Mark Dolson
Computer Audio Research Laboratory
Center for Music Experiment, Q-037
University of California, San Diego
La Jolla, California 92093 USA

The Phase Vocoder: A Tutorial

Introduction

For composers interested in the modification of natural sounds, the phase vocoder is a digital signal processing technique of potentially great significance. By itself, the phase vocoder can perform very high fidelity time-scale modification or pitch transposition of a wide variety of sounds. In conjunction with a standard software synthesis program, the phase vocoder can provide the composer with arbitrary control of individual harmonics. But use of the phase vocoder to date has been limited primarily to experts in digital signal processing. Consequently, its musical potential has remained largely untapped.

In this article, I attempt to explain the operation of the phase vocoder in terms accessible to musicians. I rely heavily on the familiar concepts of sine waves, filters, and additive synthesis, and I employ a minimum of mathematics. My hope is that this tutorial will lay the groundwork for widespread use of the phase vocoder, both as a tool for sound analysis and modification, and as a catalyst for continued musical exploration.

Overview

The phase vocoder has its origins in a long line of voice coding techniques aimed at minimizing the amount of data that must be transmitted for intelligible electronic speech communication. Indeed, the word "vocoder" is simply a contraction of the term "voice coder." The *phase vocoder* is so named to distinguish it from the earlier *channel vocoder,* which is described in this paper as well. Commercial analog "vocoders" for electronic music are of the channel-vocoder type. The phase vocoder was first described in an article by Flanagan and Golden (1966), but it is only in the past ten years that this technique has become popular and well understood.

The phase vocoder is one of a number of digital signal processing algorithms that can be categorized as *analysis-synthesis* techniques. Mathematically, these techniques are sophisticated algorithms that take an input signal and produce an output signal that is either identical to the input or a modified version of it. The underlying assumption is that the input signal can be well represented by a model (i.e., a mathematical formula) whose parameters vary with time. The *analysis* is devoted to determining the values of these model parameters for the signal in question, and the *synthesis* is simply the output of the model itself.

The benefits of analysis-synthesis formulations are considerable. Since the synthesis is based on analysis of a specific signal, the synthesized output can be virtually identical to the original input; this can occur even when the signal in question bears little relation to the assumed model. Furthermore, the parameter values derived from the analysis can be altered to synthesize useful modifications of the original signal. In the case of alteration, however, the perceptual significance and musical utility of the result depends critically on the degree to which the assumed model matches the signal to be modified.

In the phase vocoder, the signal is modeled as a sum of sine waves, and the parameters to be determined by analysis are the time-varying amplitude and frequency for each sine wave. These sine waves are not required to be harmonically related, so this model is appropriate for a wide variety of musical signals. For example, wind, brass, string, speech, and some percussive sounds all lend themselves well to phase-vocoder representation and modification. Other percussive sounds (e.g., clicks) and certain signal-plus-noise sound combinations are not well represented as a sum of sine waves. These sounds can still be perfectly resynthesized, but attempts to modify them can lead to unanticipated sonic results. Since the musical utility of the phase vocoder lies in its ability to modify sounds predic-

Computer Music Journal, Vol. 10, No. 4, Winter 1986,

Fig. 1. *Filterbank interpretation of the phase vocoder.*

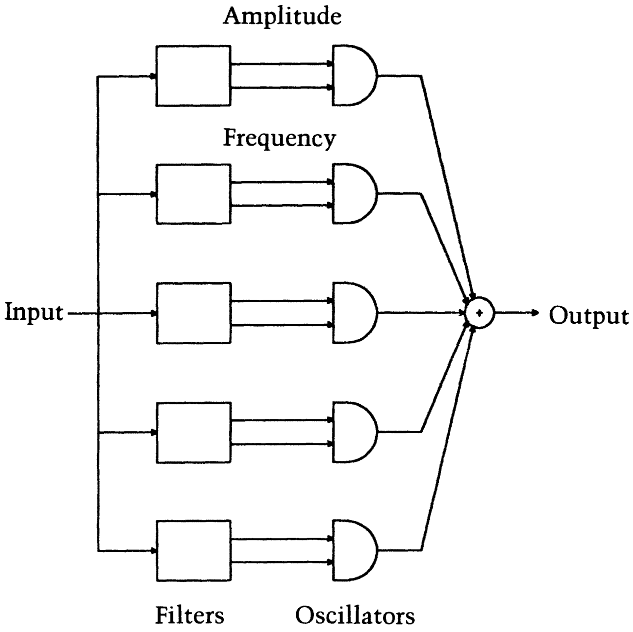
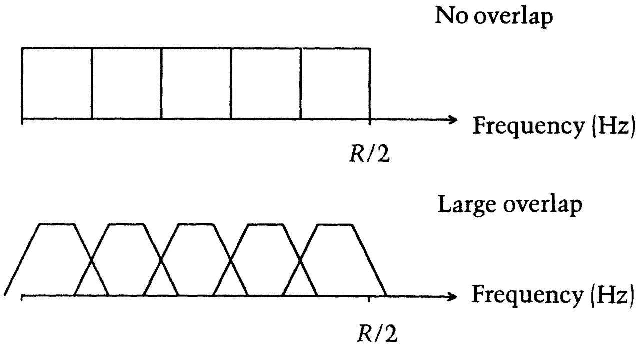


Fig. 2. *Idealized frequency response of bandpass filters.*



tably, signals that fail to fit the model are a potential source of difficulty.

In the sections that follow, I show in detail how the phase-vocoder analysis-synthesis is actually performed. In particular, I show that there are two complementary (but mathematically equivalent) viewpoints that may be adopted. I refer to these as the *filterbank* interpretation and the *Fourier-transform* interpretation, and I discuss each in turn. Lastly, I show how the results of the phase-vocoder analysis can be used musically to effect useful modifications of recorded sounds. In all cases, the input to the phase vocoder is assumed to be a discrete (i.e., sampled) signal with a sampling rate R of at least twice the highest frequency present in the signal. Thus, for a typical high fidelity application, R = 50 KHz. The frequency range of interest is then from 0 Hz to $R/2$ Hz.

The Filterbank Interpretation

The simplest view of the phase-vocoder analysis is that it consists of a fixed bank of bandpass filters with the output of each filter expressed as a time-varying amplitude and a time-varying frequency (Fig. 1). For resynthesis, the amplitude and frequency outputs serve as the control inputs to a bank of sine-wave oscillators. The synthesis is then literally a sum of sine waves with the time-varying amplitude and frequency of each sine wave being obtained directly from the corresponding bandpass filter. If the center frequencies of the individual bandpass filters happen to align with the harmonics of a musical signal, then the outputs of the phase-vocoder analysis are essentially the time-varying amplitudes and frequencies of each harmonic. But this alignment is by no means essential to the analysis.

The filterbank itself has only three constraints. First, the frequency response characteristics of the individual bandpass filters are identical except that each filter has its passband centered at a different frequency. Second, these center frequencies are equally spaced across the entire spectrum from 0 Hz to $R/2$ Hz as shown in Fig. 2. Third, the individual bandpass frequency response is such that the combined frequency response of all the filters in parallel is essentially flat across the entire spectrum. (This is akin to the situation in a good graphic or parametric equalizer when the gain in each band is set to 0 db; the resulting sound has no added coloration. In the phase vocoder, this condition ensures that no frequency component is given disproportionate weight in the analysis.) As a consequence of these constraints, the only issues in the design of the filterbank are the number of filters and the individual bandpass frequency response.

The number of filters should always be sufficiently large so that there is never more than one

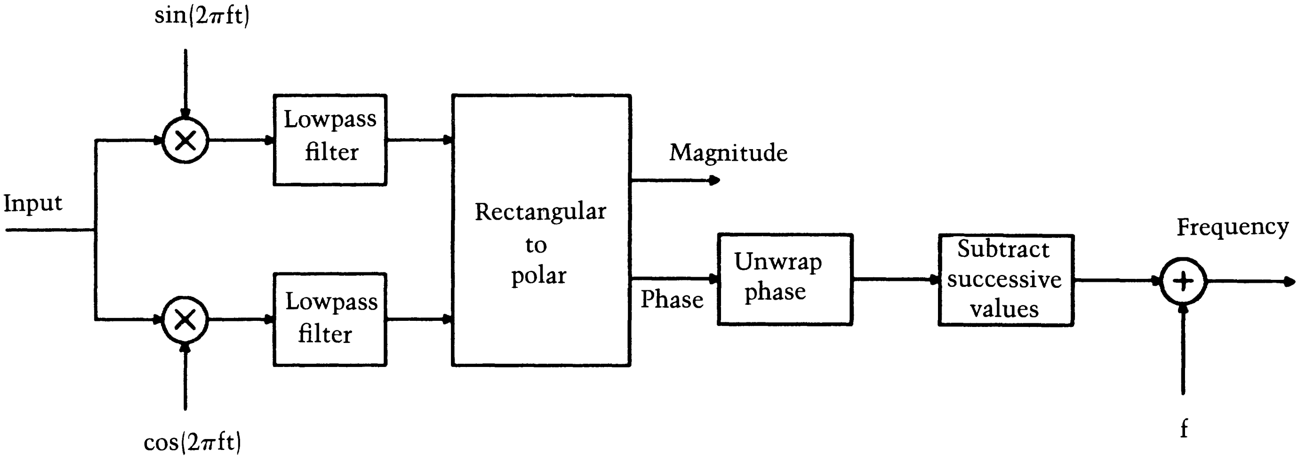


Fig. 3. An individual bandpass filter.

partial within the passband of any single filter. (In general, more filters mean narrower passbands for each filter.) For harmonic sounds, this means that there must be at least one filter for each harmonic in the frequency range from 0 Hz to $R/2$ Hz. Thus, for example, if the sampling rate R is 50 KHz, and the fundamental frequency of the sound is, say, 500 Hz, then there are partials every 500 Hz. Consequently, there are (25000/500) = 50 partials in all, and we need 50 filters in our bank.

For inharmonic and polyphonic sounds, the number of filters is usually much greater because the partials are no longer equally spaced. If the number of filters is too small, then the phase vocoder does not function as intended because the partials within a single filter constructively and destructively interfere with each other (i.e., they will "beat" with each other), and the information about their individual frequencies is coded as an unintended temporal variation in a single composite signal.

The design of the representative bandpass filter is dominated by a single consideration: the sharper the filter frequency response cuts off at the band edges (i.e., the less overlap between adjacent bandpass filters), the longer the filter will *ring* in response to a single nonzero input sample (i.e., the longer the filter *impulse response*). Another way of saying that is: the sharper the filter frequency response, the longer it takes the filter to respond to changes in the input signal. This is a fundamental tradeoff in the design of any filter—a sharp frequency response comes only at the expense of a slow time response.

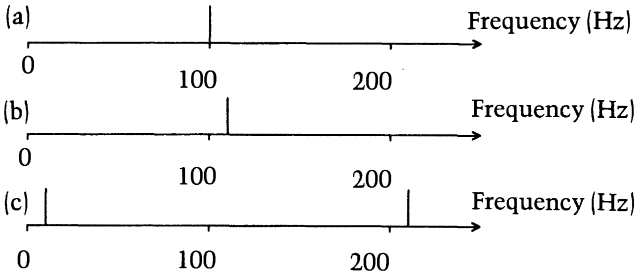
In the phase vocoder, the consequence of this tradeoff is that to get sharp filter cutoffs with minimal overlap, one must use filters whose time response is very sluggish. For slowly-varying sounds, a sluggish time response in the individual filters may be perfectly acceptable. For more rapidly varying sounds, however, it may be more desirable for the individual filters to have a rapid time response. In this case, there can be large frequency overlap between adjacent bandpass filters. In practice, the best solution is generally discovered experimentally by simply trying different filter settings for the sound in question.

A Closer Look at the Filterbank

The previous paragraphs provide an adequate description of the phase vocoder from the standpoint of the user, but they omit some important details. In particular, what needs to be clarified is that the actual implementation of the filterbank in the phase vocoder is rather different from that which might be employed in a commercial analog filterbank. In this section, I show in detail how the output of a single phase vocoder bandpass filter is

Fig. 4. Spectral plot of the result of multiplying two sinusoids: (a) sinusoid 1, (b) sinusoid 2, (c) result.

$$\cos(2\pi\ 110\ t)\cos(2\pi\ 100\ t)$$
$$= \cos(2\pi\ (110 - 100)t) + \cos(2\pi\ (110 + 100)t)$$



expressed as a time-varying amplitude and a time-varying frequency.

A diagram of the operation of a single phase-vocoder bandpass filter is shown in Fig. 3. This picture may appear a bit complicated, but it can easily be broken down into a series of fairly simple steps.

In the first step, the incoming signal is routed into two parallel paths. In one path, the signal is multiplied by a sine wave with an amplitude of 1.0 and a frequency equal to the center frequency of the bandpass filter; in the other path, the signal is multiplied by a cosine wave of the same amplitude and frequency. (The reader should recall that sine and cosine waves are both sinusoidal waveforms that differ only in their starting point of *phase.* The cosine wave is always 90 degrees ahead of the sine wave.) Thus, the two parallel paths are identical except for the phase of the multiplying waveform.

The next step in Fig. 3 is that, in each of the two paths, the result of the multiplication is fed into a lowpass filter. To understand the significance of this operation, first one must better understand the multiplication operation itself.

Suppose we have a complex signal consisting of many frequency components. Multiplying that signal by a sinusoid of constant frequency has the effect of shifting all the frequency components by both plus and minus the frequency of the sinusoid. An example of this is shown in Fig. 4 in which a single component at 110 Hz (represented as a cosine wave of constant amplitude $\cos(2\pi 110t)$), is multiplied by a cosine wave at 100 Hz (represented as $\cos(2\pi 100t)$). The effect of this multiplication is to "split" the 110 Hz component into a low frequency component at 10 Hz (i.e., 110 Hz − 100 Hz) and a high frequency component at 210 Hz (i.e., 110 Hz + 100 Hz). (A musical example of multiplying two signals can be found in the common effect of *ring modulation.* But in typical applications, neither the input signal *nor* the multiplying signal is a simple sinusoid; the result of this is that every sinusoidal component of the input signal is split by every sinusoidal component of the multiplying signal. Another musical example, the familiar phenomena of *beats,* is essentially the other side of the same coin: when two sinusoids are added together, the result can be expressed as a product of two sinusoids—an amplitude modulation of one sinusoid by the other.)

Now, if the result of the multiplication in Fig. 4 is passed through an appropriate lowpass filter, only the 10 Hz sinusoid will remain. This sequence of operations (i.e., multiplying by a sinusoid of frequency f and then lowpass filtering) is useful in a variety of signal processing applications and is known as *heterodyning.*

In the phase vocoder, heterodyning accomplishes two things. First, any input frequency components in the vicinity of frequency f are shifted down to the vicinity of 0 Hz and allowed to pass; input frequency components not in the vicinity of frequency f are similarly shifted but not by enough to get through the lowpass filter. Thus, heterodyning implements a type of bandpass filtering in which the passband is frequency-shifted down to very low frequencies. A bandpass filter of this type can easily be designed to have any desired center frequency simply by choosing the frequency of the heterodyning sine and cosine waves to equal the desired center frequency. Indeed, this is precisely how each of the filters in Fig. 2 is actually implemented. The result is not a "true" bandpass filtering because of the frequency-shifting of the passband, but it does effectively separate frequency components in the vicinity of frequency f from all other frequency components. Furthermore, as will be shown shortly, the frequency-shifting is straightforwardly taken into account in subsequent steps within the phase vocoder.

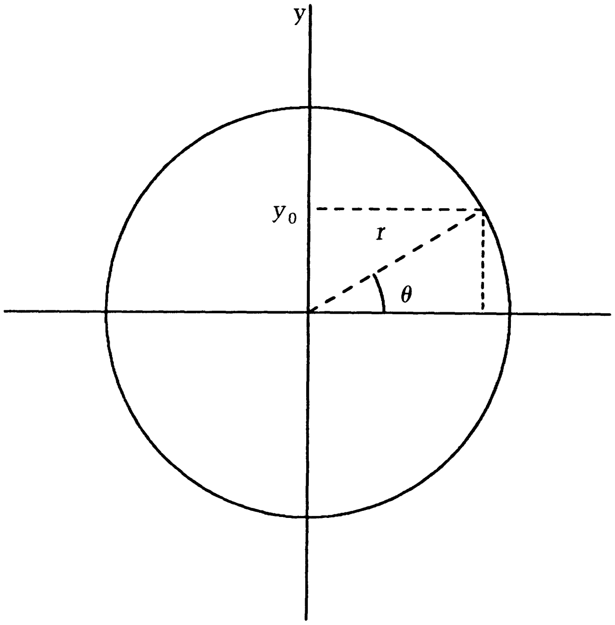
The second important consequence of heterodyning in the phase vocoder is that it provides a simple mechanism for computing the time-varying amplitude and frequency of the resulting signal. In Fig. 3, heterodyning is performed in each of the two parallel paths. But since one path heterodynes with a sine wave while the other path uses a cosine wave (and since the relative phase is preserved by the process), the resulting heterodyned signals in the two paths are out of phase by 90 degrees. Thus, in the above example, both paths produce a 10-Hz sinusoidal wave at the outputs of their respective lowpass filters, but one of the two sinusoids is 90 degrees ahead of the other. What we really want, however, are not two sinusoids, but rather the amplitude and frequency of a single sinusoid. This leads us to the next operation in the sequence of Fig. 3: the transformation from rectangular (i.e., Cartesian) coordinates to polar coordinates.

Rectangular Coordinates versus Polar Coordinates

Sinsuoidal motion is often introduced to students as a projection of uniform circular motion. I now show how this same point of view can be adopted to understand the transformation from "two sinusoids" to "amplitude and frequency of a single sinusoid" within the phase vocoder.

Suppose that we wish to plot the position of a point on the rotating wheel in Fig. 5 as a function of time. We have a choice of using *rectangular* coordinates (e.g., horizontal position and vertical position) or *polar* coordinates (e.g., radial position and angular position). With rectangular coordinates we find that both the horizontal position and the vertical position of our point vary sinusoidally, but the maximum vertical displacement occurs one quarter cycle later than the maximum horizontal displacement (e.g., the point is at its highest one-quarter cycle after being at its furthest to the right). Thus, the horizontal and vertical signals are sinusoids with a 90-degree phase difference between them. On the other hand, if we represent a circularly rotating point in terms of polar coordinates, we simply have a linearly increasing angular position and a constant radial position.

Fig. 5. Rectangular and polar coordinates.



$$r = \sqrt{x_0^2 + y_0^2}$$

$$\theta = arctan\left(\frac{y_0}{x_0}\right)$$

The situation in the phase vocoder is directly analogous. Since the result of the heterodyning and lowpass filtering operations is a pair of sinusoids with a 90-degree phase difference between them, the two parallel paths in the phase vocoder can be viewed equivalently as the rectangular (i.e., horizontal and vertical) coordinates of a single point on a rotating wheel. The position of this point can be represented equally well via polar coordinates.

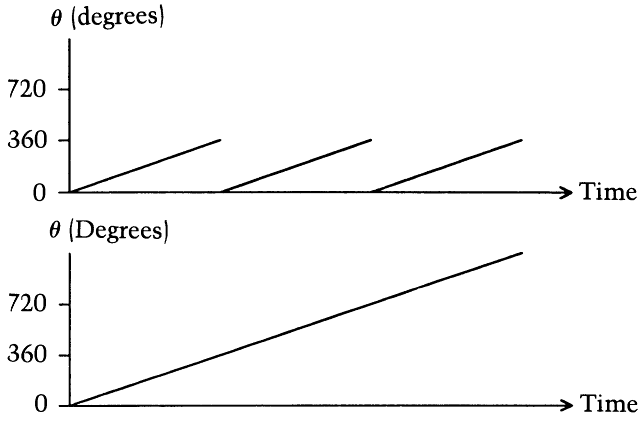
Actually, this choice between rectangular and polar coordinates occurs frequently in signal processing but in a variety of different guises. In the terminology of communications systems (where heterodyning is most frequently encountered), the horizontal and vertical signals are the two parallel heterodyning paths, and the two resulting lowpass-filtered signals are known as the *in-phase* and *quadrature* signals. In the literature of Fourier transforms, the horizontal and vertical signals are known as the *real* and *imaginary* components, and the radial position and angular position are known respectively as *magnitude* and *phase.* (Note that this usage of the term "phase"—to indicate angular position as a function of time—is more general than the usage in which phase is simply the *initial* angular position, as when two signals are said to be "out of phase.") The magnitude-and-phase representation is also the standard for describing filter frequency response, but in this case it is customary to plot only the magnitude, as in Fig. 2.

To actually perform the conversion from rectangular to polar coordinates, we can use the formulas in Fig. 5. The radial position is the hypotenuse of the right triangle with the horizontal and vertical positions as the two sides. In phase vocoder terms, the time-varying radial position is the desired time-varying amplitude (i.e., magnitude). Thus, the amplitude at each point in time is simply the square root of the sum of the squares of the two heterodyned signals. Similarly, the time-varying angular position is the time-varying phase. Thus, the phase at each point in time is the angle whose tangent is (i.e., the arctan of) the ratio of the vertical position to the horizontal position.

But relating the time-varying angular position to the desired time-varying frequency requires an additional operation (see Fig. 3). *Frequency* is the number of cycles which occur during some unit time interval. In terms of the rotating wheel, this can be stated as the "number of revolutions per unit time." Thus, to determine the time-varying frequency we need to determine the rate of rotation of the wheel. To do this, we can simply measure the change in angular position between two successive samples, and divide by the time interval between them. Hence, the frequency at each point in time is the difference between successive angular position values divided by the sample period.

It turns out, though, that there is one additional problem here: the arctan function produces answers only in the range of 0 to 360 degrees. Thus, if we examine successive values of angular position, we may find a sequence such as 180, 225, 270, 315, 0, 45, 90. This may appear to suggest that the instantaneous frequency (i.e., rate of angular rotation) is not constant (because the difference between successive values is not always 45). What has actually happened is that we have gone through more than a single cycle. Therefore, if we want our frequency calculation to work properly, we should really write the sequence as 180, 225, 270, 315, 360, 405, 450. Then the difference between successive values is indeed 45 in all cases. This process of adding in 360 degrees whenever a full cycle has been completed is known as *phase unwrapping.* A comparison of the angular-position signal obtained directly from the rectangular-to-polar conversion and the unwrapped angular position-signal is illustrated in Fig. 6.

Fig. 6. Phase unwrapping.



The operations of phase unwrapping and calculating rate-of-rotation (i.e., frequency) from successive unwrapped angular-position values are the two next-to-last operations in the sequence of steps in Fig. 3. But note that the frequency actually refers only to the difference frequency between the heterodyning sinusoid (i.e., the filter center frequency) and the input signal. (Remember that the rectangular-to-polar conversion and the rate-of-rotation calculation are all performed only on the lowpass-filtered results of the heterodyning operation—e.g., on the 10-Hz component that resulted from "splitting" the 110-Hz component into components at 10 Hz and 210 Hz.) Therefore the very final step in the phase-vocoder-analysis sequence is to add the filter center frequency back in to the instantaneous frequency signal calculated above.

In summary, the internal operation of a single phase vocoder bandpass filter, as diagrammed in Fig. 3, consists of (1) heterodyning the input with both a sine wave and a cosine wave in parallel, (2) lowpass filtering each result, (3) converting the two parallel lowpass-filtered signals from rectangular to polar coordinates, (4) unwrapping the angular-position values, (5) subtracting successive unwrapped angular-position values and dividing by the time interval to obtain a frequency signal, and (6) adding the filter center frequency back in to the frequency signal. This sequence of operations produces the desired time-varying parameter values.

The Fourier-Transform Interpretation

A complementary view of the phase-vocoder analysis is that it consists of a succession of overlapping Fourier transforms taken over finite-duration windows in time. It is interesting to compare this perspective to that of the filterbank interpretation. In the latter, the emphasis is on the temporal succession of magnitude and phase (or frequency) values in a single filter band. In contrast, the Fourier-transform interpretation focuses attention on the magnitude and phase values for all of the different filter bands or *frequency bins* at a single point in time. A graphic representation of this distinction is shown in Fig. 7.

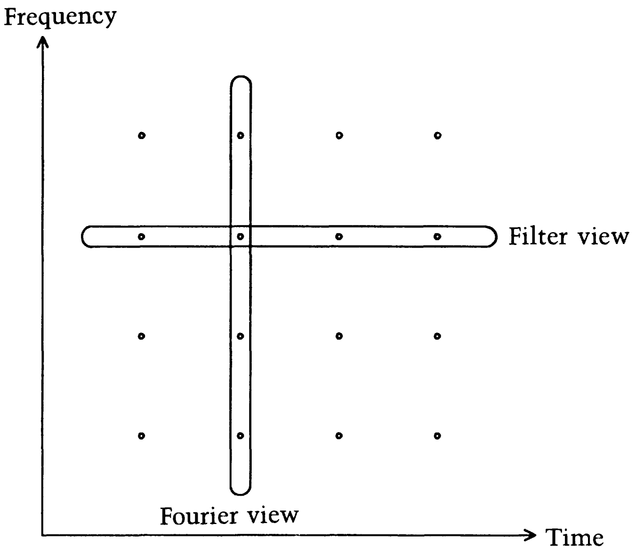


Fig. 7. Filterbank interpretation versus Fourier-transform interpretation.

It is important to understand that each of these viewpoints is really concerned only with the implementation of the bank of bandpass filters. In either case, it is still necessary to convert the filter outputs from rectangular to polar coordinates as described previously. So the real question here is how a sequence of overlapping Fourier transforms can behave like a bank of filters.

Certain features of the Fourier transform have fairly simple interpretations in terms of a filterbank. For example, the number of filter bands is simply the number of bins in the Fourier transform. Similarly, the equal spacing in frequency of the individual filters is a fundamental feature of the Fourier transform. But how does the Fourier view incorporate the shape of the bandpass filters (e.g., the steepness of the cutoff at the band edges) or the in-phase and quadrature signals?

The Fourier transform can be seen as a generalization of the Fourier series. The Fourier series specifies the amplitudes of the various harmonics that must be added together to create a complex periodic waveform. If the waveform is perfectly periodic (i.e., it repeats itself indefinitely with no changes in period or waveshape), it is always possible to synthesize it as the sum of appropriately weighted sinusoids with frequencies that are integer multiples of the fundamental frequency. In musical terms, the individual sinusoids are the *harmonics;* in mathematical terms, they are the *components* of the Fourier series. But since the initial phase of the periodic signal is arbitrary, it generally requires a sum of both sine and cosine components to match this initial phase correctly. Musical discussions of harmonics tend to ignore this subtlety because the initial phase is itself of little perceptual significance. However, when the signal is not perfectly periodic, then the phase can no longer be ignored.

The *discrete short-time Fourier transform* is essentially a means of computing the Fourier series for signals that are not perfectly periodic. The basic idea is that the signal in question can be multiplied by a *window function* so that only a certain section of the signal is nonzero, and the Fourier series can be computed assuming that the windowed section repeats indefinitely. The exact shape of the window turns out to be important because the windowing operation smears the signal's spectrum so that each bin of the Fourier transform (i.e., each component in the Fourier series) includes some energy from other bins nearby. The general rule is that the amount of smearing increases as the window duration gets shorter. Thus, the window function turns out to have exactly the same role as the filter impulse response in the filterbank interpretation of the phase vocoder (e.g., making the impulse response shorter increases the filter overlap or *smearing*).

Similarly, the fact that each bin has both a sine component and a cosine component is equivalent to each filter in the filterbank interpretation having both an in-phase and quadrature signal. The polar representation of these two signals as a rotating vector (i.e., a point on a rotating wheel) allows the precise determination of the time-varying frequency within a single bin; this is done by comparing the orientation of the vector in successive Fourier transforms. For example, when the frequency of a particular signal is exactly equal to that of a particular Fourier transform bin, then successive Fourier transforms will always find the rotating vector for that bin in the same orientation. Successive phase values are thus equal, and their difference is zero. When the input signal frequency does not exactly match the bin frequency, the difference frequency can be calculated from the successive phase values exactly as described in the preceding section.

In summary, the two complementary views of the phase-vocoder analysis can be described as follows. In the filterbank interpretation, the filtering is accomplished by heterodyning and lowpass filtering to produce in-phase and quadrature signals. In the Fourier view, the bank of filters is a set of frequency bins, and the in-phase and quadrature signals are the real and imaginary components of the Fourier transform. In either case, it is still necessary to convert from rectangular to polar coordinates, and then to calculate the difference between successive unwrapped phase values to determine the time-varying frequency within each filter (or bin). We may note that it is precisely this calculation of time-varying frequency that distinguishes the phase vocoder from the earlier channel vocoder. The channel vocoder computes only the time-varying amplitude within each filter (or bin) and makes no attempt to determine the time-varying frequency. In other respects, it is essentially identical to the phase vocoder.

These two differing views of the phase-vocoder analysis suggest two equally divergent interpretations of the resynthesis. In the filterbank interpretation, the resynthesis can be viewed as an additive-synthesis procedure with time-varying amplitude and frequency controls for each of a bank of oscillators. In the Fourier view, the synthesis is accomplished by adding together successive inverse Fourier transforms, overlapping them in time to correspond with the overlapping of the analysis Fourier transforms. (Note, however, that this requires converting back from polar coordinates to rectangular coordinates prior to calculating the inverse Fourier transform.)

Although the filterbank and Fourier views are mathematically equivalent, a particular advantage of the Fourier interpretation is that it leads to the implementation of the filterbank via the fast Fourier transform (FFT) technique. The FFT is an algorithm for computing the Fourier transform with fewer multiplications than would otherwise be required provided that the number of bins is a power of 2. The FFT produces an output value for each of N bins with (on the order of) $N \log_2 N$ multiplications, while the direct implementation of the filterbank requires N^2 multiplications. Thus, the Fourier interpretation can lead to a substantial increase in computational efficiency when the number of filters is large (e.g., if $N = 1024$, the savings is approximately a factor of 100).

It is also the Fourier interpretation that has been the source of much of the recent progress in phase-vocoder-like techniques. Mathematically, these techniques are described as short-time Fourier-transform techniques (Crochiere 1980; Portnoff 1980; 1981a; 1981b; Griffin and Lim 1984). Such algorithms can also be referred to as multirate digital signal processing techniques for reasons that will be made clear later (Crochiere and Rabiner 1983).

Sample-Rate Considerations

The input and output signals to and from the phase vocoder are always assumed to be digital signals with a sampling rate of at least twice the highest frequency in the associated analog signal (e.g., a speech signal with a highest frequency of 5 KHz might be digitized—at least in principle—at 10 KHz and fed into the phase vocoder). However, the sample rates within the individual filter bands of the phase vocoder do not need to be nearly so high. This is most easily understood via the filterbank interpretation.

Within any given filter band, the result of the heterodyning and lowpass-filtering operation is a signal whose highest frequency is equal to the cutoff frequency of the lowpass filter. For instance in the above example, the lowpass filter may only pass frequencies up to 50 Hz. Thus, although the input to the filter was a speech signal sampled at 10 KHz, the output of the filter can be sampled (at least in the ideal case) at as little as 100 Hz without any error. This is true for each of the bandpass filters, because each filter operates by heterodyning a certain frequency region down to the 0–50 Hz region.

In practice, the lowpass filter can never have an infinitely steep cutoff. Therefore, it is generally advisable to sample the output of the filter at four times the cutoff frequency (e.g., 200 Hz) rather than two. Still, this represents an enormous savings in computation (e.g., the filter output is calculated 200 times per second instead of 10,000 times per second).

If we now seek to resynthesize the original input from the phase-vocoder-analysis signals, we face a minor problem. The analysis signals, which in the filterbank interpretation are thought of as providing the instantaneous amplitude and frequency values for a bank of sine-wave oscillators, are no longer at the same sample rate as the desired output signal. Thus, an interpolation operation is required to convert the analysis signals back up to the original sample rate. Even with this interpolation, this is much more computationally efficient than omitting the sample-rate reduction in the first place.

In the Fourier-transform interpretation, an interpolation is also required for resynthesis, but the details are less apparent. In the previous example, where the internal sample rate is only 2% (200 Hz/10000 Hz) of the external sample rate, we simply skip 10000/200 = 50 samples between successive FFTs. As a result, the analysis FFT values are computed only 10000/50 = 200 times per second. It turns out, though, that the process of adding together the successive overlapping inverse FFTs for resynthesis automatically accomplishes the necessary interpolation.

Last, it should be noted that we have so far considered the bandwidth of the output of the lowpass filter without any mention of the conversion from rectangular to polar coordinates. This conversion involves highly nonlinear operations which (at least in principle) can significantly increase the bandwidth of the signals to which they are applied. Fortunately, this effect is usually small enough in practice that it can generally be ignored.

Applications

The basic goal of the phase vocoder is to separate (as much as possible) temporal information from spectral information. The operative strategy is to divide the signal into a number of spectral bands, and to characterize the time-varying signal in each band. This strategy succeeds to the extent that this bandpass signal is itself slowly varying. It fails when there is more than a single partial in a given band, or when the time-varying amplitude or frequency of the bandpass signal changes too rapidly. "Too rapidly" means that the amplitude and frequency are not relatively constant over the duration of a single FFT. Equivalently, a signal is varying too rapidly if the amplitude or frequency changes considerably over durations that are small compared to the inverse of the lowpass filter bandwidth. (Recall that the duration over which the transform is taken is inversely proportional to the lowpass filter bandwidth.)

To the extent that the phase vocoder does succeed in separating temporal and spectral information, it provides the basis for an impressive array of musical applications. Historically, the first of these to be explored was that of analyzing instrumental tones to determine the time-varying amplitudes and frequencies of individual partials. This application was pioneered by Moorer and Grey at Stanford University in the mid 1970s in a landmark series of investigations of the perception of timbre (Grey 1977; Grey and Moorer 1977; Grey and Gordon 1978;

Moorer 1978). (The *heterodyne filter* technique developed by Moorer and used in those investigations is essentially a special case of the phase vocoder.)

More recently, interest in the phase vocoder has focused more on its ability to modify and transform recorded sound materials in musically useful ways. The possibilities in this realm are myriad. However, two basic operations stand out as particularly significant. These are *time scaling* and *pitch transposition.*

Time Scaling

It is always possible to slow down a recorded sound simply by playing it back at a lower sample rate; this is analogous to playing a tape recording at a lower playback speed. But this kind of simplistic time expansion simultaneously lowers the pitch by the same factor as the time expansion. Slowing down the temporal evolution of a sound without altering its pitch requires an explicit separation of temporal and spectral information. As noted above, this is precisely what the phase vocoder attempts to do.

To understand the use of the phase vocoder for time scaling, it is helpful once again to consider the two basic interpretations described above. In the filter bank interpretation, the operation is simplicity itself. The time-varying amplitude and frequency signals for each oscillator are control signals that carry only temporal information. Expanding the duration of these control signals (via interpolation) does not change the frequency of the individual oscillators at all, but it does slow down the temporal evolution of the composite sound. The result is a time-expanded sound with the original pitch.

The Fourier transform view of time scaling is very similar, but with one additional catch. The basic idea is that in order to time-expand a sound, the inverse FFTs can simply be spaced further apart than the analysis FFTs. As a result, spectral changes occur more slowly in the synthesized sound than in the original. But this overlooks a critical detail involving the magnitude and phase signals.

Consider a single bin within the FFT for which the signal within that bin is increasing in phase at a rate of 1/8 cycle (i.e., 45 degrees) per time interval (where the time interval in question is the time between successive FFTs). This means that the successive phase values within the bin are incremented by 45 degrees. Spacing the inverse FFTs further apart means that the 45-degree increase now occurs over a longer time interval. But this means that the frequency of a signal has been inadvertently altered. The solution is to rescale the phase by precisely the same factor by which the sound is being time-expanded. Thus for time expansion by a factor of two, the 45-degree increase should be rescaled to a 90-degree increase, because it occurs over twice the time interval of the original 45-degree increase. This ensures that the signal in any given filter band has the same frequency variation in the resynthesis as in the original (though it occurs more slowly).

The reason that the problem of rescaling the phase does not appear in the filterbank interpretation is that the interpolation there is assumed to be performed on the frequency control signal as opposed to the phase. This is perfectly correct conceptually, but the actual implementation generally conforms more closely to the Fourier interpretation. Also, by emphasizing that the time expansion amounts to spacing out successive "snapshots" of the evolving spectrum, the Fourier view makes it easier to understand how the phase vocoder can perform equally well with nonharmonic material.

To be sure, the phase vocoder is not the only technique that can be employed for this kind of time scaling. A number of time-domain procedures (i.e., not involving filtering or Fourier transforms) can be employed at substantially less computational expense. But from the standpoint of fidelity (i.e., the relative absence of objectionable artifacts), the phase vocoder is by far the most desirable.

Pitch Transposition

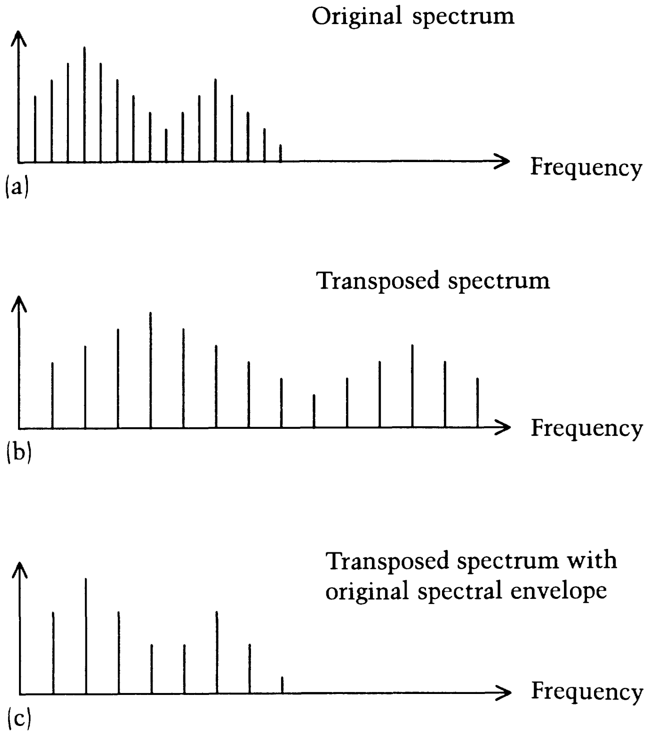
Since the phase vocoder can be used to change the temporal evolution of a sound without changing its pitch, it should also be possible to do the reverse (i.e., change the pitch without changing the duration). Indeed, this operation is easily accomplished. The procedure is simply to time-scale by the desired pitch-change factor, and then to play the resulting sound back at the "wrong" sample rate. For example, to raise the pitch by an octave, the sound is first time-expanded by a factor of two, and the time-expansion is then played at twice the original sample rate. This shrinks the sound back to its original duration while simultaneously doubling all frequencies. In practice, however, there are also some additional concerns.

First, instead of changing the clock rate on the playback digital-to-analog converters, it is more convenient to perform sample-rate conversion on the time-scaled sound via software. Thus, in the above example, we would simply designate a higher sample rate for the time-expanded sound, and then sample-rate convert it down by a factor of two so that it could be played at the normal sample rate. (It is possible to embed this sample-rate conversion within the phase vocoder itself, but this proves to be of only marginal utility and will not be discussed further here.)

Second, upon closer examination it can be seen that only time-scale factors that are ratios of integers are actually allowed. This is clearest in the Fourier view because the expansion factor is simply the ratio of the number of samples between successive analysis FFTs to the number of samples between successive synthesis FFTs. However, it is equally true of the filterbank interpretation because it turns out that the control signals can only be interpolated by factors that are ratios of two integers. Of course, this has little significance for time scaling because, while it may be impossible to find two suitable integers with precisely the desired ratio, the error is perceptually negligible. However, when time scaling is performed as a prelude to pitch transposition, the perceptual consequences of such errors are greatly magnified (by virtue of the ear's sensitivity to small pitch differences), and considerable care may be required in the selection of two appropriate integers.

An additional complication arises when modifying the pitch of speech signals because the transposition process changes not only the pitch, but also the frequency of the vocal tract resonances (i.e., the formants). For shifts of an octave or more, this considerably reduces the intelligibility of the speech. (This same phenomenon occurs in the pitch transposition of nonspeech sounds as well, but for these sounds intelligibility is not an issue. Consequently, the change in sound quality is not nearly so objectionable.) To correct for this, an additional operation can be inserted into the phase-vocoder algorithm as shown in Fig. 8. For each FFT, this additional operation determines the spectral envelope (i.e., the shape traced out by the peaks of the harmonics as a function of frequency), and then distorts this envelope in such a way that the subsequent sample-rate conversion brings it back precisely to its original shape.

Fig. 8. Spectral envelope correction. (a) original spectrum. (b) transposed spectrum. (c) corrected spectrum.



Conclusion

The descriptions previously discussed in this article address only the most elementary possibilities of the phase-vocoder technique. In addition to simple time scaling and pitch transposition, it is also possible to perform time-varying time scaling and pitch transposition, time-varying filtering (e.g., *cross synthesis*), and nonlinear filtering (e.g., noise reduction), all with very high fidelity. The phase vocoder analysis capabilities alone can be extremely useful in applications ranging from psychoacoustics to composition (e.g., individual harmonics of a sound can be separated in space).

Acknowledgments

I am grateful to both Jerry Balzano and Richard Boulanger for numerous helpful suggestions that have considerably enhanced the clarity of this presentation.

References

Crochiere, R. E. 1980. "A Weighted Overlap-Add Method of Fourier Analysis-Synthesis." *IEEE Transactions on Acoustics, Speech, and Signal Processing* ASSP-28(1): 55–69.

Crochiere, R. E., and L. R. Rabiner. 1983. *Multirate Digital Signal Processing.* Englewood Cliffs: Prentice-Hall.

Flanagan, J. L., and R. M. Golden. 1966. "Phase Vocoder." *Bell System Technical Journal* 45:1493–1509.

Grey, J. M. 1977. "Multidimensional Perceptual Scaling of Musical Timbres." *Journal of the Acoustical Society of America* 61:1270–1277.

Grey, J. M., and J. A. Moorer. 1977. "Perceptual Evaluations of Synthesized Musical Instrument Tones." *Journal of the Acoustical Society of America* 62:454–462.

Grey, J. M., and J. W. Gordon. 1978. "Perceptual Effects of Spectral Modifications on Musical Timbres." *Journal of the Acoustical Society of America* 63:1493–1500.

Griffin, D. W., and J. S. Lim. 1984. "Signal Estimation from Modified Short-Time Fourier Transform." *IEEE Transactions on Acoustics, Speech, and Signal Processing* ASSP-28(2):236–242.

Moorer, J. A. 1978. "The Use of the Phase Vocoder in Computer Music Applications." *Journal of the Audio Engineering Society* 24(9):717–727.

Portnoff, M. R. 1980. "Time-Frequency Representation of Digital Signals and Systems Based on Short-Time Fourier Analysis." *IEEE Transactions on Acoustics, Speech, and Signal Processing* ASSP-28(1):55–69.

Portnoff, M. R. 1981a. "Short-Time Fourier Analysis of Sampled Speech." *IEEE Transactions on Acoustics, Speech, and Signal Processing* ASSP-29(3):364–373.

Portnoff, M. R. 1981b. "Time-Scale Modification of Speech Based on Short-Time Fourier Analysis." *IEEE Transactions on Acoustics, Speech, and Signal Processing* ASSP-29(3):374–390.

Appendix

While there is much to be said for a nonmathematical description of the phase vocoder, there is also a limit to what can be understood in this fashion. In this Appendix I provide a brief mathematical restatement of the foregoing description.

Mathematically, the short-time Fourier transform $X(n,k)$ of the signal $x(n)$ is a function of both time n and frequency k. It can be written as

$$X(n,k) = \sum_{m=-\infty}^{\infty} x(m)h(n-m)e^{-j\frac{2\pi}{N}km} \qquad (1)$$

where $h(n)$ is a window or filter impulse response as described below.

The short-time Fourier transform is the analysis portion of the analysis-synthesis procedure. A general resynthesis equation is then given by

$$x(n) = \sum_{m=-\infty}^{\infty} f(n-m)\frac{1}{N}\sum_{k=0}^{N-1} X(m,k)e^{j\frac{2\pi}{N}km} \qquad (2)$$

where $f(n)$ is another window or filter impulse response. In the absence of modifications (and with certain constraints on $h(n)$ and $f(n)$), this equation reconstructs the input perfectly.

The analysis equation can be viewed from either of two complementary perspectives. On the one hand, Eq. (1) can be written as

$$X(n,k) = \sum_{m=-\infty}^{\infty} \left(x(m)e^{-j\frac{2\pi}{N}km}\right)h(n-m). \qquad (3)$$

This describes a heterodyne filterbank. The kth filter channel is obtained by multiplying the input $x(m)$ by a complex sinusoid at frequency k/N times the sample rate. This shifts input frequency components in the vicinity of k/N times the sample rate down near 0 Hz (and also up near twice k/N times the sample rate). The resulting signal is then convolved with the lowpass filter $h(m)$. This removes the high-frequency components and leaves only those input frequency components originally in the vicinity of k/N times the sample rate (albeit shifted down to low frequency). Thus the output $X(n,k)$, for any particular value of k, is a frequency-shifted, bandpass-filtered version of the input.

On the other hand, Eq. (1) can be regrouped as

$$X(n,k) = \sum_{m=-\infty}^{\infty} \left(x(m)h(n-m)\right) e^{-j\frac{2\pi}{N}km} \qquad (4)$$

This is the expression for the discrete Fourier transform of an input signal $x(m)$ which is multiplied by a finite-duration, time-shifted window $h(n - m)$. Thus the output $X(n,k)$, for any particular value of n, is the Fourier transform of the windowed input at time n. Instead of representing the output of a bank of parallel heterodyne filters, $X(n,k)$ now represents a succession of partially-overlapping Fourier transforms.

Computationally, this latter view is particularly significant because the transform can be implemented as a fast Fourier transform (FFT). To see this, we can note that if the nonzero extent of the window is N samples or fewer (i.e., $h(n) = 0$ for $|n| > N/2$), then Eq. (4) is simply

$$X(n,k) = \sum_{m=n-\frac{N}{2}}^{n+\frac{N}{2}-1} x(m)h(n-m) e^{-j\frac{2\pi}{N}km} \qquad (5)$$

With a change of variable ($i = m - n + N/2$), this can be rewritten as

$$X(n,k) = e^{-j\frac{2\pi}{N}k(n-N/2)} \sum_{i=0}^{N-1} x(i+n-N/2)h(-i+N/2) e^{-j\frac{2\pi}{N}ki} \qquad (6)$$

and the summation is seen to have the proper form for an FFT implementation (i.e., a sum from $i = 0$ to $i = N - 1$ of terms $x(i)\exp(-j2\pi ki/N)$). The exponential phase term preceding the summation is important to include, but it is equivalent to shifting the time origin for the terms within the summation. Hence, a cyclic rotation of the windowed signal prior to Fourier transforming eliminates the need for an explicit multiplication by the exponential.

As it turns out, this procedure is equally applicable in the case where the window duration is greater than N. The summation in Eq. (6) then assumes the general form

$$\sum_{i=0}^{M} y(i) e^{-j\frac{2\pi}{N}ki} \qquad (7)$$

where $M > N$. But the exponential terms involving $i > N$ (e.g., $i = N + 3$) are identical to the corresponding terms involving $i < N$ (e.g., $i = 3$) because (letting $i = N + l$)

$$e^{-j\frac{2\pi}{N}k(N+l)} = e^{-j\frac{2\pi}{N}kN} e^{-j\frac{2\pi}{N}kl} = e^{-j2\pi k} e^{-j\frac{2\pi}{N}kl} \qquad (8)$$

Since $e^{-j2\pi k} = 1$ for integer values of k, the summation reduces to

$$\sum_{l=0}^{N-1} \left\{ \sum_{r=0}^{rN \leq M} y(rN + l) \right\} e^{-j\frac{2\pi}{N}kl} \qquad (9)$$

Thus, even when the window is longer than the desired FFT size, the FFT formulation can still be used provided that the windowed signal is appropriately stacked and added (i.e., successive temporal segments are overlayed and added together).

Using a window duration greater than N samples, though, does entail some additional complication. First, it turns out that the condition for perfect resynthesis is that $h(n) = 0$ for $n = lN$ (where l is any nonzero integer). This says that the window must be zero at integer multiples of N. If the window is fewer than N samples to begin with, then this condition is trivially satisfied. If the window is greater than N samples in duration, then an easy way to satisfy the condition is to make the actual window the product of the desired window and a $\sin(x)/x$ function with period N.

Second, the duration of the analysis window $h(n)$ is an important factor in selecting an appropriate synthesis window $f(n)$. If the analysis window is shorter than N samples in duration, then it can be shown that an optimal synthesis window is simply an amplitude-scaled copy of the analysis window. If the analysis window duration is greater than N samples, then it is easier to think of the synthesis window as the impulse response of a filter that is performing interpolation. Then a good choice is to make the synthesis window the product of some simple window and a $\sin(x)/x$ function with period I (where I is the number of samples between the beginnings of successive Fourier transforms).

Additional insight into the role of the analysis window can be gained by considering a narrowband input signal to the phase vocoder given by

$$x(n) = A(n)\cos\left(\frac{2\pi}{N}knT + \theta(n)\right). \qquad (10)$$

If both the amplitude $A(n)$ and the instantaneous phase $\theta(n)$ are slowly varying, then it can be shown that the phase-vocoder analysis for the filter centered at frequency $2\pi k/N$ results in

$$|X(n,k)| = |\sum_{m=-\infty}^{\infty} A(m)h(n-m)e^{j\theta(m)}| \qquad (11)$$

$$arg[X(n,k) = arg[\sum_{m=-\infty}^{\infty} A(m)h(n-m)e^{j\theta(m)}] \qquad (12)$$

where $|X(n,k)|$ and $arg[X(n,k)]$ are, respectively, the magnitude and phase of $X(n,k)$.

Ideally Eq. (11) would simply state $|X(n,k)| = A(n)$. This would mean that the phase-vocoder analysis had perfectly extracted the amplitude of the input signal. In reality, though, we see that the phase-vocoder estimate of the amplitude is a smeared version of the true amplitude. If the input signal has a constant frequency of exactly $2\pi k/N$, then $\theta(n) = 0$ for all n, and the amplitude estimate $|X(n,k)|$ is precisely a lowpass-filtered version of the true amplitude $A(n)$. If the input signal has a constant frequency not quite equal to the filter center frequency, then the amplitude estimate is attenuated by the gain of the lowpass filter at the difference frequency.

The situation for the instantaneous phase estimate is similar but less conducive to a simple interpretation. It turns out, though, that the phase estimate is also smeared by a lowpass-filtering operation so that, for example, a sudden change in the input signal frequency results in a more gradual change in the instantaneous frequency estimate of the phase vocoder.

The situation with phase is further complicated by the fact that when two sinusoids of different frequencies lie within the same filter bandpass, the composite signal has 180-degree jumps in phase whenever the composite amplitude goes through zero. This is a consequence of the fact that

$$\cos(A) + \cos(B) = \cos((A - B)/2)\cos((A + B)/2). \qquad (13)$$

The slowly varying $\cos((A - B)/2)$ can be thought of as the amplitude envelope modulating the more rapidly varying $\cos((A + B)/2)$. But the phase-vocoder amplitude estimate is always positive. Hence, when $\cos((A - B)/2)$ changes sign, the amplitude estimate remains unchanged, and the phase estimate jumps 180 degrees.

In the past two years, considerable progress has been made on short-time Fourier analysis-synthesis formulations that do not require an explicit phase calculation (Griffin and Lim 1984). These magnitude-only techniques are potentially superior to the phase vocoder for applications such as time scaling because these new techniques are based on a mathematical formulation that guarantees an optimal result (i.e., some measure of error between actual and desired resyntheses is minimized).

In contrast, the phase-vocoder approach to time-scaling is basically a heuristic one. The phase-vocoder modifications to the short-time Fourier transform do not actually result in a valid short-time Fourier transform. The modified transform can still be inverted to accomplish resynthesis, but the short-time Fourier transform of the resynthesized signal is not the same as the transform from which it was synthesized. Nevertheless, the phase vocoder remains a very powerful tool for sound manipulation, and an understanding of the phase vocoder remains as the fundamental step from which future innovations can proceed.