Phase-locked Vocoder

Miller Puckette

Department of Music, UCSD
La Jolla, Ca. 92039-0326
msp@ucsd.edu

ABSTRACT

The phase vocoder is widely used to provide high-fidelity time stretching or contraction of audio signals such as speech or monophonic musical passages. Two problems bedevil the reconstructive phase of this technique. First, the frequency estimate is usually multi-valued and one does not know how to choose which of the possible frequencies given by the analysis to use in resynthesis. Second, a sinusoidal component in the incoming sound always excites several channels of the analysis; the reconstructed sine waves can interfere with each other, giving rise to a reverberant sound. An improved formulation of the phase vocoder is proposed here for which the first difficulty does not arise; and a means of phase-locking adjacent channels of the resynthesis is proposed which alleviates the second one.

1. PHASE VOCODER

The phase vocoder, first proposed by [Flanagan], was put into its modern, FFT-based form by [Portnoff] and explored for musical applications by [Moorer] and [Dolson]. An incoming sampled signal $x[n]$ is first analyzed using windowed DFTs. We will denote the results of this analysis by $X[s,k]$:

$$X[s,k] = \mathcal{FT}\{w[n]x[n-s]\}[k],$$

where w[n] is a windowing function and the DFT is taken as,

$$\mathcal{FT}\{x[n]\}[k] = \sum_{n=0}^{N-1} e^{-2\pi i n k/N} x[n].$$

(We will occasionally abuse the above definition by applying it for non-integral values of k; in this case we enclose k in parentheses instead of brackets. The reader is referred to [Settel] for technical details regarding the definitions given here.) The analysis need not be carried out for all possible values of the starting point s.

The phase vocoder assumes that the incoming sound may be modeled as a sum of sinusoidal components, each with relatively slowly changing amplitudes and frequencies. The components should be spaced widely enough apart that in each channel of the windowed DFT analysis, at most one component contributes more than negligibly. For each channel we can then estimate the angular frequency of the component contributing to it by measuring the rate of phase change in that channel. Suppose we measure the phase change between analyses at points s and t; i.e., the *hop size* is $H = t - s$. The frequency estimate in radians per second for the kth channel is given by,

$$\omega[k] = (\mathrm{Arg}X[t,k] - \mathrm{Arg}X[s,k] + 2\pi p)/H,$$

where p is an integer to be determined. The possible frequency estimates are more widely spaced the closer s and t are; if they differ by only one (a hop size of one sample), the possible values of ω differ by multiples of 2π, that is, the frequency is completely determined.

The maximum allowable difference $t-s$ depends on the bandwidth of the windowing function chosen. If we choose a sinusoidal input signal:

$$x[n] = e^{i\omega n},$$

then we have

$$X[s,k] = e^{i\omega s} W(k+\omega)$$

where

$$W(k) = \mathcal{FT}\{w[n]\}.$$

In other words, analyzing a complex exponential gives a frequency-shifted image of the windowing function's short-term Fourier transform. If we assume that $W(k)$ is concentrated in the interval $[-K, K]$ for some $K > 0$, then a component of the signal which contributes energy to some specific bin of the FFT could have any frequency out of an interval $4\pi K/N$ radians per second wide. The hop size must therefore satisfy the inequality $H \le N/2K$ in order to ensure that no more than one value for the integer p can give rise to an admissible frequency estimate. For Hamming or Hanning window functions we must have $H \le N/4$; for the Kaiser window function, $H \le N/6$.

We now consider analysis/resynthesis systems of the sort described in [Portnoff], [Moorer], and [Dolson]. There, the output is synthesized by overlap-adding IFFTs with a constant overlap, typically 4 or 8. We therefore wish to compute output spectra $Y[u_i, k]$ for $u_i = 0, J, 2J, ...$, where J denotes the output hop size, usually $N/4$ or $N/8$. The algorithm is recursive; the ith output spectrum depends both on the previous output spectrum as well as on two input spectra, $X[s_i, k]$ and $X[t_i, k]$. The magnitude of each channel of the output spectrum is set to the amplitude of the corresponding channel of either one of the inputs. The phase of each channel of the output spectrum is computed using the phase of the same channel of the previous output spectrum and the frequency estimate taken from the two input spectra:

$$\mathrm{Arg}Y[u_i, k] = \mathrm{Arg}Y[u_{i-1}, k] + (u_i - u_{i-1})\omega[k],$$

where $\omega[k]$ is the frequency calculated above. Since ω is multi-valued, the output phase is not necessarily well-defined;

differing values of the integer p give possible outputs whose phases differ by multiples of

$$2\pi(u_i - u_{i-1})/(t_i - s_i).$$

If we choose values of s, t, and u so that $u_i - u_{i-1}$ is an integer multiple of $t_i - s_i$, no indeterminacy results. Otherwise, if we choose a small enough hop size so that no more than one possible value for the frequency can lie within K bins of the channel's center frequency, then we can at least be assured that errors, if they occur, will not affect channels within the main lobe of a spectral peak.

Traditionally, the resynthesis step is seen as a bank of N oscillators, each operating at $1/J$ times the sample rate of the time-domain signals we are analyzing and resynthesizing. The calculation of the phase of each channel of the u_ith output spectrum from that of the corresponding channel of the u_{i-1}th one is equivalent to updating the phase of a digital oscillator. The inverse FFT has the effect of heterodyning the oscillators into their proper frequency bands and summing them. In this picture, the value of J is limited by the need to maintain a high enough sample rate to hold the bandwidth of the contents of the bins.

The algorithms described in the literature often make the additional specification that we reuse each FFT analysis by choosing $s_{i+1} = t_i$. At first glance this would seem to be a good choice since it saves an FFT calculation. But, as both [Charpentier] and [Brown] have noted, another trick exists which also avoids the calculation of the second FFT in the case that we choose a hop size of one, that is, $t_i = s_i + 1$. [Moorer], while disparaging the choice of hop 1, notes that we can also avoid taking the arctangent in this case.

In any event, it may easily occur that quality considerations outweigh the possible advantage of avoiding one FFT in every three. In the usual setup, it is not always guaranteed that the output phase is well-defined. The choice of t_i is dictated by the desired time stretching in the output. If time is not stretched by an integer factor, the output phase is multivalued, and we fall back on requiring a minimum time stretch factor (which depends on window type and output overlap). Difficulties also arise if we try to freeze a sound or reverse its direction in time.

The approach taken here is to choose

$$t_i - s_i = u_i - u_{i-1},$$

so that

$$\mathrm{Arg}Y[u_i, k] = \mathrm{Arg}Y[u_{i-1}, k] + \mathrm{Arg}X[t_i, k] - \mathrm{Arg}X[s_i, k].$$

This choice has the interesting property that it avoids *all* trigonometric calculations; if we combine the phase propagation equation given earlier with the magnitude equation, $|Y[u_i, k]| = |X[t_i, k]|$, we get

$$Y[u_i, k] = X[t_i, k]\left(\frac{Y[u_{i-1}, k]}{X[s_i, k]}\right)\left|\frac{Y[u_{i-1}, k]}{X[s_i, k]}\right|^{-1}.$$

2. PHASE LOCKING

There remains one largely unacknowledged difficulty in using the phase vocoder to reconstruct sounds in this way. A complex-exponential component of a signal to be analyzed excites not one single channel of the phase vocoder analysis but $2K$ of them. In the resynthesis step, we therefore reconstruct each component not once, but $2K$ times. The amplitude of the result will therefore depend on the relative phases of these $2K$ oscillators.

The situation is further complicated if the frequencies of the components of the original sound are allowed to change with time. In this case, the set of channels of the spectrum which are excited by a specific component may also change with time. Taking the oscillator-bank point of view, we see that as new oscillators are used in the reconstruction, they should be brought to the correct phase relative to the oscillators which are already present.

Suppose, for example, we use the Hanning window:

$$w[n] = 1/2\,(1 - \cos(2\pi n/N))\,, n = 0, ..., N-1,$$

so that

$$W(k) = e^{-j\pi k}\left(\frac{1}{2}D_N(k) + \frac{1}{4}(D_N(k+1) + D_N(k-1))\right),$$

where

$$D_N(k) = \frac{\sin(\pi k)}{\sin(\frac{\pi k}{N})}.$$

(In order to simplify this analysis, we have chosen the windowing function to be centered over the point $n = N/2$, not halfway between $N/2$ and $N/2+1$ which is the true center of the analysis window. The reader is again referred to [Settel] for clarification.) If a signal component lies directly in the center of a bin, the analysis is concentrated in three channels with relative strengths 1, 2, and 1; in this special case we excite only three bins instead of the usual four. If the signal component lies halfway between two bins, we get four components of relative strengths 1, 5, 5, and 1. In either case the peaks are alternately 180 degrees out of phase.

Following this reasoning backwards, if in the resynthesis stage we find 4 (or exceptionally 3) channels whose amplitudes and phases follow those of $W(k - \omega)$ for some offset ω, taking the IFFT—without applying any windowing function at all—will reproduce a Hanning-shaped sinusoidal burst, of angular frequency ω, centered at sample number $N/2$.

On the other hand, if we have for example the correct amplitudes but if all the channels are exactly in phase (not alternating in sign as prescribed above), we will unfortunately resynthesize the Hanning window backward, with a peak at 0, another at N, but a trough at $N/2$. At this point we would face the necessity of windowing the outputs for overlapped adding; after windowing we will be left with a much diminished amplitude.

This would suggest at first glance that we cannot win: if the phases are aligned correctly the output is already Hanning-window-shaped, but if the phases aren't arranged as we wish we must apply the Hanning window to the output. Luckily, it turns out that even in the good case, we lose nothing by windowing the outputs, provided we overlap at least by three. This is due to an odd property of the Hanning (and Hamming) windowing functions: if we overlap-add a sequence of them by a factor of $F \geq 2$, not only is the sum of the windowing functions a constant, but also the sum of any power up to $F-1$. (For the Kaiser window, we may go up to $(F-1)/2$.) We may therefore multiply the output signals by the Hanning window function, even if the output is already window-shaped, and still reconstruct the sinusoidal signal correctly.

If, however, we do not act to control the relative phases of adjacent channels of the output spectra $Y[u,k]$, there will still be errors in the output, primarily in its amplitude. As the pitch of the analyzed signal changes, the amplitude error will also change. This is the underlying cause of the phase vocoder's characteristic “reverberant” sound when it is used for time-stretching. It is simply an artifact caused by beating between adjacent channels as their relative phases change.

One approach to correcting this fault is to use only the channels of the FFT which hold peaks in amplitude. This dodge unfortunately gives rise to two new sources of discomfort. First, in any peak detection algorithm, either spurious peaks often appear, or else true peaks frequently drop out and reappear, or both. Second, the peak often migrates from channel to channel, so that heuristics are needed in order to provide the continuity from window to window in the analysis which is needed in order to propagate the phases of components correctly.

The alternative proposed here is to introduce phase locking between adjacent pairs of channels. The most effective way of doing this is surprisingly simple. Rather than using the phase of the complex number $Y[u_{i-1}, k]$ in the resynthesis formula, we concoct a weighted average of the phases of the three quantities,

$$-Y[u_{i-1}, k-1], Y[u_{i-1}, k], -Y[u_{i-1}, k+1],$$

where we use the negatives of the $k-1$ and $k+1$ channels because adjacent channels should ideally be 180 degrees out of phase. The phase we use is simply that of the complex sum of the three. This gives rise to an improved resynthesis formula:

$$Y[u_i, k] = X[t, k] \left(\frac{Z[u_{i-1}, k]}{X[s, k]} \right) \left| \frac{Z[u_{i-1}, k]}{X[s, k]} \right|^{-1},$$

where

$$Z[u_{i-1}, k] = Y[u_{i-1}, k] - Y[u_{i-1}, k-1] - Y[u_{i-1}, k+1].$$

Here we have arbitrarily chosen to weight the two adjacent channels equally with the original channel in determining the new phase. Whether or not it is optimal to weight the three phases equally cannot be decided until some quantitative measure of success is proposed. In practice, the result appears not to suffer greatly if the weights are changed to 1:2 or 2:1.

In a sum of complex numbers, the summand with the greatest modulus naturally has the strongest effect on the phase of the sum. In the case where a sinusoid is placed halfway between bins of the DFT, so that the windowed spectrum has amplitudes in the ratio 1:5:5:1, the two outlying bins quickly take on the phase of the one adjacent bin (whose amplitude is five times greater.) In this way, if a changing frequency causes a new channel to be added to the set contributing to some component, the new channel quickly aligns itself with the phase of the much stronger adjacent bin.

3. OUTPUT

The considerations above were confronted during the design of a new, real-time phase-vocoder application implemented on the ISPW system [Lindemann] using the Max computer music environment [Puckette]. The application is capable of altering the pitch and duration of a recorded sound independently; both the transposition and the time-stretching factor may vary in time as desired. It is possible to freeze sounds or change their direction in time.

It proved in the context of the ISPW that a vector arctangent was far more expensive than using an extra FFT, so the reformulation of the resynthesis stage shown here worked far better than the previously published method of evaluating ω explicitly. Moreover, the step of explicitly reevaluating the two windowed DFTs each time made it easy to precess or recess in time at any rate desired. The analyses were done on the output of a granular sampler so that in each frame of the analysis a transposition up or down could be included at no additional cost.

The phase locking mechanism described above was included as a switch. Testing the algorithm on a sung voice demonstrated that the time-altered sounds were much more faithful to the original recorded sound when phase-locking was used than not. The difference between the two was that when phase locking was not used the sound output had the phase vocoder's characteristic reverberant quality, while in the phase-locked case the resynthesized sound had the same presence as the original.

Whether using phase locking between adjacent channels of the phase vocoder is truly the best solution to the problems cited here cannot be answered yet. We lack an effective measure of the quality of the resynthesized sound. However, at least the question of fidelity of resynthesis has now been raised, and it has been established here that the phase vocoder is built upon assumptions whose ramifications are not completely understood. If we can put the phase vocoder on a steadier foundation, we might have more control over what comes out of it.

References

[Brown] Brown, J.C., and Puckette, M.S., 1993. “A high resolution fundamental frequency determination based on phase changes of the Fourier transform.” J. Acoust. Soc. Am. 94: pp. 662-667.

[Charpentier] Charpentier, F.J. 1986. “Pitch detection using the short-term phase spectrum.” Proc. International Conference on Acoustics, Speech, and Signal Processing, IEEE, New York, N.Y.: pp. 113-116.

[Dolson] Dolson, M., 1986. “The phase vocoder: a tutorial.” Computer Music Journal, 10/4: pp. 14-26.

[Flanagan] Flanagan, J.L., and Golden, R.M., 1966. “Phase vocoder." Bell Syst. Tech. J. **45** 1493-1509; Reprinted in *Speech Analysis*, R.W. Schaefer and J. D. Markel, Eds. New York: IEEE Press, 1979.

[Lindemann] Lindemann, E. et al., 1991. "The Architecture of the IRCAM Music Workstation." Computer Music Journal 15/3, pp. 41-49.

[Moorer] Moorer, J.A., 1976.. "The use of the phase vocoder in computer music applications." J. Audio Engineering Soc. 26/1: pp. 42-45.

[Portnoff] Portnoff, M.R., 1976. "Implementation of the digital phase vocoder using the fast Fourier transform." IEEE trans. Acoust., Speech, Signal Processing, Vol. ASSP-24, pp. 243-248; Reprinted in *Speech Analysis*, R.W. Schaefer and J. D. Markel, Eds. New York: IEEE Press, 1979.

[Puckette] Puckette, M., 1991. "Combining Event and Signal Processing in the MAX Graphical Programming Environment." Computer Music Journal 15/3: pp. 68-77.

[Settel] Settel, J., and Lippe, A.C., 1995. "Real-time Musical Applications using Frequency Domain Signal Processing." These Proceedings.