resyn vst

spectral resynthesis by xoxos

Specification of time, pitch and spectrum/formants
Cross synthesis between files
Phase smearing for whisper and ensemble effects
Stereo and high quality modes

Resynthesis works better with samples shorter than 32 * window length samples
eg. 11 window = 2048 samples * 32 = 65536 samples = 1.48s at 44100Hz

parameters

shift

Continuous adjustment of spectral contour, spectral or "formant shift". See spectrum parameter.

read

Position of read pointer in file.

smear

Adds random value to phase increment between frames, creating varied effects depending on window length.

file

Loads wav file for analysis. Second button loads wav file for cross synthesis.

lifter

Specifies separation of spectral contour data. Higher values increase resolution of formants, but pitch colouration occurs above a certain point. Leave it about a 1/4 way up if you don't know.

read2

Position of read pointer in cross synthesis file.

mix

Applies spectrum of cross synthesis file to primary file.

spectrum

off	bypasses spectral shift calculation for efficiency and precision
fix	retains constant spectral contour when repitched
on	spectral shift active, no correction for pitch
fix cross	fix + cross synthesis of spectral contour
on cross	on + cross synthesis of spectral contour

mode

mono	standard phase vocoder implementation with 4 overlapped frames for rendering
hifi	mono, 8 overlapped frames
stereo	stereo processing of stereo files at 4 overlaps

Cross synthesis is not available in stereo mode (they use the same buffer)

Hifi mode improves quality with long window lengths. It is clearly audible at 12 but only drops the occasional sideband at 11. I haven't observed any difference in quality for lower window lengths.

window

Applies window length of 8-12 ^ 2 (eg. 256 to 4096). Longer windows provide better frequency resolution but less temporal resolution during resynthesis. I generally use 11 (2048 samples) which, with 4x overlapping, renders a new frame every 512 samples, or updates about 86 times a second at 44100 host rate.

Frequency resolution of the fourier transform is determined by samplerate divided by window length (eg. 44100/2048 = 21.533 Hz).

Phase vocoders analyse the phase increment between successive fft analysis frames to estimate the pitch of each band, or bin. Resyn uses 128 frames. Less distance between frames improves accuracy. Bin aliasing, or sidebands, occur when frequencies are between central frequencies of the analysis window bins. Classic phase vocoders are known for sounding reverby.

The other fundamental issue with the phase vocoder is that windowed resynthesis modulates the signal, producing sidebands equally as loud as the signal. 4x overlapping cancels them to a level canonically considered to be tolerable, I think around -24 or -30dB. The current build of this vst is limited to forms that run in realtime on my single core.

practical information - analysis

Operations requiring file analysis or reading take time. Analysis occurs when lifter, mode, or window are changed. Loading wave files can take several seconds and will occur when changing patches. If you are not using cross synthesis, make sure to remove any filename.

practical implication - playback rate

All playback is performed with modulation of the read parameter. Linear playback speed is achieved with a linear envelope or lfo (contour 0). There is no function to automatically recall the original playback speed.

technical notes

This is a "classic phase vocoder" implementation, eg. there are no algorithms to align phase by peaks or other improvements apart from choice of windowing method (exact blackman for resynthesis, a half windowing scheme for analysis).

Reconstruction of spectra from cepstral decomposition worked amazingly well. In this build, "spectrum off" uses the original magnitude, and any spectral shift operations of course use the reconstructed signal. Try switching back and forth while viewing a spectral analyser. Even low resolution liftering seems to fit very well.

Phase vocoder implementation is well documented but can take some observation to become sensible. Cepstral derivation of the spectral contour is accomplished by a fourier transform of the log10 spectral magnitude, then separating lower bins ('filter') from upper ('source'). Further transforms are then used to take the data back into spectral and then temporal space. The spectrum of the source must be "whitened" (spectrum flattened) so it can be multiplied/convolved with the filter (or another filter). All development and coding took place over two weekends, often with a sandwich in one hand. Once I had the basic phase vocoder plus cepstral process, it was trivial to add more buffers for 8x overlap and stereo. Once that was done, it was again trivial to add cross synthesis, being one more file read, and a crossfade to magnitude values. This powerful resynthesis method only takes a few hundred lines in straightforward, C style code. I stopped there, before eg. separate formant shifting for the cross synthesis, because my prefab environment needs more fuss with extra gui parameters and might have run into a third weekend.

Filters adapted from source by Robin Schmidt (SV forms) and Neotec.