Phase-bashed packet synthesis: a musical test

Miller Puckette *

Abstract

A classic piece of computer music by Charles Dodge is studied by making an approximate regeneration using the phase-bashed packet synthesis technique. The result is available as part of the Pd Repertory Project.

1 Introduction

Many researchers have proposed techniques for synthesizing vocal and other musical timbres by assembling a series of wave packets, laid out in time with one packet per period (Templaars 1977; Rodet 1984). If the packets in question are obtained by windowing a pre-existing recording, their phases must be aligned and careful attention must be paid to issues of windowing and overlap (Puckette 2005). The resulting synthesis technique, which could be called 'phase-bashed packet synthesis', is now a fully developed musical resource.

Before applying this technique to the composition of new music, however, an important step in validating it and making it workable has been to test it against an existing piece of music. This is in some ways a more revealing test of a synthesis technique than putting it to a new piece of music would be, since pre-existing music can't be automatically adjusted to take advantage of the best qualities of the synthesis technique, or to avoid its worst limitations.

An ideal piece to use as a benchmark of vocal synthesis is Charles Dodge's *Speech Songs* (1972). *Speech Songs* maintains a clarity of musical exposition which draws keen attention to the vocal sounds used, which are mostly presented in a single voice without accompaniment—the most exposed situation possible. The vocal sounds are subjected to a variety of transformations, ranging from fairly natural to extremely unnatural. These include separate presentation of vocal fragments; separately controlled changes of speed and pitch; exaggeration of sonic features; and denaturing of formant structure. The next section will describe the current status of the technique, and following ones will describe its use to resynthesize the first of Dodge's four Songs.

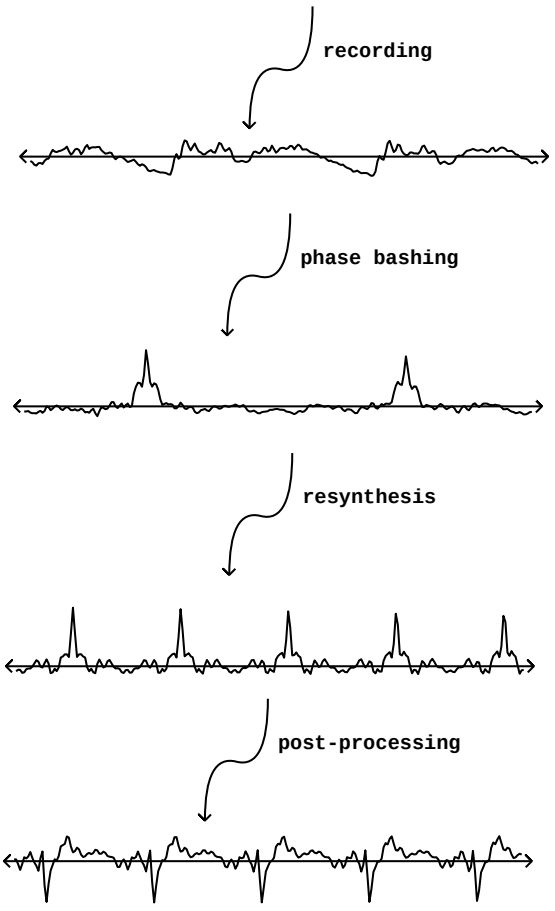

Figure 1: Phase bashed packet synthesis as a process with four steps: recording, phase bashing, resynthesis, and post-processing.

*CRCA, $\mathrm{Cal(it)}^2$, UCSD. Reprinted from *Proceedings*, ICMC 2006.

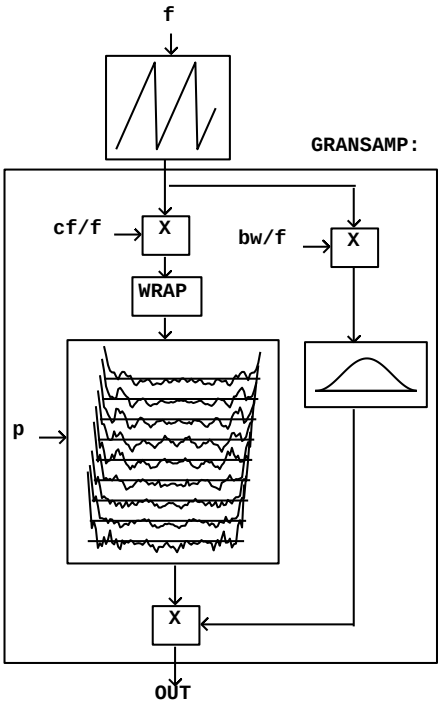

Figure 2: Granular sampling in the resynthesis step.

2 The synthesis technique and its implementation

Phase-bashed packet synthesis is a four-step process (Figure 1. First, a natural sound is recorded. Next, and still prior to synthesis, the recording must be 'phase bashed' to yield a series of single waveforms, laid out end to end. In the third step ('resynthesis'), the packets are arranged at a desired period to synthesize a pitched sound. Finally, post-processing can effect frequency shifts and/or modulate the signal to make it unvoiced.

The preliminary steps, recording and phase bashing, are as described earlier (Puckette 2005). The result is a concatenation of wavetables (which may be stored as a soundfile between sessions), consisting of phase-aligned packets laid end to end. Typical packet sizes range from 512 to 2048 samples, with the usual tradeoff between time and frequency resolutions.

The preparation of the phase-bashed wavetable and its use in synthesis may be done simultaneously, in which case the technique essentially acts as a real-time transformation with a delay of twice the analysis period. Alternatively, the phase-bashed wavetables may be prepared ahead of time, as is done in the example below.

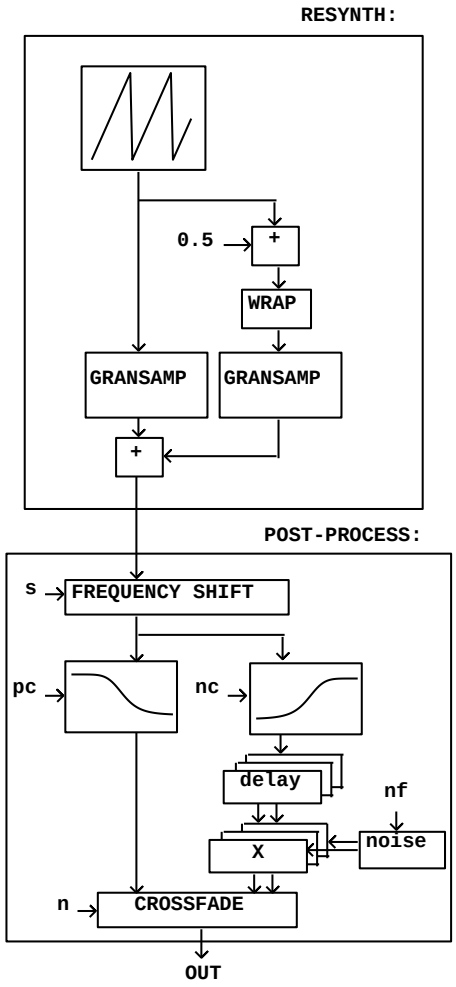

Figure 3: Overlap-add arrangement of two GRANSAMP modules, followed by the post-processing step.

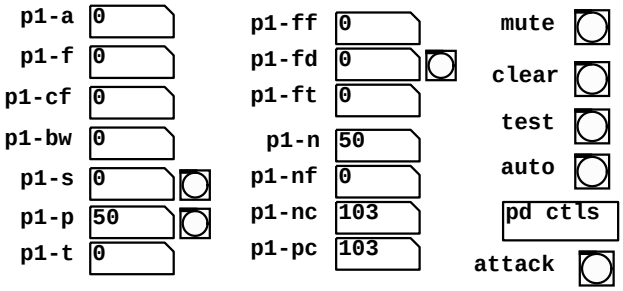

Figure 4: Control panel for one voice of phase-bashed packet synthesis.

The resynthesis step relies on a granular sampler shown in Figure 2. At its heart is a two-dimensional table lookup. Rows of the table are successive phase-bashed windows of the sound that was analyzed. The signal coming from above the table indexes the table from left to right (thereby scanning within a window) and the signal coming from the left side indexes it from top to bottom (thereby choosing which window, from beginning to end of the analyzed sound, to play from). Top-to-bottom interpolation is linear (so that if we sit between two analysis windows we get a linear mixture of the two) and left-to-right interpolation is four-point (the minimum order that is practically usable for resampling audio).

There are four parameters to specify. The fundamental frequency f of the driving sawtooth oscillator controls the periodicity of the output (the final output is heard at twice this frequency since there will be two overlapped copies of the granular sampler).

The "center frequency" cf could be more generally described as a spectral shift (it becomes the true center frequency of the single formant generated if the table contains a pure sinusoid). The cf parameter essentially controls the factor at which phase-bashed windows of the analyzed sound are squeezed in time as they are scanned. The cf parameter is divided by f to specify the precession per sample independently of fundamental frequency.

The "bandwidth" bw has the function of squeezing the window shape. At its minimum the raised cosine window fills an entire period of the driving oscillator. As bw is increased, the raised cosine window is progressively squeezed. The bw parameter is also normalized by dividing by f; the quotient must be at least one so that the raised cosine window does not take more than the single period.

The "position" (the p parameter) controls the onset within the analysis file. To play through the analysis file, p is ramped from minimum to maximum over the time duration of the original sample. Alternatively the evolution may be stretched or contracted in time, frozen, or run backward.

Figure 3 shows how two copies of the granular sampler are combined, one half cycle out of phase, to complete the synthesis step of the technique. Also shown here is the post-processing step, which has two functions. First, the signal is optionally frequency shifted. The frequency shift (parameter name s) is specified relative to the fundamental frequency, so that, for example, a frequency shift of -0.5 shifts down an octave and gives only odd harmonics. Other values of s can make inharmonic sounds.

Next, the signal is optionally de-pitched. This is accomplished through the network of delays and multiplications by noise; the process is called "shaking". The shaker noise has a controllable cutoff frequency (the nf parameter) to control the degree of destabilization of the sound. Three other pa-

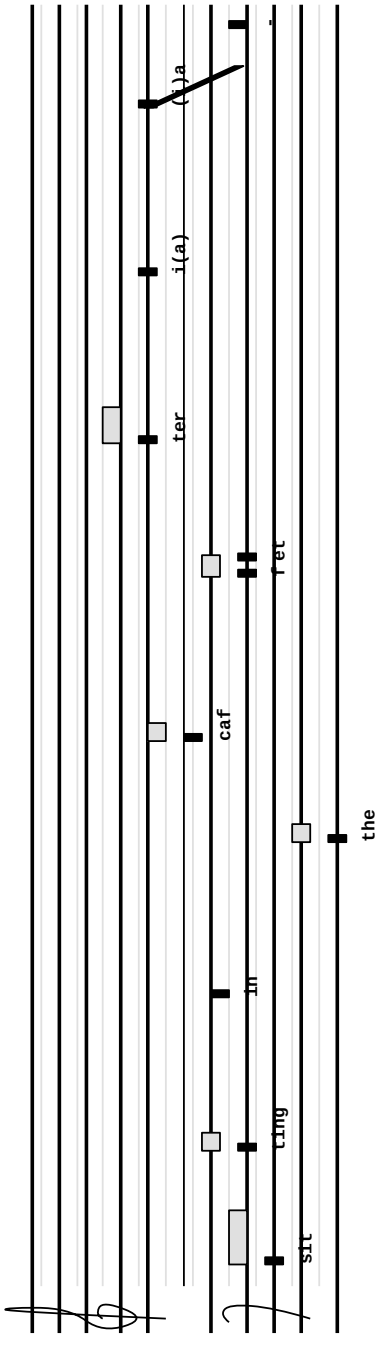

Figure 5: The phrase "sitting in the cafeteria" set as a graphical score in Pd.

rameters control the region and amount of de-pitching. First, the signal is separately low-pass filtered (the "pitched cutoff" frequency is controlled by the pc parameter) and high-pass filtered for sending to the shaker (the high-pass filter cutoff is set by the nc parameter). The pitched and "shaken" signals are then mixed, with the balance controlled by the n parameter. If the two cutoff frequencies are set equal and the pitched and shaken signals are balanced equally, the effect is to render anything above the cutoff frequency as noisy and everything below it as pitched. If, on the other hand, both filters are set to pass the entire signal, then the balance control becomes a pitched/noisy fader. (This second choice is taken for the reconstruction of the *Speech Song* described below).

Figure four shows a control panel in Pd for controlling one voice of synthesis. The parameters a, f, cf, bw, p, and s control amplitude, frequency, center frequency, bandwidth, position, and shift. The four controls for the shaker stage (n, nc, pc, nf) are as described above. There is also a vibrato unit; the parameters f, fd, ft control vibrato frequency, depth and function table. Buttons at right set to a known test configuration, mute the voice, or set the position parameter p automatically on a trajectory from beginning to end of the soundfile for testing. The "attack" button causes all the phases to be reset to zero, useful for deterministically shaping an attack.

3 Realizing the musical example

Charles Dodge's first *Speech Song* is acoustically transparent enough that none of the original materials were needed in order to approximately reconstruct it. Dodge recorded his own voice for analysis (Dodge 1989), but rather than ask him to recite the poem anew I used my own voice. (Dodge mentioned the possibility of hearing his own regional accent in the original piece; mine is audible in my reconstruction.) The 'notes' of the piece have steady pitch with one exception (a glissando toward the middle of the song) and the pitches all belong to the tempered Western scale with one apparently unintentional deviation. The pitches and times of all the notes were easily read off using the fiddle~ pitch tracker.

Next, the spoken text of the poem was separated into 36 segments. Many segments comprised a single syllable of the text, but in a few places a syllable was separated into more than one segment; for instance, in the last word, "fake", the consonant "k" was segmented separately in order to control its timing exactly. Each segment was given a mnemonic name such as "sit".

To create the computer score, each sonic event (syllable or shorter) was entered as a separate line in a text file, and in the Pd patch a sequencer was built around the "textfile" object to play the events. Each event had only six parameters: time, pitch, mnemonic, noise duration, and two parameters for a possible glissando. For purposes of visualization the data was read into a graphical editor using Pd data structures. An extract is shown in Figure **??**. The small black rectangles mark the beginning of events and give their pitch and mnemonic, and the gray rectangles indicate noisy portions of the events. The slanted segment is a glissando.

4 How it sounds

The first three Speech Songs used a formant tracking technique by J.P. Olive based on the Newton-Raphson technique (Dodge 1989). The results sound primitive by today's standards. As also happens with the "more advanced" LPC technique, the voice sounds as if the 'singer' has a bad cold. In one spot the resynthesis of the word "one" sounds like the nonsense syllable "lun". These defects were used quite deliberately by Dodge and are part of the fabric of the songs.

The recreation of the first song presented here (which can be heard by downloading the Pd Repertory Project (Puckette 2001) and starting the "dodge-song" patch) can not be considered the same piece of music; it is for study purposes only. That it sounds more "modern" is not necessarily an advantage; and indeed, comparing the two, one can easily imagine that, in another three decades, many of the sounds of today's computer music will sound equally antique. However, certain early computer music pieces—among which Dodge's *Speech Songs* should be included—rise above the primitiveness of the techniques available at the time to make musical statements that have not become dated as they age. How those pieces achieve that is a deep mystery.

References

Dodge, C. (1989). On speech songs. In M. V. Mathews and J. R. Pierce (Eds.), *Current Directions in Computer Music Research*, pp. 9–17. Cambridge: MIT Press.

Puckette, M. S. (2001). New public-domain realizations of standard pieces for instruments and live electronics. In *Proceedings of the International Computer Music Conference*, Ann Arbor, pp. 377–380. International Computer Music Association.

Puckette, M. S. (2005). Phase bashing for sample-based formant synthesis. In *Proceedings of the International Computer Music Conference*, Ann Arbor, pp. 733–736. International Computer Music Association.

Rodet, X. (1984). The chant project: from the synthesis of the singing voice to synthesis in general. *Computer Music Journal 8*(3), 15–31.

Templaars, S. (1977). The vosim signal spectrum. *Interface 6*, 81–96.