PyTSMod: A Python Implementation of Time-Scale Modification Algorithms

Sangeon Yong **Soonbeom Choi** **Juhan Nam**

Graduate School of Culture Technology, KAIST, South Korea

{koragon2, cjb3549, juhan.nam}@kaist.ac.kr

ABSTRACT

Time-scale modification (TSM) is a digital audio effect that adjusts the length of an audio signal while preserving its pitch. The TSM audio effect is widely used in not only sound production but also music and audio research such as for data augmentation. In this paper, we present PyTSMod, an open-source Python library that implements several different classical TSM algorithms. We expect that PyTSMod can help MIR and audio researchers easily use the TSM algorithms in the Python-based environment.

1. INTRODUCTION

TSM is a popularly used digital audio effect that adjusts the length of audio waveforms in the time domain. The goal of TSM is to shorten or lengthen various sound sources naturally in time. There are many TSM algorithms designed to be suited for different sound sources. For example, pitch synchronous overlap-add (TD-PSOLA) is specialized for monophonic speech sounds [1] and phase vocoder based TSM [2] is more effective for preserving the phase continuity of polyphonic sounds. Despite the long-term research and use of TSM algorithms, there is currently a lack of high-quality TSM algorithm packages written in Python, which is a mainstream programming language for music and audio research. To address the issue, we present PyTSMod, a Python package that implements several different types of TSM algorithms. The source code is available at this link [1] .

2. TSM ALGORITHMS

The PyTSMod package includes the following TSM Algorithms.

2.0.1 Overlap-Add (OLA)

OLA is the simplest TSM algorithm that changes the length of the signal through modifying the hop size between analysis and synthesis frames. This approach is most intuitive and has influenced other TSM algorithms, but on the contrary, the quality is very low because it does not consider the characteristics of the signal at all.

2.0.2 Pitch-Synchronous Overlap-Add (TD-PSOLA)

TD-PSOLA [1] is one of the algorithm that shows high quality results based on OLA. PSOLA analyzes the original waveforms to create pitch-synchronous analysis windows and synthesize the output signals both for modifying time-scale and pitch-scale. This type of modification is effective especially for pitch shifting of human speech because it preserves the formant of the input signal while changing pitch [3]. However, it is quite difficult to use PSOLA because the pitch prediction errors such as octave error can cause abrupt glitches.

2.0.3 Waveform-Similarity Overlap-Add (WSOLA)

Another algorithm that improves the basic OLA is WSOLA, which was proposed to solve the low signal sensitivity of OLA [4]. WSOLA maximizes the waveform similarity by allowing the analysis frame to find the most similar position through cross correlation. This method is effective for monophonic sources, but it cannot connect phases of all frequency components of polyphonic sources, causing artifacts called transient doubling or stuttering.

2.0.4 Phase Vocoder (PV)

To preserve the phase continuity for all frequency bin components while modifying the time-scale, using phase vocoder [2] is one of the solution. Phase vocoder estimates instantaneous frequency and it is used to update the phases of frequency components of the input signal in short-time Fourier transform. Although TSM results with phase vocoder has high phase continuity, it causes transient smearing for percussive audio sources and a coloring artifact called phasiness.

2.0.5 Harmonic-Percussive Source Separation (HPSS) with Phase Vocoder

A solution to reduce the artifacts from TSM with phase vocoder is using HPSS. Driedger and Müller proposed a novel TSM algorithm that applies phase vocoder to only harmonic sources and OLA to only percussive sources [5]. This can preserve both phase and transients to achieve the high quality TSM results for polyphonic signals.

[1] https://github.com/KAIST-MACLab/PyTSMod

Function	Main parameters	Remarks
OLA	audio, $scale\_factor^*$	Basic overlap-add method.
WSOLA	audio, $scale\_factor^*$	TSM using waveform similarity.
PV	audio, $scale\_factor^*$	TSM using PV. Phase locking is also supported.
HP-TSM	audio, $scale\_factor^*$	TSM using HPSS, PV, and OLA.
TD-PSOLA	audio, $scale\_factor(pitch)$, $scale\_factor(time)$, f_0	f_0 from external pitch tracker is needed.

* scale factor can either be a fixed value or anchor points.

Table 1. Available TSM functions and main parameters.

3. IMPLEMENTATION

We implemented PyTSMod using NumPy [6] and SciPy [7], which are commonly used for signal processing in Python. We also used libROSA [8] for HPSS. Most algorithm functions are based on the TSM toolbox [9], preserving the same input parameters. Therefore, the input signals and time-scale factor are required as main parameters for OLA, WSOLA, and phase vocoder. The scale factor can be either a fixed stretching rate alpha or an anchor point array for non-linear TSM. The anchor points are the pair of time position which contain the value of knee points of input and the output signal.

In case of TD-PSOLA, frame-level pitch estimation of the input audio is required for windowing. Since PyTSMod does not provide pitch tracker, users should use an external pitch estimation algorithm. TD-PSOLA also needs two scale factors alpha and beta as a main parameter, which represent the time stretching rate and pitch shifting rate, respectively. They are required because both the pitch and the length of the signal can be manipulated at the same time. To support non-linear pitch shifting, an absolute pitch value can be used as an input parameter instead of the fixed pitch scale factor.

For users who want to modify and save the result easily, a command-line interface is also supported. Users can save the result of TSM algorithm with a simple command named *tsmod*. The command-line interface is supported for all algorithms except TD-PSOLA which needs pitch estimation from an external pitch tracker.

4. REFERENCES

[1] F. Charpentier and M. Stella, "Diphone synthesis using an overlap-add technique for speech waveforms concatenation," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 11, Tokyo, Japan, 1986, pp. 2015–2018.

[2] J. L. Flanagan and R. M. Golden, "Phase vocoder," *Bell System Technical Journal*, vol. 45, no. 9, pp. 1493–1509, 1966.

[3] U. Zölzer, *DAFX: Digital Audio Effects*, 2nd ed. John Wiley & Sons, 2011, pp. 205–209.

[4] W. Verhelst and M. Roelands, "An overlap-add technique based on waveform similarity (WSOLA) for high quality time-scale modification of speech," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 2, Minneapolis, USA, 1993, pp. 554–557.

[5] J. Driedger and M. Müller, "Improving time-scale modification of music signals using harmonic-percussive separation," *IEEE Signal Processing Letters*, vol. 21, no. 1, pp. 105–109, 2013.

[6] S. V. D. Walt, S. C. Colbert, and G. Varoquaux, "The NumPy array: A structure for efficient numerical computation," *Computing in Science & Engineering*, vol. 13, no. 2, pp. 22–30, 2011.

[7] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. Jarrod Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. Carey, İ. Polat, Y. Feng, E. W. Moore, J. Vand erPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, and SciPy 1.0 Contributors, "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python," *Nature Methods*, vol. 17, pp. 261–272, 2020.

[8] B. McFee, C. Raffel, D. Liang, D. P. Ellis, M. McVicar, E. Battenberg, and O. Nieto, "librosa: Audio and music signal analysis in python," in *Proceedings of the 14th Python in Science Conference*, vol. 8, 2015, pp. 18–25.

[9] J. Driedger and M. Müller, "TSM toolbox: MATLAB implementations of time-scale modification algorithms," in *Proceedings of the 17th International Conference on Digital Audio Effects*, Erlangen, Germany, 2014.