Post-mix Vocoding and the making of All You Need Is Lunch

Miller Puckette
University of California at San Diego
msp@ucsd.edu

Kerry L. Hagan
University of Limerick
Kerry.Hagan@ul.ie

Reprinted from Proceedings, Sound and Music Computing*, 2023*

ABSTRACT

A six-minute-long audiovisual presentation, *All You Need Is Lunch*, was produced using a novel vocoding technique in which the spectrum of a sung word is altered from within a finished stereo mix, avoiding the need for blind source separation. In the piece, snippets of pop music tunes containing the word "love" are altered to say "lunch" instead, as in "where is lunch", "tainted lunch", "saving all my lunch for you", etc. To do this the utterance "lunch" is analyzed using an additive-synthesis model, and the musical recording to be altered is selectively filtered in specific, time-varying frequency ranges and left untouched elsewhere. The depth of alteration is frequency-dependent and time-varying. The target ("lunch") utterance must be time-morphed to fit optimally onto each individual source utterance ("love"). It proved particularly important, and often difficult, to either suppress or hide the sibilant portion of the "v" consonant. Since over 100 occurrences of the source word were altered, production tools were developed for editing and managing the many time-varying parameters that had to be chosen through critical listening and, ultimately, painstaking trial and error.

1. INTRODUCTION

Music production, both historical and present-day, can be very roughly divided into two paradigms, "studio" and "live". The (roughly 100-year-old) studio paradigm aims for a perfect finished product, and typically entails repeated, incremental editions to a static document. In this respect it resembles Western common-practice music composition.

On the other hand, the live paradigm aims at repeated performance, prepared in practice rooms and rehearsal spaces. Whatever technologies are used are usually considered as musical instruments whose behavior is in some sense reproducible, so that an onstage performance can resemble a rehearsed one. As an added bonus, if the production process hews carefully to certain safety protocols, the piece can be made reproducible over long enough spans of time to allow for future performances and/or musicological study.

Current work along these lines is underway at IRCAM [1, 2] and elsewhere [3].

Here we present a production of a fixed-medium piece of the sort that is usually developed in a studio paradigm, but using tools designed for performance, so that the techniques used can be readily studied and adopted by others. Our methods call to mind recent reconstructions of work by John Chowning and Hildegard Westercamp [4]. That we created a fixed-medium piece using a performance-oriented workflow was an accident of circumstances; our practice was formerly purely live and we only resorted to making a fixed-medium work because live performances were impractical at the time.

The piece was motivated by a headline that appeared in the Guardian newspaper about getting children to eat, headlined "Lunch is a battlefield". Not only is the pun irresistible, but it clearly has legs: almost any of the hundreds of popular songs that feature the word "love" (used as a noun, not a verb) becomes oddly funny when the word is changed to "lunch".

We made a montage of 29 of the juiciest examples we could find and set about vocoding each of 118 instances of the word "love" to "lunch". This would be straightforward if we had used a plug-in to separate the voices out from the instrumental parts, but we chose to do the vocoding in-place, that is, directly onto the unaltered stereo commercial recordings, for at least two reasons: first, blind source separation algorithms give results of uncertain and variable quality; and second, we wanted to use our habitual production environment, oriented toward live performance, and based on open-source tools (Audacity and Pure Data). As a side benefit, the lion's share of the audio manipulations—all but the montage—run in real time within a Pd patch and are thus both transparent and reproducible.

After creating the montage, we used a vocoding technique in Pd to perform 118 minor surgeries on it, attempting in each case to turn the word "love" to "lunch", sometimes quite successfully, sometimes less so. The final product, a 6-minute video, uses this soundfile as a soundtrack, and contains a video component produced separately. It can be seen on
msp.ucsd.edu/media/music/
2021.01.13-higgs-whatever-lunch.mp4
—the reader might want to look at it to decide whether to read the rest of this paper.

2. VOCODING TECHNIQUE

Our situation differs from most vocoder setups, in which a non-verbal sound such as a guitar is filtered following the

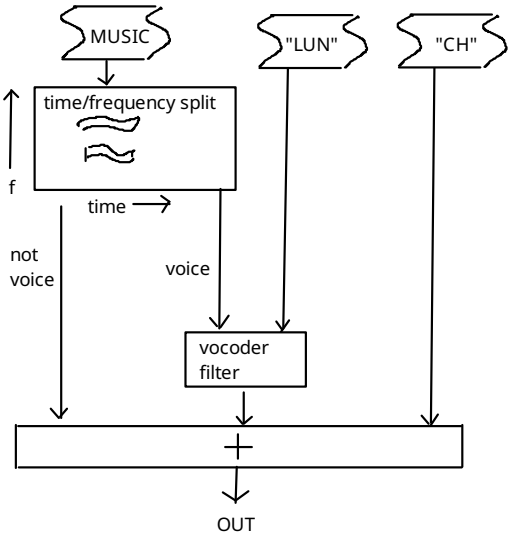

Figure 1. Simplified block diagram of the word-changing process.

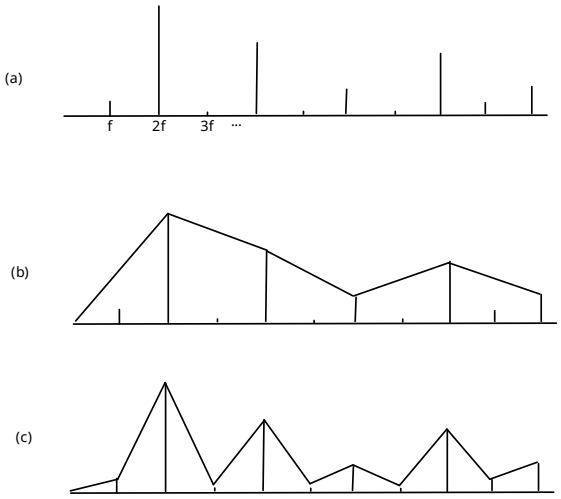

Figure 2. A hypothetical spectrum (a), gives rise to spectral envelope (b) or (c) depending on what we take as a hypothetical filter source signal's fundamental frequency.

spectral envelope of a *control sound*, in three ways. First, the sound to be vocoded contains both instruments and one or more layers of singing, and we ideally want to change the sound of the singing without affecting the instruments. Second, the vocal utterance is short and entirely voiced (the "ch" consonant is handled separately as described below) so we can obtain a highly reliable pitch track for it and get amplitude envelopes for all the harmonics within the frequency range we care about. Finally, the music to be altered, which we call the *target* sound, is already vocal, not instrumental, so our problem is to alter one vocal sound to another rather than to impose vocality on a non-vocal sound.

Changing the word "love" to "lunch" is best understood in terms of the individual phonemes to be altered. The voiced part of both words, the nonsense syllables "luh" and "lun", start similarly but have different formant trajectories. One can be transformed into the other. Fortunately, because of the structure of the vowel of "love", the filter need not apply gains of more than 10 or 15 dB in any frequency region so this is a reasonably stable transformation. (If, for instance, a higher gain were needed this would also boost any noise or parasitic sound uncomfortably.)

The most important spectral difference between the two vowels is that a formant located about 1500 Hz. in "luh" stays largely put as the "v" consonant appears, whereas the "n" of "lun" consists partly of a rise in the formant frequency over time to reach about 2100 Hz. Other spectral changes in the 500-Hz. range are more variable from one singer to another, and are sometimes salient and at other times buried beneath the accompaniment. Often a relatively constrained frequency range of spectral alteration suffices to change the one vowel sound to the other.

The consonant "ch" of "lunch" can't be formed using filtering since it depends on a noisy source. Even though there is some air turbulence to be heard in the "v" of "love" it is not at all isolatable from the simultaneously heard voiced part of the word. So here the only workable strategy will be to add sampled, unvoiced "ch" sounds of appropriate loudness and duration.

Finally, and most difficult, is the removal of the "v" consonant. It has relatively low amplitude and its frequency range is not as wide as the "ch" sound we are adding, and so we simply cheat and try to cover it with the sample. The overall procedure is diagrammed (in very simplified form) in figure 1.

The figure is oversimplified in one important respect: the recombination of the non-vocoded and vocoded segments is actually done in the short-time frequency representation as windowed DFTs. In this way the splitting and recombination of to-be-processed and to-be-left-alone FFT channels is reduced to setting a time-variable, per-bin depth coefficient. If the depth is zero the magnitude is left unchanged, and as the depth increases to 100% the magnitude is increasingly replaced by one determined by the spectral envelope of the control syllable "lun". The depth coefficient can be ramped up and down so that the transition between original and processed sound is less abrupt.

If we wished we could use a blind source separation algorithm to carve out the spectral areas we wish to vocode, since most such algorithms work by creating spectral masks of the same sort. But since we would have had to hand-correct the masks anyway, we chose the simpler path of defining the masks entirely by hand.

None of the operations we are carrying out is completely novel, of course. In particular, the voice-to-voice problem was already recognized and attacked in the earliest musically useful phase vocoder [5]. That phase vocoder has an option, when making a transformation that caused an input signal to be transposed, to re-impose the spectral envelope of the original sound upon the transformed one. This paper does not claim to lay out any techniques that are novel in themselves, but rather to show an example in which an integrated approach proved useful in a particular musical

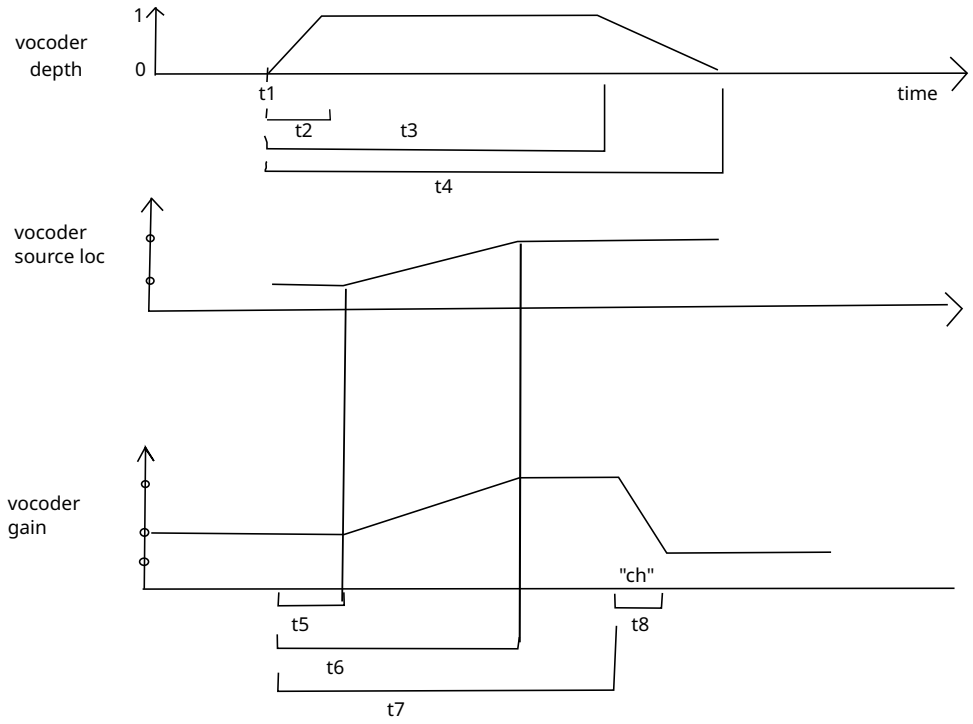

Figure 3. Parameter definitions used to control filtering.

situation.

2.1 Measuring spectral envelope

Given a short-duration recorded sound such as a few milliseconds of speech, a first problem will be to measure its spectral envelope. This is often done using either linear prediction or Fourier analysis, each of which has its own limitations. As a thought experiment, suppose we wish to find the spectral envelope of a periodic signal whose spectrum is as in Figure 2.

Assuming as in part (a) of the figure the sound consists of several sinusoids tuned to multiples of a base frequency f, there are two possible pitches we could perceive, depending on the auditory context and past trajectories of the sinusoids. (We are tacitly assuming that we hear this complex as a fused sound to begin with). Depending on whether the frequency is f or its octave $2f$, we would come to radically different conclusions about the spectral envelope, as shown in parts (b) and (c) of the figure. If we assume that we are hearing a broadband glottal pulse train through a filter, we do not know *a priori* which envelope better represents the filter. This is important because if we wish to apply this filter to a different sound we will get radically different results depending on our estimate of the fundamental frequency.

The situation is further complicated by the presence of noise, as is ubiquitous in normal speech and singing, because we would need yet another oracle to tell us how strong the noise source is compared to the glottal pulse. The standard FFT-based vocoder resolves this problem by smoothing the measured magnitude spectrum out, for example by blurring it using a convolution kernel whose bandwidth becomes a free parameter to be guessed.

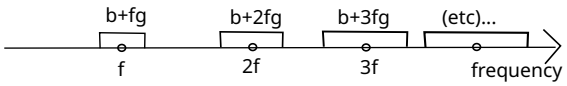

Figure 4. Selection of frequencies over which to apply vocoding. This is determined by the fundamental frequency f and two parameters, b and g that specify a constant bandwidth b plus a frequency-relative bandwidth g.

In our case, the utterance "lun" was carefully spoken and recorded and Pd's `sigmund~` object tracked its fundamental frequency easily. We analyzed the sound by resampling individual cycles (at time points determined in real time) to a fixed buffer size and applying a DFT whose nth channel magnitude gives the nth harmonic strength of the recorded sound. (The individual cycle is obtained by Hann windowing two cycles and additively wrapping the result down to a single cycle.)

The spectral envelope at any frequency is then estimated by interpolating between the magnitudes of four nearest harmonics. This makes a smooth spectral envelope without any holes between the harmonics but with a minimum of blurring.

2.2 Customizing the filter

The vowel-to-vowel vocoding is done by a time-variable filter, implemented by applying a windowed STFT, altering magnitudes while preserving phases, and windowing and overlap-adding the inverse STFTs. The window length is 42.7 msec for a nominal frequency resolution of 23.4 Hz. The amplitude at each frequency that falls within the domain of filtering is adjusted to a level proportional to the amplitude at that frequency of the analyzed control word.

This procedure requires that we supply three time-varying filter parameters: first, the vocoder depth, which must be made to rise from nothing to full strength and back to catch only the moment where the vowel must be modified. Second, the source location at which the control sound is analyzed, which should sweep forward through the diphthong that forms the "n" consonant.

Last, and most delicate, is a time-varying multiplier that is applied to the control-word amplitude to give a new, proposed amplitude for the filtered sound; adjusting this multiplier sets the overall amplitude at which we will hear the new vowel. This can't be easily set automatically since there will inevitably be other sounds mixed in at the same frequencies, and so the contribution of the voice itself to the overall signal strength can't be directly measured.

Figure 3 diagrams the time-varying trajectories of these three signals, depending on eight time parameters, a beginning and final source location, and beginning, middle, and final vocoder gains. The start time and duration of the "ch" sample is also pictured; these are assumed to coincide with a final ramp in the vocoder gain.

Meanwhile, the particular frequencies to be vocoded are specified using three parameters f, b, and g, so that bands centered about frequencies $f, 2f, \cdots$, each with a constant bandwidth b and a proportional bandwidth g are affected by the vocoder (figure 4). For each instance of the word "love" a new triple (f, b, g) is specified. Vocoding within each band is applied according to the time-varying vocoder depth, so that vocoding is limited to specific ranges in both time and frequency. The reasoning for this setup is that we would always expect to alter at least two frequency bins for each partial, but to account for variations in pitch (often a half-tone or more over the duration of the word) an additional proportional factor g is needed.

Sometimes the word is sung by two or three voices in harmony; in this case the regions to be vocoded are combined. Finally, the overall range is limited to lie between a minimum and maximum frequency, for example between 400 and 2000 Hz.

3. IMPLEMENTATION

The audio for the piece was created first, and as performance venues gave way to online platforms, a decision was made to create a video to accompany it. The audio is created automatically by starting a Pure Data patch, which may be downloaded from

`msp.ucsd.edu/media/ideas/lunch/`

This patch reads an already prepared montage of the music to be altered, and outputs in real time the final version of the piece.

The patch also served as the development environment for the piece. The piece can be run backward and forward at any speed and/or scrubbed. This proved essential in setting the thousands of hand-chosen parameters that govern the vocoding. These are stored in a text file that is read statically by the patch, so that the file may be reloaded while sound is being produced, allowing careful tweaking of the parameters at millisecond granularity. Final values of parameters for five of the surgeries are shown in figure 5.

4. EVALUATION

The reader is invited to download the patch and check our observations firsthand (this offer would be much harder to make if, for instance, we used a deep learning model and/or external data collections). Pd's portability and stability make it a good platform for performing repeatable experiments.

As we would expect, instances of "love" where the voice is high-pitched (so that the overtones are widely spaced and we only need treat the file at a relatively sparse set of frequencies) are usually easier to modify than low-pitched ones. So for instance "where is lunch" (47.390 seconds into the piece) works very well. This is fortunate since, being the first moment the change is made, it sets the listener up for some ensuing, less successful ones.

This condition is not met at all in "lunch, sweet lunch" (second word "lunch", 318.770") where the singer's melisma ranges over a major third, so that isolating fixed frequency ranges for harmonics becomes pointless. In this case, rather than invent a mechanism for tracking moving harmonics, we simply throw up our hands and treat an entire frequency range, from 400 to 1500 Hz. Surprisingly, in this particular case the result is not bad. Similarly, Freddy Mercury's contributions (four repasts starting at 165.467") are so strongly sung that even though the pitch ranges widely through individual words, the results are good.

Results were not uniformly good for "good lunch" (56.463" and onward) - these four lunches are sung at quite stable pitches but in 3-part harmony (294, 370, 440 Hz) so that their harmonics crowd the entire spectrum. Here, the voices hew closely to their nominal pitches and we were able to choose fairly tight bandwidths (20 Hz. plus 4% of frequency), but we still hear that the vowel change is stamped on the accompanying music as well as the voices. The third of the four meals is particularly unsuccessful; we hear the accompaniment heavily filtered but still don't hear the intended vowel clearly.

Covering the "v" consonant was often hard to do without stepping audibly on both the voiced aspect of the voice and the instrumental accompaniment, all of which often occupied similar frequency ranges around 100-200 Hz. For example in "You've got to hide your lunch away" (73.936") , suppressing the "v" sound required such a brutal filter that the whole mix takes a momentary dive.

Frank Sinatra's "lunch and marriage" (253.767' and 255.938") proved especially difficult, and the end result is unsatisfactory. He holds his pitch very stably and his diction is very clear. But his "v' consonant lies atop the word "and", sung at a pitch a fifth, and then a minor sixth, higher than "love". Attempts at filtering to quiet the "v" had the knock-on effect of destroying the short "a" of "and". Instead, we formed the "un" diphthong out of his "love" (without any "v" to worry about) and simply tried to bury the "v" under our "ch" sample, which also then fell late, on top of "and". The same obnoxious, mansplaining vocal style that makes Sinatra an essential contribution to our bit of satire makes it quite difficult to deal with acoustically.

Most difficult of all, and least successful of all, are the ending "lunch, lunch, lunch is all you need" (341.700")

```
0 t1    t2  t3  t4     t5 t6 t7  t8  g1 g2 g3 v1   v2     cha/s/p/lp  bw  lf  hf  sq  f1  f2 f3 lf/pcbw
0 addicted to lunch
1 69384 30 350 400     0 400 500 300 25 25 25 600  900    55  2 4 50  70  600 2500 70 57   0 0  10   1
0 hide your lunch away
1 73936 30 350 400     0 350 250 200 25 15 15 1700 2150   45  5 2 50  70  500 3200 70 52.4 0 0  10   9
0 the look of lunch
1 80960 50 700 800     0 800 700 300 35 35 35 550  1100   40 11 3 50  70  600 2500 70 66   0 0  10   2
0 im in the mood for lunch
1 90870 30 1400 1452   0 350 1871 300 25 25 20 550 1100   25 15 3 50  70  600 1700 50 55   0 0  10   1
0 oh my lunch
1 99840 150 850 920    0 850 1004 300 35 35 35 550 1100   35  4 5 50  70  600 1700 23 60   0 0  10   2
```

Figure 5. Parameter sets for five surgeries (of 118 total).

and its many hocketed repetitions at the end of the original recording. Here we resorted to a ruse added at the last minute: since the chords behind the first three lunches are each repeated one beat before or after the sung word, we added an option in which the un-sung-over chord is filtered down to the frequency range in which the alteration is made and layered on top of it to cover for the brutality of the filtering. In these lunches and in the closing repetitions, the word "love" is highly compressed in time and liaisoned onto the following word; and, even worse than in the Sinatra example, the "v" liaisons to the "s" of the following word "is". Worse yet, during the hocketing, each final word of the phrase ("need") overlaps in time with the following "lunch". The results are not as convincing as in the rest of the piece.

5. CONCLUSION

Much of electronic music happens behind the scenes and it is often hard to borrow ideas explored in other musicians' past work for one's own ends. Here we present an example of a piece of studio-produced electronic music that can be studied in detail, and the authors would welcome such borrowings.

This work can also be taken as evidence of the usefulness of open-source music platforms as foundations for building a body of common practice in electronic music, both studio-based and live.

Meanwhile, the piece itself appears as a companion musical submission to this conference. Ideally that piece would be programmed in a late morning concert...just before lunchtime.

6. REFERENCES

[1] S. Lemouton, A. Bonardi, L. Pottier, and J. Warnier, "On the documentation of electronic music," *Computer music journal*, vol. 42, no. 4, pp. 41–58, 2018.

[2] S. Lemouton, "The electroacoustic repertoire: Is there a librarian?" *array.*, pp. 7–14, 2020.

[3] M. S. Puckette, "New public-domain realizations of standard pieces for instruments and live electronics," in *Proceedings of the International Computer Music Conference*. Ann Arbor: International Computer Music Association, 2001, pp. 377–380.

[4] M. Clarke, F. Dufeu, and P. Manning, *Inside Computer Music*. Oxford University Press, USA, 2020.

[5] M. Dolson, "The phase vocoder: A tutorial," *Computer Music Journal*, vol. 10, no. 4, pp. 14–27, 1986.