Instrument Timbre Transformation using Gaussian Mixture Models and Centroid and Matrix Transformation

Yu-Chen Huang
Carnegie Mellon University
California, CA 94035
yuchenhu@andrew.cmu.edu

Hung-Kuang Han
Carnegie Mellon University
California, CA 94035
hungkuah@andrew.cmu.edu

Yi Wang
Carnegie Mellon University
Pittsburgh, PA 15213
wangyi@andrew.cmu.edu

1 Introduction

Timbre is the characteristic of instruments. People use timbre to identify different musical instruments. Sometimes, people may want to hear a piece with different instrumentation. Timbre transformation would expand the scope of sounds of any existing music, for instance, we could transform a piece of classical guitar performance into one performed with electric guitar without having the musician playing it all over. Together with the technique of instrumentation separation, we could apply timbre transformation on each individual instrument and yield a high-quality remix of any existing piece of music.

2 Related Work

In the previous work of [1], Settel, et al.(1994) use FFT/IFFT in real time to conduct digital signal processing in Max programming environment, which requires no compilation for digital signal processing(DSP). They use what's called overlap-add technique, including the following steps: (1) windowing input signal (2) transformation of the input signals into the spectral domain using FFT (3) operate on signal's spectra (4) resynthesis of modified spectra using IFFT (5) windowing the output signal. Their operation in the spectral domain includes convolution, addition, square root. We want to apply similar procedures for our timbre transformation project on data from Megenta's NSynth.

3 Dataset

After loading NSynth Dataset of guitar family[4], we obtain train, test and validation sub directories. Within each sub directory, audio files is formatted as: guitar_source_identifier_pitch_velocity.wav, (e.g.,guitar_acoustic_000-021-025.wav)

In each sub directory, we have 3 sources: acoustic, electronic and synthetic; number of identifiers varies between sources, pitch ranges from 21 to 108 and 5 velocity options: 25, 50, 75, 100, 127.

4 Method

Our goal is to transform audio of original source S (acoustic) to target source T (electronic). To simplify the problem for now, we take S and T to be of the same family, which is the guitar and keyboard family in our experiment.

Submitted to 34th Conference on Neural Information Processing Systems (NeurIPS 2020). Do not distribute.

4.1 Approach 1: Gaussian Mixture Models

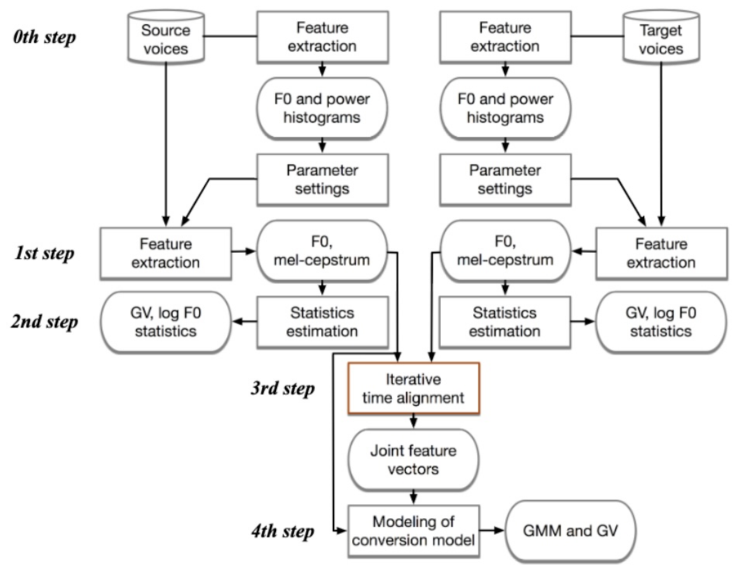


Figure 1: Gaussian Mixture Models

We apply a Voice Conversion toolkit “sprocket”[5] to our dataset. Sprocket first extracts acoustic features like F0 and the global variance(GV), then it estimates statistics of the extracted acoustic features, such as the mean and standard deviation.

To model a joint probability density function based on the GMM, frame-aligned joint feature vectors are extracted with an iterative Dynamic Time Wrapping(DTW) in sprocket. The joint probability density function based on the GMM is trained as the conversion model for the conversion process using the refined joint feature vectors.

Sprocket transforms acoustic features with GMM from the original sound. Then it converts the original sound into the converted sound by utilizing excitation generation and the mel log spectral approximation (MLSA) filter (i.e., a vocoder) based on transformed acoustic features.

4.2 Approach 2. Centroid and Matrix Transformation

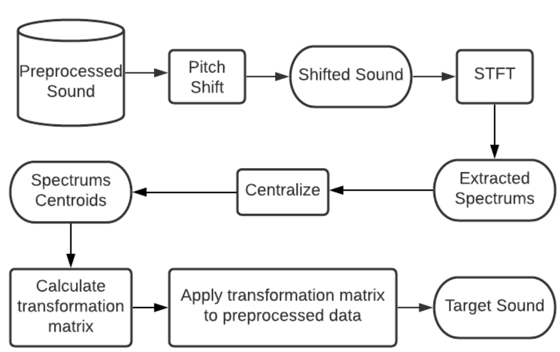


Figure 2: Centroid and Matrix Transformation

We model our goal as we would like to obtain a transformation matrix W that would transform the original source S to an approximated spectrum of target source as T' which would be close to T.

To obtain such transformation matrix, our idea is to find spectrum centroid of S and T, s and t. Then use s and t to compute the transform matrix W between the two spectrum centroids. Our method to compute such transformation W is taking the pseudo inverse of one:

$$W = Pinv(s) \times t. \tag{1}$$

Since we want our centroids to focus on timbre information and not be diverged by varying pitches in our training data set. We make an assumption: performing pitch-shift on our data would not cause too much loss on its timbre information. Based on this assumption, we pitch-shifted our training set to pitch center, and used a range of pitch-shifted data with the original pitch centered around the pitch center for computing centroids.

In our experiment, we tried different pitch centers and different range of pitch range around the pitch center to obtain better centroid representation. The centroid representation would give us a better transformation matrix. To choose the best parameter for pitch center and range, we were making our judgments based on the subjective listening experience of the timbre transformed version and the target timbre audio samples.

Our method for computing centroids is quite naive and this is definitely the part we want to improve on. For now, the centroid of each source is computed by adding all spectrum in the pitch-shifted training data set then take its mean. The resulting centered audio suffered recognizable loss phase cancellation.

5 Evaluation metrics

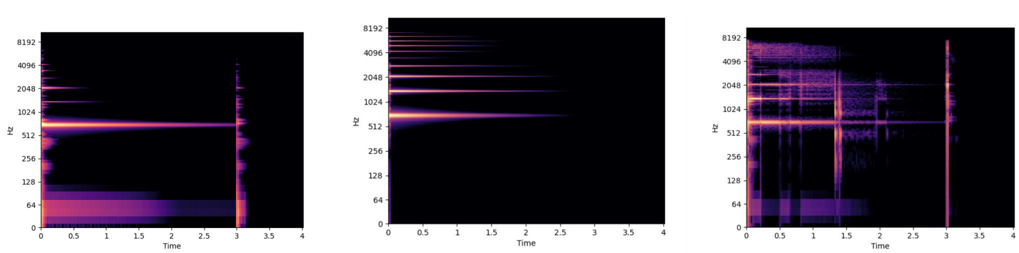
We use Mel cepstral distortion (MCD) [6] as evaluation metric because it's a popular objective measure for evaluating the timbre similarity [7]. MCD represents the distance between the MFCC feature of spectrum of the transformed electric guitar sound set and the standard electric guitar sound set, and the formula is as follows:

$$MCD(y - \hat{y}) = \frac{10\sqrt{2}}{ln10} \|y - \hat{y}\|_2 \tag{2}$$

Where y is the standard sound, $\hat{y}$ is the transformed sound and the coefficient in front of the norm is to convert the unit to decibels.

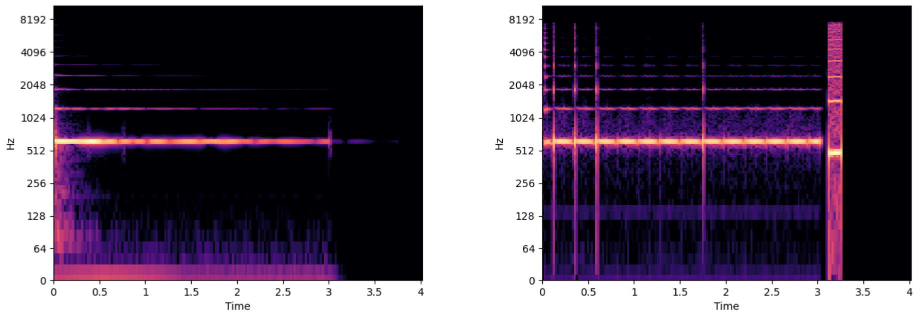
6 Results

6.1 Result of Gaussian Mixture Models



(a) source sound (b) target sound (c) transformd sound

Figure 3: guitar family



(a) target sound (b) transformd sound

Figure 4: keyboard family

We trained the sprocket on the parallel dataset to learn the model. The result generated from the sprocket has successfully erased the lower frequency of the source sound (Figure 3 (a)).

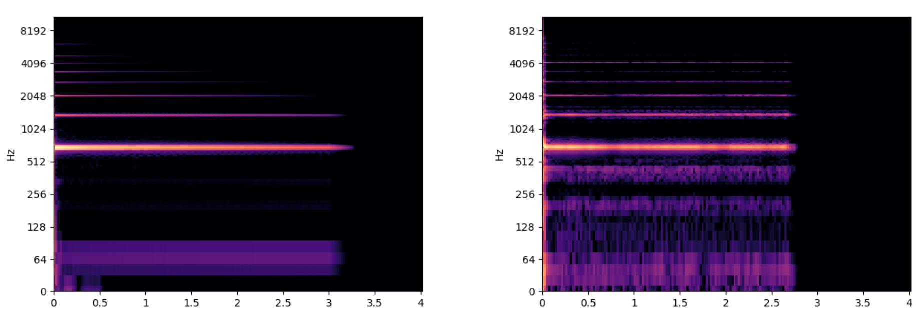
We then calculated the average norm distance between MFCC feature of the source sound S and the target sound of same pitch, T.The average MCD distance between S and T is 2450.

Since the source sound has a spike near its end, sprocket moved as much lower frequencies to the higher frequencies as possible (Figure 3 (c)).

However, almost all of our acoustic guitar source sounds contains a clap-ish sound at their own timing; sprocket would falsely learn to generate an arbitrary spike in the middle of the spectrogram for all transformed sounds.

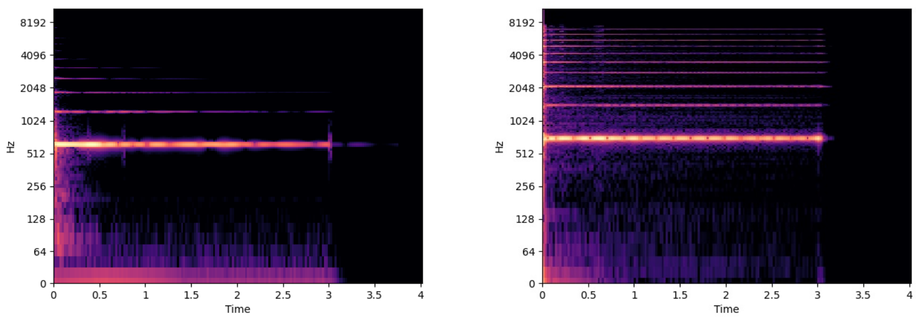
We also did the transformation on the keyboard sounds (See figure4 (a) and (b)).

6.2 Result of Centroid and Matrix Transformation



(a) target sound (b) transformed sound

Figure 5: guitar family



(a) target sound (b) transformed sound

Figure 6: keyboard family

In our experiment, we take the original source S as acoustic guitar and keyboard samples and target source T as electronic guitar and keyboard samples.

After getting the transformation matrix W between spectrum centroids of s and t, we would apply W to transform acoustic source to electronic source in the testing data set.

We tested our transform on the testing set by randomly selecting acoustic audio file S and electronic audio file T of the same pitch. Then conduct the transform on the S and collect the transformed audio T'. We then calculated the average norm distance between MFCC feature of T' and the electronic source audio of same pitch, T. The average MCD distance between T and T' of guitar is 849. The average MCD distance of keyboard family is 1708.

Figure 5(a) and 6(a) are the spectrum of one of the electronic guitar and keyboard source audio from testing data set, figure 5(b) and 6(b) is the spectrum of the transformed electronic guitar and keyboard from one of the acoustic source audio of the same pitch from testing data set.

7 Discussion and analysis

Table 1: Compare 2 approach in different instrument family

MCD Distance	Keyboard	Guitar
GMM (sprocket)	3574	1708
Centroid and Matrix Transformation	2450	849

In this timbre transformation task, the difficulty lies not only in how to extract the feature that are sufficient to represent timbre for transformation, but also in the need to use these feature to synthesize the transformed audio. In table 1, we can see that our Centroid and Matrix Transformation (see section 4.2) has better performance than GMM(sprocket) (see section 4.1) in keyboard and guitar family(both transform from acoustics source to electronics source).

One reason is that our Centroid and Matrix Transformation uses all the frequency spectrum as the feature, and there is almost no loss of information during the transform process. Therefore, except that the synthesized transformed audio sounds dull because of pitch distortion due to some aliasing(because of pitch shifting), most of the timbre-related information is reserved to make the audio sound more fidelity.

On the other hand, we find it to be difficult to train a GMM to model all different guitars because even in the same instrument family, different guitars have different timbre.

For the next improvement, using other ways of centralize, such as K-means on dimension reduction data samples to obtain multiple centroids, which could enhance the generalizing ability of Centriod and Matrix Transformation method.

References

[1] Settel, Z., & Lippe, C. (1994). Real-time timbral transformation: FFT-based resynthesis. Contemporary Music Review, 10 (2), 171-179.

[2] Wakabayashi, Y., Fukumori, T., Nakayama, M., Nishiura, T., & Yamashita, Y. (2017, March). Phase reconstruction method based on time-frequency domain harmonic structure for speech enhancement. In 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 5560-5564). IEEE.

[3] Yoshii, K., Tomioka, R., Mochihashi, D., & Goto, M. (2013, November). Beyond NMF: Time-Domain Audio Source Separation without Phase Reconstruction. In ISMIR (pp. 369-374).

[4] Engel, J., Resnick, C., Roberts, A., Dieleman, S., Norouzi, M., Eck, D. & Simonyan, K.. (2017). Neural Audio Synthesis of Musical Notes with WaveNet Autoencoders. <i>Proceedings of the 34th International Conference on Machine Learning</i>, in <i>Proceedings of Machine Learning Research</i> 70:1068-1077 Available from https://proceedings.mlr.press/v70/engel17a.html.

[5] Kobayashi, K., Toda, T. (2018, June). sprocket: Open-Source Voice Conversion Software. In Odyssey (pp. 203-210).

[6] Kubichek, R. (1993, May). Mel-cepstral distance measure for objective speech quality assessment. In Proceedings of IEEE Pacific Rim Conference on Communications Computers and Signal Processing (Vol. 1, pp. 125-128). IEEE.

[7] Kim, J. W., Bittner, R., Kumar, A., & Bello, J. P. (2019, May). Neural music synthesis for flexible timbre control. In ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 176-180). IEEE.