The Phase Vocoder

January 19, 2017

This paper follows the development of Reiss and McPherson [1].

Introduction

Most audio effects are performed in the time domain. This report focuses on a group of audio effects which are performed in the frequency domain. The phase vocoder is a term used to describe these audio effects. The phase vocoder process consists of the following steps.

1. Select the first N samples of the input signal which may have arbitrary length. These N samples are the first frame.
2. Multiply the frame by a window function of length N.
3. Apply a fast Fourier transform (FFT) of length M≥N to the windowed signal.
4. Perform the particular phase vocoder effect to frequency domain signal.
5. Perform an inverse fast Fourier transform IFFT on the processed frequency domain signal.
6. Add the M output samples to the output buffer.
7. Hop to the next frame by hop size H which is typically less than N as the overlap add method is uses to stitch the frames into one uninterrupted signal.

The windowing, the FFT, IFFT, and overlap add method are not covered in detail in this report. The focus will be on effects that can be performed using the phase vocoder technique. In particular robatization, whisperization, time scaling, and pitch shifting will be discussed.

Robotization and Whisperization

Robotization and whisperization are instructive, if not musical, effects. Both these effects are intended to operate on a voice and operate by modifying the phase in each bin of the frequency domain signal.

Robotization gets its name from the monotone robot like sound it results in. In step 4 of the above procedure the phase in each bin is set to zero while the magnitude is left alone. Since magnitude is unchanged, the overall shape of the spectrum is also unchanged, thus the vocal formants are preserved and the voice is still intelligible. However, by setting the phase in each bin to zero each frequency component will restart from zero phase with each frame rather than having a smooth continuous transition between frames. The result is a constant pitch related to the hop size between frames where $f_{robot} = f_s/H$ where f_s is the sample rate and H is the hop size. The quality of the effect is also dependent on the window size. For longer windows the resetting of phase is less frequent hence the effect is less noticeable, and with very short windows the clarity of the output is reduced making the voice hard to understand. A medium sized window on the order of 256-1024 samples yields the best results.

While robotization attempts to regulate the pitch of a voice, whisperization attempts to remove any sense of pitch at all. Whisperization is implemented in almost the same way as robotization but the phase is set to a random value between 0 and 2π. A random phase is selected for each bin in the frame resulting in the loss of any sense of periodicity. Again the magnitude is unaffected in order to preserve vocal formants. For whiserpization shorter frames on the order of 64-256 samples work best.

Because both robotization and whisperization take advantage of artifacts of the phase vocoder process the constant overlap-add criterion for choosing the hop size in relation to the window size is not important.

Time Scaling and Pitch Shifting

A useful application of the phase vocoder is scaling time or shifting pitch while leaving the other unaffected. First we will look at time scaling. In order to perform time scaling without affecting pitch the relationship $\Phi=\omega t$ or phase = frequency multiplied with time. By allowing phase to be adjust time can be changed while leaving frequency unaffected. For time scaling the first 3 steps of the phase vocoder are performed as usual but steps 5 through 7 are modified to accommodate the change in time. A stretching ration R relates the hop size for the analysis steps (1-3) and the synthesis steps (5-7) denoted as h_a and h_s respectively such that $h_s = Rh_a$. The change in frequency and amplitude for each bin is calculated by the following equations. These phase and amplitude increments are proportional to the instantaneous frequency. The variable h_a is the hop size for the analysis portion of the algorithm.

$$\Delta\Phi_i(n,k) = \frac{\omega_k h_a + arg(\Phi_d(n,k))}{h_a}\omega\Delta$$

$$\Delta \mathrm{A}_i(n,k) = \frac{A_i(n,k) - A_i(n-1,k)}{h_a}\omega\Delta$$

These phase and amplitude increment values will be used in the synthesis part of the algorithm. The phase for the re-synthesis stage can be calculated at each sample m using

$$\theta(m,k) = \theta(m-1,k) + \Delta\Phi_i(n,k)$$

The phase for the synthesis or output signal is incremented by the same amount as was calculated above for analysis but for the synthesis hop size. The same method cannot be used to scale the amplitude because it would lead to serious artifacts. Instead the increment is calculated using the synthesis hop size to ensure that the amplitude will be the same at the beginning and end of the hops in both synthesis and analysis.

$$\Delta \mathrm{A}_{is}(n,k) = \frac{\mathrm{A}_i(n,k) - \mathrm{A}_i(n-1,k)}{h_s}$$

The amplitude over the synthesis hop is then calculated.

$$A_k(m,k) = A_k(m-1,k) + \Delta \mathrm{A}_{ks}(n,k)$$

The sinusoidal components are then summed to synthesize the output which has been scaled in time.

Pitch shifting without changing the time of the signal can be used leveraging the above algorithm for time scaling. Suppose a time scaling factor of R is applied to each block of N samples of input resulting in R**.**N output samples. If these samples were played at a sample rate of R**.**$\mathrm{f_s}$the result would be a higher pitched version of the signal with the same time duration. In practice it doesn't make a lot of sense to have a different sample-rate for input and output. An alternative is to achieve a similar result using interpolation to fit R**.**N samples in the space of N.

Phase Vocoder Artifacts

The effects mentioned above often rely on linear interpolation. This presents a problem because so many elements of music are non-linear. Problems also arise with the distortion of phase relationships because phase error is cumulative. A note being struck can be considered a transient. Transient smearing occurs when phase coherence is lost at a transient, the result is transient smearing which softens the attack of the note and gives it an unnatural sound. A third problem is "phasiness" which adds a reverb-like quality to the sound.

Conclusion

The phase vocoder is a fairly simple yet powerful tool to manipulate frequency components of an audio signal. It is tempting initially to focus on the magnitude of a signal but it becomes clear that the phase of the signal, and how it is affected, has huge implications on the quality of the overall effect.

References

[1] Reiss, Joshua D., and Andrew McPherson. Audio effects: theory, implementation and application. CRC Press, 2014.