Reunderstand PSOLA

Kanru Hua

December 2015

1 Introduction

PSOLA (Pitch Synchronous OverLap and Add)[1], an algorithm able to change the duration or shift the pitch of speech signals, invented around 1990 (arguably) by France Telecom, remains being *the* most widely used speech modification algorithm as of 2015. Staying in time-domain from the beginning to its end, this incredibly simple algorithm outputs speech at pretty good quality and blazingly fast speed. Given such a pleasant compromise between complexity, speed and quality, no wonder those top-tier softwares (e.g. Melodyne® and Auto-Tune®) in music production industry are using PSOLA as their 'secret' weapon to conquer the market; no wonder those top-tier universities all over the world more or less put PSOLA in their speech processing courses.

What leaves me wondering is: seems like there's always a blind spot in all tutorials, slides and papers about PSOLA. In a few sentences they tell you something like, [1]

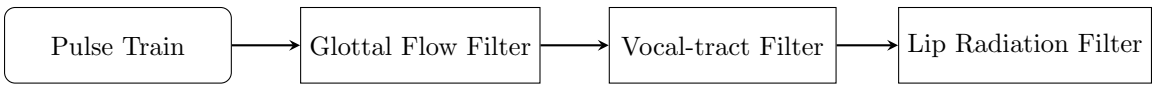
> PSOLA basically marks the "epochs" (some call them Glottal Closure Instants or Instants of Significant Excitation), puts a bunch of 2-period-long hanning windows on these epochs, cuts the signal into pieces, and finally adds the pieces back together (at some different epoch positions determined by pitch and time stretch).

Now you can clutch onto your favourite editor, implement the whole thing in a few hundred lines. They've taught you *how does it work*, but they hide the more intriguing fact from you - *what does this mean?*, or they themselves either don't know what PSOLA does in frequency domain.

The goal of this quite informal article is to fill in the blank, i.e., to gain a deep understanding of PSOLA, discover a few interesting properties, and see how to ultimately improve the algorithm using these properties.

2 PSOLA as a Source-Filter Model

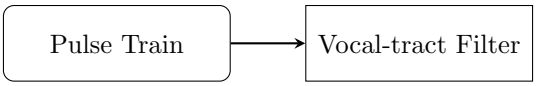
You may have heard of this a hundred times, but for completeness I still have to mention it again. The very classical source-filter model of speech production is an acoustic simplification of the actual, physical process of passing quasi-periodic volume velocity flow through vocal-tract and going out from lip. After making a great deal of (stupid) assumptions such as the organs are stationary during phonation and the whole process is linear (which of course is not true), the speech production process can be expressed as a pulse train[2] going through a chain of cascaded filters resembling each organ,



An even simpler model is achieved by combining the three filters into one,

[1] A much more detailed yet easy-to-understand video introduction can be found on Professor Simon King's website, `http://www.speech.zone/td-psola-the-hard-way/`

[2] A pulse train is a bunch of Dirac Delta functions shifted and added together.



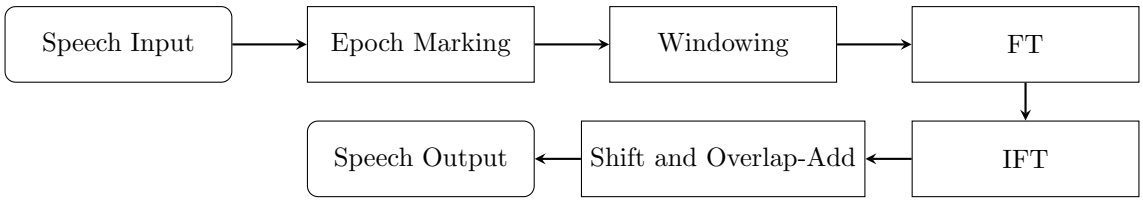
which suggests that your vocal fold suddenly bursts open and closes again in no time, being an even more ridiculous assumption. Afterall, only by making these assumptions the model becomes tractable, and can be implemented on a digital computer. Some recent researches are trying to narrow down the assumptions, but that's out of the scope for our discussion.

Back to PSOLA, consider the process of extracting lots of pieces, each covering 2 periods of speech signal, and adding them back. This perfectly fits into the simplified source-filter model, **where the vocal-tract filter's impulse responses, slowly varying along time, are the pieces extracted**. In time-domain, filtering corresponds to convolution, and convolution with a pulse train is equivalent to shifting and adding the impulse response onto each pulse (or epoch) position.

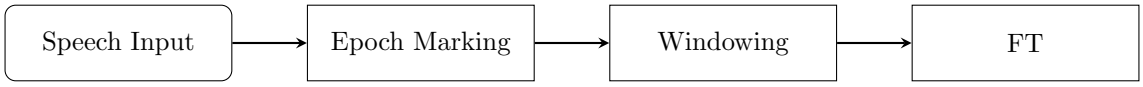
This further implies that the analysis procedure - marking the epochs and windowing the signal - is to **deconvolve** the speech signal into separated source (pulse train) and filter (vocal-tract filter) component!

3 PSOLA from a Frequency Domain Perspective

Now let's take a step further by examining PSOLA and the underlying source-filter model in frequency domain. An interesting way of doing this is to insert a pair of Fourier Transform and Inverse Fourier Transform[3] into the middle,



This is actually a variant known as FD-PSOLA (Frequency-Domain PSOLA), which allows you to carry out frequency domain filtering between FT and IFT blocks in the above flowchart. Without such filtering, obviously the insertion of FT/IFT doesn't change anything in the result. But from now on we can split the algorithm into two parts: the analysis part,



and the synthesis part,

In the following we tackle down each part respectively. It's good to start with the analysis part.

[3]When dealing with discrete signals, one can use Discrete Time Fourier Transform, or Discrete Fourier Transform in a more practical sense. Note that we're doing the transform for each windowed piece, in a way similar to STFT, but being pitch-synchronous.

3.1 Analysis Subprocess

We shall first review the concept of narrowband and wideband spectrum. These two terms have multiple definitions regarding to different contexts. In speech processing, a narrowband spectrum displays well-resolved harmonic structure (for voiced speech) while a wideband spectrum only shows a rough contour as if joining all harmonics by a smooth curve.

PSOLA's window is only two periods long so we can safely assume that the fundamental frequency is constant around each analysis instant. Since the analysis window fades to zero on both sides, it doesn't matter if the pitch outside of the window changes or not, so we can still consider the constant frequency and infinite length pulse train as the excitation signal. The Fourier Transform of such a pulse train is also a pulse train in frequency domain, exhibiting a harmonic structure,

$$\sum_n \delta(t - nT_0) \leftrightarrow \frac{2\pi}{T_0} \sum_n \delta(\omega - \frac{2\pi n}{T_0}) \tag{1}$$

When filtered by the vocal-tract filter it becomes,

$$v(t) * \sum_n \delta(t - nT_0) \leftrightarrow \frac{2\pi}{T_0} \sum_n V(\omega)\delta(\omega - \frac{2\pi n}{T_0}) \tag{2}$$

where $v(t)$ and $V(\omega)$ are the impulse response and frequency response of the vocal-tract filter, respectively. This is a very, very narrow-band spectrum since each harmonic is infinitely narrow and super nicely resolved.

Now consider the effect of windowing. Modulating (multiplying) the infinite length signal by a finite length window is to convolve the spectrum of two signals in frequency domain. When one of the spectrums is a pulse train, this becomes shifting and adding the window's spectrum onto each pulse. Here the vocal-tract frequency response is reduced to a vector of phasors V_n since equation (2) is only non-zero on each harmonic,

$$w(t)\left(v(t) * \sum_n \delta(t - nT_0)\right) \leftrightarrow \frac{2\pi}{T_0} \sum_n V_n W(\omega - \frac{2\pi n}{T_0}) \tag{3}$$

The magical power behind PSOLA lies in the use of 2-period-long hanning window in its analysis part. It has a main-lobe radius of $\frac{2\pi}{T_0}$ radians[4], which exactly fills in the gap between neighbouring harmonics ($\omega - \frac{2\pi n}{T_0}$ in equation (1)-(3)). This guarantees that harmonics are not resolved and the spectrum is hence wideband. In summary, PSOLA's analysis subprocess is a wideband spectrum estimation method[5] whose output is smooth in both magnitude and phase.

3.2 Synthesis Subprocess

The synthesis part simply consists of two linear operations: inverse Fourier Transform, and shift and overlap-add. We can swap the order of execution by first shifting and adding in frequency domain and then transforming to time domain. Given the estimated wideband spectrum $V(\omega)$, the output $y(t)$ is expressed in a manner similar to equation (1)-(3). Again, we can assume the fundamental being constant because it hardly changes within an analysis frame.

[4]Professor Julius O. Smith has written a nice book[2] covering some topics on window properties.

[5]Some call this "Spectral Envelope Estimation" but it mostly refers to magnitude response estimation only.

$$y(t) = \mathrm{F}^{-1}\{\sum_n V(\omega)e^{jn\omega T_0}\} \tag{4}$$

$$Y(\omega) = V(\omega)\sum_n e^{jn\omega T_0} \tag{5}$$

$$\hat{Y}(\omega) = V(\omega)\lim_{N\to\infty}\frac{1}{N}\sum_n^N e^{jn\omega T_0} = V(\omega)|_{\omega=\frac{2\pi n}{T_0}} \tag{6}$$

The right hand side of equation (5) is the summation of many harmonically-related complex sinusoids. When $\omega \neq \frac{2\pi n}{T_0}$, the complex values cancel out to give a zero, otherwise it pops up to infinity. By normalizing $Y(\omega)$ by the range of summation, the normalized output spectrum equals to $V(\omega)$ at harmonic frequencies. In other words, $V(\omega)$ is subsampled at harmonic frequencies, generating a narrowband spectrum $\hat{Y}(\omega)$.

3.3 Preliminary Conclusions

So far we've examined both the analysis and synthesis part. By cascading them together, PSOLA first *implicitly* estimates an array of wideband spectrum from the input signal, at each analysis instant synchronized to period; then it *implicitly* reconstructs narrowband spectrum by subsampling the wideband spectrum at a modified fundamental frequency. Without pitch modification (i.e. time-scale modification only), the narrowband spectrum can be perfectly preserved throughout the process.

In the context of source-filter model, the wideband spectrum estimated from the input represents the vocal-tract transfer function. Note that not only magnitude but also phase response of the vocal-tract are estimated, being different from most spectral envelope estimators (e.g. STRAIGHT, SEEVOC) which only consider the magnitude component. However, PSOLA has a critical flaw that the vocal-tract impulse response is time-limited to 2 periods. When the pitch is shifted down by less than half, the separation between two pulses becomes longer than the impulse response, and a small region in the middle is left blank.

4 Interference Issues and Epoch Marking

In section 3.1, equation (3) assumes that both the pulse train and the window are centered at time zero. In a more realistic case, we need to add a time shift representing the location of epoch or analysis instant.

$$w(t)\left(v(t) * \sum_n \delta(t - nT_0 - \Delta T)\right) \leftrightarrow \frac{2\pi}{T_0}\sum_n e^{-j\Delta T\omega}V_n W(\omega - \frac{2\pi n}{T_0}) \tag{7}$$

Note that V_n is a phasor which follows the definition $V_n = a_n e^{j\theta_n}$; the spectrum of a symmetric window is real. Thus the phase component for each harmonic in the summation is $\theta_n - \Delta T\omega$.

Here arises an interference problem in the wideband spectrum: when neighbouring harmonics have different phases, the overlapping spectral content in the middle is attenuated. This is depicted in Figure 1 where a 9-period speech signal is synthesized from the wideband spectrum (red) at 300Hz. In the left figure the synthetic signal is windowed at a local extrema; in the right figure the window is shifted by 20% of the period. A new wideband spectrum (green) is calculated from the windowed signal. In the right figure we can find a few valleys in inter-harmonic regions spanning from 1kHz to 4.5kHz.

This example shows that epoch marking has a critical effect on the quality of wideband spectrum, and the undesirable valleys can be prevented, to some extent, by centering the window at a local maximum of absolute value of the signal. Indeed, time-domain peak picking is a commonly used epoch marking method for PSOLA[6].

[6]The PSOLA implementation in Praat uses local extrema based epoch marking ("Sound - To manipulation...").

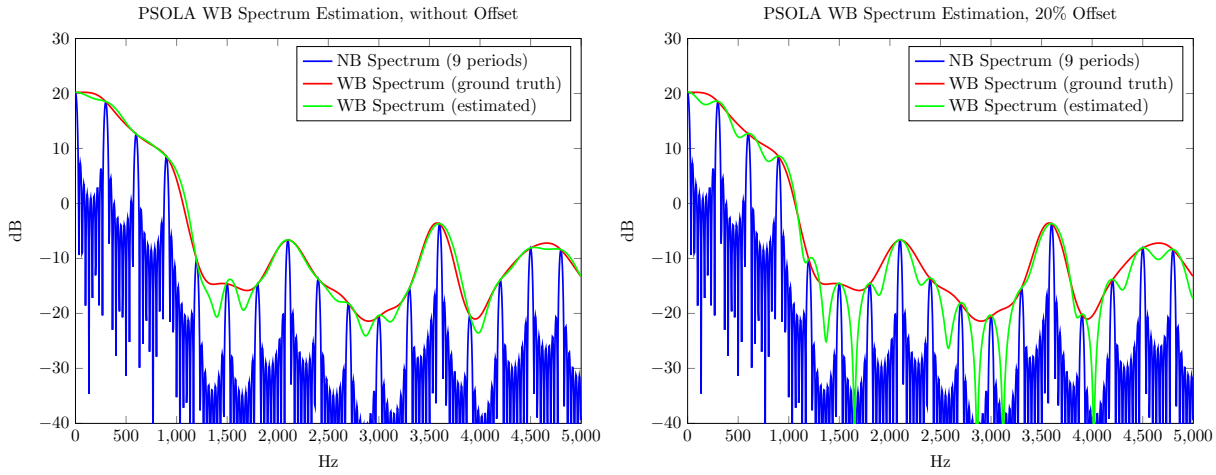


Figure 1

To show why peak picking works, we can consider an alternative time-domain representation of equation (2), taking account of time shift,

$$\frac{2\pi}{T_0}\sum_{n=-\infty}^{\infty} e^{-j\Delta T\omega}V(\omega)\delta(\omega-\frac{2\pi n}{T_0}) \leftrightarrow \frac{2\pi}{T_0}\sum_{n=\infty}^{\infty} V_n e^{j2\pi n\frac{t-\Delta T}{T_0}} \tag{8}$$

$$=\frac{4\pi}{T_0}\sum_{n=0}^{\infty} a_n\cos(2\pi n\frac{t-\Delta T}{T_0}+\theta_n) \tag{9}$$

$$=\frac{4\pi}{T_0}\sum_{n=0}^{\infty} a_n\cos(2\pi n\frac{t}{T_0}+\theta_n-2\pi n\frac{\Delta T}{T_0}) \tag{10}$$

which is a harmonic model. The phase shift of the n-th harmonic is $\theta_n - 2\pi n\frac{\Delta T}{T_0}$, in accordance with the phase component in equation (7). When $\theta_n - 2\pi n\frac{\Delta T}{T_0} = 0$ or $\pi\ \forall n$, the waveform sufficiently has a sharp local maximum or minimum at $t = 0$. Although having a local extrema at time zero doesn't in turn imply perfect phase alignment, the shape of time-domain waveform and phase coherency are still highly correlated.

4.1 Optimal Epoch Marking Method

Our next question is how to design an epoch marking method that minimizes the phase interference, i.e., how to find ΔT such that phase differences for neighbouring harmonics are minimized. Once the question is stated, its answer becomes obvious,

$$\Delta T^* = \underset{\Delta T}{\operatorname{argmin}}\sum_n \left|\theta_n - 2\pi n\frac{\Delta T}{T_0} - \theta_{n-1} + 2\pi(n-1)\frac{\Delta T}{T_0}\right|_{\text{mod } 2\pi} \tag{11}$$

$$= \underset{\Delta T}{\operatorname{argmin}}\sum_n \left|\theta_n - \theta_{n-1} - 2\pi\frac{\Delta T}{T_0}\right|_{\text{mod } 2\pi} \tag{12}$$

which requires prior knowledge of θ_n, which can be measured from the harmonic peaks of a narrowband spectrum. This coincides with MFPA (Maximally Flat Phase Alignment) algorithm[3] for sinusoidal modelling, as a pre-processing step before sinusoidal analysis. To implement MFPA, we first generate a bunch of candidate ΔT uniformly sampled from range $[-\frac{T_0}{2}, \frac{T_0}{2}]$, then directly evaluate the sum of absolute difference modulus 2π and pick the candidate corresponding to the smallest sum.

In comparison with Figure 1, Figure 2 shows the wideband spectrum calculated at a position predicted by MFPA algorithm. Valleys around 1.5kHz and 3kHz are successfully reduced in the later plot.

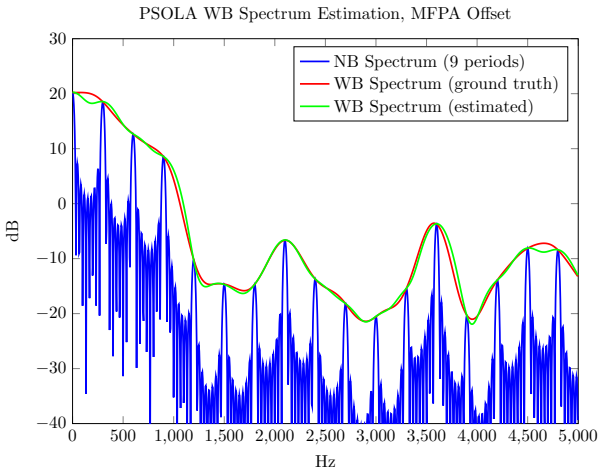


Figure 2

4.2 When Everything Fails

Looking carefully at Figure 2, we still find small mismatches between the ground truth and the estimated spectrum. MFPA is only able to minimize the phase difference, but in most of the cases it's impossible to achieve zero phase difference, except when the vocal-tract filter is zero phase (which is also impossible). So PSOLA more or less introduce a distortion when pitch is modified, and the distortion depends on speaker's characteristics.

I'd like to conclude this article with the following figure - PSOLA wideband spectra of synthetic speech generated from a zero-phase filter and a random-phase filter. The left plot shows the most ideal case where the minimal phase difference can be zero. On contrary, in the right, the estimated wideband spectrum poorly matches the ground truth, despite the fact that interference has been minimized.

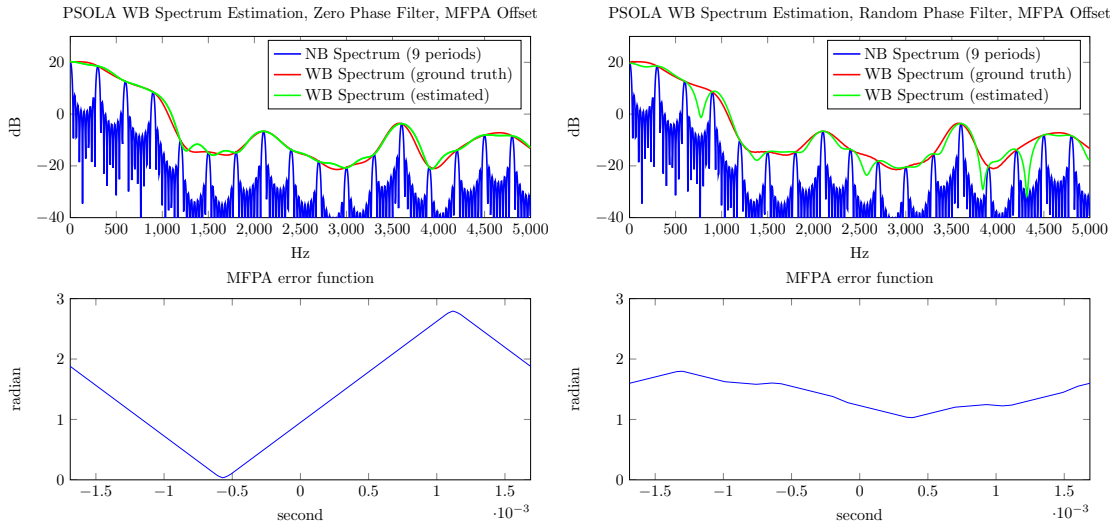


Figure 3

We've been using synthetic signal throughout this article, but the same method and conclusion apply for real life speech signals, provided that the fundamental frequency doesn't change so abruptly.

References

[1] Moulines, Eric, and Francis Charpentier. "Pitch-synchronous waveform processing techniques for text-to-speech synthesis using diphones." Speech communication 9.5 (1990): 453-467.

[2] Smith, Julius O. "Spectral Audio Signal Processing." W3K Publishing. ISBN 978-0-9745607-3-1.

[3] Bonada, Jordi. "High quality voice transformations based on modeling radiated voice pulses in frequency domain." Proc. Digital Audio Effects (DAFx). 2004.