

+phasemash

CC 1	**phase shift**
CC 7	**gain**
CC 12	**band shift**
CC 13	**bands**
CC 16	**phase null**
CC 17	**phase noise**
CC 64	**scramble**
CC 69	**phase reset**

+pitchsift

CC 1	pitch
CC 12	octave
CC 13	bands
CC 16	cents
CC 17	threshold

+spiralstretch

CC 1	stretch
CC 12	rate
CC 13	bands
CC 16	voices
CC 17	grain size
CC 18	grain shimmer

+pvocloop

CC 1	stretch
CC 7	gain (MIDI channel mapped to voice)
CC 70	bands
CC 72	release
CC 73	attack