PROJECT SPECIFICATION FOR CSC 475 2025: TRANSVOX

Isaiah Doyle
University of Victoria
isaiahdoyle@uvic.ca

Aileen Klassen
University of Victoria
aileenklassen@uvic.ca

Elijah Larmer
University of Victoria
elijahlarmer@uvic.ca

ABSTRACT

The state of programs designed to support voice training are lacking with respect to timbral nuance. For trans people undergoing voice training, available applications tend to favour pitch as the primary – or in some cases sole – measure of progress. To accommodate the many parameters that factor into voice perception, we propose a tool to allow users to mimic a resynthesized version of their voice using applied timbral descriptors.

1. INTRODUCTION

This project is a component of a larger application that supports people undergoing voice training in developing their preferred vocal timbre by mimicking a synthesized version of their voice with any desired timbral modifications. Current applications [1], [2], [3], [4] have relied largely on pitch to distinguish vocal characteristics. Vocal perception is more complex than this, and as such timbral development of those undergoing voice training is paramount to the users' success [5]. Semantic timbral descriptors (e.g., breathier, huskier, higher) will be given by the user to apply to a recording of their voice, and the resulting output can be tweaked further using additional descriptors.

By using the user's voice as the primary input source, our goal is to promote a healthy and informed way to explore the timbral possibilities of one's voice. The long term goal is to package this into an acessible vocal cooach, acting as a tool for speech pathologists and the public (e.g., trans people seeking a more feminine/masculine/neutral voice, people with speech impairments [6], voice actors).

This document first outlines project goals, then tools and resources used during development. A predicted timeline of work to be completed follows, concluding with a log of work done to date and their results.

Attribution: Doyle, I., Klassen, A., & Larmer, E., "Project Specification for CSC 475 2025: TransVox", in *Proc. of the 26th Int. Society for Music Information Retrieval Conf.,* Daejeon, South Korea, 2025.

2. GOALS

1. Basic goals include being able to have code that can adjust and change the user's voice per their specifications. We want to include different options to re-synthesize the given audio and have it return a natural-sounding voice. This would include pitch, breathiness and other textures as mentioned previously.

2. Realistic goals include having a method that lets the user upload their audio to our program so that they can adjust their audio. This can be done using a simple HTML page where a user can upload and choose which adjustments they would like to do. Adjustments would be made in set intervals where the user would be able to choose their own levels of adjustment.

3. A stretch goal would to be able to create a detailed and customized interface for the user to be able to interact with. This would include more specific JavaScript interactive objects such as dials, fine tune levels and options. Design would be initially done in Figma and then would be translated to HTML/CSS/JS format to create a memorable, interactive interface.

3. METHODOLOGY

Python is the driving coding language for this project. FreeVC [7] has proven to be a wealth of information regarding voice cloning, featuring the implementation of libraries like pytorch [8] and Librosa [9] to support speech analysis and synthesis using a fine-tuned checkpoint of Microsoft's WavLM model [10]. FreeVC first extracts a generalized timbre (i.e., mel-frequency cepstral coefficients) spectrogram from a sample of a 'target' voice, then uses WavLM to extract the content (i.e., words spoken) from a sample of a 'source' voice. A custom transformer is then used to infer a .wav file containing the content from the source speaker using the timbre of the target speaker.

We have used FreeVC as a guideline for our own implementation of speech resynthesis, and we plan on using similar libraries to train a separate model to allow timbral adjustments to be made to the output (by intercepting and modifying the MFCC spectrogram before resynthesis).

4. TIMELINE

There will be a significant learning curve for all members while implementing this project, so the precise details are subject to change throughout the term. The following timeline represents the work undergone as of March 18, 2025, and expected work to be completed throughout the rest of the term:

1. Before commencing the project, it was important that all members are on the same page and agree to and understand the project details and distribution of work. Amendments would be made to this timeline as research progresses and implementation details are decided on.

2. There are two major components to the project - the first being effectively cloning a vocal sample by any means. We eventually decided on using FreeVC as a primary reference. We anticipated having a working model by early March, but the following dates are when related work was completed:

   - Mar. 10: discovered FreeVC (Isaiah)
   - Mar. 13: got FreeVC working, effectively cloning a speech sample via resynthesis (Isaiah)
   - Mar. 18: developed a proof of concept for adjustment of timbral quality by manual intervention (Isaiah)

3. The second component involves the development of a trained model to modify vocal timbre. This includes deciding on the number of descriptors we want to use (if a finite number), and labelling a dataset of speech samples with those descriptors. We found that using effects that mimic timbral changes (e.g., breathiness by linear predictive coding [11]) would be an effective way to generate this dataset. This was anticipated to be completed by mid-March.

   - Mar. 8: determined which descriptors to use (Aileen, Elijah, Isaiah)
   - Mar. 15: implemented a pitch shifting effect (Elijah)
   - Mar. 17: implemented an effect to adjust perceived ’roughness’ of a voice (Aileen)

4. What remains is to complete a labelled dataset to train an MFCC classification model with, and use it to infer the adjustments neeed to be made to an aribtrary MFCC spectrogram (representing a voice) in order to mold into a desired timbre.

   - Mar. 28: complete a labelled dataset mapping MFCC spectrograms to timbres (Aileen, Elijah, Isaiah)
   - Apr. 1: train a model on said dataset, and use it to infer adjustments to any particular MFCC spectrogram (Isaiah)
   - Apr. 4: create a minimal UI (Aileen)
   - Apr. 4: final testing (Aileen, Elijah, Isaiah)
   - Apr. 11: final adjustments (Elijah)

5. EXPERIMENTS

We began by exploring a number of potential methods in trying to get past the initial phase of resynthesizing a recording of human speech. Going in we knew it would be critical, due the intent behind the project, for the output to be realistic (i.e., imitable by a human within reason), and to have manual control over the timbre of the output. The first method we explored was the application of linear predictive coding, as we were all familiar with historical example of using LPC for vocal modifications like pitch shifting [12]. As we explored using LPC, it became clear that the output audio sounded rough and robotic, which was a deal-breaker since we required that the output sound as natural as possible.

Mel frequency cepstral coefficients were considered early on due to their generalized, but effective, characterization of timbre, and its proven applications in speech processing [13]. Although MFCCs are indeed able to extract vocal features and represent an audio file’s timbral characteristics - even going as far as to be able to identify the speaker’s emotion [14] - it seemed to us that MFCCs lacked the granularity required to faithfully re-synthesize an audio sample. Unlike frequency spectra, choosing a discrete number of MFCCs to analze results in loss of information, making it impossible to invert the operation to obtain the unchanged input. That is, using a sample’s MFCCs to re-synthesize the audio results in a thin sound, coarse and robotic due to the temporal information lost in the process.

As neither liner predictive coding or mel frequency cepstral co-efficients seemed to be able to retrieve the information we needed from the audio as well as provide enough information to re-create the audio, we detoured toward a simpler approach. In the interest of getting a tangible proof of concept, we decided to program distinct functions for adjusting the pitch, tone, breathiness, and roughness associated with input audio. With these functions, we proposed that users would adjust a number of sliders mapping to the prominence of the aforementioned effects, thus removing the need for any kind of speech synthesis.

We were still keen on researching speech re-synthesis though, so we allocated time and effort into diving deeper. We eventually stumbled upon source code claiming to effectively re-synthesize input speech using the content from one speaker using the timbre of another. FreeVC [7] works by first encoding speech information (i.e., the words spoken) by providing the raw ’source’ waveform to a pre-trained WavLM model and bottleneck extractor. The timbre of another ’target’ waveform is then extracted by computing a mel-spectrogram, such that the content of the source waveform and mel-spectogram of the target waveform are used as input to a HiFi-GAN v1 vocoder [15] for synthesis.

The approach Li et. al [16] employ allow for the modification of MFCC timbral information during inference, thus opening the possiblity of timbral adjustment. This motivated us to get back on track with our original plans, pursuing the use of MFCC modification as a means for timbral adjustment. We spent some time tinkering with the FreeVC source code to familiarize ourselves with its inner workings and explore the timbral possibilities of this approach.

Each frame of the extracted mel-spectrogram from the target waveform, by FreeVC's defaults, contains 80 MFCCs taken at intervals of 320 samples with a window size of 1280 samples. These are organized in a pytorch `tensor` object, which takes a form similar to a 3-dimensional numpy array, supporting the same operations for data access and modification (e.g., `tensor([[[A1, A2, ..., AN], [B1, B2, ..., BN], ...]])` represents a mel-spectrogram of some number of N-coefficient mel-spectra). Thus, we were able to manually modify the mel-spectrogram before resynthesis to observe its effects. The following is an example of halving the magnitude of the 40 upper MFCCs (the upper half of the mel-spectrum) for all spectra in the extracted mel-spectrogram:

```
mel_spectrogram[:, 40:, :] *= 0.5
```

The quality of the results were varied, which is to be expected given the brute-force nature of the experimental procedure. There was undeniably, however, considerable timbral change in the synthesized output, so we are optimistic that a more informed approach to MFCC modification will allow the program to adjust the synthesized timbre in an intuitive way.

6. FUTURE WORK

In order to implement intuitive exploration of timbral parameters, the simplification of timbral adjustment is critical. We don't want users to have to consider the relative weights of thousands of MFCCs to adjust timbre, so the next critical step is to construct a dataset that can be used to train a model on MFCCs and their perceived timbral properties. In order for this to be effective, the dataset will need to be of significant size. Thus, it may also prove beneficial to create an interface we can use to speed up the process of labelling data. More importantly, how exactly timbre will be described (i.e., what descriptors? how many?) will need to be set in stone to ensure that labelling is consistent.

We will also need to consider how we want users to give input to how they want to affect their vocal timbre. An advanced final product may implement a natural language processing model to support textual specification of any number of timbral descriptors. A cheaper alternative may be to resort to a series of sliders corresponding to particular timbre descriptors (e.g., 'breathy', 'nasally', etc.) which reflect exactly the descriptors used during labelling.

Once we have solidified a mapping of user input to MFCC-represented timbre, a trained model will be used to infer adjustments to make to the source mel-spectrogram in order to produce a waveform that is close enough to the user's voice to be realistically imitable, but still desirably affect the timbre in a meaningful way.

7. REFERENCES

[1] DevExtras. (2018). Voice Tools. Accessed: Feb. 18, 2025. [Online]. Available: https://devextras.com/voicetools/

[2] D. Seek & C. Nitz (2020). Voice Pitch Analyzer. Accessed: Feb. 18, 2025. [Online]. Available: https://voicepitchanalyzer.app

[3] C. Antoni and C. Speechtools Ltd. (2013). ChristellaVoiceUp. Accessed: Feb. 18, 2025 [Online]. Available: https://www.christellaantoni.co.uk/transgender-voice/voiceupapp/

[4] I.L. Alter, K.A. Chadwick, K. Andreadis, R. Coleman, M. Pitti, J.M. Ezell, A. Rameau. "Developing a mobile application for gender-affirming voice training: A community-engaged approach" *Laryngoscope Investig Otolaryngol*. 2024. doi: 10.1002/lio2.70043. Accessed: Feb. 13, 2025. [Online]. Available: https://pmc.ncbi.nlm.nih.gov/articles/PMC11645500

[5] J.L. Hawley and A.B. Hancock. "Incorporating Mobile App Technology in Voice Modification Protocol for Transgender Women." Journal of Voice. 2024. DOI: 10.1016/j.jvoice.2021.09.001. Accessed: Feb. 11, 2025. [Online]. Available: https://www.sciencedirect.com/science/article/abs/pii/S089219972100299X

[6] J.M. Barkmeier-Kraemer, J.N. Craig, A.B. Harmon, R.R. Hillman, J. Jacobson, R.R. Patel, B.H. Ruddy, J.C. Stemple, Y.A. Sumida, K. Tanner, S.M Theis, M.R. van Mersbergen, & L.P. Verdun. "Voice Disorders." asha.org. Accessed: Feb. 12, 2025. [Online]. Available: https://www.asha.org/practice-portal/clinical-topics/voice-disorders

[7] J. Li, W. Tu, and L. Xiao. "FreeVC: Towards High-Quality Text-Free One-Shot Voice Conversion," *arXiv preprint arXiv:2210.15418*, 2022.

[8] PyTorch, "pytorch," github.com. Acessed Mar. 18, 2025. [Online]. Available: https://github.com/pytorch/pytorch

[9] Librosa, "librosa," github.com. Acessed Mar. 18, 2025. [Online]. Available: https://github.com/librosa/librosa

[10] S Chen, C Wang, Z Chen, et al., “Wavlm: Large-scale self-supervised pre-training for full stack speech processing,” *IEEE Journal of Selected Topics in Signal Processing*, 2022.

[11] K. I. Nordstrom, G. Tzanetakis, and P. F. Driessen, "Transforming Perceived Vocal Effort and Breathiness Using Adaptive Pre-Emphasis Linear Prediction," *IEEE TRANSACTIONS ON AUDIO, SPEECH, AND LANGUAGE PROCESSING*, vol. 16, no. 6, 2008.

[12] Y. Sasahira and S. Hashimoto. “Voice Pitch Changing by Linear Predictive Coding Method to Keep the Singer’s Personal Timbre.” *International Computer Music Conference*, (September 3-7) 1995. Available: https://quod.lib.umich.edu/cgi/p/pod/dod-idx/voice-pitch-changing.pdf?c=icmc;idno=bbp2372.1995.118;format=pdf

[13] C. Ittichaichareon, S. Suksri, and T. Yingthawornsuk. "Speech Recognition Using MFCC." *International Conference on Computer Graphics, Simulation and Modeling*, (July 28-29) 2012. Available: https://www.researchgate.net/publication/281446199_Speech_Recognition_using_MFCC

[14] S. Lalithaa, D. Geyasrutia, R. Narayanana, & M. Shravani. "Emotion Detection using MFCC and Cepstrum Features." *International Conference on Eco-friendly Computing and Communication Systems*. (2015) Available: https://www.sciencedirect.com/science/article/pii/S1877050915031841?ref=cra_js_challenge&fr=RR-1

[15] J Kong, J Kim, et al., “Hifi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis," *arXiv preprint arXiv:2010.05646*, 2020.