Tempo Adjustment with Waveform Similarity based Overlap-Add (WSOLA)

by Mitja Schmakeit
Outline

- Motivation
- OLA
- WSOLA
- Algorithmic Complexity
- Recent R&D
Motivation

Music Mixing requires two songs to be at the same tempo.
If the tempo differs, it needs to be adjusted.
For that task, there exist many different algorithms.
Resampling

[D11]
Algorithms

- OLA (Overlap and Add)
- WSOLA (Waveform-similarity based OLA)
- Phase Vocoder
OLA (Overlap and Add)

Basic algorithm in digital signal processing
All more specialized algorithms utilize OLA
\( x \in \mathbb{R}^M : \text{input signal of size } M \)
OLA — Partitioning into segment 1 and 2

[Diagram showing amplitude vs. sample number for segments 1 and 2, with added zeros indicated]
OLA — Partitioning

[S03]
OLA — Result

\[ S_03 \]
OLA — Usage for tempo adjustment

[Diagram showing tempo adjustment with labels $y_1$, $y_2$, $y_3$, $y_4$, $x_1$, $x_2$, $x_3$, $x_4$, and arrows indicating slower ($H_a < H_s$) and faster ($H_a > H_s$) tempo adjustments.]

[DMDP16]
OLA does not preserve phase relations between consecutive frames. This means that in the worst case, heavy cancellation effects can occur.

[D11]
OLA – Example

OLA produces significant artifacts in the output signal, which is especially noticeable in harmonic structures.

Play original music

Play music made 20% faster with OLA

[S13]
Waveform-similarity based OLA (WSOLA)

Developed by W. Verhelst and M. Roelands in 1993 at Vrije Universiteit Brussel [VR93]

Still used today via various audio processing libraries that are used in programs such as Foobar2000, Audacity, Rhythmbox, Firefox and Chrome [DMDP16]
WSOLA — $\delta$ windows for similarity matching

Idea: Move each pair of overlapping frames around a bit before merging them, such that their waveforms are as similar as possible.

$$H_{a_i} = i \cdot H_a + \delta_i$$

[DMDP16]
WSOLA – Complexity

Space complexity $\mathcal{O}(n)$ (with $n$ being the frame size)

Time complexity $\mathcal{O}(n \cdot \log_2 n)$ \cite{DMDP16}

Therefore, with the right equipment, suited for real-time usage.

There exist many proposals for further reduction of WSOLA complexity (e.g. by estimating the optimal shift \cite{KLK+10})
WSOLA – Audio example

Play original music

Play music made 20% faster with OLA

Play music made 20% faster with WSOLA

[S13]
Recent R&D

- Time Stretching algorithms are numerous, the implementations on different devices are the current problem.
- As many applications move to the web, so do audio editing tools like DJ mixing software.
- Currently, there exist only few JavaScript implementations that can be used by web audio applications.

<table>
<thead>
<tr>
<th>Name</th>
<th>Algorithm</th>
<th>Audio Artifacts</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vexwarp</td>
<td>Phase Vocoder</td>
<td>Metal Tunnel</td>
</tr>
<tr>
<td>tempo-sox.js</td>
<td>WSOLA</td>
<td>Unknown</td>
</tr>
<tr>
<td>PhaseVocoder.JS</td>
<td>Phase Vocoder</td>
<td>Smeared Transients</td>
</tr>
<tr>
<td>OLA-TS.JS</td>
<td>Modified OLA</td>
<td>Modulation in harm. struct.</td>
</tr>
</tbody>
</table>

[DMDP16]
References


References


(https://youtu.be/J1FX7Knafng)

Thank You

Questions?