GCT535: Sound Technology for Multimedia

Time Scale Modification and Pitch Shifting

Graduate School of Culture Technology

Juhan Nam

Outlines

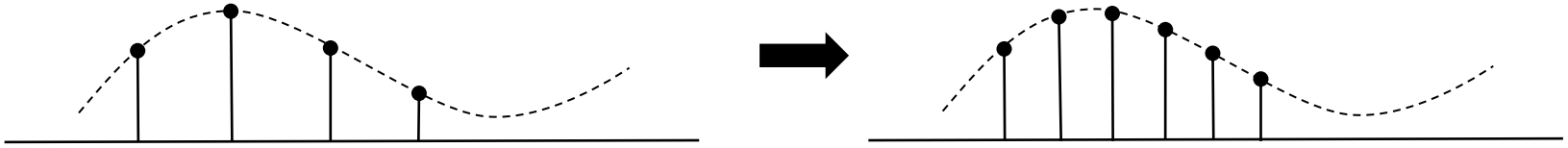
- Understanding the algorithms to change pitch or length (or time-scale) of audio waveforms
  - Resampling
  - Time-scale modification (or time stretching)
  - Pitch shifting

Resampling (Playback Rate Conversion)

- “Playback rate” is not necessarily equal to the recording rate
- Adjusting the playback rate given the recorded audio change the tone
  - Sliding tapes on the magnetic header in a variable speed
  - Speeding down: “monster-like”
  - Speeding up: “chipmunk-like”
  - It can be even negative rate: reverse playback
- Demo
  - https://musiclab.chromeexperiments.com/Voice-Spinner

Playback Rate Conversion (Resampling)

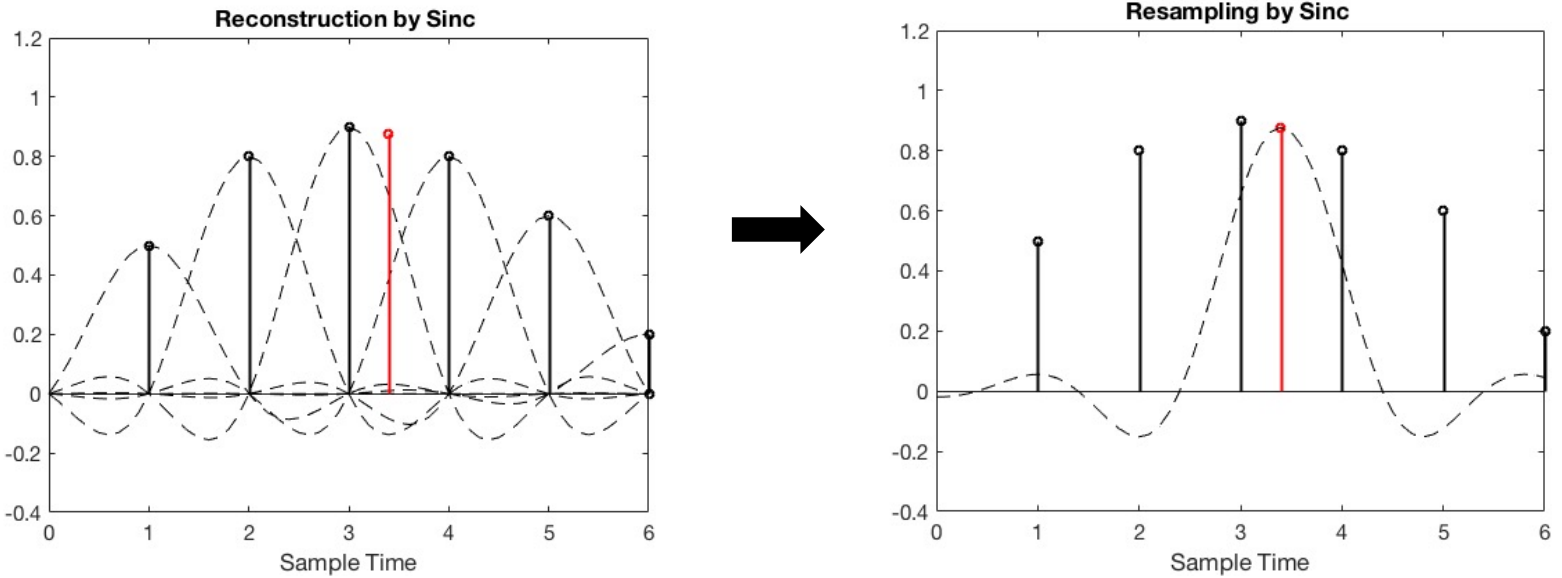
- Reconstruct the original signal and sample it with a new rate



- For a digital system with a constant playback rate
  - **Up-sampling** makes the original sound **played slower**
  - **Down-sampling** makes the original sound **played faster**

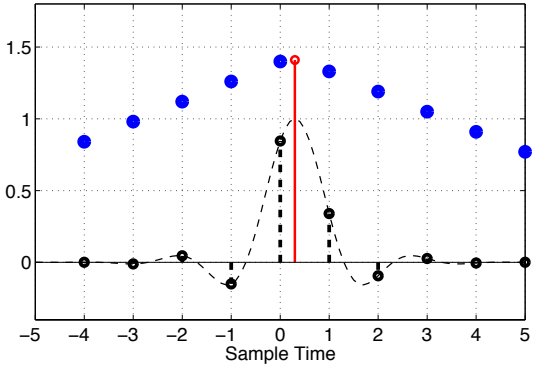
Resampling by the Reconstruction Lowpass Filter

- As you recall from the topic of digital audio, the original signal can be reconstructed by the sinc function
  - Resampling on the reconstructed signal is equivalent to interpolation with the reconstruction filter

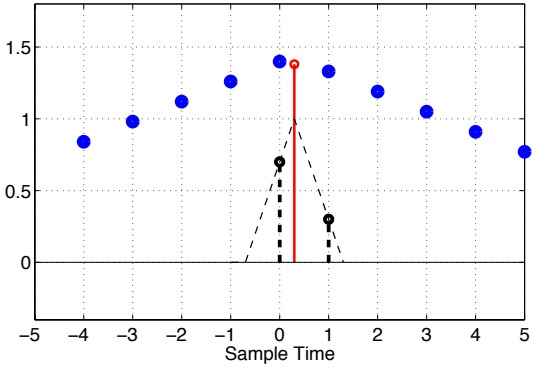


Reconstruction Lowpass Filters (Interpolation Filters)

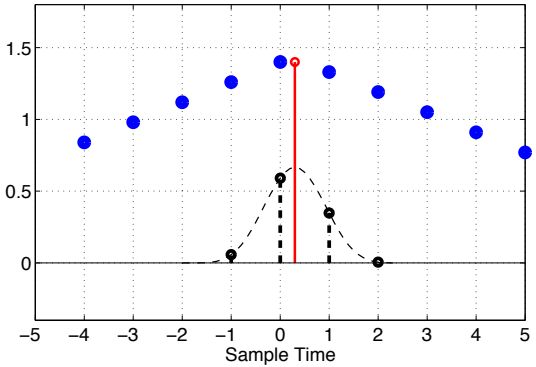
Windowed Sync



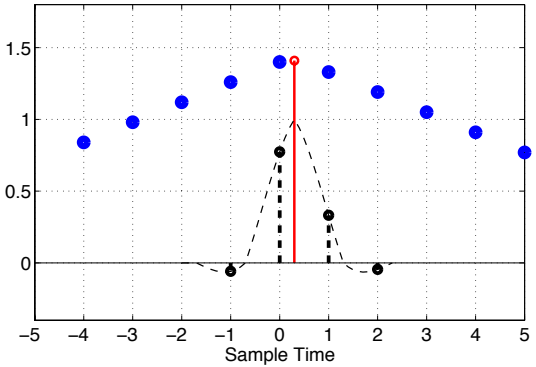
Linear Interpolation



3rd order B-spline Interpolation



3rd order Langrange Interpolation



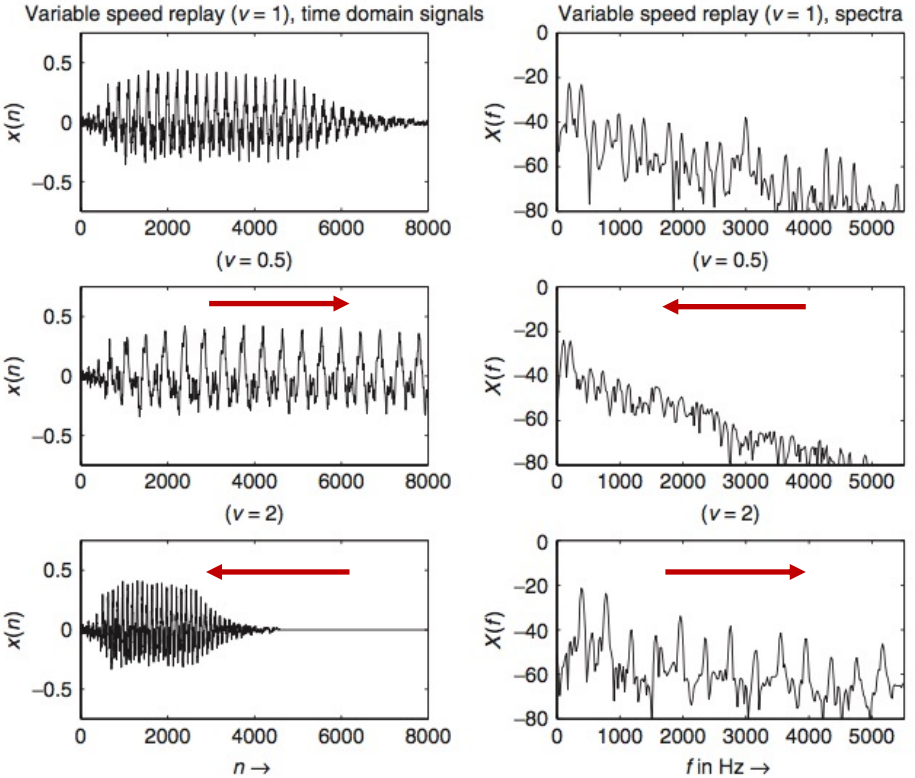
Resampling

- Resampling changes pitch, length and timbre at the same time!

Original

Speed Down (Up-sampling)

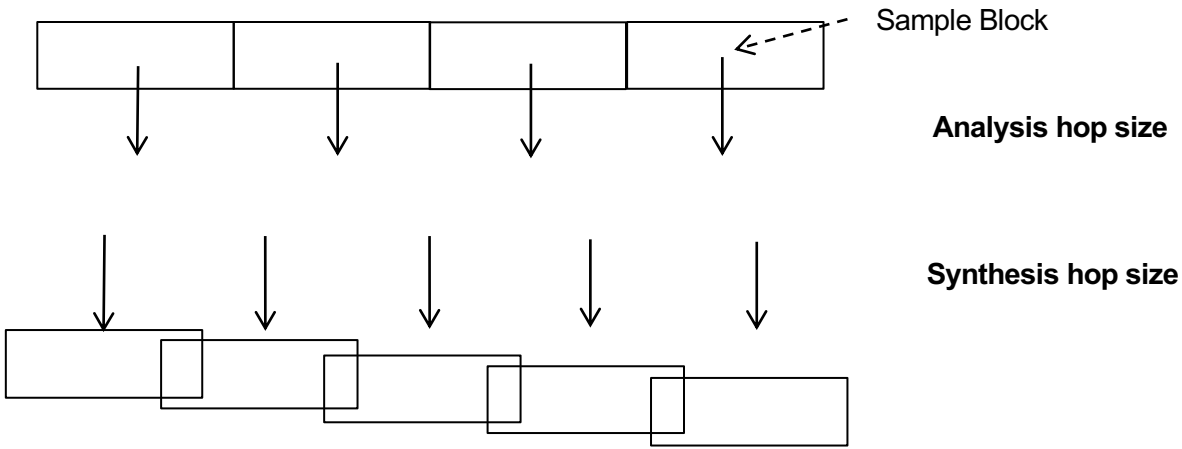
Speed Up (Down-sampling)



[The DaFX book]

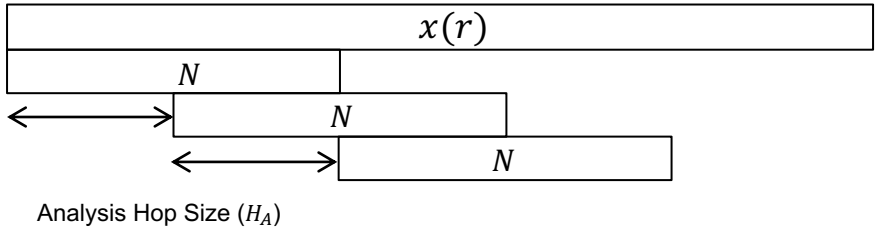
How can we control pitch and length independently?

- The secret is processing samples in frame-level instead of sample-level
  - The waveform is locally preserved within the frame
  - Analysis hop size and synthesis hop size are distinguished



Fundamental of Time-Scale Modification

- Analysis-Resynthesis approach in a frame-by-frame manner



$$x_m(r) = \begin{cases} x(r + mH_A), & \text{if } r \in [-\frac{N}{2} : \frac{N}{2} - 1] \\ 0, & \text{otherwise.} \end{cases}$$

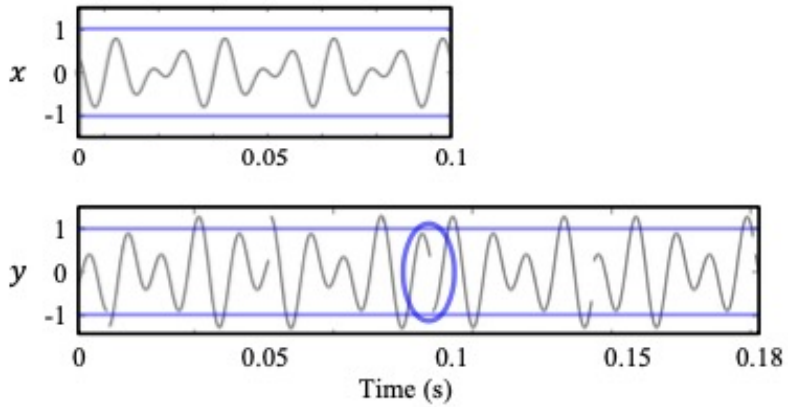
$$\mathcal{F}: x_m(r) \rightarrow y_m(r)$$

$$y(r) = \sum_{m \in \mathbb{Z}} y_m\,(r - mH_S)$$

- Time-stretching ratio: $\alpha = \frac{H_S}{H_A}$ (H_S: synthesis hop size, H_A : analysis hop size)
- If $\alpha > 1$, increase the length, If $\alpha < 1$, reduce the length

Fundamental of Time-Scale Modification (TSM)

- If the analysis frame $x_m(r)$ is the same as the synthesis frame $y_m(r)$?
  - Discontinuity at the boundary of the unmodified frames
  - Overlapping of the synthesis frame changes the amplitude



Time-scale modification with α=1.8 [Driedger and Müller, 2016]

OverLap-and-Add (OLA)

- Enforce a smooth transition between frames and compensate the amplitude change
  - Applying a window function w to the analysis frame: e.g. Hann window

$$w(r) = \begin{cases} 0.5(1 - \cos(\frac{2\pi(r + N/2)}{N - 1})), & \text{if } r \in [-\frac{N}{2} : \frac{N}{2} - 1] \\ 0, & \text{otherwise.} \end{cases}$$

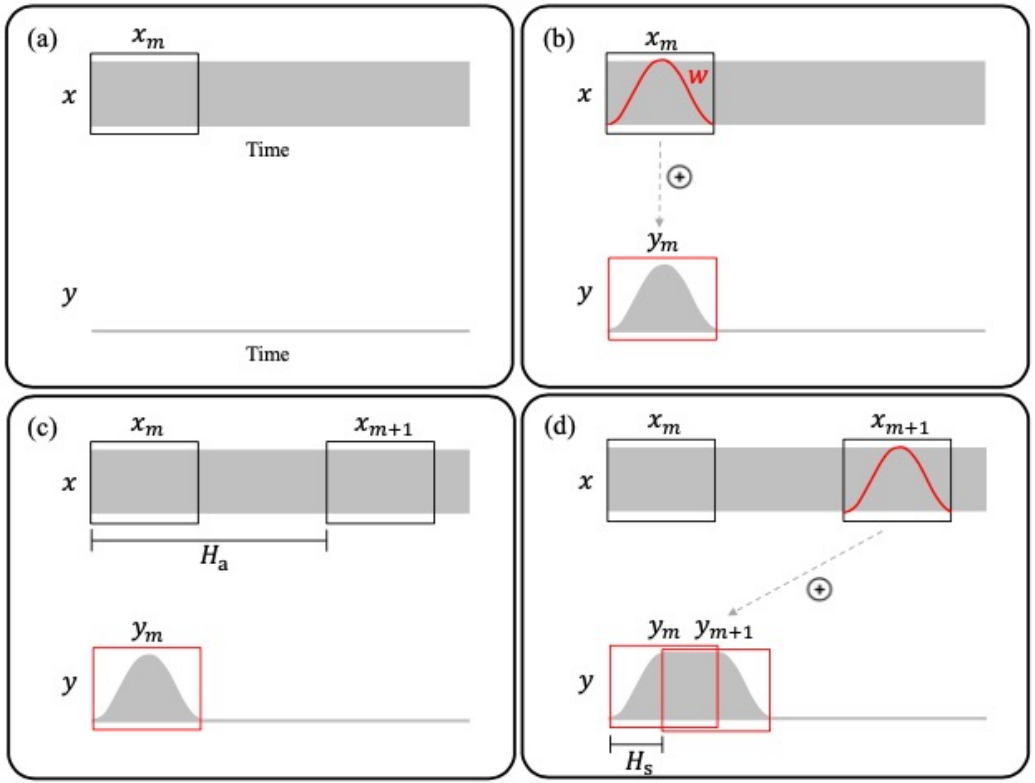
  - The Hann window has the following property for all r $\in \mathbb{Z}$

$$\sum_{n \in \mathbb{Z}} w\,(r - n\frac{N}{2}) = 1$$

  - The synthesis frame is computed as a windowed analysis frame with the amplitude normalization

$$y_m(r) = \frac{w(r)\, x_m(r)}{\sum_{n \in \mathbb{Z}} w\,(r - nH_S)}$$

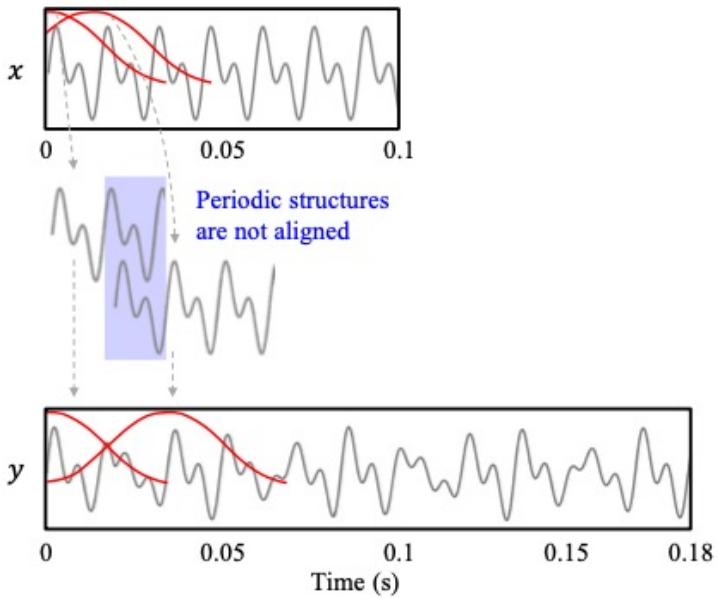
OverLap-and-Add (OLA)



Overlap and Add TSM [Driedger and Müller, 2016]

OverLap-and-Add (OLA)

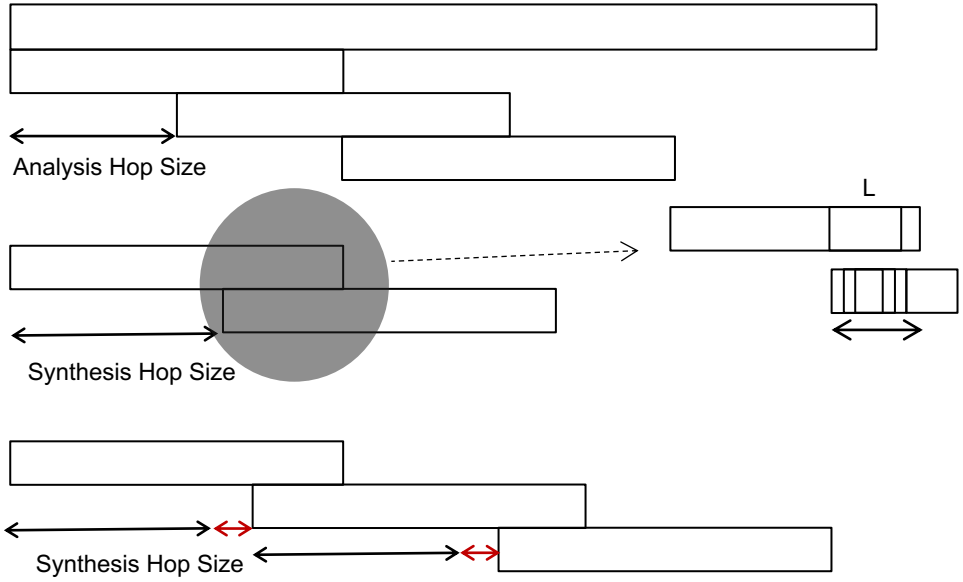
- However, OLA has a problem for periodic signals
  - They are called **phase jump artifacts**



Overlap and Add TSM [Driedger and Müller, 2016]

Synchronized OverLap-and-Add (SOLA)

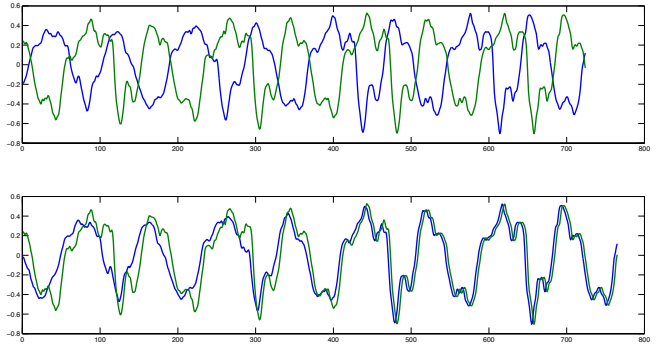
- Reduce artifacts in OLA by shifting the overlapped region such that the two adjacent frames are maximally correlated



Shift the next frame by the lag

Synchronization by cross-correlation

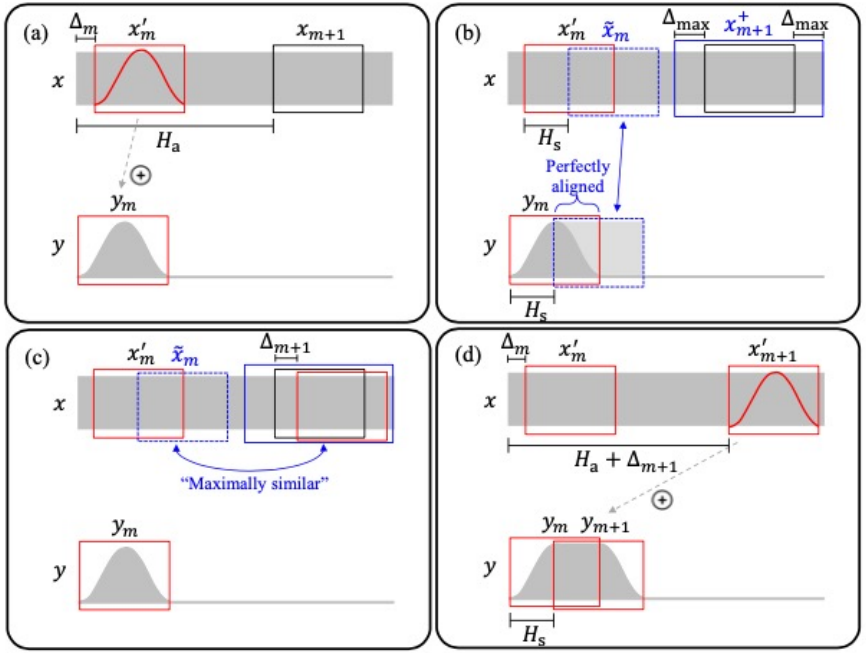
$$X_{corr}(l) = \sum_{n=0}^{n=L-1} x_1(n)x_2(n+l)$$



Find the lag (*l*) where the cross correlation is maximum

Waveform Similarity OverLap-and-Add (WSOLA)

- A variant of SOLA that adjusts the analysis hop size



WSOLA TSM [Driedger and Müller, 2016]

Phase Vocoder TSM (PV-TSM)

- A time-frequency processing based on Short-Time Fourier Transform
  - The analysis frame has a hop size of H_a

$$X(m,k) = \sum_{r=-N/2}^{N/2-1} x_m(r)w(r)\exp(-2\pi jkr/N) = \underbrace{|X(m,k)|}_{\text{Magnitude}}\exp(2\pi j\underbrace{\varphi(m,k)}_{\text{Phase}})$$

- PV-TSM uses a modified DFT of the analysis frame to reconstruct the synthesis frame

$$x_m^{mod}(r) = \frac{1}{N}\sum_{k=0}^{N-1} X^{mod}(m,k)\exp(2\pi jkr/N) \longrightarrow y_m(r) = \frac{w(r)x_m^{mod}(r)}{\sum_{n\in\mathbb{Z}} w^2\,(r-nH_S)}$$

Phase Vocoder TSM (PV-TSM)

- Phase Vocoder refines the the coarse frequency estimate of STFT
  - Estimate the instantaneous frequency using the difference of phase between two adjacent frames

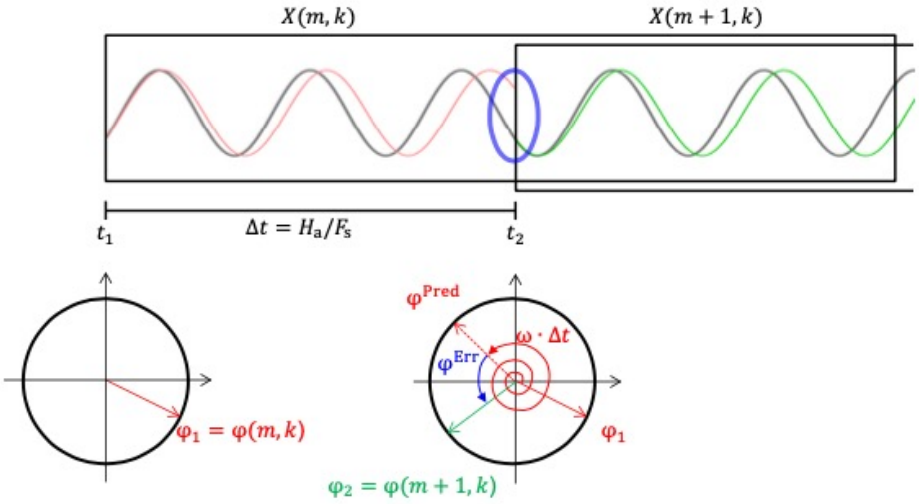
$$IF(\omega) = \omega + \frac{\varphi^{err}}{\Delta t}$$

- The modified phase is computed as followed

$$X^{mod}(m,k) = |X(m,k)|\exp(2\pi j\varphi^{mod}(m,k))$$

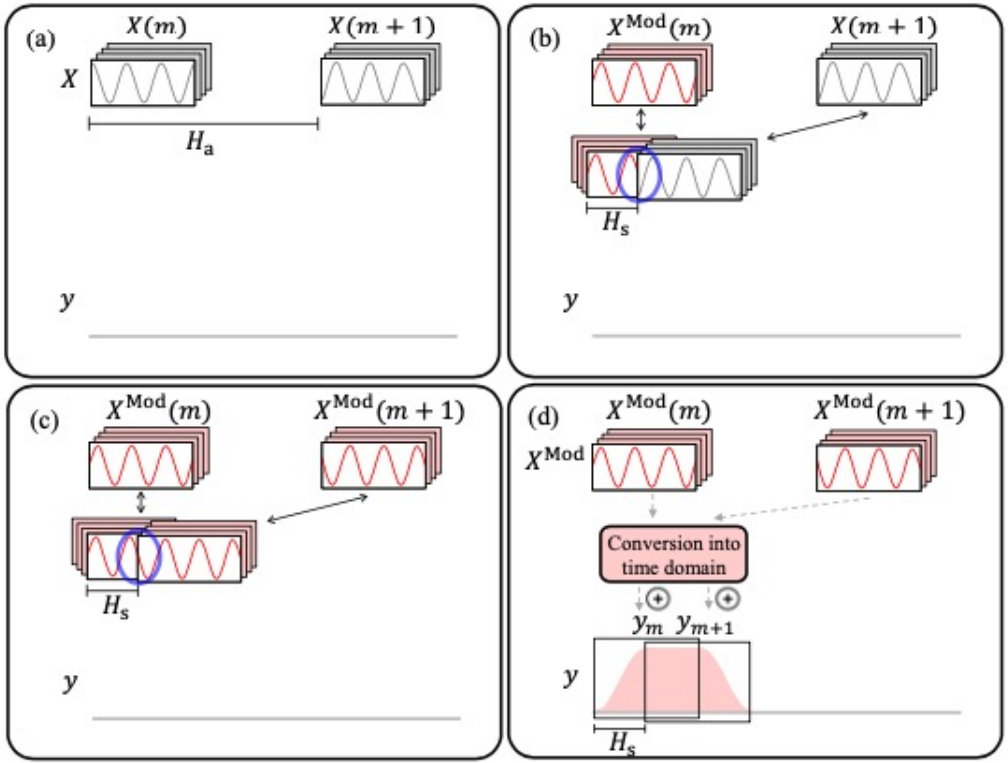
$$\varphi^{mod}(m+1,k)) = \varphi^{mod}(m,k) + IF(\omega)\frac{H_S}{F_S}$$

F_S= sampling rate



The phase jump between two adjacent frames [Driedger and Müller, 2016]

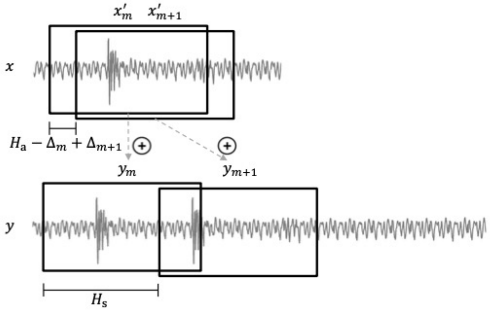
Phase Vocoder TSM (PV-TSM)



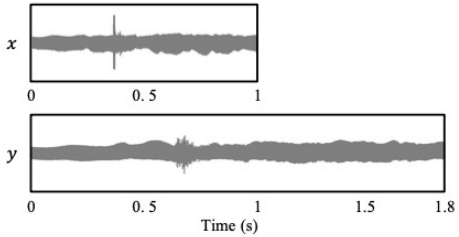
PV-TSM [Driedger and Müller, 2016]

TSM with Harmonic-Percussion Source Separation

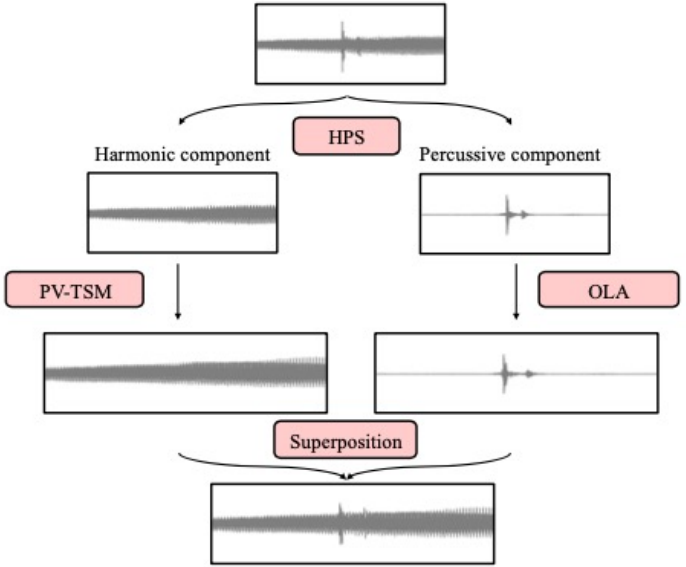
- Use different TSM methods for harmonic and percussive components of the input signals



Transient doubling (WSOLA)



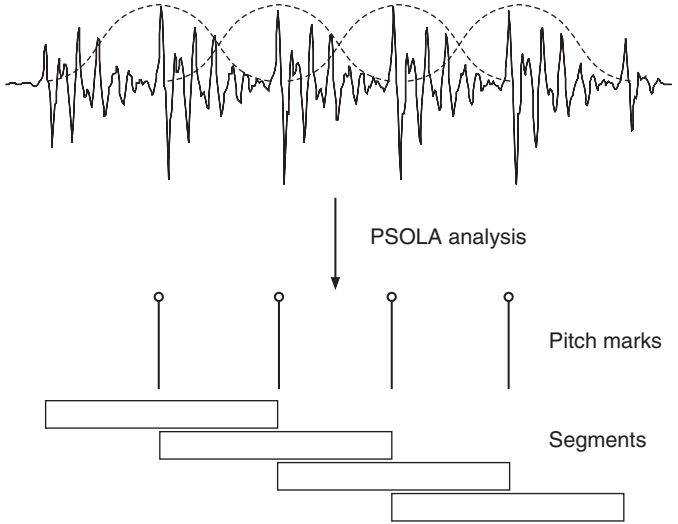
Transient smearing (PV-TSM)



TSM with HPSS [Driedger and Müller, 2016]

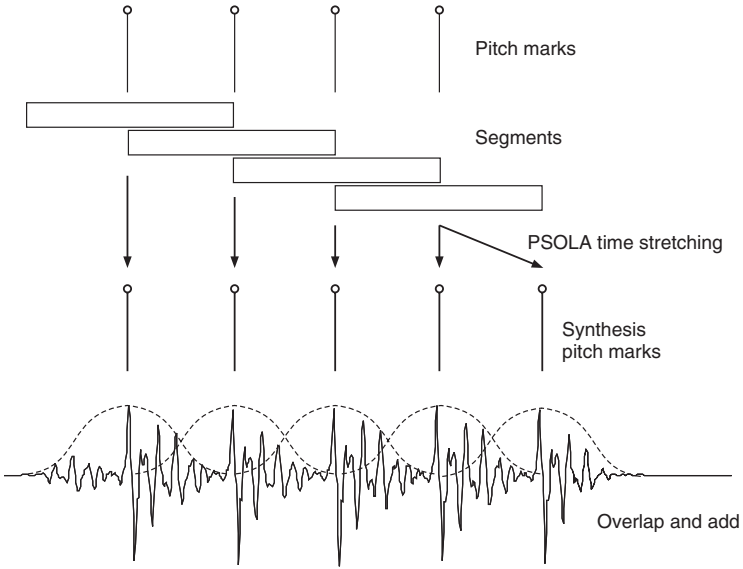
Pitch-Synchronous OLA (PSOLA)

- The analysis and synthesis hope size is synchronized to estimated pitch
- Analysis
  - Perform block-based pitch detection and find pitch marks t_i
    - Pitch period: $P(t) = t_{i+1} - t_i$
  - Extract a segment centered at every pitch mark t_i using a Hanning window with length $L_i = 2P(t_i)$ to ensure fade-in and fade-out



Pitch-Synchronous OLA (PSOLA)

- Synthesis for time-stretching
  - For every synthesis pitch mark $\tilde{t}_k$, search the corresponding t_i that minimizes $|\alpha t_i - \tilde{t}_k|$
  - Overlap and add the selected segment
    - If $\alpha > 1$, some segments will be repeated
    - If $\alpha < 1$, some segments will be discarded
  - The next synthesis pitch mark $\tilde{t}_k$ is determined to preserve local pitch
    - $\tilde{t}_{k+1} = \tilde{t}_k + \tilde{P}(\tilde{t}_k) = \tilde{t}_k + P(t_i)$



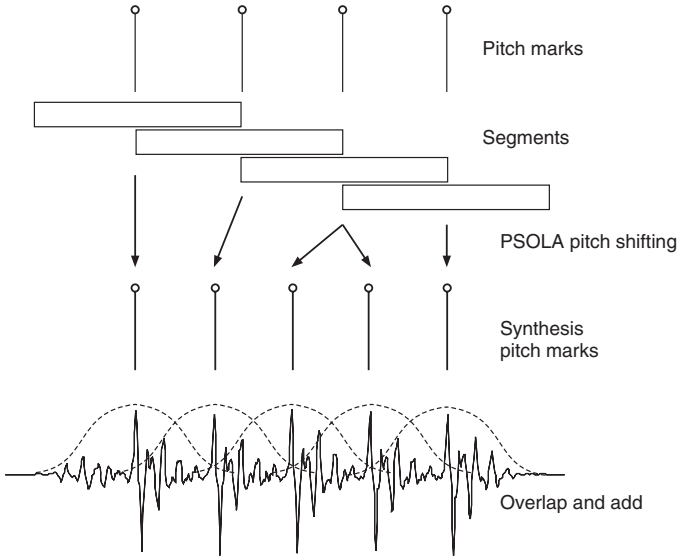
Pitch-Shifting

- TSM followed by Resampling
  - First, perform time-stretching with a ratio of α
  - Second, resampling the output with the same ratio of α

- Problem
  - Timbre (i.e. formant) changes by the resampling
  - This is quite audible for human voice (e.g. speech or singing)

Pitch-Synchronous OLA (PSOLA)

- PSOLA can be used for pitch-shifting
  - For every synthesis pitch mark $\tilde{t}_k$, search the corresponding t_i that minimizes $|t_i - \tilde{t}_k|$
  - Overlap and add the selected segment
    - If $\beta > 1$, some segments will be repeated
    - If $\beta < 1$, some segments will be discarded
  - The next synthesis pitch mark $\tilde{t}_k$ is determined to preserve local pitch
    - $\tilde{t}_{k+1} = \tilde{t}_k + \tilde{P}(\tilde{t}_k) = \tilde{t}_k + P(t_i)/\beta$
  - It is possible to combine the time-stretching (with the term $|\alpha t_i - \tilde{t}_k|$) and pitch-shifting
  - This preserves the formant of the input sound!



Resources

- PyTSMod
  - Time-scaling modification code using WSOLA (waveform similarity OLSA) and phase vocoder
  - https://github.com/KAIST-MACLab/PyTSMod

Ambient Sound Design Using Paul's Extreme Sound Stretch

- Extreme time scale modification (e.g. α = 50) with spectral smoothing can transform any sound/music into a texture or ambient sound
  - http://hypermammut.sourceforge.net/paulstretch/
  - https://cdm.link/2018/02/free-plug-brings-extreme-paulstretch-stretching-daw/