석 사 학 위 논 문
Master's Thesis

가창 표현 이식 알고리즘

Transferring Singing Expressions from One Voice to Another

2017

용 상 언 (龍 相 彦 Yong, Sangeon)

한 국 과 학 기 술 원

Korea Advanced Institute of Science and Technology

석 사 학 위 논 문

가창 표현 이식 알고리즘

2017

용 상 언

한 국 과 학 기 술 원

문화기술대학원

가창 표현 이식 알고리즘

용 상 언

위 논문은 한국과학기술원 석사학위논문으로
학위논문 심사위원회의 심사를 통과하였음

2017년 6월 22일

심사위원장 남 주 한 (인)

심 사 위 원 노 준 용 (인)

심 사 위 원 이 성 희 (인)

Transferring Singing Expressions from One Voice to Another

Sangeon Yong

Advisor: Juhan Nam

A dissertation submitted to the faculty of
Korea Advanced Institute of Science and Technology in
partial fulfillment of the requirements for the degree of
Master of Science in Engineering in Culture Technology

Daejeon, Korea
June 22, 2017

Approved by

Juhan Nam
Professor of Graduate School of Culture Technology

The study was conducted in accordance with Code of Research Ethics[1].

[1] Declaration of Ethical Conduct in Research: I, as a graduate student of Korea Advanced Institute of Science and Technology, hereby declare that I have not committed any act that may damage the credibility of my research. This includes, but is not limited to, falsification, thesis written by someone else, distortion of research findings, and plagiarism. I confirm that my thesis contains honest conclusions based on my own careful research under the guidance of my advisor.

MGCT 20154466 용상언. 가창 표현 이식 알고리즘. 문화기술대학원 . 2017년. 30+iv 쪽. 지도교수: 남주한. (영문 논문)
Sangeon Yong. Transferring Singing Expressions from One Voice to Another. Graduate School of Culture Technology . 2017. 30+iv pages. Advisor: Juhan Nam. (Text in English)

초 록

본 논문에서는 자동적으로 가창 표현을 한 목소리 신호에서 다른 목소리 신호로 이식하는 오디오 신호처리 시스템을 제안한다. 가창자의 능력에 따라 같은 노래를 부르더라도 음의 시작점, 음정, 에너지와 같은 부분에서 큰 변화가 발생할 수 있다. 이 시스템은 이러한 가창자의 고유의 음색을 제외한 음악적인 표현들을 추출 및 적용하는 것에 중점을 두었다. 이러한 가창 표현 이식 행위는 노래 부르기를 어려워하는 사람들의 음악 활동에 도움을 주고, 새로운 가창 표현을 학습하려는 사람들에게 보다 직관적인 가이드라인을 제공해 줄 수 있다. 이 시스템은 차례대로 음의 타이밍 정보와 음정, 그리고 에너지를 일치시키는 방식으로 표현을 이식한다. 본 연구에서는 이를 위해 음의 타이밍 정보를 일치시키는 알고리즘, 음정과 에너지 정보를 일치시키는 알고리즘, 그리고 해당 알고리즘의 성능을 최대한 개선시키고 최적화하는 방법을 제안한다. 그리고 이러한 세부 방법들을 기반으로 가창 표현 이식 시스템을 제안하여 가창 표현 수정에 대한 새로운 접근법을 제시하려고 한다.

핵 심 낱 말 가창, 표현 이식, 시간축 변환, 동적 시간 워핑

Abstract

This paper presents an audio signal processing system that automatically transfers singing expressions from one voice to another. Depending on singers' skills, a song is sung with great variations in terms of note onset time, pitch and energy. The system focused on extracting and transferring musical expressions, excluding the timbre of singers. This singing expression transfer system can provide more intuitive guidance to those who want to learn new vocabulary expressions and help the music activities of those who have difficulty in singing. The system transfers expressions in the order of tempo, pitch, and energy. In this study, we propose an algorithm to align the tempo of the note, a method to match pitch and energy information, and a method to optimize the performance of these processes. Based on these methods, we propose a new singing expression transfer system and propose a new approach to singing voice modification.

Keywords Singing voice, expression transfer, time-scale modification, dynamic time warping

Contents

List of Tables

List of Figures

Chapter 1. Introduction

Singing is a popular musical activity that many people enjoy, for example, in the form of karaoke. Depending on singing skills, a song can be rendered into touching music or just noisy sounds. What if my bad singing can be transformed and so sound like a professional? In this research, we present a vocal processing system that automatically transfers singing expressions from one voice to another.

Commercial vocal correction tools such as Autotune[1], VariAudio[2] and Melodyne[3] mainly focus on modifying pitch of singing voice. Some of them are capable of manipulating note onset timing or other musical expressions by editing transcribed MIDI notes. Although they provide automated controls, the correction process is often tedious and repetitive until satisfactory results are achieved. There are some previous work that attempted to minimize the manual effort in modifying musical expressions. Bryan et. al proposed a variable-rate time-stretching system that allows users to modify the stretching ratio easily [1]. Given a user-guided stiffness curve, the system automatically computed time-dependent stretch rate via a constrained optimization program. Roebel et. al proposed an algorithm to remove vibrato expressions [2]. They operated entirely based on spectral envelope smoothing without manipulation of individual partial parameters. While these methods provide more convenience to process singing voice signals, they still require user guide or parametric control to some extent.

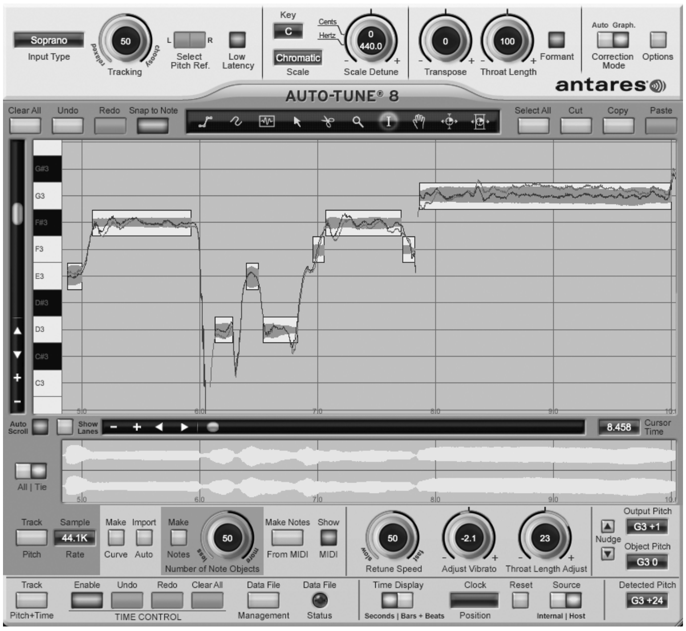

figure 1.1: Antares Autotune 8 Graphical Mode.[5]In the existing vocal correction tools, users should manipulate features in manual to modify singing voice signals.

In this thesis, we propose an audio signal processing system that modifies musical expressions of singing voice in a fully automatic manner with a target singing voice as a control guide. Assuming that both source and target voices sing the same song, the system transfers three musical expressions

[1]http://www.antarestech.com/products/index.php
[2]https://www.steinberg.net/en/products/cubase/cubase_pro.html
[3]http://www.celemony.com/en/melodyne/what-is-melodyne
[5]http://www.antarestech.com/products/detail.php?product=Auto-Tune_8_66

from target to source: tempo, pitch, and dynamics. First, it temporally synchronizes two singing voices using dynamic time warping on vibrato-suppressed mel-scale spectrum and a formant feature. Second, it extracts pitch ratio between the two voices and modifies the pitch of source voice using pitch-synchronous overlap-add algorithm (PSOLA). Finally, it modifies dynamics of the source voice by extracting the ratio of amplitude envelops. In the series of process, the system does not use any user guide or additional information such as lyrics and music scores beside a target voice. Since the system modifies only technical elements in singing and preserves the timbre of source voice, it will be useful for not only sound production but also vocal training.

Chapter 2. Research Background

2.1 Time-Scale Modification Algorithm

Time-scale modification (TSM) algorithm is the process that manipulates the length of the audio signal. [3] The ideal TSM algorithm should modify only the tempo of the signal, and preserve any other properties such as pitch and timbre. This TSM method is commonly used in sound producing area to synchronize the duration of audio sources to other media source, or change the pitch of audio sources with resampling without changing the duration of audio sources.

There are two main issues in TSM procedures. The first one is degradation of percussive transients. [4] While modifying audio sources with TSM algorithms, percussive transients often disappeared or are doubled. The other problem is phase discontinuity in mixed audio sources. Because the phase of each sources are different, phases in the mixed sources are discontinued with overlap-add based TSM method.

The key idea of TSM algorithm is decomposing the audio signal in the short length with the analysis hop size H_a, modifying the decomposed audio frames and recomposing those modified frames with the synthesis hop size H_s. The m^{th} decomposed frame x_m is derived as:

$$x_m[n] = \begin{cases} x[n + mH_a], & \text{if } -N/2 \leq n < N/2, \\ 0, & \text{otherwise.} \end{cases} \tag{2.1}$$

where N is the size of the frame. While $H_a < H_s$, the audio signal will be stretched after the procedure, and will be compressed if $H_a > H_s$.

In this section, we introduce some frequently used TSM algorithms and advantages and limitations of each methods.

2.1.1 Overlap-Add Method (OLA)

Overlap-add (OLA) method is the most simple and basic structure of TSM algorithm. In the OLA, the decomposed analysis frames x_m is used in recomposing without any modification. The result of OLA is derived as:

$$y[n] = \sum_{m=0} x_m[n - mH_s]w[n - mH_s] \tag{2.2}$$

where w is a window function with size N to help frames to be smoothly connected. Typically, Hann window function is used.

In OLA method, there is no aligning process to preserve local periodic structure of the input signal while recomposing the output signal. Therefore, phase jump artifacts, which means the distortion of periodic structures in the signal, occurs in the output signal, and it causes warbling sound if there is a harmonic component in the input signal.

2.1.2 Waveform-Similarity Overlap-Add Method (WSOLA)

Waveform-Similarity Overlap-Add (WSOLA) algorithm is the improved version of OLA. [5] WSOLA is focused on the reduction of phase jump artifacts, which is the main problem of OLA.

To reduce the artifacts and preserve the periodic structure of the signal, WSOLA tries to decide the next analysis frame based on the similarity of the previous frame. To find the proper frame, a small amount of shifting Δ_m is applied to the m^{th} analysis frame, where Δ_m is an integer in the range of $-\Delta_{\max} \le \Delta_m < \Delta_{\max}$. This adjusted analysis frame $x_m^{'}$ is derived as:

$$x_m^{'}[n] = \begin{cases} x[n + mH_a + \Delta_m], & \text{if } -N/2 \le n < N/2, \\ 0, & \text{otherwise.} \end{cases} \tag{2.3}$$

To find the next frame that is most naturally connected with the m^{th} adjusted analysis frame $x_m^{'}$, WSOLA tries to find $(m+1)^{\text{th}}$ analysis frame most similar to the frame that follows $x_m^{'}$. This following frame $\widetilde{x}_m[n]$ is called natural progression of the adjusted analysis frame and derived as:

$$\widetilde{x}_m[n] = \begin{cases} x[n + mH_a + \Delta_m + H_s], & \text{if } -N/2 \le n < N/2, \\ 0, & \text{otherwise.} \end{cases} \tag{2.4}$$

In WSOLA, cross-correlation is used for checking similarity between frames to find the best value for shifting. To decide the best value for Δ_{m+1}, the system calculates cross-correlation from $-\Delta_{\max}$ to $\Delta_{\max}$.

$$\text{xcorr}(\widetilde{x}_m, x_{m+1}, \Delta) = \sum_{n=0} \widetilde{x}_m[n]x[n + (m+1)H_a + \Delta] \tag{2.5}$$

$$\Delta_{m+1} = \underset{-\Delta_{\max} \le \Delta < \Delta_{\max}}{\text{argmax}} \ \text{xcorr}(\widetilde{x}_m, x_{m+1}, \Delta) \tag{2.6}$$

After finding the adjusted analysis frames, WSOLA uses the adjusted analysis frame instead of the analysis frame to synthesize the output in the same way as the OLA.

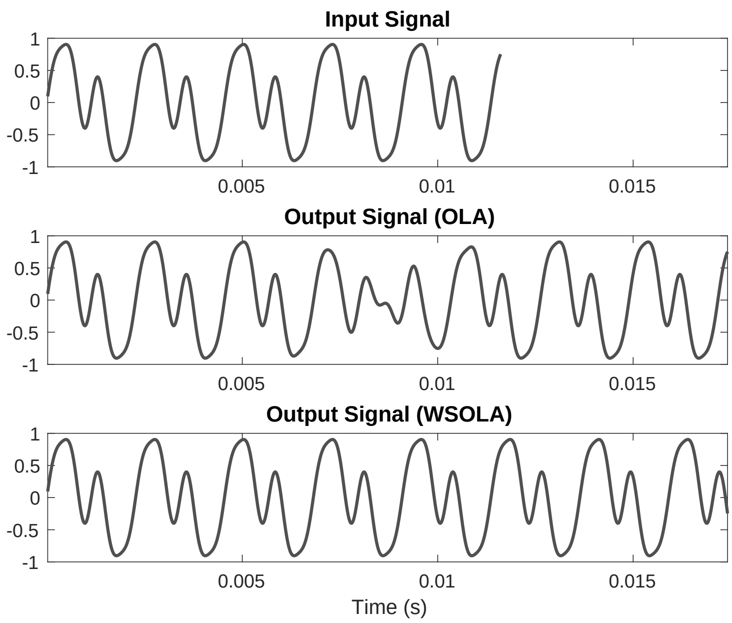

figure 2.1: Example result with OLA and WSOLA. There is no phase jump artifacts in the result with WSOLA.

Because WSOLA uses the frame that is most naturally connected with the previous frame, phase jump artifacts of the result are greatly reduced comparing to the result using OLA. However, WSOLA still has a few problems. The first one is transient doubling or stuttering. When we choose the adjusted analysis frame $x_m^{'}$, the short transient signal can be located in multiple frames, or just skipped. Another problem is that WSOLA cannot preserve phases in all frequency range in polyphonic signals because WSOLA is trying to connect only the conspicuous phase of the signal.

2.1.3 TSM with Phase Vocoder Method (PV-TSM)

TSM method with phase vocoder (PV-TSM) is a method to preserve the phase of all signal elements. [6, 7] The basic concept of PV-TSM is using short-time Fourier transform (STFT) to extract frequency-domain information including phase, and resynthesize the output signal based on the extracted data.

The core idea of PV-TSM is to find the accurate instantaneous frequency of the sinusoidal component $\mathrm{IF}(\omega)$. We can derive the instantaneous frequency through the phase error between the predicted phase and the actual phase.

If the phase from STFT ϕ is accurate, the predicted phase after Δt seconds ϕ^{Pred} is derived as:

$$\phi^{\mathrm{Pred}} = \phi + \omega \Delta t. \tag{2.7}$$

Therefore, we can calculate the phase error ϕ^{Err} with the difference between the predicted phase ϕ^{Pred} and the actual phase ϕ^{Act} by

$$\phi^{\mathrm{Err}} = \phi^{\mathrm{Act}} - \phi^{\mathrm{Pred}} - 2\pi \cdot \mathrm{round}(\phi^{\mathrm{Act}} - \phi^{\mathrm{Pred}}) \tag{2.8}$$

where $2\pi \cdot \mathrm{round}(\phi^{\mathrm{Act}} - \phi^{\mathrm{Pred}})$ is for adjusting the phase into the range $[-\pi, \pi]$.

Because the phase error ϕ^{Err} means the error in Δt seconds, the instantaneous frequency $\mathrm{IF}(\omega)$ is derived as:

$$\mathrm{IF}(\omega) = \omega + \frac{\phi^{\mathrm{Err}}}{\Delta t}. \tag{2.9}$$

In PV-TSM, the system modifies the phase of spectrogram based on the instantaneous frequency to connect the phase of all signal elements. PV-TSM modifies the phase of STFT-ed input signal X to X^{Mod}, and calculate inverse STFT to create the modified signal.

To calculate the instantaneous frequency of the m^{th} analysis frame x_m, we find the phase error with the phase of $(m+1)^{\mathrm{th}}$ analysis frame, and modify the phase of the signal:

$$\phi_{m+1}^{\mathrm{Mod}}(\omega) = \phi_m^{\mathrm{Mod}}(\omega) + \mathrm{IF}_m(\omega)\frac{H_s}{F_s} \tag{2.10}$$

where H_s/F_s is the time difference between m^{th} frame and $(m+1)^{\mathrm{th}}$ frame in seconds. With the modified phase ϕ_m^{Mod}, the STFT of m^{th} frame is modified as:

$$X_m^{\mathrm{Mod}}(k) = |X_m|\phi_m^{\mathrm{Mod}}(k) \tag{2.11}$$

and the output signal is created with the sum of inverse STFT of the modified signal.

As mentioned above, PV-TSM has a great advantage in preserving the phase of polyphonic signals. However, it is difficult to preserve the short transient signal that is shorter than the length of analysis frame. [8] Therefore, PV-TSM is not suitable for modifying the percussive signal.

2.1.4 Pitch-Synchronous Overlap-Add Method (PSOLA)

Pitch-Synchronous Overlap-Add (PSOLA) method is another improved version of OLA. [9] In PSOLA algorithm, the pitch information of the audio signal is used to modify the audio signal.

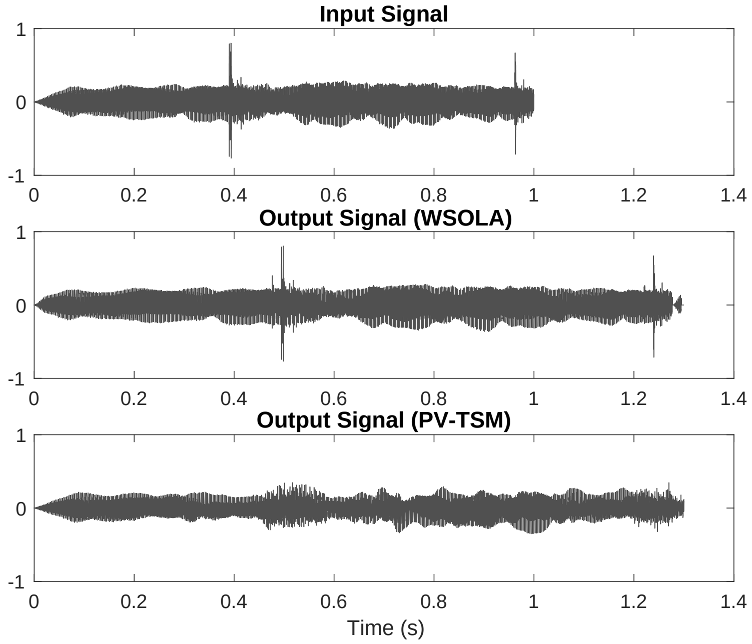

figure 2.2: Example result with WSOLA and PV-TSM. The transient signal is not preserved in the result with PV-TSM.

At first, PSOLA extracts the pitch of the input signal, and finds the position of pitch marks, which means the local maximum in the pitch period of the signal. If signal has unvoiced portions, which does not have a pitch, the pitch marks are positioned in a constant rate. One of the way to find the pitch mark is a peak-search approach. In this approach, $(m+1)^{\text{th}}$ pitch mark t_{m+1} and $(m-1)^{\text{th}}$ pitch mark t_{m-1} is derived as:

$$t_{m+1} = \max([t_m + \delta P_0, t_m + (2-\delta)P_0]) \tag{2.12}$$

$$t_{m-1} = \max([t_m - \delta P_0, t_m - (2-\delta)P_0]) \tag{2.13}$$

where P_0 is the pitch period of the signal and δ is a constant factor. δ is usually in the range of $[0.5, 0.9]$.

After finding the pitch marks, m^{th} analysis frame x_m has the center at m^{th} pitch mark t_m, and has the length of $2P(t_m)$:

$$x_m = \begin{cases} x[n + t_m], & \text{if } -P(t_m) \le n < P(t_m), \\ 0, & \text{otherwise.} \end{cases} \tag{2.14}$$

To synthesize the output signal, PSOLA defines the synthesis pitch mark first. The k^{th} synthesis frame is determined by the m^{th} analysis frame that minimizes the pitch mark distance $|\alpha t_m - \widetilde{t}_k|$ where α is the time-stretching rate and $\widetilde{t}_k$ is the k^{th} synthesis pitch mark. The k^{th} synthesis pitch mark is located in a section away from the $(k-1)^{\text{th}}$ pitch mark by a pitch period at the corresponding point in time:

$$\widetilde{t}_k = \widetilde{t}_{k-1} + P(t_i). \tag{2.15}$$

PSOLA is the high cost TSM algorithm because it needs a high-quality pitch tracking algorithm to use. [5] Nevertheless, PSOLA is frequently used because it has an advantage in pitch-shifting. In PSOLA, if the distance between a synthesis pitch mark is narrowed or widened, the pitch of the input signal is changed. This pitch shifting with PSOLA is different from pitch-shifting based on resampling because it preserves formant (spectral envelope) of the original signal. [10] Because the broken formant causes the artificial timbre especially in human voice, PSOLA is greatly used for changing the pitch of the human voice signal.

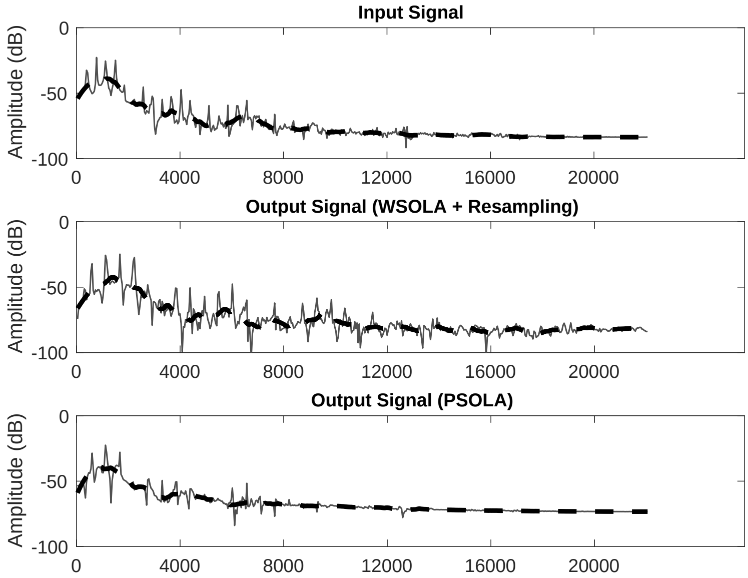

figure 2.3: Example result of pitch shifting with WSOLA + resampling and PSOLA. The formant (spectral envelope) is preserved in the result with PSOLA.

2.2 Dynamic Time Warping

Dynamic time warping (DTW) algorithm is the well-known algorithm that measures the temporal similarity between two signals and finds the optimal path that aligns the two signals. [11] DTW has been used in aligning temporal signals such as speech, music, and video.

The basic idea of DTW is using dynamic programming to find the path that has the lowest cost. Before starting to find the optimal path, the similarity matrix C is calculated as:

$$C(i,j) = \text{distance}(X(i), Y(i)) \tag{2.16}$$

where X and Y are features of two signal x and y. In the audio signal, STFT [12] and Mel-Frequency Cepstral Coefficient [13] are frequently used as a feature in creating the similarity matrix, and Cosine distance is used for calculating the distance.

To find the path, the DTW calculates the accumulated similarity matrix D first. The first row and the first column of the accumulated similarity matrix D is calculated as:

$$D(n,1) = \text{sum}(C(1:n,1)), n = [1, N] \tag{2.17}$$

$$D(1,m) = \text{sum}(C(1,1:m)), m = [M, 1] \tag{2.18}$$

where M and N is the size of the similarity matrix. After the initialization of the first row and the column, the rest part of the accumulated similarity matrix D is derived as:

$$D(n,m) = C(n,m) + \min \begin{cases} D(n-1,m) \\ D(n,m-1) \\ D(n-1,m-1) \end{cases} \tag{2.19}$$

After calculating the accumulated similarity matrix D, the path is obtained to track the path from destination to starting point of the similarity matrix backward.

2.3 Pitch Tracking Algorithm

Pitch tracking algorithm is the algorithm that measures the pitch of the given audio signal. There are two ways to measure the pitch, time-domain method and frequency-domain method.

In frequency domain, the fundamental frequency F_0 is estimated by observing features in STFT-converted signal. Methods like harmonic pattern matching [14], cepstrum [15], and harmonic-product-sum [16] are the frequency-domain pitch tracking algorithm.

In time domain, auto-correlation [17] and average magnitude difference function [18] is the representative methods to track the pitch. Time-domain pitch tracking algorithm is analyzing the periodic pattern of the signal to find the fundamental frequency F_0.

Chapter 3. Related Works

As mentioned in the introduction, our goal is to extract and transfer the musical features from source signal to target signal which sings the same song. In this chapter, we are focused on the research about transferring musical expressions with additional information such as score and lyrics, and modifying musical features of the signal.

3.1 Changing Musical Expressions with Extracting Features

There are some works in digital audio effects field about changing musical expressions of singing voices and musical instruments with extracting features. Some studies tried to manipulate musical features like vibrato [19], pitch, tempo [1, 20], and spectral envelope [21] for changing musical expressions. These studies are the most basic research to change musical expressions, but because they manipulate variables artificially to change musical expressions instead of using actual recorded examples, they are cumbersome and sometimes unnatural.

3.2 Transfer Musical Styles to Synthesized Sources

Also, there are some studies to extract styles from recorded examples to transfer musical styles to synthesized sources. [22, 23, 24] However, in this case, it requires the additional information such as lyrics and scores, and it does not transfer expressions from audio to audio directly.

3.3 Aligning the Source Signal and the Target Signal

Because this system directly transfers expressions from audio to audio, it is important to align the source signal and the target signal. In previous studies, some researchers tried audio-to-audio alignment to align audio and additional information such as lyrics and scores. [25, 26] Because this additional information contains onset data, the system does not have to align every frame by frame accurately. Also, there are some studies to align the temporal alignment of two voice signals, [27, 28] but in this case, they do not have to align every frame by frame because they used lyrical information to align them. However, in this system, we try to align two audio signals without any additional information.

Chapter 4. Proposed Architecture and Implementation

4.1 System Overview

Figure 4.1 illustrates the overall processing pipeline of the proposed system. It is composed of three modules that extract acoustic features from both voice signals and process the source. The source signal is transformed through the three modules in sequence, and the target signal is delivered to the three modules to provide musical expressions.

The first module extracts the timing information from both signals, and align the tempo of the source signal. In this process, we use both musical feature and lyrical feature to measure the timing of the singing signal more accurate for every frame. After the feature extraction, the time-scale modification algorithm is applied to modify the source signal.

After the temporal alignment procedure, the system extracts the pitch information of both signals and align the pitch of the source signal. To extract the pitch, YIN algorithm [18] is used in this system. After the feature extraction, the pitch synchronous overlap-add (PSOLA) algorithm is used to align the pitch of the signal without distortion in the formant.

At last, the dynamics alignment module works. In this step, the system extracts dynamics feature from both signals with envelope detector, and multiply the difference of both envelope signals to the source signal to align the dynamics.

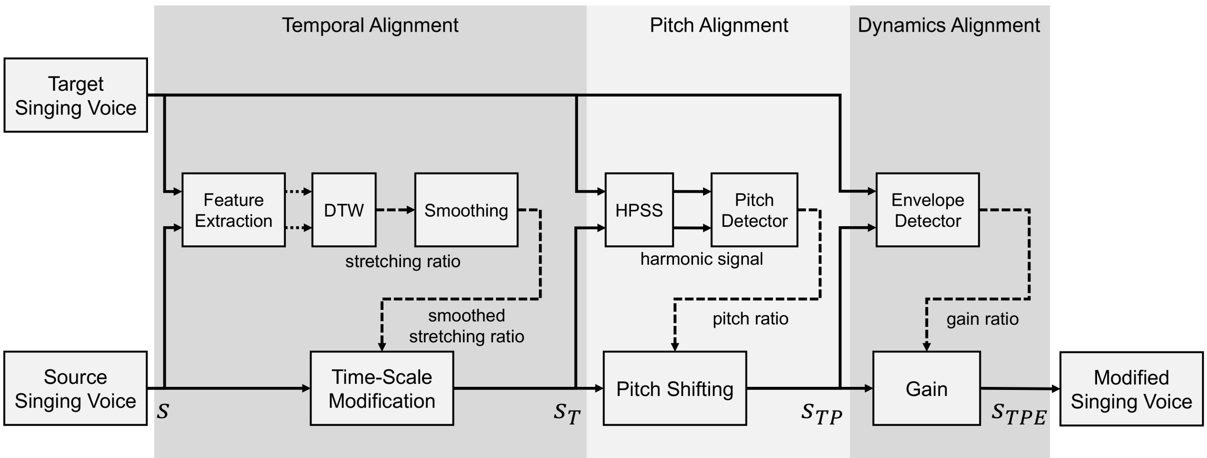

figure 4.1: System overview.

4.2 Temporal Alignment

The first step of the system is temporal alignment that synchronizes note timings between the source voices. This is actually the most important step because the subsequent steps relies on the aligned source for pitch and dynamics processing. We basically use dynamic time warping (DTW), a dynamic programming algorithm which is popularly used for temporal alignment of music and audio data [29]. The issue here is what type of features will be used as input for DTW.

4.2.1 Feature Extraction

Considering that the source and target voices are rendered from the same song, one straightforward approach is transcribing the audio signals into MIDI notes and use the melody notes for DTW [30]. However, this approach can be affected by performance of the transcription module and, moreover, misses exploiting the phonetic information from lyrics which is another common part in the two singing voices. Thus, we instead extract audio features from the signals and use them for DTW.

Our initial approach was simply using the magnitude spectrum of two singing voices as audio features. However, the DTW algorithm often failed to find a correct alignment path when either one voice has vibrato and pitch bending. The left-top in Figure4.2 shows the similarity matrix where each element was computed from cosine distance between every pair of the two magnitude spectra. The alignment path in red returned from the DTW algorithm tended to find the onset and offset of note quite successfully. However, it has severe detour, for example, that in the range of 300 to 350 time frames where the target voice has strong vibrato. This detour caused audible artifacts when the system modifies the time scale of the source signal.

To solve the detour problem and improve the path accuracy, we tried three methods. The first method is leveraging the phonetic information shared in lyrics of the song. The phonetic information tends to be less affected by musical expressions such as vibrato and pitch and so can allow more stable alignment. Since the phonetic information is related to the voice formant, we extracted the formant features using linear predictive coefficients (LPC). We chose the filter order according to [31]. Using LPC, we compute a separate similarity matrix. The left-middle in Figure 4.2 shows the similarity matrix and alignment path by DTW. Compared to the DTW path by the spectrum, the detour in the segment with strong vibrato become more diagonal. However, when we listened to the processed sound, the path by LPC-only similarity matrix did not cause artifacts but often misses right note timings.

Considering these advantages and disadvantage of STFT and LPC, we create a new matrix by averaging the two similarity matrices as follows:

$$S(i,j) = rS_{\mathrm{STFT}}(i,j) + (1-r)S_{\mathrm{LPC}}(i,j) \tag{4.1}$$

where r is the mix rate of the two matrices. In this study, we used 0.7 for r.

The left-bottom in Figure 4.2 shows that it successfully reduces the detour problem and, at the same time, finds more accurate path comparing to the similarity matrix with one of STFT or LPC.

The second method is converting the frequency domain of STFT to mel-scale. Mel-scale is a scale that maps frequency to perceptual pitches Since the perceptual pitch is proportional to the logarithmic scale of the frequency [32], the similarity matrix with mel-scale STFT is less sensitive to the pitch difference of the source signal and the target signal, and the amplitude of vibrato is transformed equally in all harmonics. Figure 4.3 shows the spectrogram and mel-scaled spectrogram of an audio signal. To convert spectrogram to mel-scaled spectrogram, filterbank approach is used. [33]

The last method is applying the maximum filter to the spectrum. The maximum filtering is effective in suppressing vibrato or other pitch variations [2] and so this can help the detour problem. We used the maximum filter to the magnitude spectrum of both source and target before computing the similarity matrix as follows.

$$X_{max}(i,j) = \max(X(i, j-l : j+l)) \tag{4.2}$$

where j corresponds to the frequency axis and l is the order of the maximum filter. In this paper, we used 3 for l to compensate the error within 3 semitones.

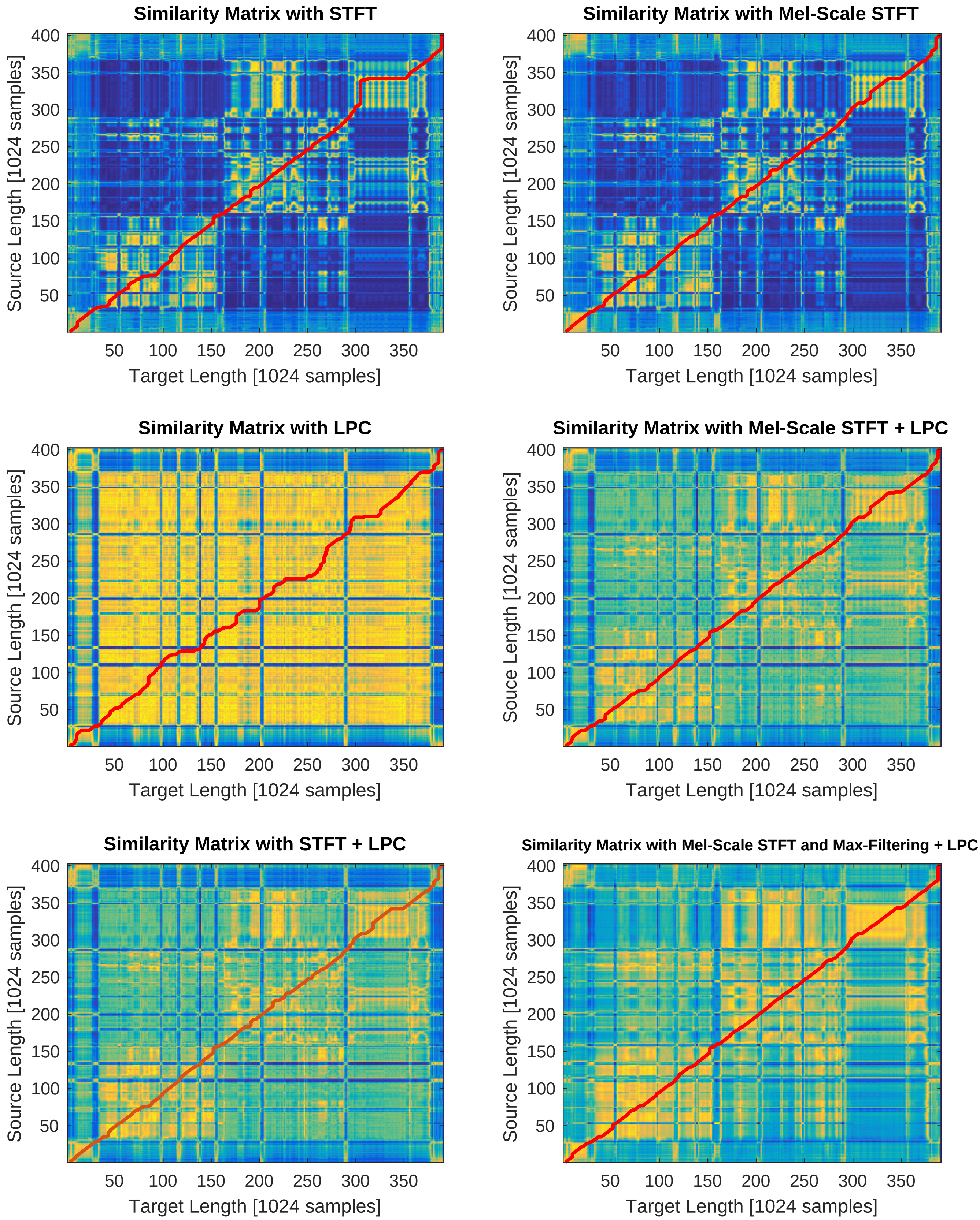

figure 4.2: *DTW path results with similarity matrices.*

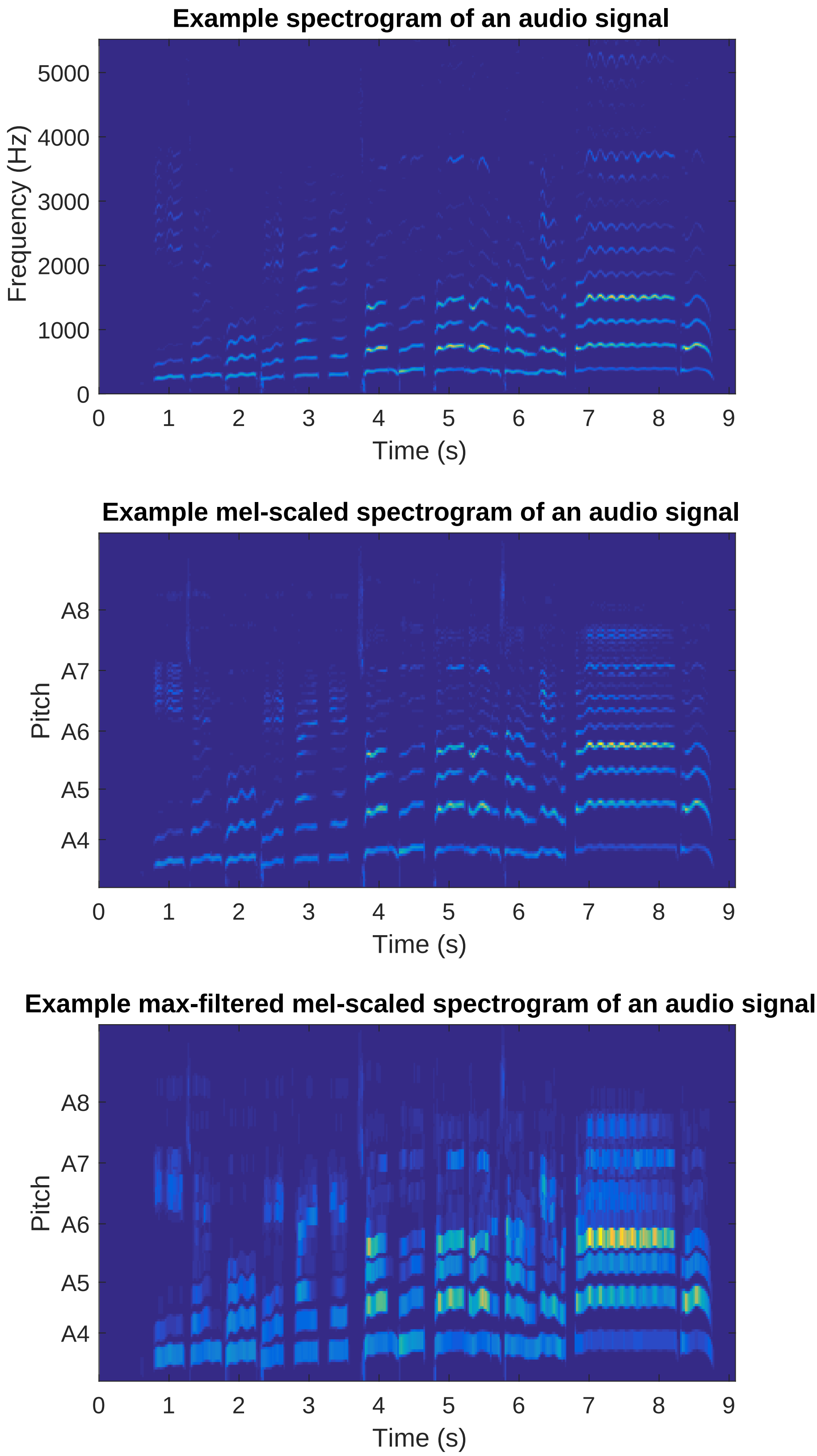

figure 4.3: *The spectrogram of an audio signal (top), the mel-scaled spectrogram of an audio signal (middle), and the max-filtered mel-scaled spectrogram of an audio signal.*

4.2.2 Smoothing Time Stretch Ratio

Given the alignment path, we need to find a sequence of stretching ratio to apply them for a time-stretching modification algorithm. Since the alignment path moves only three directions, upward, rightward, and diagonal direction every frame, we need to smooth the path such that the stretching ratio is within a reasonable range.

To apply a TSM algorithm to the source signal, we changed the DTW path into an explicit function because an explicit function is easier to apply filters and calculate time-stretching rate. To convert DTW path to an explicit function, we removed the vertical interval in the path as follows.

Algorithm 1 Removing vertical interval in the DTW path

```
expPath ← q(1), i ← 2
while i ≤ length(p) do
    if p(i) ≠ p(i − 1) then
        expPath.append(q(i))
    i ← i + 1
```

To reduce the minor detours and to make the path smoother to reduce artifacts, we first tried constrained least squares method. The basic idea of smoothing with constrained least squares is that dividing path curve into short frames and finding the polynomial curve that minimizes the difference with the path. [34]

At first, we tried constrained least squares with linear function, and define the smoothed curve p_l as the sum of the slopes derived by constrained least squares:

$$\begin{aligned} \text{minimize} \quad & \|AX - b\|_2 \\ \text{subject to} \quad & \sum A = y_{\text{end}} \end{aligned} \tag{4.3}$$

where $X(i,j)$ is the j^{th} element of the i^{th} frame, $A(k)$ and $b(k)$ is the slope and constant of the linear function optimized for k^{th} frame, and y_{end} is the last y value of the DTW path. After the optimization, we defined the smoothed curve p_l as:

$$p_l(n) = \sum_{i=1}^{n} A(i) \tag{4.4}$$

while the hop size of the frame is 1.

This optimization with constrained least squares successfully removing angled part of the original DTW path, but it also decreases path accuracy. Therefore, it was not appropriate for the system because the system needs high accuracy to transfer the pitch and the dynamics.

The second method with constrained least squares is optimizing with quadratic function. In this case, we used to connect quadratic functions derived from the optimization system. The condition of constrained least squares is as follows:

$$\begin{aligned} \text{minimize} \quad & \|AX^2 + BX - c\|_2 \\ \text{subject to} \quad & 2AX + B > 0. \end{aligned} \tag{4.5}$$

The constraints $2AX + B > 0$ means that the optimized quadratic function should not decrease in the range of the frame that the function is optimized. In this case, the optimized DTW path p_q is derived as:

$$p_q(n) = A(i) * n^2 + B(i) * n - c, i = \text{floor}(n/N) \tag{4.6}$$

while N is the hop size of the frame.

This quadratic optimization has several problems. First, it was a very high cost algorithm. Therefore, it is difficult to apply it to the system because of too much time consuming. Also, there is a disconnection in the boundary of every frames because we map different quadratic function for each frames. Finally, the system cannot find the perfect optimization with the constraint $2AX + B > 0$. In that case, the optimized curve has a decreasing range, which cannot be used for time-stretching.

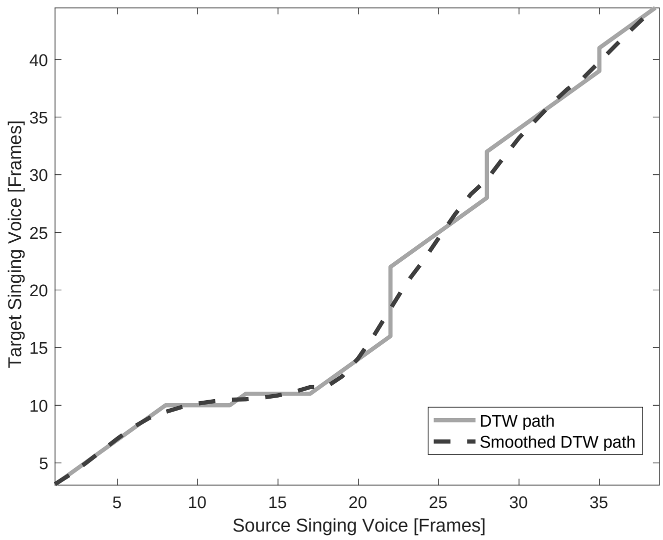

figure 4.4: *Raw path (blue) and filtered path with Savitzky-Golay filter (red).*

To smooth the DTW path with minimizing the problems of constrained least square methods, we applied 3rd-order Savitzky-Golay filter [35] to the path. Savitzky-Golay filter is a kind of filter that uses linear least squares. With the algorithm proposed by Abraham Savitzky and Marcel J. E. Golay, linear least squares of the signal can be obtained by a convolution. Therefore, we can apply a high-order least squares optimization for every frame with low cost. The effect of Savitzky-Golay filter is shown in fig. 4.4.

To calculate the time-stretching rate α, the system simply used the slope of filtered path. Since one path value corresponds to one frame, we could apply the path slope to the time-stretching rate of each frame.

When correcting rhythm based on the path information, Time-Scale Modification(TSM) algorithm is used. In this system, we used TSM Toolbox, the open source MATLAB TSM algorithm code. [20] We tried both PV-TSM and WSOLA for modifying the source signal, but the WSOLA gives the better result because WSOLA has an advantage in preserving the unvoiced signal.

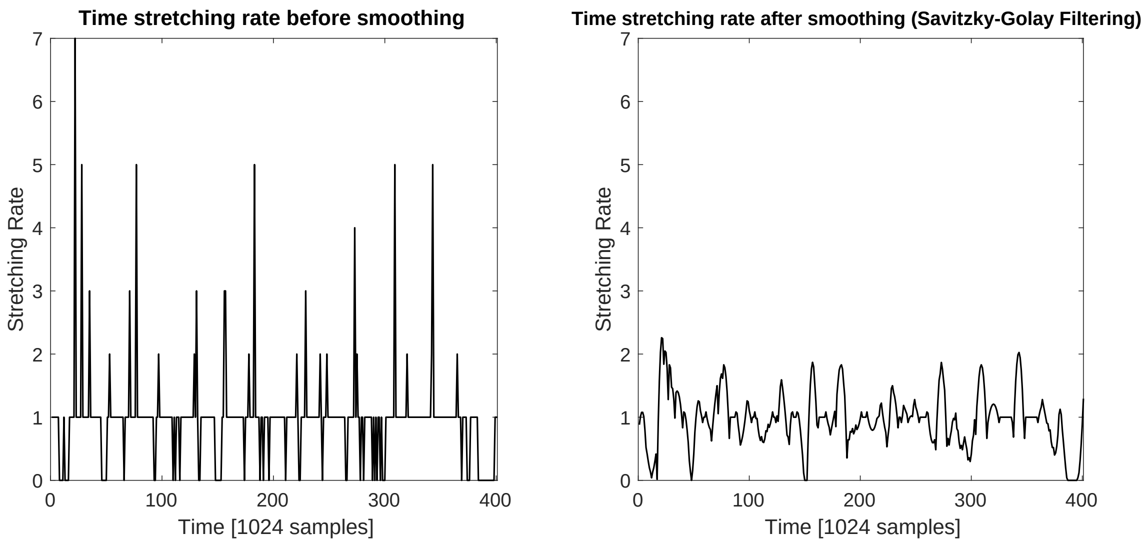

figure 4.5: *Time stretching rate before (left) and after (right) the Savitzky-Golay filtering.*

4.3 Pitch Alignment

To transfer the pitch of the target signal to the source signal, we used YIN algorithm [18] to analyze the pitch of the signals. When we analyze the pitch, the unvoiced signal is excepted because it is difficult to measure, and is natural without changing the pitch.

To separate the unvoiced signal and voiced signal, the system uses aperiodicity of the signal. The part of the signal where the aperiodicity falls below 0.2 is regarded as an unvoiced signal and excluded from the pitch analysis and transplantation.

To reduce the unvoiced signal and get the stable pitch, the system uses harmonic-percussive source separation (HPSS) with median filtering [36] to separate the percussive signal and the harmonic signal from the singing voice. The system applies YIN algorithm to the harmonic signal to extract the more stable pitch.

Since the timing problem has already been solved in the rhythm phase, it is simple to calculate the pitch that needs to be changed based on the extracted pitch information. The beta value, the pitch amount that should be changed, is calculated as follows.

$$\beta(i) = \begin{cases} f0_t(i)/f0_{s_T}(i) & \text{if } aperiodicity > 0.2 \\ 1 & otherwise \end{cases} \tag{4.7}$$

$f0$ means the fundamental frequency of the signal, and $source*$ means the rhythm modified source signal.

The PSOLA algorithm is used to modify a pitch based on the extracted pitch information because it is an algorithm that can change the pitch without resampling. Since resampling causes the the formant break and changes the timbre of the voice, using PSOLA can retain the voice timbre of the signal.

figure 4.6: *Pitch alignment.*

4.4 Dynamics Alignment

The source signal, in which both the rhythm and pitch information are modified, is finally transplanted power of the target signal. The power of the signal is extracted envelope detector, which uses rms value. In this system, we use rms value to extract envelope instead of peak value because the envelope with peak value often outputs a negative number.

$$s_{TPE}[n] = s_{TP}[n] * env_t[n]/env_{s_{TP}}[n] \tag{4.8}$$

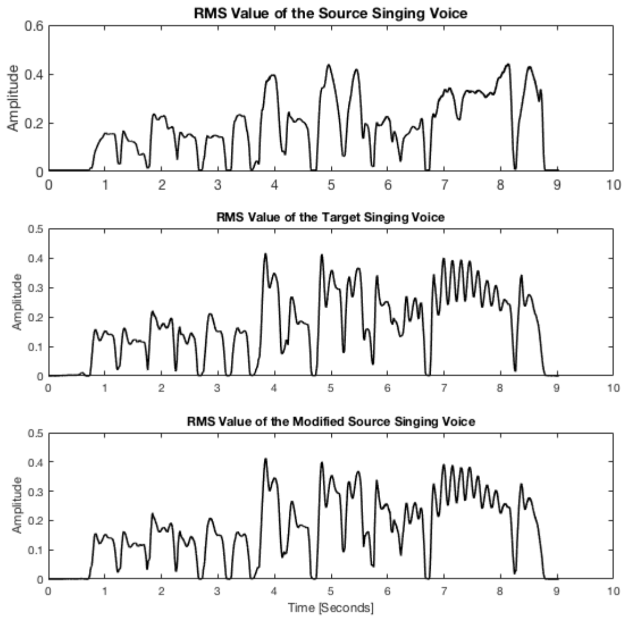

figure 4.7: *Energy alignment.*

Chapter 5. Evaluation

5.1 Datasets

In this experiment, 4 songs were used as experimental data, and totally 12 modified signals were generated using one target signal and three source signals for each song. The length of each song was about 10 seconds to 20 seconds, and only the chorus of the original song was used. All singing is recorded in the same place with same equipment. To verify that the system works well in various styles of songs, we chose the 4 songs with different styles. One of the four songs was the song for female vocals and the other three songs were male vocals. Two of the three songs with male vocal was the song with swing rhythm, and other one was the song with low pitches.

table 5.1: The list of songs used for experiment.

	song1	song2	song3	song4
gender	male	male	male	female
# of source	3	3	3	3
Remarks	swing rhythm	swing rhythm	low pitch	high pitch

5.2 Alignment Evaluation of the Converted Signal

To evaluate how well the alignment works, we tried to extract the note onset of the source, the target, and the modified source. If the average difference of note onset timing with the target decreases after the modification, it means that the alignment works well.

To extract the note onset, we used Tony, a program that extracts onset and pitch based on pYIN and hidden Markov model. [37] Because the onset detection performance of Tony is not perfect, we fine-tuned the onset timing and the number to calculate the accurate average onset difference.

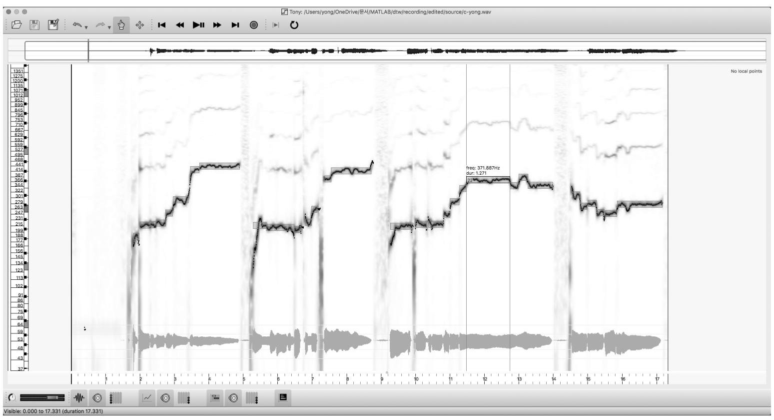

figure 5.1: *Tony, the pitch tracking and onset detection tool.*

In this experiment, we tested four cases: original source signal, aligned signal with STFT and LPC, aligned signal with mel-scale STFT and LPC, and aligned signal with mel-scale STFT, LPC, and max-filter. For each case, we have 12 examples, three per one song.

Figure 5.2 shows that aligning signal with mel-scaled STFT has less onset error comparing to aligning signal with STFT in most cases, but there are some exceptional points when there is no max-filtering. In all cases, max-filtered mel-scale STFT shows better onset error than STFT. Therefore, mel-scale STFT improves the onset error a lot, and max-filtering makes mel-scale STFT more stable.

Figure 5.4 is the average of average onset difference depending on the aligned feature. In this figure, we can see that mel-scaled STFT, max-filtering, and LPC reduces the average onset difference to 54.38% comparing to the source, and to 44.64% comparing to STFT and LPC.

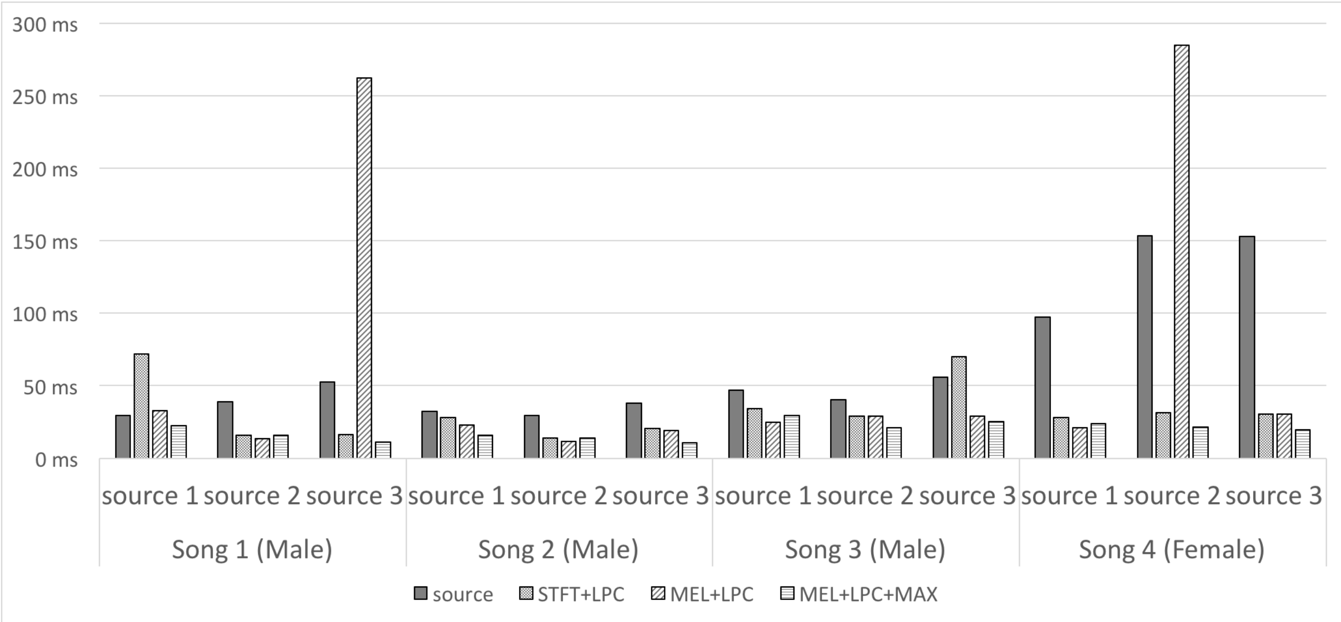

figure 5.2: *Average onset difference with target signal.*

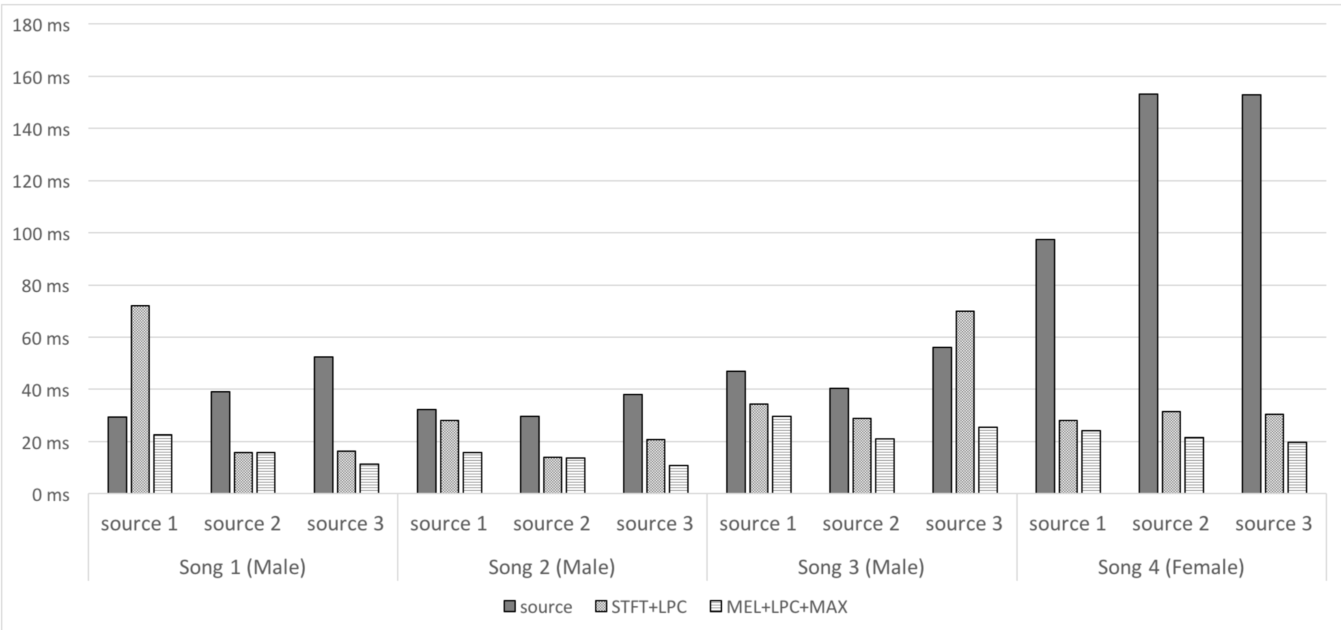

figure 5.3: *Average onset difference with target signal. (except the aligned signal with mel-scale STFT + LPC)*

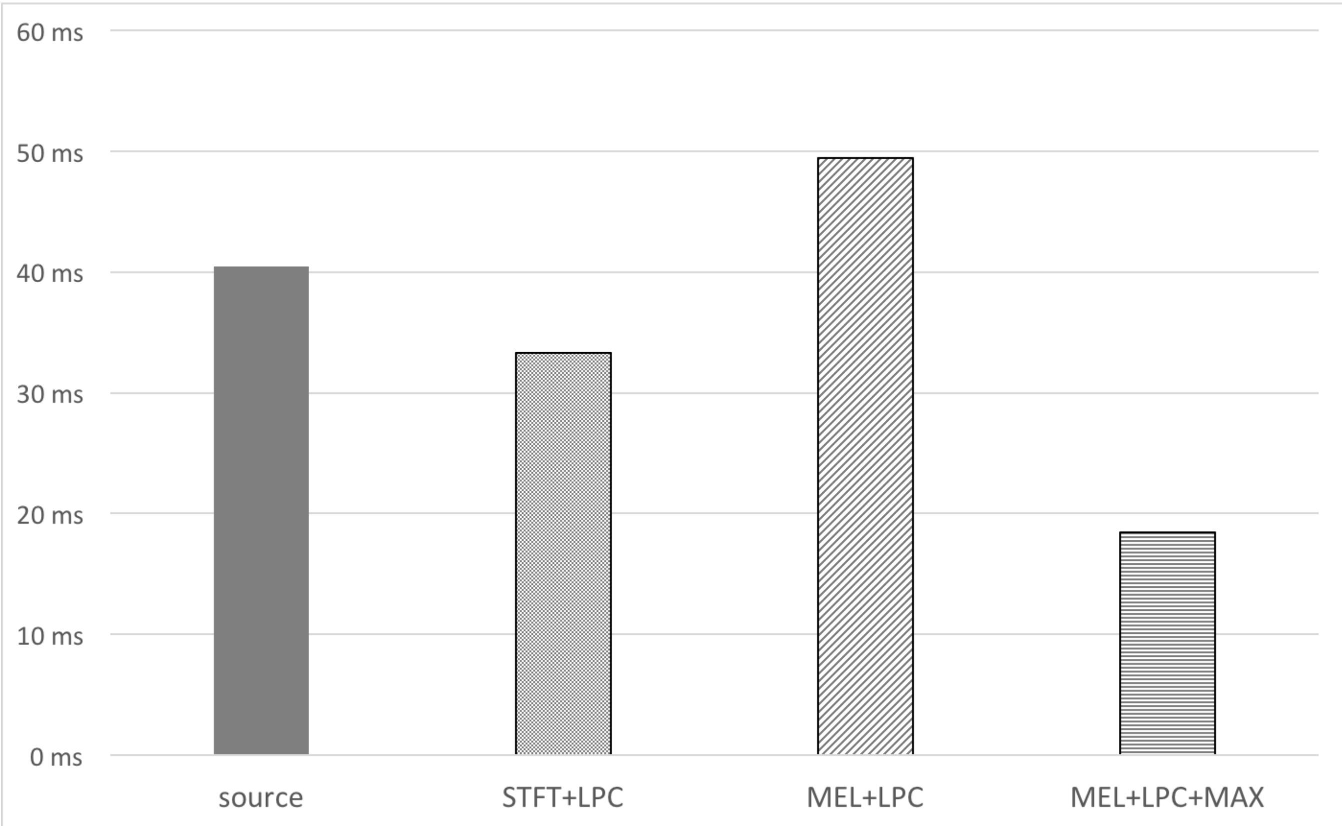

figure 5.4: *Average of average onset difference with target signal depending on the aligned feature.*

Chapter 6. Conclusion

In this paper, we proposed the system that improves the vocal expressions of the source signal through the alignment with the target signal. The system tries to improve the expressions of vocal by adjusting the alignment of the three elements of rhythm, pitch and energy using DTW, TSM algorithm and envelope detector without any additional information while maintaining the original vocal tone.

This proposed system is mainly focused on the improvement of the temporal alignment because temporal alignment is the most difficult process in the system, and the other two modules cannot work properly if the temporal alignment is not achieved.

We used STFT as a feature for finding the optimal align path first. However, the path with STFT was not accurate and has a detour problem, which means the detour path within one note because of musical expressions like vibrato. To solve this problem, we proposed a three method to reduce it.

The first one is mixing the similarity matrices S_{STFT} and S_{LPC}. Because S_{LPC} represents lyrical information, in the audio signal, S_{LPC} helps the system to find more accurate path and reduce the detour problem.

The second one is converting STFT to mel-scale. Because the perceptual pitch is proportional to the mel-scale, it is easier in mel-scaled STFT to analyze the signal in pitch range, and the amplitude of vibrato becomes equal in all harmonics. Mel-scaled STFT reduced the onset difference with the target signal effectively, but there are some exceptions that mel-scaled STFT does not work well.

To reduce the exceptions in mel-scaled STFT, we applied maximum filter in mel-scaled STFT to suppress the vibrato of the signal and reduce the effect of the pitch difference between the source signal and the target signal. Because the maximum filter improves the stability of the system, the average offset difference of examples that were not well aligned with the mel-scaled STFT decreased to the average onset difference of well aligned examples when using the maximum filter.

Before aligning the timing of the source signal with DTW path, path smoothing is needed because the DTW path is composed with only three directions: vertical, horizontal and diagonal. If the system uses DTW path directly without smoothing, only a small number of frames are excessively deformed, and it causes lots of artifacts. To solve this problem, we first tried constrained 1^{st} and 2^{nd} order least squares method, but it was too much time-consuming and there was a slight timing mismatch. Therefore, we used 3^{rd} order Savitzky-Golay filtering, which simplifies the least squares method, to smooth the path fast with high-order least squares method. To modify the signal with smoothed time stretch rate, WSOLA is used instead of PV-TSM because the singing voice is monophonic signal, and contains both harmonic and percussive components.

For transferring pitch expressions, we extracted the pitch of the temporally aligned source signal and the target signal with YIN algorithm. Before extracting the pitch contour, we applied harmonic-percussive source separation to the audio signal to separate harmonic component and percussive component of the audio signal. Pitch tracking with harmonic component slightly increases the accuracy of the extracted pitch contour. To modify the pitch of singing voice, we used PSOLA instead of other TSM methods because only PSOLA can preserve the formant of singing voices when changing the pitch of the signal.

The last process was dynamics transfer. For dynamics transfer, we extracted rms value and apply it to the source signal that the timing and the pitch is modified.

For the future works, hidden Markov model (HMM) can be applied to improve the temporal alignment. Because LPC is not enough to represent the lyrical information in the audio signal, separating the unvoiced signal and the voiced signal through HMM can provide a useful guide for aligning the note timing of the source signal. [38]

Furthermore, this research can be extended to the general musical expression transfer in different songs. Because this research is focused on the case that only the source and the target is the same song, it can be used in a limited situations. In this case, deep neural networks (DNN) can be a solution for it. Like a study for the artistic style transfer of fine art through DNN [39], the musical style transfer may be possible with DNN.

Bibliography

[1] Nicholas J. Bryan, Jorge Herrera, and Ge Wang, *User-Guided Variable-Rate Time-Stretching Via Stiffness Control*, Proc. of the 15th Int. Conference on Digital Audio Effects (DAFx), 2012.

[2] Sebastian Böck and Gerhard Widmer, *Maximum filter vibrato suppression for onset detection*, Proc. of the 16th Int. Conference on Digital Audio Effects (DAFx), 2013.

[3] Jonathan Driedger and Meinard Müller, *A Review of Time-Scale Modification of Music Signals*, Applied Sciences, 6(2), 57, 2016.

[4] Jonathan Driedger, Meinard Muller, and Sebastian Ewert. *Improving time-scale modification of music signals using harmonic-percussive separation*, IEEE Signal Processing Letters 21.1, 105-109, 2014.

[5] Werner Verhelst and Marc Roelands. *An overlap-add technique based on waveform similarity (WSOLA) for high quality time-scale modification of speech." Acoustics, Speech, and Signal Processing*, IEEE International Conference on Acousitcs, Speech and Signal Processing (ICASSP), 1993.

[6] James L. Flanagan and R. M. Golden. *Phase vocoder.* Bell Labs Technical Journal 45.9, 1493-1509, 1966.

[7] Mark Dolson. *The phase vocoder: A tutorial.* Computer Music Journal 10.4, 14-27, 1986.

[8] Jean Laroche and Mark Dolson. *Phase-vocoder: About this phasiness business.* Proc. IEEE ASSP Workshop on Applications of Signal Processing to Audio and Acoustics, 1997.

[9] Eric Moulines and Francis Charpentier. *Pitch-synchronous waveform processing techniques for text-to-speech synthesis using diphones.* Speech communication 9.5-6, 453-467, 1990.

[10] Patrick Bastien. *Voice specific signal processing tools.* 23rd International Audio Engineering Society Conference, Signal Processing in Audio Recording and Reproduction. Audio Engineering Society, 2003.

[11] Meinard Müller. *Information retrieval for music and motion.* Springer Science & Business Media, 2007.

[12] Arno Zinke and Dessislava Mayer. *Iterative multi scale dynamic time warping.* Computer graphics technical reports, #CG-2006-1, ISSN 1610-8892, 2006.

[13] Lindasalwa Muda, Mumtaj Begam, and Irraivan Elamvazuthi. *Voice recognition algorithms using mel frequency cepstral coefficient (MFCC) and dynamic time warping (DTW) techniques.* arXiv Preprint, 1003.4083, 2010.

[14] Miller S. Puckette, Miller S. Puckette Ucsd, and Theodore Apel. *Real-time audio analysis tools for Pd and MSP.* Proceedings of the International Computer Music Conference, 109-12, 1998.

[15] A. Michael Noll. *Cepstrum pitch determination.* The journal of the acoustical society of America, 41.2, 293-309, 1967.

[16] A. Michael Noll. *Pitch determination of human speech by the harmonic product spectrum, the harmonic sum spectrum, and a maximum likelihood estimate.* Symposium on Computer Processing in Communication, ed. 19, 779-797, 1970.

[17] Mohan Sondhi. *New methods of pitch extraction.* IEEE Transactions on audio and electroacoustics, 16.2, 262-266, 1968.

[18] Alain De Cheveigné and Hideki Kawahara. *YIN, a fundamental frequency estimator for speech and music.* The Journal of the Acoustical Society of America , 111.4, 1917-1930, 2002.

[19] Axel Roebel, Simon Maller, and Javier Contreras, *Transforming vibrato extent in monophonic sounds*, Proc. of the 14th Int. Conference on Digital Audio Effects (DAFx), 2011.

[20] Jonathan Driedger and Meinard Müller, *TSM Toolbox: MATLAB Implementations of Time-Scale Modification Algorithms*, Proc. of the 17th Int. Conference on Digital Audio Effects (DAFx), 2014.

[21] Matthew Roddy and Jacqueline Walker, *A Method of Morphing Spectral Envelopes of the Singing Voice for Use with Backing Vocals*, Proc. of the 17th Int. Conference on Digital Audio Effects (DAFx), 2014.

[22] Chih-Hong Yang, Pei-Ching Li, Alvin W. Y. Su, Li Su, and Yi-Hsuan Yang, *Automatic Violin Synthesis Using Expressive Musical Term Features*, Proc. of the 19th Int. Conference on Digital Audio Effects (DAFx), 2016.

[23] Pei-Ching Li, Li Su ,Yi-Hsuan Yang, and Alvin W. Y. Su, *Analysis of Expressive Musical Terms in Violin Using Score-Informed and Expression-Based Audio Features*, Proceedings of the International Symposium on Music Information Retrieval, 809-815, 2015.

[24] Tomoyasu Nakano and Masataka Goto, *VocaListener: A singing-to-singing synthesis system based on iterative parameter estimation*, Proceedings of the Sound and Music Computing Conference, 343-348, 2009.

[25] Namunu C. Maddage and Khe Chai Sim and Haizhou Li, *Word level automatic alignment of music and lyrics using vocal synthesis*, ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM), 6(3), 19, 2010.

[26] Simon Dixon, *Live tracking of musical performances using on-line time warping*, Proc. of the 8th Int. Conference on Digital Audio Effects (DAFx), 2005.

[27] Shimpei Aso, Takeshi Saitou, Masataka Goto, Katsutoshi Itoyama, Toru Takahashi, Kazunori Komatani , Tetsuya Ogata, and Hiroshi G. Okuno, *Speakbysinging: Converting singing voices to speaking voices while retaining voice timbre*, Proceedings of the 13th International Conference on Digital Audio Effects (DAFx), 2010.

[28] Takeshi Saitou, Masataka Goto, Masashi Unoki, and Masato Akagi, *Speech-to-singing synthesis: Converting speaking voices to singing voices by controlling acoustic features unique to singing voices*, IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, 2007.

[29] Meinard Müller, *Fundamentals of Music Processing: Audio, Analysis, Algorithms, Applications*, Springer, 2015.

[30] Roger B. Dannenberg, *An on-line algorithm for real-time accompaniment.*, International Computer Music Conference, Vol. 84, 1984.

[31] Xuedong Huang, Alex Acero, and Hsiao-Wuen Hon, *Spoken Language Processing:A Guide to Theory, Algorithm and System Development*, 1st ed, Prentice Hall, 290, 2001.

[32] Stanley Smith Stevens, John Volkmann, and Edwin B. Newman. *A scale for the measurement of the psychological magnitude pitch.* The Journal of the Acoustical Society of America, 8.3, 185-190, 1937.

[33] Ben J. Shannon and Kuldip K. Paliwal. *A comparative study of filter bank spacing for speech recognition.* Microelectronic engineering research conference, Vol. 41, 2003.

[34] Stephen Boyd and Lieven Vandenberghe. *Convex optimization.* Cambridge university press, 2004.

[35] Abraham Savitzky and Marcel JE Golay. *Smoothing and differentiation of data by simplified least squares procedures.* Analytical chemistry, 36.8, 1627-1639, 1964.

[36] Jonathan Driedger, Meinard Muller, and Sebastian Ewert, *Improving time-scale modification of music signals using harmonic-percussive separation*, IEEE Signal Processing Letters, 21(1), 105-109, 2014.

[37] Matthias Mauch, Chris Cannam, Rachel Bittner, George Fazekas, Justin Salamon, Jiajie Dai, Juan Bello, and Simon Dixon. *Computer-aided melody note transcription using the Tony software: Accuracy and efficiency.* Proc. 1st International Conference on Technologies for Music Notation and Representation, pp. 25-30, 2015.

[38] Joseph Picone. *Continuous speech recognition using hidden Markov models.* IEEE ASSP Magazine, 7.3, 26-41, 1990.

[39] Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. *A neural algorithm of artistic style.* arXiv Preprint, 1508.06576, 2015.

Acknowledgments in Korean

어린 시절부터 선망의 대상이었던 카이스트에 몸을 담은 지도 벌써 6년 하고도 절반이 지나가고 있습니다. 카이스트에 입학하기 이전에는 이곳에 오면 진심을 다해서 공부하고 무엇이든지 다 이루어낼 수 있을 것만 같다고 생각했었는데, 보면 볼수록 부족함과 아쉬움이 느껴지는 저의 석사 졸업 논문을 읽고 있으면 저는 과연 그때의 초심을 얼마나 간직하고 있나 되돌아보게 됩니다.

비록 보잘것없는 논문이지만, 이 글 한 편을 작성하는 데에 정말 많은 분의 도움이 있었습니다. 가장 먼저, 성심성의껏 논문 지도를 해주신 저의 지도교수님 남주한 교수님께 가장 큰 감사를 표합니다. 저의 연구와 인생을 위해 큰 비전을 제시해주신 덕분에 무사히 석사과정을 마칠 수 있었습니다. 또한, 저와 함께 해주신 MAC Lab.의 동료분들께도 큰 감사를 전합니다. 제일 먼저 이장원 선배님, 연구뿐만 아니라 연구 외적으로도 친한 동생처럼 잘 챙겨주셔서 정말 감사했습니다. 정다샘 선배님, 대학원에 입학하기 이전부터 진로 상담으로 큰 도움을 주시고 입학 이후에도 연구를 포함한 여러 가지 부분에서 많이 도와주셔서 정말 감사합니다. 천경수 선배님, 같은 프로젝트를 수행하면서 많이 도와주시고 챙겨주셔서 정말 감사합니다. 금상은 선배님, 여러 가지 어려운 일들이 있을 때 늘 웃으면서 도와주셔서 정말 감사합니다. 김태형 선배님, 항상 친동생처럼 챙겨주시고 연구를 여러 가지 조언들을 해주셔서 정말 감사합니다. 김근형 선배님, 알고 계신 것이 대단히 많으셔서 많은 가르침과 조언을 얻을 수 있었습니다. 정말 감사합니다. 박새별 선배님, 늘 막내처럼 귀여워해 주시고 챙겨주셔서 정말 감사합니다. 오창현 선배님, 연구실장으로서 연구실을 잘 이끌어 가주시고 모르는 부분에 있어서 많은 조언을 주셔서 정말 감사합니다. 김승훈 선배님, 졸업하시기 이전에도, 졸업하신 이후에도 모르는 부분들을 많이 도와주셔서 정말 감사합니다. 이종필 선배님, 항상 연구실에 긍정적인 에너지를 북돋워 주시고, 배울만한 점을 보여주셔서 정말 감사합니다. 박승순 선배님, 비록 이야기를 많이 나눠본 적은 없지만, 연구실의 구성원으로서 함께 길을 걸어주셔서 정말 감사합니다. 최순범 군, 모범적인 모습을 많이 보여줘 자극을 주고, 같이 프로젝트를 진행하며 많이 도와줘서 정말 감사합니다. 권태균 군, 모르는 부분이 있을 때 여러 가지 도움을 주었던 점 정말 감사합니다. 그리고 마지막으로 박지영 양, 옆에서 늘 열심히 하며 본받을만한 모습을 보여주고, 힘들 때 옆에 있어줘서 정말 감사합니다.

제가 여기까지 달려올 수 있었던 데에는 가족들과 친척들의 도움도 빼놓을 수 없습니다. 가장 먼저, 저를 낳아주신 부모님께 깊은 감사의 말씀을 전합니다. 저를 위해서라면 그 어떤 어려움도 불사하시고 도와주시며 나아갈 길을 열어주신 아버지, 정말 감사합니다. 저를 무한한 사랑으로 보듬어주시고 언제나 제 생각, 제 걱정을 해주시는 어머니, 정말 감사합니다. 늘 뒤에서 응원하고 있는 친동생 용자윤 양, 정말 감사합니다. 언제나 저를 응원해주시고 찾아 뵐 때면 늘 반겨주시는 할아버지 할머니, 정말 감사합니다. 언제나 많은 응원을 해주시고 절 믿어주시는 외할아버지 외할머니, 정말 감사합니다. 그 외에도 저를 응원해주시고 물심양면으로 도와주셨던 모든 친척분들 정말 감사합니다.

그 외에도 학교 생활을 하며 많은 것들을 가르쳐 주고 도와주신 덕분에 제가 한 층 더 넓은 세계를 보게 해준 김준휘 선배님, 정말 감사합니다. 입학할 때부터 기쁠 때나 슬플 때나 늘 함께 해준 룸메이트 이상혁 군, 정말 감사합니다. 처음 이 학교에 발을 디뎠을 때부터 함께 해주며 적응을 도와준 11학번 새터 13반 친구들, 정말 감사합니다. 행복한 대학원 생활을 만들어 준 문화기술대학원 동기들, 그리고 선후배 분들, 정말 감사합니다. 이 밖에 저와 함께 해주신 모든 분들께도 진심으로 감사의 말씀을 전합니다.

이 논문을 마지막으로 저의 석사과정은 끝을 맺게 되지만, 동시에 박사과정이라는 새로운 여정이 시작됩니다. 아쉬움이 많았던 석사과정을 돌아보며 박사과정이라는 까마득히 먼 길을 제가 잘 나아갈 수 있을지 새로운 길에 대한 설렘만큼이나 많은 걱정이 들지만, 지금까지 도와주셨던 모든 분의 은혜를 잊지 않고 생각하면서 끝까지 진심으로 나아가겠습니다. 정말 감사합니다!

Curriculum Vitae in Korean

이 름: 용 상 언

학 력

2008. 3. – 2011. 2. 고양외국어고등학교

2011. 2. – 2015. 8. 한국과학기술원 전기및전자공학부 (학사)

2015. 9. – 2017. 8. 한국과학기술원 문화기술대학원 (석사)

학 회 활 동

1. **Sangeon Yong**, E.J. Lee, R. Peiris, L. Chan, and J. Nam, *ForceClicks: Enabling Efficient Button Interaction with Single Finger Touch.*, Proceedings of the Tenth International Conference on Tangible, Embedded, and Embodied Interaction. ACM, Yokohama (Japan), March., 2017.
2. E.J. Lee, **Sangeon Yong**, S. Choi, L. Chan, R. Peiris, and J. Nam, *Use the Force: Incorporating Touch Force Sensors into Mobile Music Interaction.*, Proceedings of the 13th International Symposium on Computer Music Multidisciplinary Research, Proto and Matosinhos (Portugal), September., 2017 (to be published).