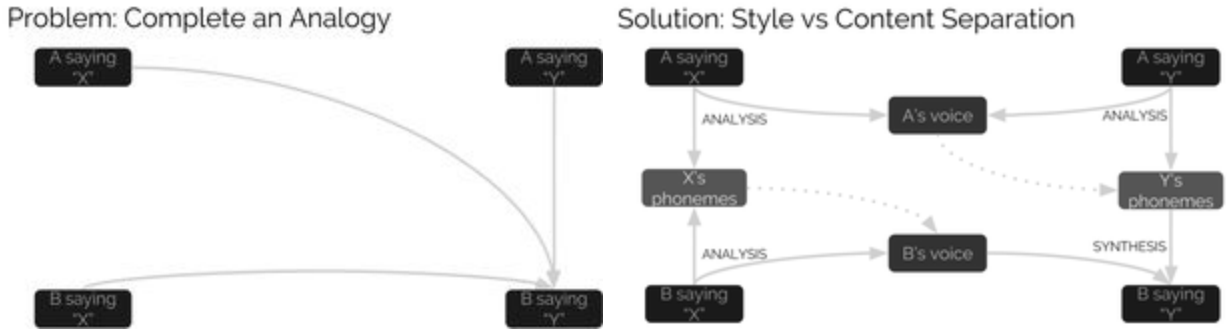
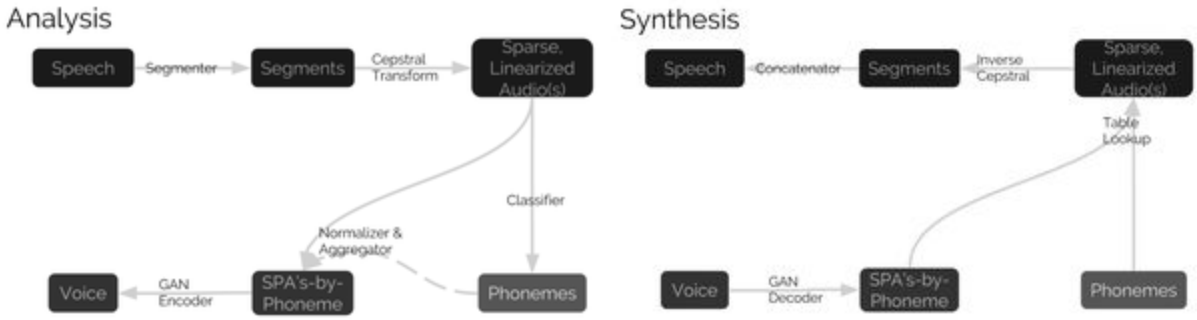
Voice Translator (Project Report #1)
group: ian calloway, heming han, seth raker, samuel tenka
date: 2016-11-04
descr: Project report for Eecs 351 Project, in response to Nate's comments.

0. Introduction
The Voice Translator teams aims to imitate human speech. More precisely, we seek to solve the *voice morphing* problem, which we view as a problem in style/content separation and analogy completion:



In our project proposal, we outlined an architecture to analyze and synthesize style/content in the domain of natural voices:



Nate gave us feedback on our project proposal, recommending in particular that we:

> . . . . **Settle on your database** . . . **collect particular phrases of your own** . . . **demonstrate [imitable stylistic] differences between [sample recordings]** . . . . **Read about voice morphing** . . . . **Suggest what languages/packages** . . . .

In this report, we follow up on those points. We begin with an overview of data collection, list what pre-existing software we'll use to analyze and synthesize voice into psychologically meaningful characteristics, move on to summarize our analysis of how raw data reflects those characteristics, and conclude with reflections on prior work in voice morphing.

1. Data

1.0. Large official datasets

We will use the freely available CMU Arctic dataset. ARCTIC consists of paired speech clips of ~1000 sentences, each read by the same 7 humans. Of those 7 humans, 3 speak with standard american accents, and 4 speak with ``other accents'', providing variety suitable to training a speech-transcriber. Though we are also interested in certain larger research databases containing snippets of natural conversation, our requests for access have not yet been granted, and therefore we will plan to develop around our smaller databases.

1.1. Homemade Recordings

In addition to downloading standard linguistics datasets, we are also building our own problem-tailored dataset for analysis and preliminary training. Here, we control voice and text (style and content), for now fixing the latter to isolate variations in the former. We were able to record and pre-process ~25 records in an hour, a rate which scales to our goal dataset size of low hundreds. Below, we copy verbatim our documentation for a size-23 set of voice samples we've taken of ourselves:

1.1.0. What filenames mean

Each file is a stereo .wav containing at least one recording of a fixed speaker using a fixed voice style on a fixed text. Durations range from ~4 to ~12 seconds.

Consider the filename "Sam_nasal_quickfox_3.wav". This indicates the speaker (Sam) using a voice (Sam's fake nasal voice) uttered the sentence ("The quick brown fox jumped over the lazy dog.") 3 times.

1.1.1. Dataset size

We made 23 recordings of "The quick brown fox..." in Shapiro 1152 on Thursday, 2016-11-03:

	Standard	Noisy	Nasal	Deep	Total
Heming	2	3	1	1	7
Ian	0	0	0	0	0
Sam	3	0	3	3	9
Seth	2	3	1	1	7
Total	7	6	5	5	23

1.1.2. Recording process

Participants read "The quick brown fox jumped over the lazy dog", an accidentally altered version of the classic sentence that contains all 26 letters (and hence an unusually broad range of phonemes). Speakers are instructed to read *clearly* and at *moderate speed and volume*.

The recordings are performed on Sam's Chronos 7 laptop using Audacity, in an enclosed room (Shapiro 1152) with quiet but audible air conditioning. Speech occurs 1 to 2 feet away from the screen, facing the screen, unless otherwise specified. Our 4 voice styles are defined as follows:

0. [**standard**] --- ordinary clear speaking voices. This is the voice style to which speakers default in daily life, modified by the requirement for "clarity". Thus, good articulation replaces mumbling and slurring.
1. [**noisy**] --- in these and only these recordings, speech occures at distance roughly 4-6 feet from screen. This emphasizes background noise. Other than background noise, this style should match [standard].
2. [**nasal**] --- fake nasal voice. Nasality is achieved (unconsciously) by restricting mouth airflow and hence increasing airflow through the nose, for instance by clenching the jaw or bunching the back of the tongue.
3. [**deep**] --- fake deep voice. Depth, in contrast to nasality, is achieved by enlarging resonant cavities, for instance by relaxing the throat and depressing the tongue. This emphasizes lower frequencies.

Post-processing involved:

0. Manual removal of extraneous sounds such as inter-record clickings and voice-work instructions.
1. Amplification (maximum possible without clipping. This is a standard Audacity tool with no parameters).
2. Noise reduction (with the standard Audacity settings, namely (12, 6.00, 3) for (decibels, sensitivity frequency, smoothing)).
3. Manual splicing of "silence" to align each record within a single file to be spaced roughly every 4 seconds. Since each file has at most 3 records, each file is less than 12 seconds long. For comparison, each record is audible for around 3 seconds.

1.1.3. Future Recording Plans

We hope to expand the dataset in several directions. Ranked in descending importance:

0. new speakers

1. new sentences
2. more records of current speakers/voice styles/sentence(s)
3. new voice styles

Also important will be a distinction between voice timbre and phrasing; these aspects of voice style are currently treated as one, but the former is local, the latter global, and hence our system will likely treat the two as different.

2. Pre-existing Software Tools

We narrowed down our previous list of technologies. Listed from input-side to output-side of our data pipeline:

0. **Audacity**: Record voice samples, preprocess, and create spectrograms.

1. **audioop**: Raw audio manipulation in Python. Multitude of useful functions, including audioop.avg(fragment, width) to average samples for a given width and audioop.findfit(fragment, reference) to match a reference to a portion of a fragment.

2. **Numpy**: The fundamental Python package for scientific computing. Our linear algebra workhorse for comparing high-level voice features
.
3. **Keras**: A high-level neural-network library in Python built on Tensor Flow. Tensor Flow was originally developed by Google's machine intelligence research team to streamline development of networks for such problems as image classification and machine translation. Neural nets may provide the requisite degrees of freedom to model realistic voices.

4. **Matlab**: The language of our translator and design-phase analysis. Industrial-strength matrix algorithms (applicable, for instance, to quantified voice qualities) will form a base for our final system. The Audio System Toolbox provides a variety of functions, including filtering and equalization.

3. Data Analysis

We here examine and discuss spectrograms of our own speech. Each team member read aloud “The quick brown fox jumped over the lazy dog” in voices ranging from “nasal” to “standard” to “deep”, with multiple trials per voice (see Section 1.1.2 for data-acquisition details).

3.0. Summary

Our main observations we summarize as follows:

- As seen in class, visual inspection of spectrograms more easily reveals text (content) than voice (style). Indeed, different speakers’ spectrograms are indistinguishable from afar.
- White noise has a uniform-magnitude spectrogram. Upon noise reduction via standard tools, gaps between formants become defined.
- Harsh consonants such as plosives and fricatives, being similar to white noise, have distinctly taller spectra than vowels and liquids. Thus, the “x”, “j” and “z” stand out as peaks in our recordings of the “quick fox” sentence.
- Formant separation corresponds to a “brightness” or “nasality” of voice, while the lack thereof corresponds to a “richness” or “depth” of voice. Since perfectly periodic signals should have perfectly banded spectrograms (“harmonics”), we interpret the formant separation as a measure of periodicity. Therefore: periodic signals sound “bright”.

The above observations are supported by data in the following discussion.

3.1. Details

We begin with the spectrograms of our standard voices:

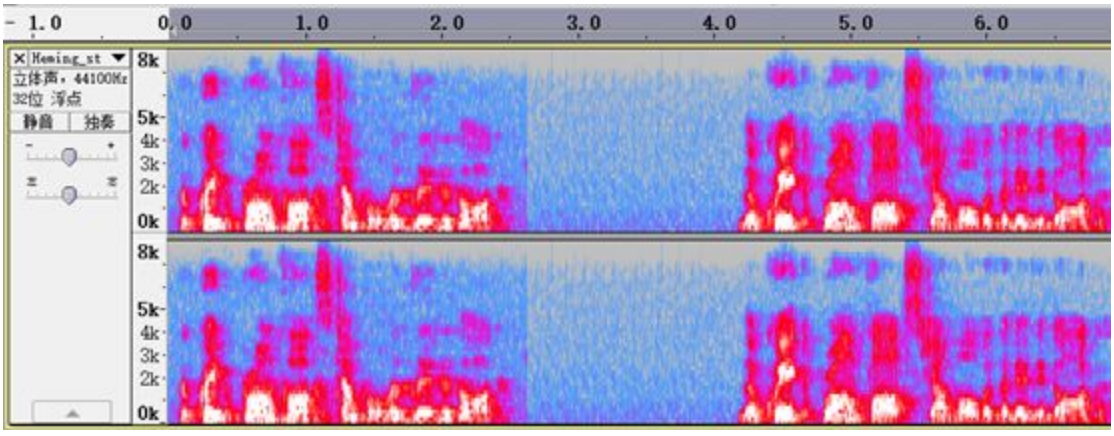


Figure 1: Heming – standard

The spectrogram of Heming Han – standard contains two times of speaking “the quick brown fox jumped over the lazy dog.”

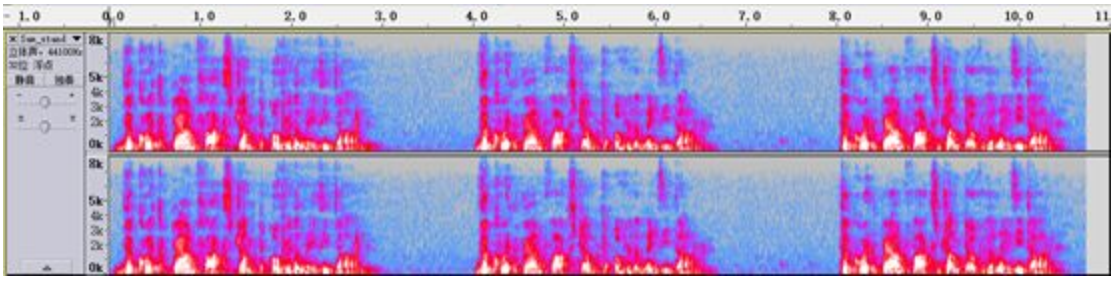


Figure 2: Sam – standard

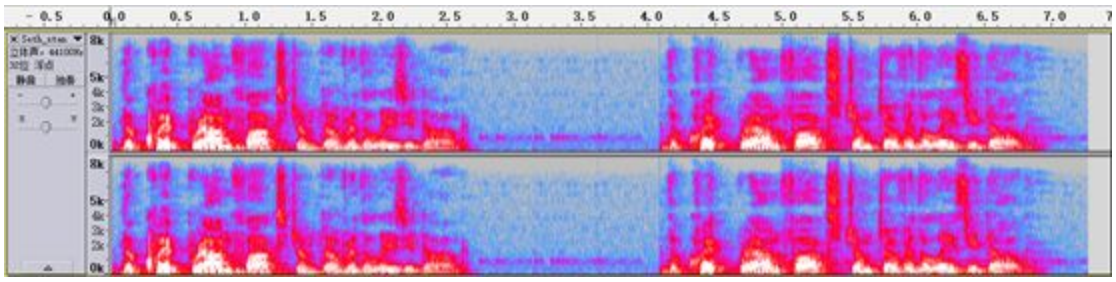


Figure 3: Seth – standard

Above three spectrograms are all standard speech, so we can use them as references when compared with nasal/ deep voice of a single group member.

Our initial observation is that white noise is reflected as a uniform-magnitude addition to the spectrogram. We are able to eliminate the environmental noise from the original sound track, to improve the audio quality for higher precision of possible recognition.

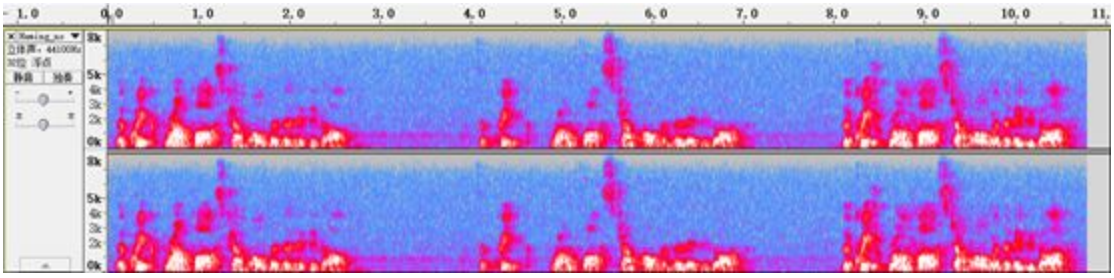


Figure 4: Heming – noisy

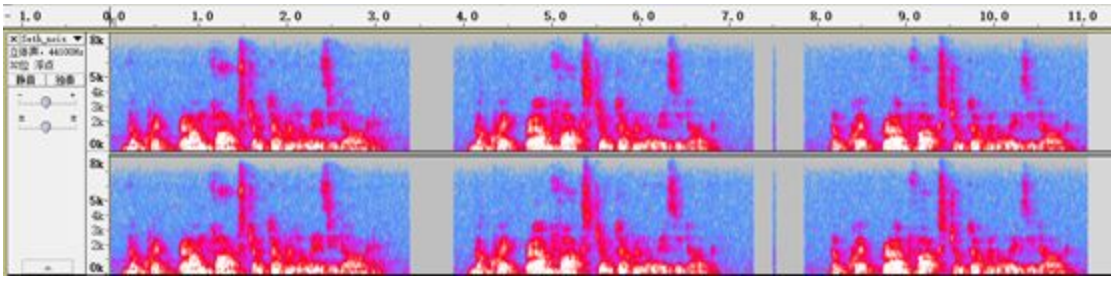


Figure 5: Seth - noisy

What is the influence of noise? We can compare our spectrogram in Figure 1 and Figure 4, as well as Figure 2 and Figure 5. What we can discover is that in high frequency region (frequency that is higher than 3kHz), the resolution of noisy sound file is much poorer than standard non-noisy sound. The reason is obvious: the intensity of high frequency harmonics of human voice is far lower than the intensity in 0~3000Hz. By subtracting the noise of the environment (recorded sound when nobody is speaking), we can get a sound file with higher high frequency performance. Besides, according to what we do from previous assignments (comparing the piano notes with single sinusoidal sound), the higher frequency harmonics preserve the sound quality of our voice. The elimination of the noise, makes it available for us to carry out a transformation between different style.

The second part is the difference between different people's voice. Pay attention to Figure 1and Figure 2 with Figure 3. There is a very interesting point and we make some assumption: the gap (or the richness) in high frequency domain are different: the record of Seth, whose sound is more deep and rich in daily life, has less gap in frequency; while for Heming and Sam (especially Sam because Heming is not a native speaker and his poor performance in high frequency domain may due to the pronunciation skill), the more light/ bright sound results in a more fragmented frequency pieces. By comparing the continuity of the spectrogram in higher frequency, we (or computer) may catch the difference of voice style, make it possible to find out the style difference and do something about it.

To make sure there do exist some difference in continuity in high frequency if the sound sounds different in richness, we also record the same sentence trying to use deep voice and nasal voice. One thing worth attention is that except Sam, the record for nasal voice only works for first several words because we are not able to keep using nasal sound. Following are our records:

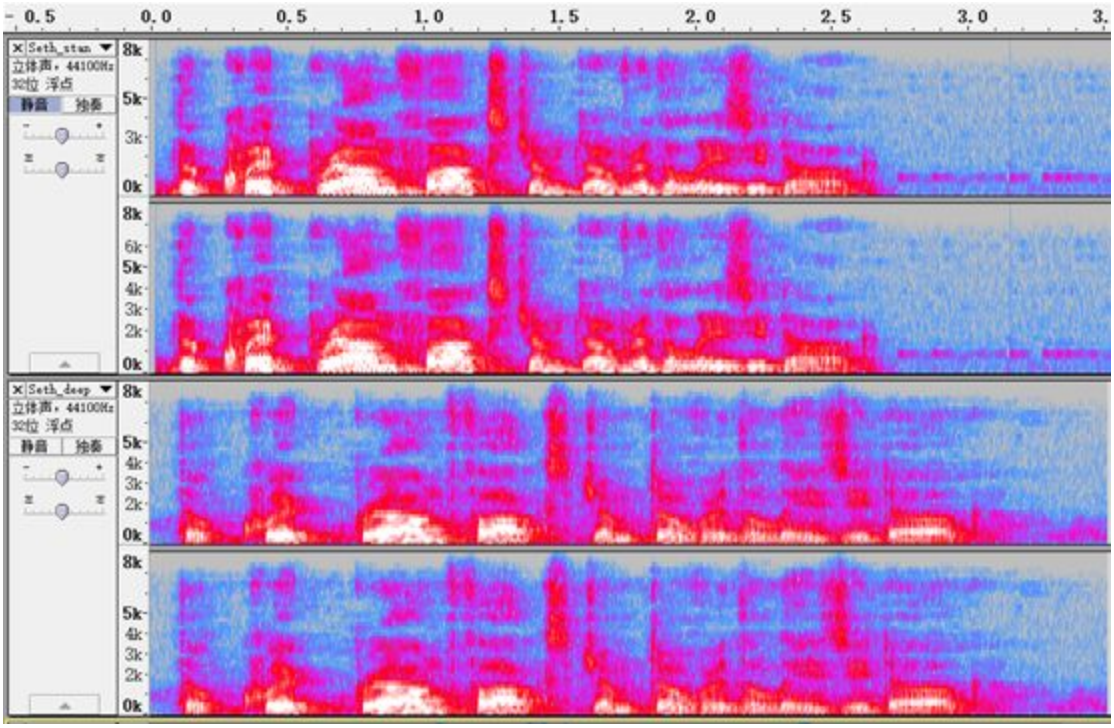


Figure 6: Seth – standard vs. Seth - deep

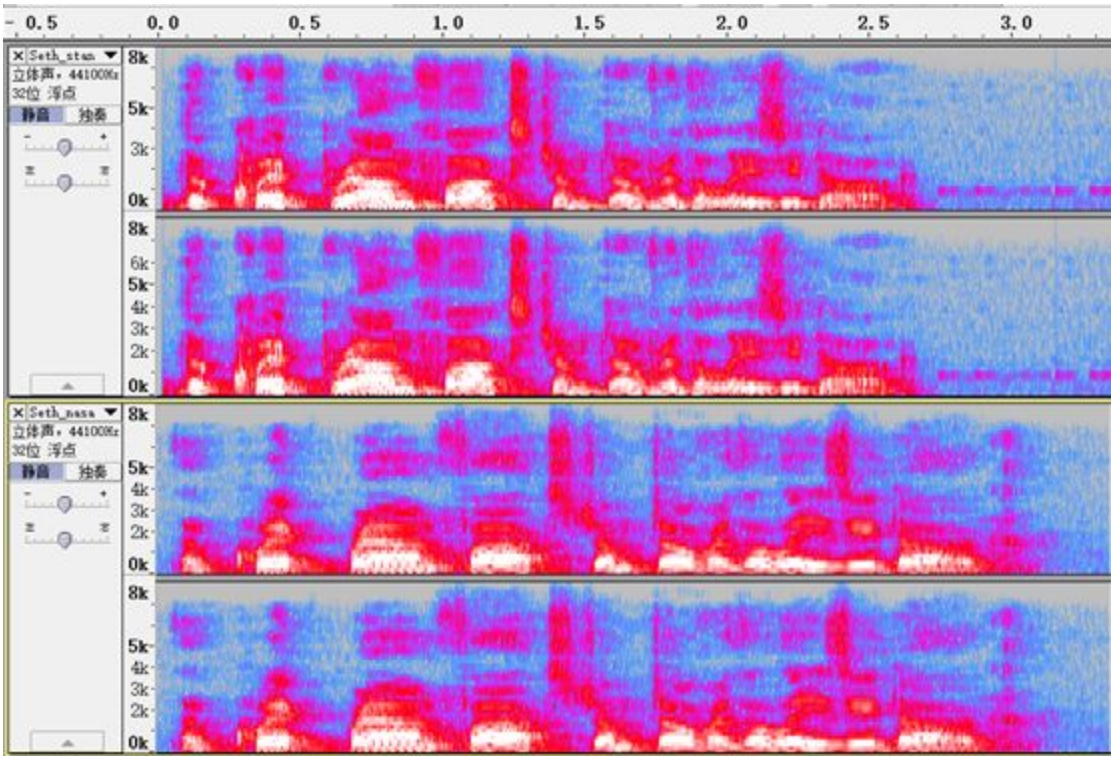


Figure 7: Seth - standard vs. Seth - nasal

What we can see from above two figures are:

1. Compared with standard speech, nasal speech has more gap (discrete frequency pattern).
2. Compared with standard speech, deep voice doesn't not differ much from standard speech, though deeper voice does have a more extensive distribution of frequency. Actually, when we try deep voice, we only tend to lower our frequency as well as making our voice full. As a result, we may conclude that the frequency does not affect much about the overall pattern of the spectrogram; only the quality, the richness matters. By comparing the degree of their frequency distribution of spectrograms, a computer can get the style.

Following are Sam's comparison. It proves our thought.

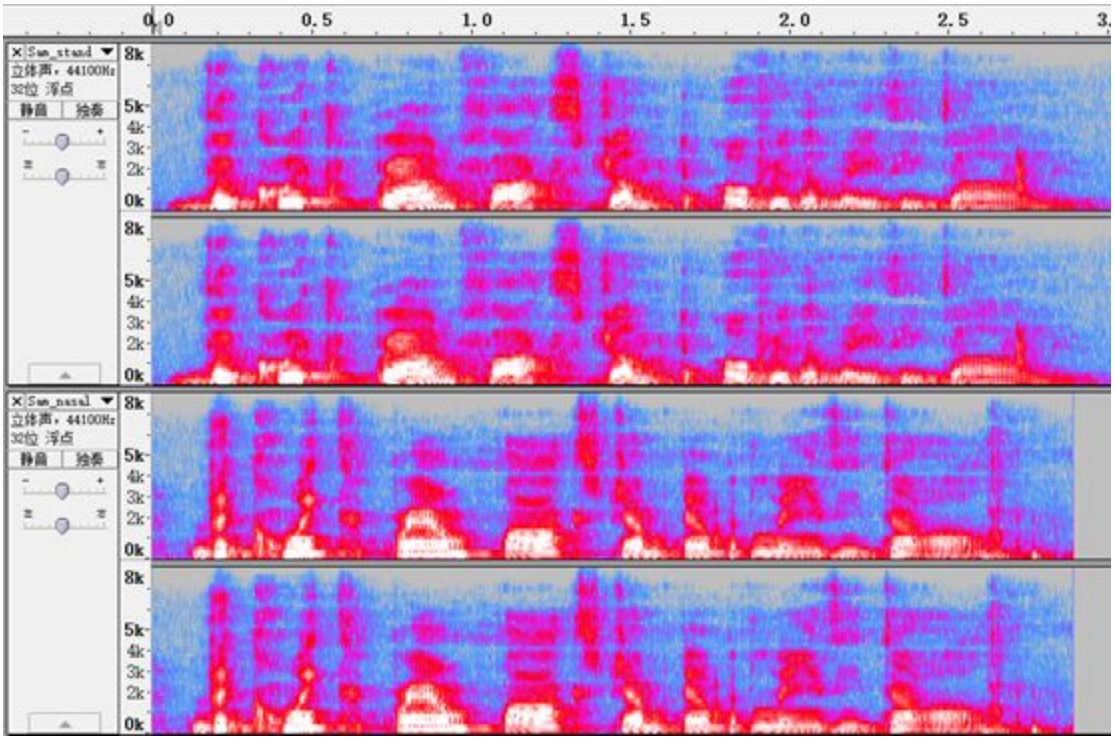


Figure 8: Sam - standard vs. Sam – nasal

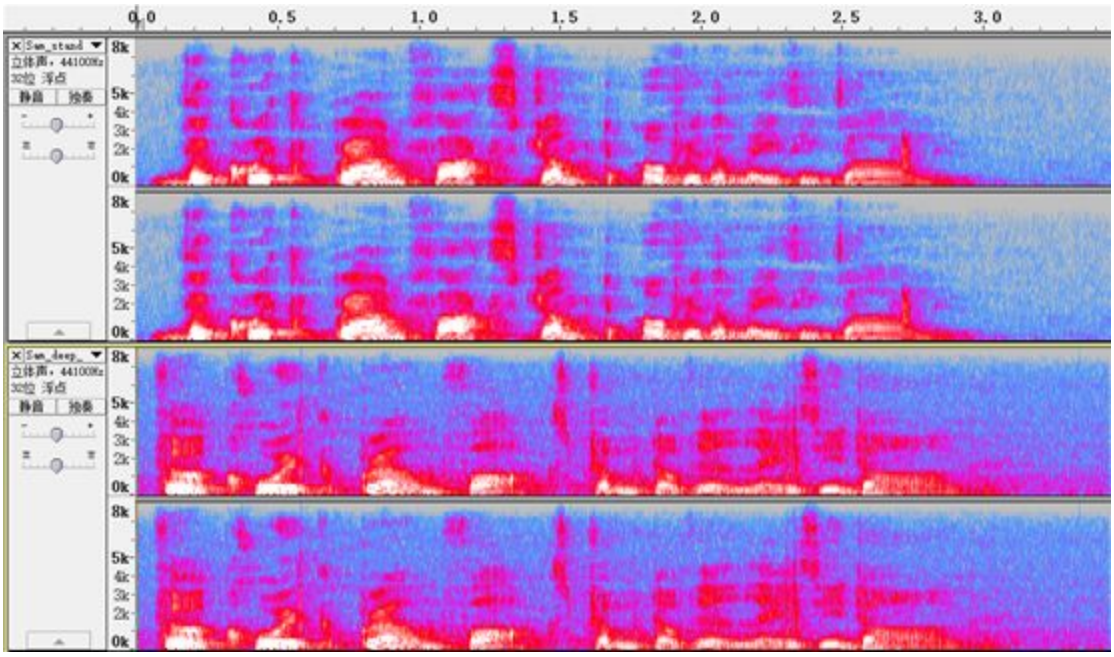


Figure 9: Sam - standard vs. Sam-deep

Third part is the frequency analysis for words. We can see that though we three have totally different voice quality, though we are talking about the same sentence, we have a very similar pattern of the spectrogram.

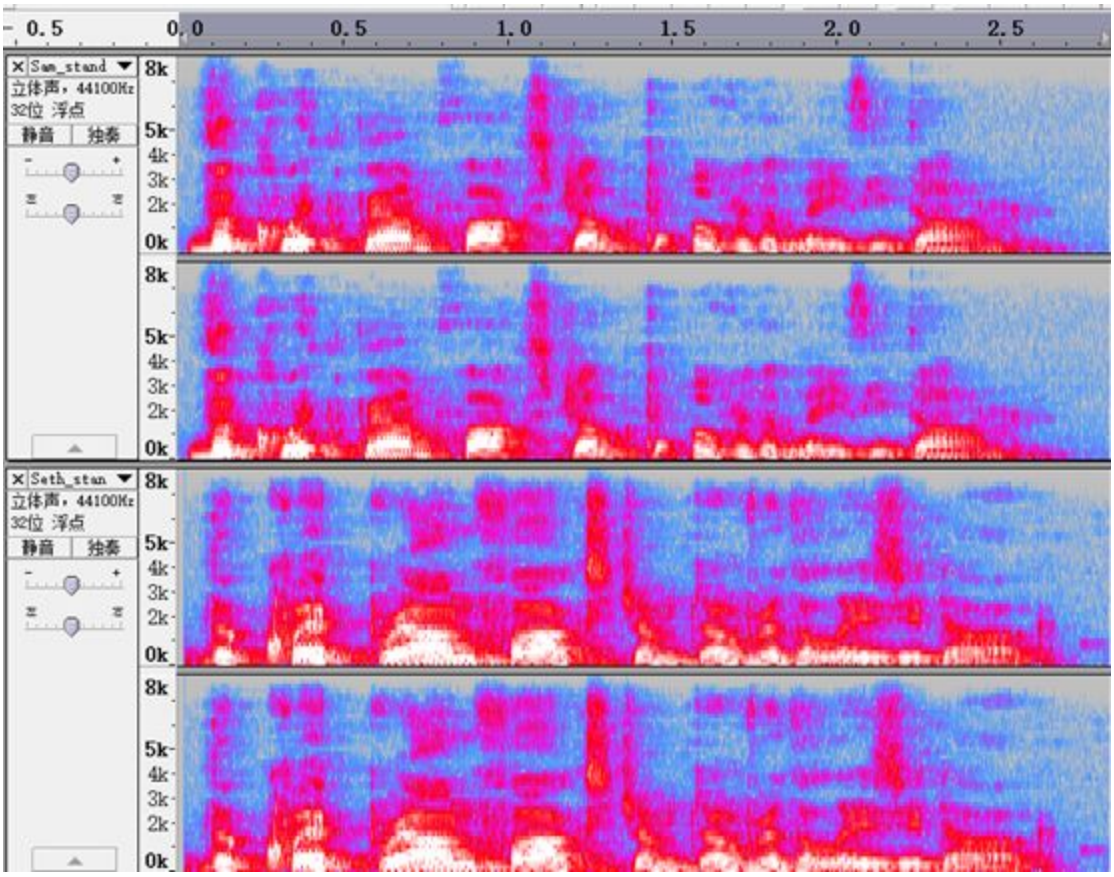


Figure 10: Sam vs. Seth

What we can conclude from our figure and our record is that: the most intensive low frequency domain (which is shown as white in our figure), is determined by the vowel; “th” in “the”, “n” in “brown”, “j” in “jump”, “x” in “fox”, “z” in “lazy”, in our spectrogram, has a very extensive frequency distribution and easy to tell. We believe the computer would have the ability to tell them and achieve recognition.

In conclusion:

For words, we can look at the acoustics characteristic, then compare with what we can get in the database to identify the letters/letter-groups (, then may go to the meaningful word).

For style, one interesting discover is the gaps/continuity/distribution in harmonics: brightness versus richness.

4. Voice Morphing: Reflections on Prior Work

Popa et al briefly survey work (up to 2012) on the voice morphing problem, suggesting that all prior approaches suffer from (A) artifacts due to unnatural windowing in the time domain and (B) "over-smoothing" of synthesized features.

Such over-smoothing we are familiar with in the context of least-squares models: if such a model is uncertain, it will output an average of its perceived possibilities. While such models work well when reality has a convex loss function, they perform poorly for more complex situations or more raw feature representations. For instance, is what is "the" average of a baby and a senior citizen? Linear models on the most accessible features would answer: "a white-haired human that crawls with the aid of a cane, and has completed half of high school", while a model based on deeper features might answer the more natural "a young adult". We are struck at the precision of the human ear: when realism and variety are our goals, it seems an even more challenging task to synthesize audio than images. Plain linear models are inadequate, and Popa et al's survey confirms this.

An initial step toward nonlinearity is to have linear weights adapt to the input; another is to condition successive algorithm steps on a classifier's output. Those two ideas form the main new development of the Popa paper; they call their method "Local Linear Transformation". However, their method improves only modestly on prior work on a subjective, relative scale, and they do not report any absolute evaluation criteria. We are confident vast additional improvements can be made.

Especially exciting is the prospect of applying the recent idea of Generative Adversarial Networks (Goodfellow 2014) to speech synthesis. GAN's learn a complicated loss-function in parallel with the generative model's parameters, and have been shown to perform well precisely on the open-ended, complex, non-convex tasks that Popa argues include the problem of voice morphing.

We've drawn a few non-algorithmic tools from the literature, too: Popa et al test on the CMU Arctic dataset, confirming that our data acquisition efforts are on the right track. And the Ye slides suggest some good quantitative metrics for voice similarity, for instance "Spectral Distortion".

In short, the prior work provides insight and inspiration not only into techniques applicable to our problem, but also into the challenges that remain to be solved.

5. References

We learned about formants:

- Formant (Wikipedia Article, 2016-11-03)

We read about the Voice Morphing work of the past 1.5 decades from:

- High Quality Voice Morphing Seminar (Ye 2004 --- Slides)
- High Quality Voice Morphing (Ye and Young 2004)
- Local Linear Transform for Voice Conversion (Popa et al. 2012)

We'll train and test on:

- CMU Arctic: Databases for Speech Synthesis (Komenek & Black 2003)

We're excited about this algorithm:

- Generative Adversarial Networks (Goodfellow et al. 2014)