Vocal Hacker

Ojas Bhargave
College of Engineering
Carnegie Mellon University
Pittsburgh, PA 15213
obhargav@andrew.cmu.edu

Ravi Kiran Vadlamani
College of Engineering
Carnegie Mellon University
Pittsburgh, PA 15213
rvadla@andrew.cmu.edu

Urvil Kenia
College of Engineering
Carnegie Mellon University
Pittsburgh, PA 15213
ukenia@andrew.cmu.edu

Wallace Dalmet
College of Engineering
Carnegie Mellon University
Pittsburgh, PA 15213
wdalmet@andrew.cmu.edu

Abstract

The vocal music hacker is a speech-to-singing synthesis system that can synthesize singing voice given a speaking voice reading the lyrics of a song and a singing voice of a song. The speaking or singing voice can be same or different. The system is based on speech manipulation system WORLD and models controlling the acoustic features - fundamental frequency(F0), Phoneme duration, spectral envelope. Given the singing voice, the F0 control model extracts F0 contour from reading audio and replaces it with that of singing audio. The duration control model uses synchronization information to control duration of phonemes. If the singing voice is different from speaking voice, timbre transfer is done by matching the unaligned phonemes along with the F0 control model and the duration control model. Our experiments show that reading lyrics can be converted to singing which is almost similar to a real singing voice.

1 Introduction

The goal of this project is to generate singing voice by reading the lyrics of a song and controlling three acoustic features - fundamental frequency (F0), phoneme duration and spectral envelope. In terms of a superposition of sinusoids, the fundamental frequency (F0) is the lowest frequency sinusoidal in the sum of harmonically related frequencies or the frequency of the difference between adjacent frequencies. The spectral envelope is the envelope of the amplitude spectrum. Periodicity indicates the duration for which each phoneme was spoken. Based on our research and experiments by manipulating these three features we can generate a song by just reading its lyrics given the original song.

Recent approaches mainly focus on text-to-song synthesis by using neural networks and GANs to generate a singing voice from scratch, however, our project deals with converting a speech audio reading the lyrics of a song to the tune of singing voice given the song. This problem is interesting as it helps us to delve deeper to investigate and understand the acoustic differences between speaking and singing voices. It also has practical applications for computer-based music productions where the pitch of singing voices is often manipulated (corrected or intentionally modified but their naturalness is sometimes degraded) [1]. Most of the current work done on this problem use neural networks that have millions of parameter and several hyper-parameters to tune. In this project, we have made an attempt to follow the traditional machine learning approaches to manipulate the reading voice for converting it into a song which tremendously reduces the model complexity and the number of parameters when compared to a deep neural network. Furthermore, our project will also enable people to sing their favorite songs just by reading its lyrics, even if they do not know how to sing.

2 Literature Review

Speech analysis/synthesis systems have been used in various kinds of applications such as voice conversion and statistical parametric speech synthesis[1]. These applications use a high-quality system based on a vocoder. Traditionally, STRAIGHT vocoder was mainly used to convert speech to features. However, the STRAIGHT vocoder was surpassed in terms of performance by the WORLD vocoder. The MUSHRA-based evaluation result showed that the current version of WORLD could achieve the best performance. In the analysis of a speech signal, there are several speech waveforms for which STRAIGHT cannot estimate the fundamental frequency (F0). YANG vocoder could not totally achieve natural speech because of the low-accuracy of the spectral envelope.[2]

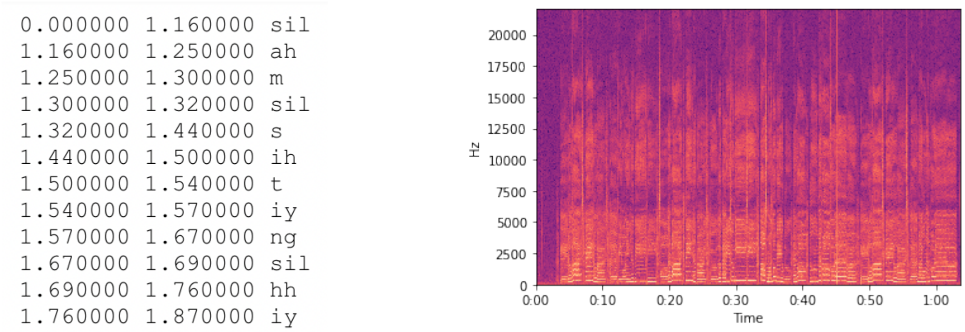
WORLD consists of three algorithms to obtain three speech parameters and a synthesis algorithm that takes these three as input[3]. The first algorithm estimates the fundamental frequency (F0) contour based on period detection of the vocal fold vibration, so it does not require expensive computation such as STFT or autocorrelation[4]. The second algorithm estimates the spectral envelope. The concept of the algorithm is to obtain an accurate and temporally stable spectral envelope. The algorithm uses fundamental frequency (F0) and consists of F0-adaptive windowing, smoothing of the power spectrum, and spectral recovery in the frequency domain[5]. To develop a high-quality vocal synthesizer, a voice synthesis system that can manipulate pitch and timbre without sound quality deterioration is required. The third algorithm achieves this goal through the signal excitation method[6].

Further, speech to singing conversion is possible by manipulating three acoustic features obtained from WORLD. The duration control model lengthens the duration of each phoneme in the speaking voice by considering the duration of its musical note. The spectral control model converts the spectral envelope of the speaking voice into that of the singing voice by controlling both the singing formant and the amplitude modulation of formants in synchronization with vibrato[1].

3 Datasets

In this project, the chosen dataset is the NUS-48E Sung and Spoken Corpus developed at Sound and Music Computing Laboratory at National University of Singapore. The corpus is a 169-min collection of 48 (20 unique) English songs comprising 25,473 phone instances with complete set of transcriptions and duration annotations at the phone-level for all the audio recordings.

The recordings correspond to 12 unique singers, each singing and reading various songs which are present as '.wav' files and their time-aligned phoneme-level annotations as '.txt' files. An example of the annotation '.txt' file can be seen in Fig. 1.(a), where the first column is the start duration (in seconds) of the phoneme in the third column and the second column indicates the stop duration (seconds). An input speech '.wav' file transformed into a spectrogram is shown in Fig.1.(b).



```
0.000000 1.160000 sil
1.160000 1.250000 ah
1.250000 1.300000 m
1.300000 1.320000 sil
1.320000 1.440000 s
1.440000 1.500000 ih
1.500000 1.540000 t
1.540000 1.570000 iy
1.570000 1.670000 ng
1.670000 1.690000 sil
1.690000 1.760000 hh
1.760000 1.870000 iy
```

(a) Annotation File (start (s), stop (s), phoneme)

(b) Spectrogram for speech audio

Figure 1: Example of dataset

4 Evaluation Metrics

We are using two evaluation metrics for our project. First one is a similarity score obtained by calculating a normalised dot product between the magnitude of the spectrogram of the expected audio and the magnitude of the spectrogram of the re-synthesized output. The similarity score ranges from 0 to 1 with 1 (since $cos(0) = 1$) being most similar to the original audio file.

The Similarity score is computed as follows:

$$Similarity\ \ score = \frac{STFT(OriginalAudio).STFT(Re-synthesizedAudio)}{|STFT(OriginalAudio)| \times |STFT(Re-synthesizedAudio)|} \tag{1}$$

Similarity score only indicates how similar two spectrograms are but does not give any information about the perceived naturalness of a song through the human senses. To overcome this problem, we used the naturalness index as our second evaluation metric. It is the degree of the naturalness of singing voices synthesized by our models. For every song, 10 graduate students, with normal hearing ability, rated the naturalness of the synthesized output between 0 to 1. The average value of their ratings is our naturalness indicator.

However, using only naturalness index may make our results look subjective and may induce errors due to bias. Therefore, we decided to use both similarity score and naturalness index together as our evaluation metric.

5 Method

A block diagram of our vocal hacker system is shown in Fig. 2. The system takes two audio files as input. The first audio is of a person reading the lyrics of a song and the second audio is of a person singing the same song. The annotation files in the NUS-48E dataset provides us with the duration of the audio files where each phoneme occurs. The number of phonemes in the spoken audio may or may not match with that of the singing audio so we use the information in the annotation file to synchronize and match the phonemes in the spoken audio as per the sung song based on our algorithm. The fundamental frequency (F0) control model and the duration control model that we have used in our vocal hacker system are described below:

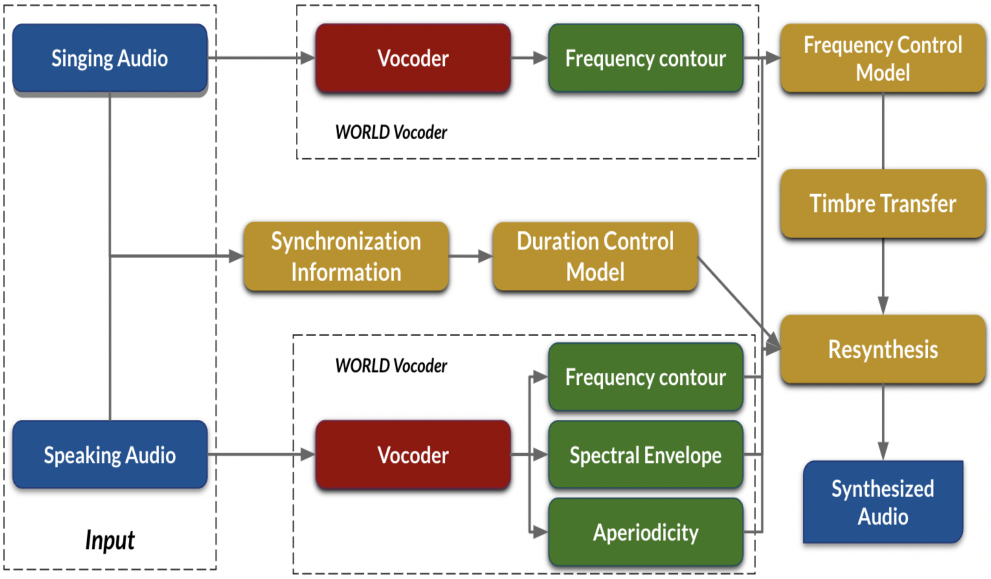
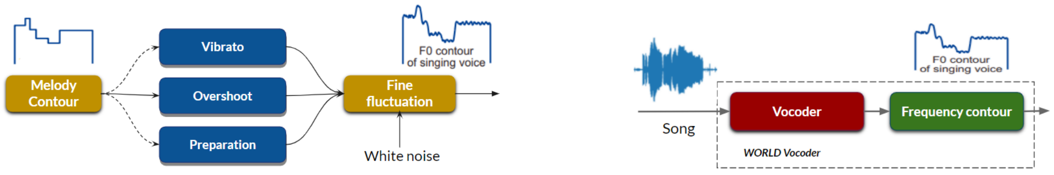


Figure 2: Block Diagram of the Vocal Music Hacker System

5.1 F0 Control Model

In our speech to singing synthesis, we first discarded the F0 contour of the speaking voice and replaced it by the F0 contour of the singing voice. In general, if the target song is not available, the F0 contour of the singing voice can be obtained by adding F0 fluctuations such as overshoot, vibrato, preparation to a melody contour of artificially created notes. However, in our case, the target song is available and we can directly get the F0 contour from it using the WORLD vocoder. This creates a mismatch in the size of the F0 contours thus obtained. We further shift the frequency base to the reader's audio frequency and fuse that frequency contour with the reading voice. A general F0 control model when the target song is not available is shown in Fig .3(a) and our F0 control model for the case when target song is available is shown in Fig. 3(b). The model is then passed through the duration control model to match each of the phonemes with the corresponding music.



(a) When target song is available

(b) When target song is not available

Figure 3: F0 control models

5.2 Duration Control Model

In duration control we used the synchronization data available in the dataset and the algorithm we developed, to extract the phoneme samples from the audio files and control the phoneme duration. Because the duration of each phonemes are different in singing and reading, we need to elongate or shrink the duration of each phonemes. For this, we developed an algorithm based on the dataset that identifies the amount of elongation/shrinkage needed for each phoneme in the reading audio. Based on the phoneme duration in the reading and singing data, the algorithm calculates the ratio with which the phoneme in the reading audio file needs to be changed and then maps the phonemes duration from the reading data to that of the singing data. The phonemes duration is changed as per the ratio below:

$$T_n = T_o * (R_t / S_t) \tag{2}$$

T$_n$: Duration to which we need to elongate or shrink each phoneme

T_o: Current duration of the phoneme

R_t: Duration of phoneme while reading

S_t: Duration of phoneme while singing

6 Experiments

6.1 Vanilla NMF for timbre transfer

Timbre transfer is when we are trying to control the voice of person A into that of the song sung by person B. In our naive approach, we found a transformation matrix T to transform the spoken voice clip to song sung by the same person. To achieve this, we computed the product of pseudo-inverse of spoken audio file's magnitude spectrogram matrix and singing magnitude spectrogram matrix, to get the best approximation of the transformation matrix as shown in equation (3) below:

$$T = Mr^{+}Ms \tag{3}$$

Using this transformation matrix, we calculated the reconstructed output as follows:

$$M_{out} = TMr \tag{4}$$

This method did not yield good results and the L2 norm of error between the reconstructed song and the original song turned out to be $1.23 * 10^8$.

We then experimented with the Non-negative matrix factorization (NMF) technique on the training data to obtain the basis and weight matrices of the song as follows:

$$Ms_1 = Bs_1Ws_1 \tag{5}$$

Similarly, we obtained basis and weight matrices of the spoken lyrics audio file.

$$Mr_2 = Br_2Wr_2 \tag{6}$$

From equations (1) and (2), we got the reconstructed audio output as shown in equation (7) below:

$$Ms_2 = Br_2Ws_1 \tag{7}$$

Using the NMF technique. the final L2 error turned out to be $5.58 * 10^6$.

6.2 Vocal Music Hacker System with F0 control model

First, we convert both the singing voice and speaking voice into three acoustic features - F0 contour, spectral envelope and aperiodicity. We then discard the F0 contour of the reading voice. If there is a change of gender, the frequencies of female voices are generally higher. Therefore, we took the difference of the normalized mean and set a maximum threshold of 5. Here is the pseudo code for the vocal music hacker with only F0 control model:

Algorithm 1 F0 control model

```
$f01,\ sp1,\ ap1 \leftarrow WORLD(Reading\ Voice)$
$f02,\ sp2,\ ap2 \leftarrow WORLD(Singing\ Voice)$
if Gender Changed then
    if $f01 \geq (f02 - mean(f02))\ +\ 5$ then
        $f01 \leftarrow f01 + 5$
    end if
    if $f01 \leq (f02 - mean(f02))\ -\ 5$ then
        $f01 \leftarrow f01 - 5$
    end if
    Synthesized Output = $WORLD(f02,\ sp1,\ ap1)$
end if
```

6.3 Vocal Music Hacker System with both F0 and Duration control models

The short coming of using only the F0 control model is that the phoneme duration is unchanged during reading and singing. This resulted in the synthesized output sounding unnatural. Practically, in singing, different phonemes are supposed to be of different lengths in tune with the pitch of the song. This was not the case with the results of the above model. Hence, the output sounded more like faster reading than singing.

To overcome this issue we employed a duration control model along with the F0 control model. The duration control model makes an ordered list of all the phonemes for which the duration needs to be changed. For these phonemes the duration of the phonemes in the song is used so that it is in sync with the F0 of the song. Here is the Pseudo code for the duration control model:

Algorithm 2 F0 control model with duration control

```
$th \leftarrow 5$
$ph1 \leftarrow Read\ Audio\ Phoneme\ List$
$ph2 \leftarrow Sing\ Audio\ Phoneme\ List$
while $l \leq len(ph2)$ do
    if (phoneme in $ph1$ and $ph2$) and (location $\leq th$) then
        $ph \leftarrow$ (phoneme)
    end if
    $l \leftarrow l + 1$
end while
for all phonemes in $ph$ do
    Get duration of each phoneme from singing voice
    Elongate/Shrink corresponding reading voice
end for
```

7 Results

The results of NMF were discouraging, the NMF technique was not able to recognise the order of phonemes. The phonemes were mostly out of sync with fundamental frequency and we were unable to separate the fundamental frequencies and the phonemes into the feature matrices. The synthesized output was mostly noise and not intelligible.

The results of the Vocal Music Hacker with F0 control model were promising compared to vanilla NMF and are tabulated in Table.1:

Table 1: F0 control model

Singer	Reader	Song	Similarity	Naturalness
Adiz	Adiz	Edelweiss	0.74	0.70
Mpur	Mpur	Sitting Here	0.75	0.70
Kenn	Kenn	Goodbye	0.70	0.75

The results of the Vocal Music Hacker with both F0 control model and Duration Control Model are tabulated in Table. 2:

Table 2: F0 control model with duration control

Singer	Reader	Song	Similarity	Naturalness
Adiz	Adiz	Edelweiss	0.84	0.85
Mpur	Mpur	Sitting Here	0.89	0.85
Kenn	Kenn	Goodbye	0.75	0.70

The results of Vocal Music Hacker with both F0 control model and Duration control model for timbre transfer are tabulated in Table. 3:

Table 3: F0 & duration control model for timbre transfer

Singer	Reader	Song	Similarity	Naturalness
Mpol	Pmar	Edelweiss	0.75	0.60
Jlee	Vkow	Sitting Here	0.72	0.60
Mcur	Kenn	Goodbye	0.68	0.58

The results obtained by employing both F0 control model and duration control model are promising and for a person just reading the lyrics of a song, our algorithm will be able to convert his reading voice into a singing voice with about 74% naturalness.

8 Discussion and Analysis

Following are the areas in which our project could be improved and extended to in the future:

8.1 Spectral Envelope Control and Timber Transform

Even after applying the frequency control model and the duration control model, our algorithm does not perform well when the reader and singer are different individuals, as they have different voices which needs to be fused and re-synthesized, this results in bad outputs for timbre transform. To alleviate this issue, an attempt can be made to modify the spectral envelope of the speaking voice by controlling the spectral characteristics unique to the singing voices.

8.2 Accent Correction

One of the major challenges of our project was to capture the information from different accents. Different accents have different durations of phonemes so they have a major impact during phoneme prediction. One approach to overcome this issue would be to create a phoneme matching dictionary that would store phonemes that sound similar, as different accents have similar-sounding phonemes, we can then modify our existing algorithm to take into account the phonemes that sound similar based on our dictionary within a pre-specified value of the threshold.

8.3 Better Phoneme Prediction

Our XGBoost model had an accuracy of 67.4% for phoneme prediction. With this model's output, the re-synthesized song had a lot of noise. The model could be improved further by maybe hypertuning the parameters or using XGBoost in conjunction with other methods or using an ensemble of classifiers.

8.4 Real-time Conversion

We are not using any Deep Learning technique like GANs so the model does not have a lot of parameters to be trained and tuned. The algorithm used in our project is fairly fast which can make it possible to process real time data and make it available using API.

8.5 De-noising

Our current observation suggests that if the input audio files are noise-free, the re-synthesis does not capture much noise. Though we made sure that any additive noise does not get added during the processing, a de-noising block in the beginning of the pipeline would help the output to be more natural sounding. There could also be a benefit of adding a de-noise block after synthesizing the output.

9 Github

Vocal Hacker Github Repository

10 Division of Work

Wallace Dalmet: F0 control with duration control implementation, baseline implementation and literature review

Urvil Kenia: Dataset research, preprocessing, baseline evaluation, and XGBoost implementation

Ojas Bhargave: Timbre transfer implementation, evaluation metrics research and literature review

Ravi Kiran Vadlamani: F0 control model implementation, algorithm designing, model tuning and literature review

References

[1] M. Morise and Y. Watanabe: Sound quality comparison among high-quality vocoders by using re-synthesized speech, Acoust. Sci. Tech., vol. 39, no. 3, pp. 263-265, May 2018.

[2] M. Morise, F. Yokomori, and K. Ozawa: WORLD: a vocoder-based high-quality speech synthesis system for real-time applications, IEICE transactions on information and systems, vol. E99-D, no. 7, pp. 1877-1884, 2016.

[3] M. Morise, H. Kawahara and H. Katayose: Fast and reliable F0 estimation method based on the period extraction of vocal fold vibration of singing voice and speech, AES 35th International Conference, CD-ROM Proceeding, Feb. 2009.

[4] M. Morise: CheapTrick, a spectral envelope estimator for high-quality speech synthesis, Speech Communication, vol. 67, pp. 1-7, March 2015.

[5] M. Morise: D4C, a band-aperiodicity estimator for high-quality speech synthesis, Speech Communication, vol. 84, pp. 57-65, Nov. 2016.

[6] Ma, X., Wang, Y., Kan, M. Y., Lee, W. S. (2021). AI-Lyricist: Generating Music and Vocabulary Constrained Lyrics.

[7] Chandna, P., Blaauw, M., Bonada, J., Gómez, E. (2019, September). Wgansing: A multi-voice singing voice synthesizer based on the wasserstein-gan. In 2019 27th European Signal Processing Conference (EUSIPCO) (pp. 1-5). IEEE.

[8] Hono, Y., Hashimoto, K., Oura, K., Nankaku, Y., Tokuda, K. (2021). Sinsy: A Deep Neural Network-Based Singing Voice Synthesis System. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 29, 2803-2815.

[9] Mirbeygi, M., Mahabadi, A., Ranjbar, A. (2021). RPCA-based real-time speech and music separation method. Speech Communication, 126, 22-34.

[10] Takeshi saitou et al. SPEECH-TO-SINGING SYNTHESIS: CONVERTING SPEAKING VOICES TO SINGING VOICES BY CONTROLLING ACOUSTIC FEATURES UNIQUE TO SINGING VOICES