AccentTutor: Improved Second Language Acquisition with Phoneme Biofeedback

Acshi Haggenmiller

Faculty Adviser Steven Zucker

May 5, 2016

Abstract

AccentTutor is a Windows/Android application that visualizes vowel pronunciation in real time. A user is able to practice the sounds of a language they are learning and then see how well their vowel pronunciation matches against a template of that language's vowels. Currently, learning to improve one's accent remains one the biggest challenges in second-language acquisition. Because of its difficulty to teach, many instructors don't spend time on pronunciation at all, relying on the hope that students will be able to pick up on correct pronunciation by themselves through exposure. While students can certainly improve their pronunciation on their own, it can be very hard for the learner to distinguish between vowel sounds that do not occur in their native language. This phenomenon results in foreign languages sounding like a homogeneous mush to the untrained ear.

This program seeks to provide an additional path for students and instructors to use in improving pronunciation. By providing visual biofeedback on pronunciation, the user is able to see the differences in vowel sounds that their ears have not yet learned to distinguish. The user can then experiment with vowel production and use positive feedback to learn correct pronunciation at a faster pace.

The general efficacy of AccentTutor to distinguish between the vowels of English and Mandarin has been tested with a limited sample of Mandarin second language learners.

AccentTutor has been set up with vowel information for Standard Mandarin Chinese, but also includes tools for generating appropriate vowel information for any other language.

1 Introduction

Vowels are generally characterized by frequency bands known as formants. Formants are acoustic resonances in the vocal tract made by the shape of the mouth and auditory tract. As such, although vowels can be spoken or sung at many different pitches, the formants that characterize any one vowel will remain more or less in place. In effect, they can be viewed as the result of band-pass filters.

Figure 1 shows the spectrogram of three English vowels, with the first and second formants indicated. There are generally many more than just two formants and the overall energy and apparent number of formants can vary between vowels significantly. It should also be noted that when formants are close together they can be fairly difficult to distinguish.

Generally, vowels in Standard American English can be distinguished by looking at only the first two formants. In some cases, and possibly in other languages, the third formant is necessary as well.

As resonances or filters acting on a speaker's voice, the appearance and effect of formants are highly dependent on the speaker's tone quality. A spectrogram of a high-frequency voice will look very different from a low-frequency voice. Even at the same frequency, men and women's voices will look different. In particular, if a formant does not lie close to a harmonic in the speaker's voice, then that formant might not appear in the spectrogram at all, but it will still have affected the spectrogram in some way.

In short, determining the true formants present in an isolated sample of speech is difficult

because the formants better describe a process the speech went through than the speech itself.

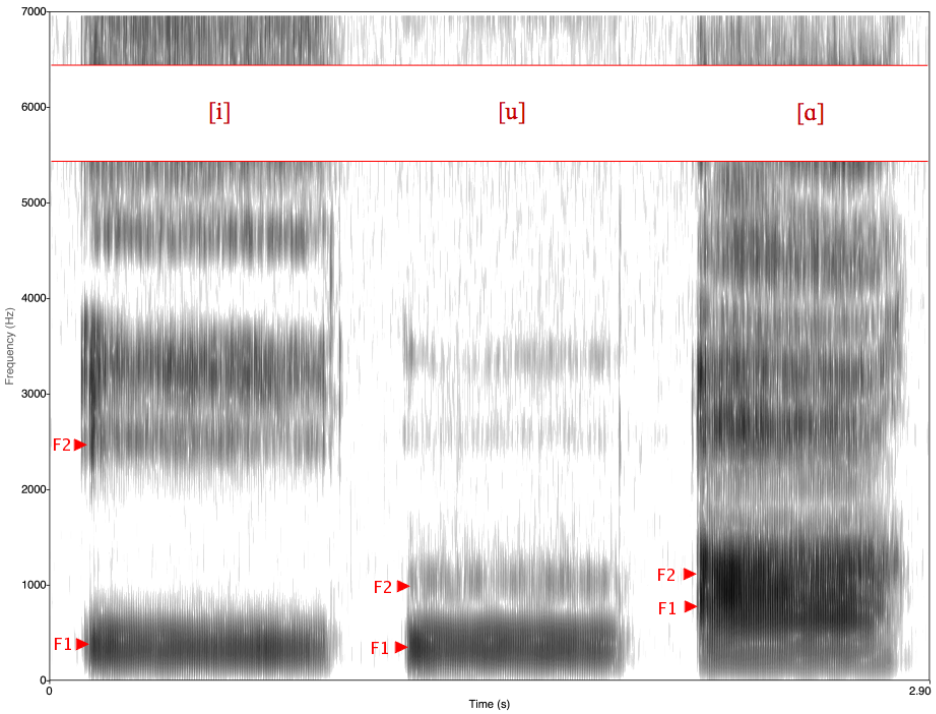

Figure 1: Spectrogram of three English vowels
Created by Wikipedia user ish_ishwar, CC-by-2.0.

The challenge for AccentTutor is to reliably extract the actual formants from speech samples in real-time.

2 Difficulties

Starting by referencing figure 1, I thought that in general, finding the formants in speech should be rather simple. Especially in the first two vowels shown, the first two formants are simply the first two local maxima. Unfortunately, this naive approach performed very poorly, and I realized that figure 1 shows a very idealized spectrogram. It was likely recorded with a high-quality microphone in a very quiet location because there is no noise in the background. The speaker also has a low fundamental frequency of only about 110Hz. These characteristics result in the spectrogram being so easy to interpret.

Recordings taken by microphones available to me, however, tend to have noise and differ-

ent frequency responses. For example, some microphones can barely pick up low frequencies, while others are extremely sensitive to them. This means that it is difficult to rely on the relative amplitudes of different formants, and also makes it hard to know when a local maximum is actually just a result of the microphone's sensitivity or some other characteristic of speech besides a formant.

Furthermore, because AccentTutor needs to run in real time on a smartphone, the length of each sample being analyzed has to be short enough for the program to compute in real time and updated often enough for the program to feel responsive, and it cannot be calculated with future sample points that have not yet been recorded (unlike the spectrogram in figure 1 which could have included sample points both in the past and in the future, because the recording was made in whole before it was analyzed). These characteristics affect the accuracy and precision of the frequency information the program analyzes.

A number of tricks were implemented to get around these various issues. First, all development and comparisons were initially done using the same microphone (the microphone embedded in a Lenovo ThinkPad T450s). This was necessary for consistency and means that the program will work most reliably for that specific microphone. Second, the program does not try to determine the formants for a single sample of sound in isolation but instead looks at a number of prior samples altogether. For each stage of execution, it finds the median value of those prior samples and then uses this value to continue its analysis.

For example, after taking the Fast Fourier Transform (FFT) of a sample of speech, the first step the program takes is determining the fundamental frequency of the speaker's voice. With this, it is able to isolate the harmonics, and then ignore any other frequency content that happens to be in the sample. Unfortunately, determining the fundamental frequency of a sample can be very difficult, and it is easy for algorithms to give the answer in the wrong octave. If the fundamental frequencies calculated in the last 6 samples were 110, 100, 210, 105, 110, and 230, respectively, we choose the median fundamental frequency of 110Hz. This method of robust estimation improves the reliability of the algorithm. The

next step would be to determine the frequency of the first formant. This robustly estimated fundamental frequency is used to reanalyze the harmonics of all the prior samples being used, and the first formant is then selected as the highest peak in the lower frequency range of the spectrogram. It is important for the fundamental to be determined correctly because if it is chosen to be too high, then possible formant peaks will be excluded from analysis. If it is selected to be too low, then every harmonic will look like a local maximum because in between each harmonic will always be a fake "harmonic" with a very low amplitude. A robustly estimated first formant is then used in the determination of the second formant in each of the previous samples, and again a robustly estimated second formant is used in the determination of the third formant.

Another technique used is the explicit preference of certain frequencies in the analysis. For example, the fundamental frequency will always be chosen to be at least 70Hz, because it is very unusual for male speakers to have voices that low. The first formant chosen must be in the range of 250Hz and 1400Hz, the second formant must be between 600Hz and 3000Hz and be at least 100Hz above the first formant, and the third formant must be greater than 1500Hz and again at least 100Hz above the second formant.

Another challenge is what to do when a formant resonance lies between two harmonics of the voice. In this case, which of the two harmonics is the local maxima is likely to oscillate over time, leading to unstable feedback to the user. To correct for this, if a local maximum is adjacent to a harmonic that has an amplitude at least 20% as high as it, the two harmonics are merged and the frequency used for that local maxima is the weighted average of the two harmonics.

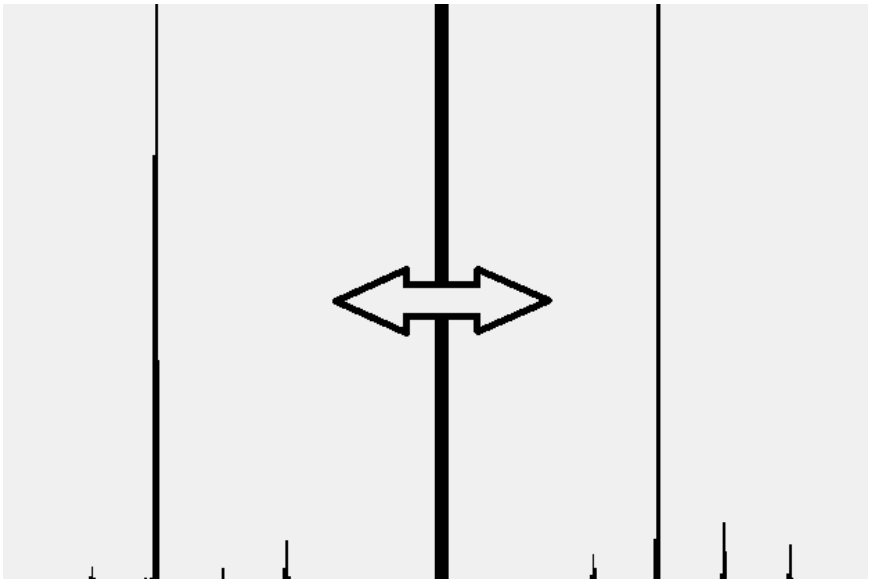

Figure 2: Comparison of Fourier transforms of the /u/ vowel at two different moments in time. The vertical axis is amplitude. The horizontal axis is frequency. The second formant is only discernible in the first plot.

Figure 2 shows a difficulty that may arise when two formants are close together and not cleanly separated by at least two harmonics. While the first formant is plainly visible as the second harmonic of each sample, the second formant rests somewhere between the third and fourth harmonics. Because of this, it only even appears as a local maximum part of the time. In the second sample, inferring the presence of the second formant would require comparing the amplitudes of the third and fourth harmonics to their expected values in the absence of that formant. Making some estimate of these values would involve knowledge of the speaker's vocal tone and the microphones frequency response. Perhaps the formant could be found as a local maximum across time instead of frequency. Additional exploration would be necessary to properly resolve this difficulty. Currently, AccentTutor relies on robust estimation of the formants from the last several samples in order to recover from this difficulty most of the time.

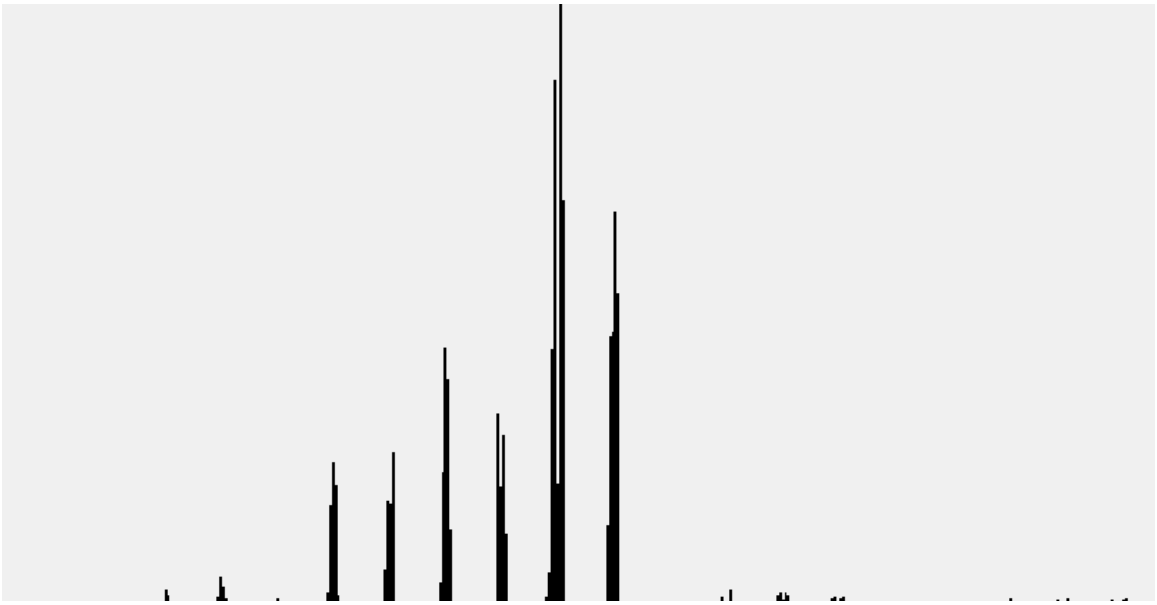

Figure 3: Fourier transform of a sample of the "ahh" vowel. The vertical axis is amplitude. The horizontal axis is frequency.

Figure 3 shows just how complicated the analysis of a frequency spectrum may become. The first two harmonics are completely absent. The fourth harmonic is a local maximum, but is not a formant. The second formant has a higher amplitude than the first formant. The second formant has two separate individual peaks. The third formant is somewhere in the cluster of the three low amplitude harmonics following the second formant.

Although in general AccentTutor handles these difficulties well the majority of the time, there are still a fair number of false formant classifications that need to be ignored by the user. There is plenty of room for further development in this area.

A final difficulty arose with the Android version of the program. For some unknown reason, the microphone would not always reliably supply the program with audio data. If I tried to run the program in the debugger or in an emulator, it would instantly crash. It is likely that these issues are specific to the computer I use for development and could easily be solved by someone else.

3 Software Overview

AccentTutor was written in C♯ with Xamarin.Android. The main AccentTutor application is divided up into three components: the portable core formant analysis component, the graphical user interface, and the audio input component. Owing to this structure, both Windows and Android versions can use the exact same core analysis code.

The AudioIn class for each version of the program is responsible for continually receiving audio data from the system microphone at a sample rate of 44,100Hz and delivering a specified number of samples (currently 8820 samples, or about 200ms worth) to the analysis code through an AudioAvailable event.

This audio data is then taken by the core FftProcessor class, which slides the new data into a moving window currently of length 16384, or about 372ms of audio. The length of this window needs to be a power of two in order to take advantage of the efficient Radix-2 FFT algorithm supplied by the MathNet.Numerics library. While the highly optimized FFTW C library was originally for development, it proved much more difficult to port to the Android platform. As a native C♯ library, MathNet.Numerics did not require porting. A Han windowing function is then applied to the window data and the result it passed through the FFT algorithm. Before this FFT data is analyzed for formants, it passes through a HandleFftData function which is different in the Windows and Android versions. This code squares the FFT data to get what is more strictly ”power” than ”amplitude”, performs some microphone specific equalization, runs the FFT through a high-pass filter to remove noise/drift, and then Z-scores it to compensate for variation in the volume of the speaker's voice. If the standard deviation taken in the Z-scoring process is too low, the sample of sound is considered to simply be silence, and the data is ignored. If not, the modified FFT data shifted into an array containing the last several FFTs. Currently, this array has 8 elements, which corresponds to a total time of 1.77 seconds. These 8 most recent FFTs are together passed to the main formant analysis code in the PerformFormantAnalysis method of the SpectrumAnalyzer class.

First, the fundamental frequency of the new FFT is determined and this value is saved to avoid repeated calculation later. The fundamental frequency is calculated with a modified Harmonic Product Spectrum (HPS). The HPS is very simple in principle. Each element of the spectrum is multiplied by each of its multiples in the spectrum. For example, the element at 110Hz is multiplied by the values at 220Hz, 330Hz, 440Hz, and so forth. The element at 200Hz is multiplied by the values at 400Hz, 600Hz, and so on. Whichever frequency has the largest value after these multiplications is determined to be the fundamental frequency.

Unfortunately, there are some difficulties with this naive implementation. First, if the value at any multiple is ever zero or very close to zero, then the HPS will fail to find the fundamental. This is especially apparent with high frequencies that have no multiples in the spectrum. In order for the comparison to be fair, each element must be multiplied by the same number of harmonics. These two sides of the problem can be fixed by having a relatively small but non-zero minimum applied to values in the spectrum and given to frequencies higher than it. Second, harmonics are only exact multiples of the fundamental in the idealized world. In practice, the acoustics of natural oscillators are far more complicated and tend to vary slightly from this ideal. To compensate for this, the HPS is modified to find the product of the maximum values close to the ideal harmonics. The higher the harmonic, the more allowance is given for drift. Finally, as mentioned above, it is not uncommon for the HPS to have octave errors. For this reason, it is constrained to find a fundamental of at least 70Hz, and the values of harmonics are weighted to favor lower numbered harmonics (so that a very high peak contributes most to itself as the fundamental).

With the fundamental frequency determined by the median of the previous values, each of the previous FFTs is then reduced to just the values of the harmonic series. This reduction follows a similar pattern to that used in the HPS, where the maximum value close to the ideal harmonic frequency is taken.

Potential formants are extracted as the local maxima of each harmonic series, and strong harmonics close to local maxima are merged together to try to determine formants that lie

between the harmonics.

Each formant is then determined in order. The highest value formant in the formant's frequency range is taken from each set of apparent formants, and then the median formant is chosen from this.

Finally, the first and second formant values are used to plot the speaker's vowel as a dot against a background of vowel shapes determined by the same process from reference vowel recordings. Because the dot is likely to bounce around somewhat with error, the last eight determined dots are all plotted, with the older dots appearing fainter. This helps the user ignore occasional misplacement errors made by the program.

The program also displays the roundness of vowels with the color of the vowel shapes and the pronunciation dot. Roundness is a measure of how much energy exists in the higher harmonics of the vowel. Because of this, rounder vowels tend to sound more mellow, while less round vowels sound brighter. The program uses cool colors to indicate round vowels, and hot colors to indicate bright vowels. Unfortunately, the current measure for roundness does not seem to have a high reproducibility.

The reference vowel shapes and colors for Mandarin were determined by using the program to analyze reference vowel pronunciations, saving the determined formant values, and then performing some statistical analysis on them.

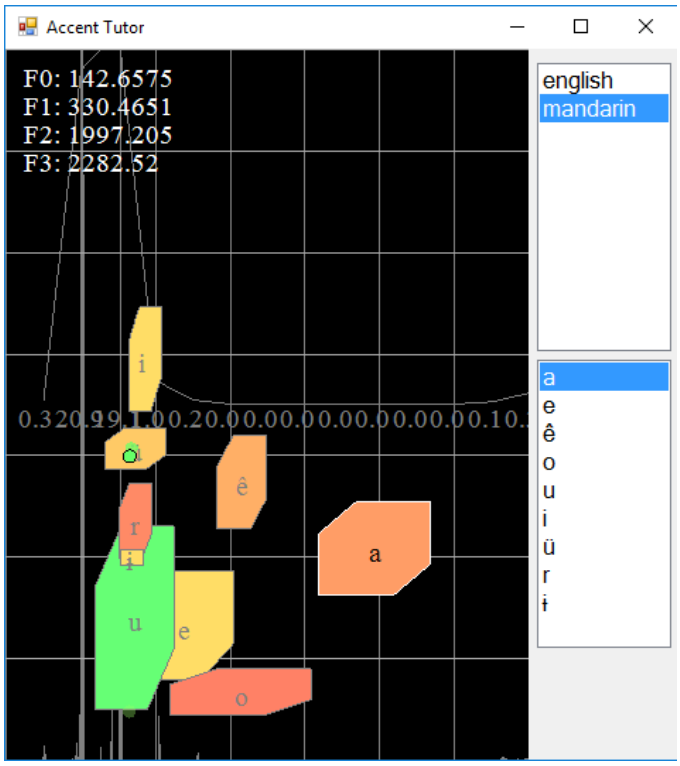

Figure 4: The Windows AccentTutor program. The user is speaking the /ü/ vowel and the program is correctly plotting the vowel, indicating good pronunciation.

4 Software Usage

The program was designed to be easy to use and encourage experimentation. The Android version has nothing to interact with directly at all, and the Windows version simply has a (largely unnecessary) selector for switching between languages. Once the program starts up, the user can immediately begin speaking and experimenting.

With a brief explanation that the program listens for vowels and plots them against a chart of Mandarin vowels, I pointed to the /e/ vowel and asked a friend who doesn't speak Mandarin to try to say it. Unsurprisingly, his first guess was incorrect, but with experimentation, he was able to use the program to find the correct pronunciation.

Some additional explanation would be helpful to the user. It should be explained that the axes of the plot roughly correspond to where the tongue is in the mouth: how far forward or backward it is, and how high or low it is. Since some of the Mandarin "vowels" plotted

are very foreign to the English speaker (e.g. /r/), these would likely need to be explained on their own or demonstrated. AccentTutor would pair well with pronunciation recordings from a native speaker. The speaker could hear the native speaker, emulate that sound, and then use the program to evaluate their pronunciation and experiment with different tongue positions to find the correct one.

5 Validation

Although AccentTutor's primary goal is to aid second language acquisition in students, performing a meaningful evaluation of its efficacy in improving pronunciation would extend beyond the scope of this project. In order to still have a more objective evaluation of the programs ability to consistently distinguish vowels, a small scale preliminary study was designed. Three native English speakers (besides the author) who had studied Mandarin Chinese and could speak relatively fluently were asked to read the text of Genesis 1:1-8 in both English and Mandarin, both at a comfortable pace and with separation between each syllable. These recordings were then analyzed by the AccentTutor program (specifically with the SpectrumDisplay project, which includes tools for saving the analysis) to extract formant characteristics. Finally, these data were analyzed with the CompareLanguageRecordings project. The reference vowel closest to each set of observed formants was found, and then a score for that observation was set equal to the sum of the differences between the first, second, and third formants. The overall score for that recording was simply the average score of each observation.

My hypothesis is that the Mandarin recordings with space in between each syllable will have lower scores than the corresponding recording from the same speaker, and that this effect will be less pronounced in the recordings spoken at natural speeds due to AccentTutor's total use of 1.77 seconds worth of data in each observation.

Table 1: Preliminary Study Result Scores

Subject	Normal English	Normal Mandarin	Slow English	Slow Mandarin
Author	912.5	606.8	872.7	892.0
1	711.9	625.5	547.1	601.2
2	1094.4	772.1	982.4	907.6
3	794.9	631.0	674.4	731.8

These scores are a measure of how much the average vowel observation in a recording differed from the closest reference Mandarin vowel. Lower scores indicate that the program thought the recording sounded "more like Mandarin".

In three of the four subjects, the "Normal Mandarin" recording had the lowest score. And with all subjects the "Normal Mandarin" recording had a lower score than the "Normal English" recording.

I find it surprising that the program found it easier to distinguish between English and Mandarin when spoken at the quick pace of normal speech. It is possible that when asked to put space between each syllable, the subjects unconsciously used less natural and less correct pronunciations for both languages. It is also possible that there are other confounding factors. The study was purposefully conducted using standard texts instead of contrived lists of vowel sounds, but it is likely that the presence of consonants interfered somewhat with the analysis. In addition, while an effort was made to take the recordings in quiet locations, there was still a fair amount of noise.

It seems that the program is able to differentiate English and Mandarin when spoken at normal comfortable speeds, but more subjects would be needed to further verify that.

6 Conclusion

Many of my peers have commented that AccentTutor seems like it would be useful in the study of a foreign language. From personal experimentation with the program, it seems effective at its job of showing the user where their pronunciation is on a vowel plot. However,

there are many areas for improvement. One of the most basic changes would be to enlist the help of a native Mandarin speaker in creating a more accurate Mandarin vowel reference. Another challenge would be to improve the reliability of the program to work with different microphones. Many other improvements to individual parts of the formant analysis process were mentioned earlier. To extend the usefulness of the program beyond just those seeking to learn Mandarin, further reference vowel information could be made for other languages. In the future when I seek to learn other foreign languages, I will work with and improve AccentTutor to aid my pronunciation.