Voice Recognition

By:
Gbenga Badipe
Adrian Galindo
Yuqiang Mu

Voice Recognition

By:
Gbenga Badipe
Adrian Galindo
Yuqiang Mu

Online:
< http://cnx.org/content/col11389/1.1/ >

C O N N E X I O N S

Rice University, Houston, Texas

Collection structure revised: December 19, 2011
PDF generated: December 20, 2011
For copyright and attribution information for the modules contained in this collection, see p. 19.

Table of Contents

Chapter 1

Introduction[1]

1.1 Project Introduction

The project detailed in this collection describes an exploration of the uniqueness and variability of the human vocal tract across different speakers and, by extension, the potential for speaker identification based on their voicing characteristics (that is, the frequency signatures of their vowels).

This group recognizes the existence of more robust voice recognition systems based on theory of greater complexity (e.g. autoregression, Gaussian mixture models) that look at more elements of the subject's speech such as pacing and speed, consonants, etc. However, the scheme detailed below was developed solely based on the characteristics of a speaker's vowels with the hypothesis that a system based on such an approach, if reliable, would potentially be computationally less intense and comparatively simpler to implement in hardware and apply.

The sections below detail, respectively: some of the theoretical concepts relevant to this project; the finer points of the recognition scheme we developed; the experiments carried out to evaluate the scheme, as well as the data and conclusions we obtained from the tests.

[1] This content is available online at <http://cnx.org/content/m41854/1.2/>.

Chapter 2

Background[1]

2.1 Background

2.1.1 Formants

In speech science, formants are defined as the regions of concentrated power in the frequency representation of a vowel. These specific peaks of power are a reflection of a specific configuration of the different parts of the vocal tract; humans produce different kinds of vowels, each one characterized by its pattern of formants, by vibrating their vocal chords and changing the configuration of their articulators (e.g. positioning of the lips, tongue, and jaw). Most vowels across a single person's speech can be completely described by their F1 and F2, the formants with the lowest and second-lowest frequency centers; notable exceptions to this are the "rhotacized" or "r-colored" vowels, which depending on the speaker's vocal tract may often feature overlapping F1 and F2 with other non-rhotacized members of the vowel space and must also be identified by their F3, and a few lip-rounded vowels as distinguished from their unrounded counterparts, which have lowered high-level formants in general but may still overlap with other vowels in F1 and F2.

[1] This content is available online at <http://cnx.org/content/m41768/1.3/>.

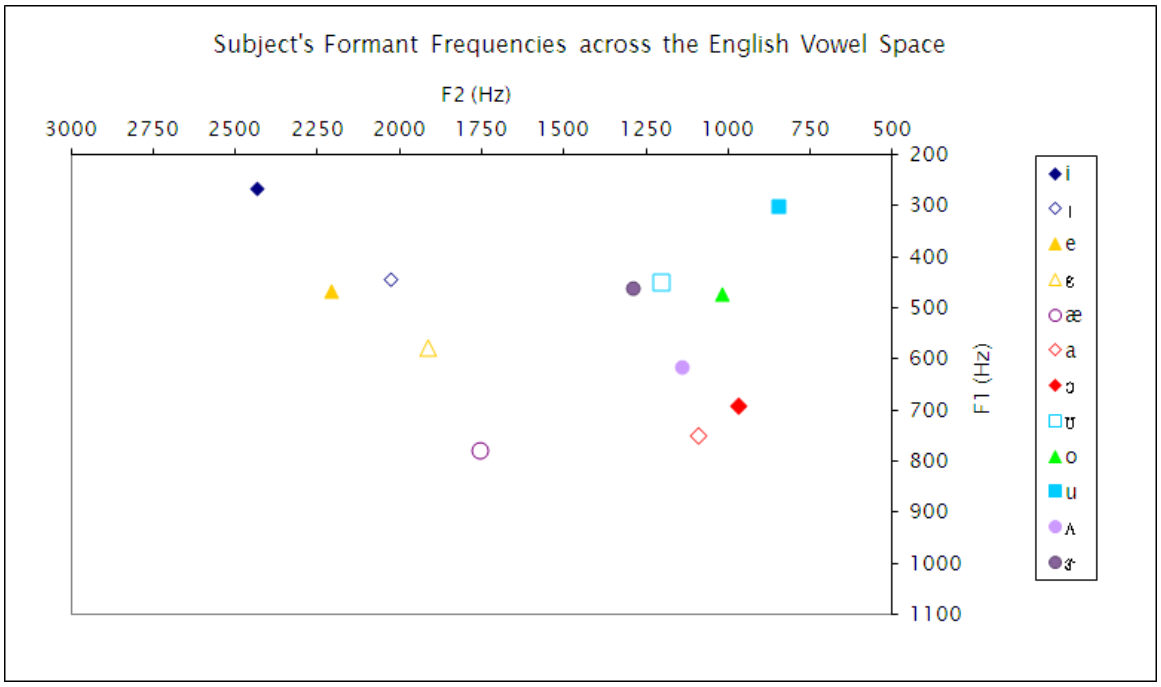

Figure 2.1: Example formant plot of subject. Note the closeness between [] and [u].

However, the actual frequencies of the formants are obviously not constant across different people; in fact, the relationship between the formant patterns of different speakers saying the same vowel is highly erratic. This can be attributed to the high degree of variability found in the physical characteristics of the human vocal tract; males often have larger and longer vocal tracts than females, two people may not have the same jaw size or shape, etc. This variability affects the resonant frequencies of the vocal tract's various components, so that even if two people are making the same articulatory gestures their formants will not only be different, but different in a decidedly nonlinear way for each vowel; that is, physical differences between speakers make their formants for multiple vowels differ in different ways. For example, pertaining to the features of rhotacized vowels mentioned above: both F1 and F2 of the rhotacized vowel [] (as in "herd") overlap with those of either [] (as in "hood") and/or [u] (as in "who'd) for many speakers; however, short preliminary measurements of our subjects (all speakers of standard American English, though with slight and different dialectical influences of his own) taken for testing purposes before running any actual trials revealed that in one there was only slight F1 overlap between [] and [], in another there was F1 overlap between [] and [u], and in a third there was no clear overlap at all. Larger studies in the ream of Linguistics on this particular kind of overlap do report that it is rather commonly found, but in varying degrees: a testament to the amount of vocal tract diversity that exists in the human population.

This variability makes it highly unlikely that two people will have anything close to a nearly identical set of formant frequencies across a large set of vowels; more topically, this means that a voice-recognition system that knows the average formants of a specific person's vowels would be difficult to dupe for other people solely on the basis of their vowel characteristics. For all intents and purposes, the more vowels used in the trials the more specific and rigorous the trials become (our project ended up utilizing all twelve members of the English vowel space).

2.1.2 "hVd" words

The "hVd" words are a class of monosyllabic English words with a single vowel nucleus that follow the specific pattern of beginning with an [h] and ending with a [d]. This set is particularly interesting because of the uninvasive nature of the word-initial and word-final consonants; for the most part they do not interfere with the frequency characteristics of whatever vowel is in between them. In contrast, most of the other consonants in the English language will, in some way or another, alter or bend the formants of the surrounding vowels (e.g. vowels with adjacent nasals like [n] or [m] may become nasalized, changing their formants considerably; formants of vowels next to approximants like [l] will "drift" toward those of the approximant targets instead of remaining steady; etc.). It is for this reason that these hVd words were used in our trials; they provide a clear picture of what a speaker's uncorrupted vowels look like and they represent a middle ground of sorts between saying pure vowels and normal speech with other, more complicated words.

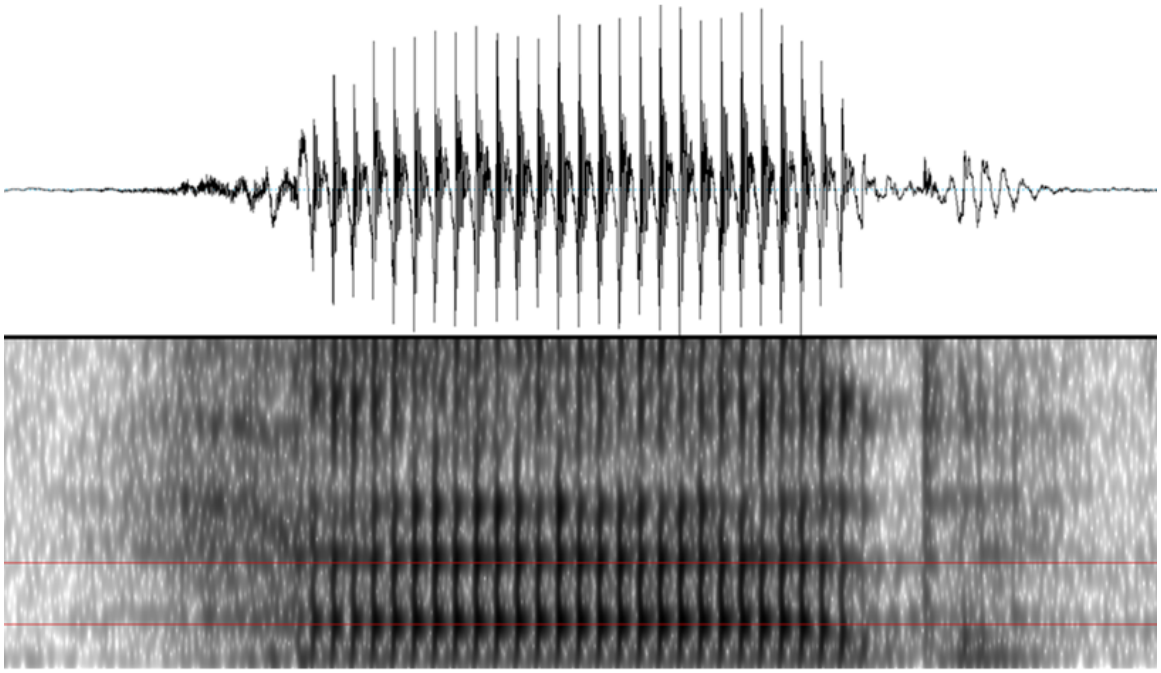

Figure 2.2: Example waveform and spectrogram of the word "had." Note the stability of the formants (centers approximated by red lines).

2.1.3 Windowing

It is quite commonplace in the realm of Fourier analysis (especially spectrum analysis of sound waveforms) to obtain a time-domain-based view of the frequency content of a signal. This is addressed by "windowing" the data in question in order to obtain a series of segments of the original waveform, each representing a particular segment in time of the (discrete-time, naturally) signal; the process of creating each of these segments is, in short, taking the product of a windowing function of specific length and the signal at that particular time.

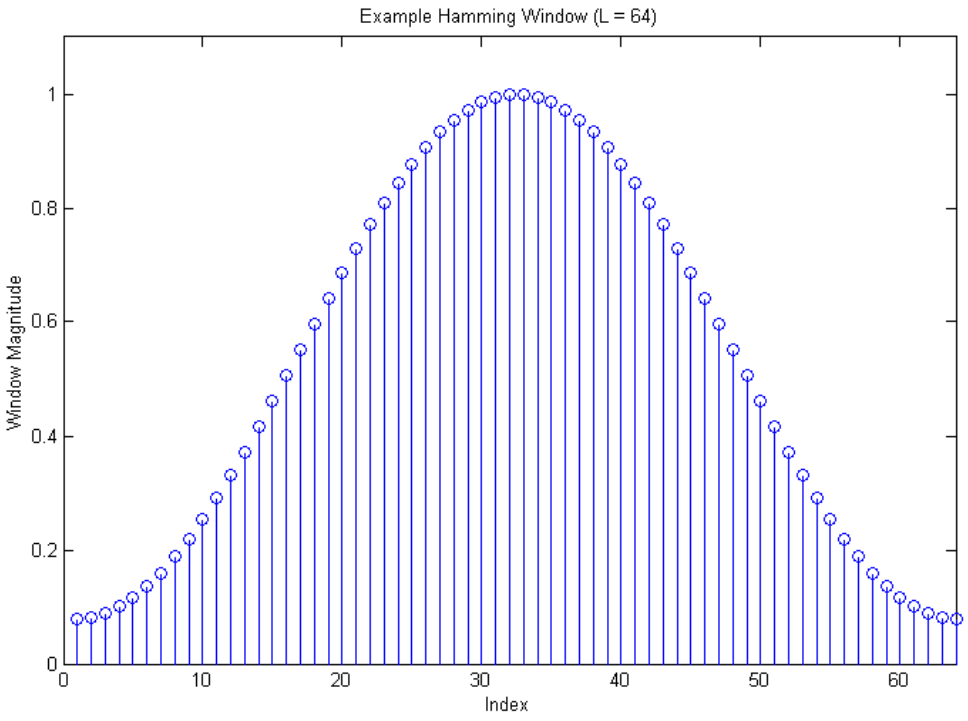

Figure 2.3: Example plot of a length 64 Hamming window (magnitude vs. discrete-time index).

The windowing function is zero outside its interval, and is usually tapered down to zero or near zero to reduce the erroneous high-frequency content that would be introduced by sharp cutoffs at the window ends; in addition, the windows are staggered in steps of half the window length to help mitigate potential information loss due to the attenuated ends of the window function. The window of choice for this project was the Hamming window, a "raised cosine" shape (see the above figure). To be brutally honest, this window function was not the optimal choice for this particular application; however, the window choice is not a large issue for us because the core of this project does not depend much on factors like high side lobe attenuation and falloff for the window function (we are only concerned with locating the positions of peaks within a certain frequency range for each window of the signal, and in fact we bandpass-filter out all the content outside the normal human range for the two formants in question).

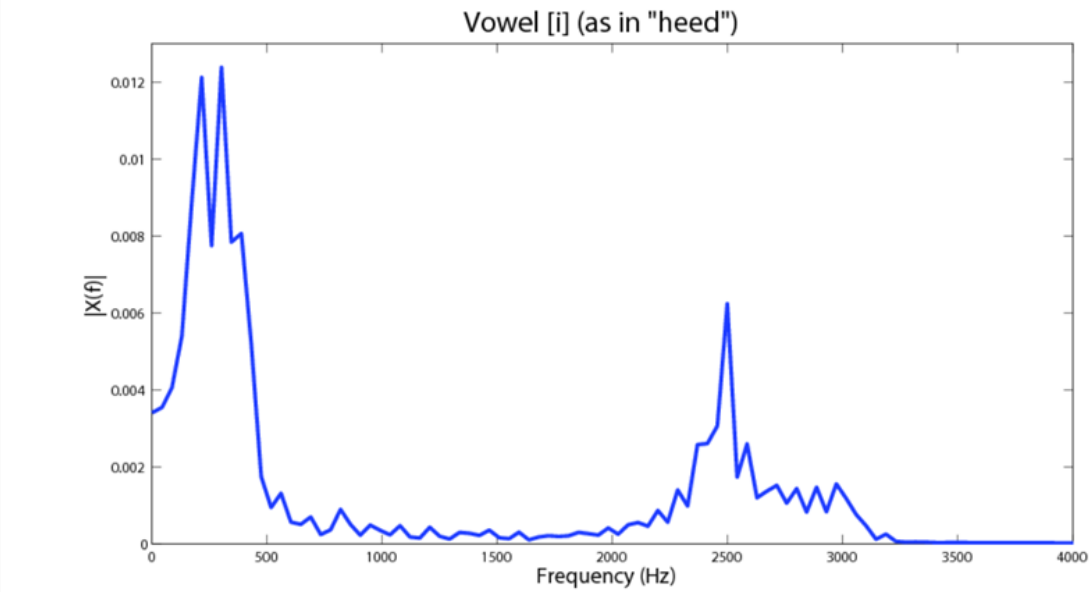

Figure 2.4: Example plot of the FFT of a length 1024 window of the vowel [i] in "heed," spoken by a test subject. Clear formant spikes observed at ∼400 Hz and ∼2500 Hz in this particular window.

A window length of 1024 samples was chosen for this project with consideration given to the nature of the time-frequency tradeoff inherent to the windowing process; with increasing time resolution (i.e. more windows for a given signal, more bins of time information) comes less frequency information, and with increasing frequency resolution comes less temporal information (i.e. less windows for a given signal, less bins of time information). The standard accepted window length used widely in linguistic analysis applications is 0.025 seconds, an interval that balances time and frequency resolutions for the windowed signal reasonably well. With sampling frequency of 44100 Hz for our waveforms, we chose the nearest power of 2 that would give us the necessary time interval (1024 samples is slightly under 0.025 seconds, but the margin is so small that it does not affect our results at all).

Chapter 3

Methodology[1]

3.1 Methodology

3.1.1 Purpose

To uniquely identify a speaker based on their vowel formants using frequency domain analysis of a speaker's voice.

[1] This content is available online at <http://cnx.org/content/m41762/1.14/>.

3.1.2 Method

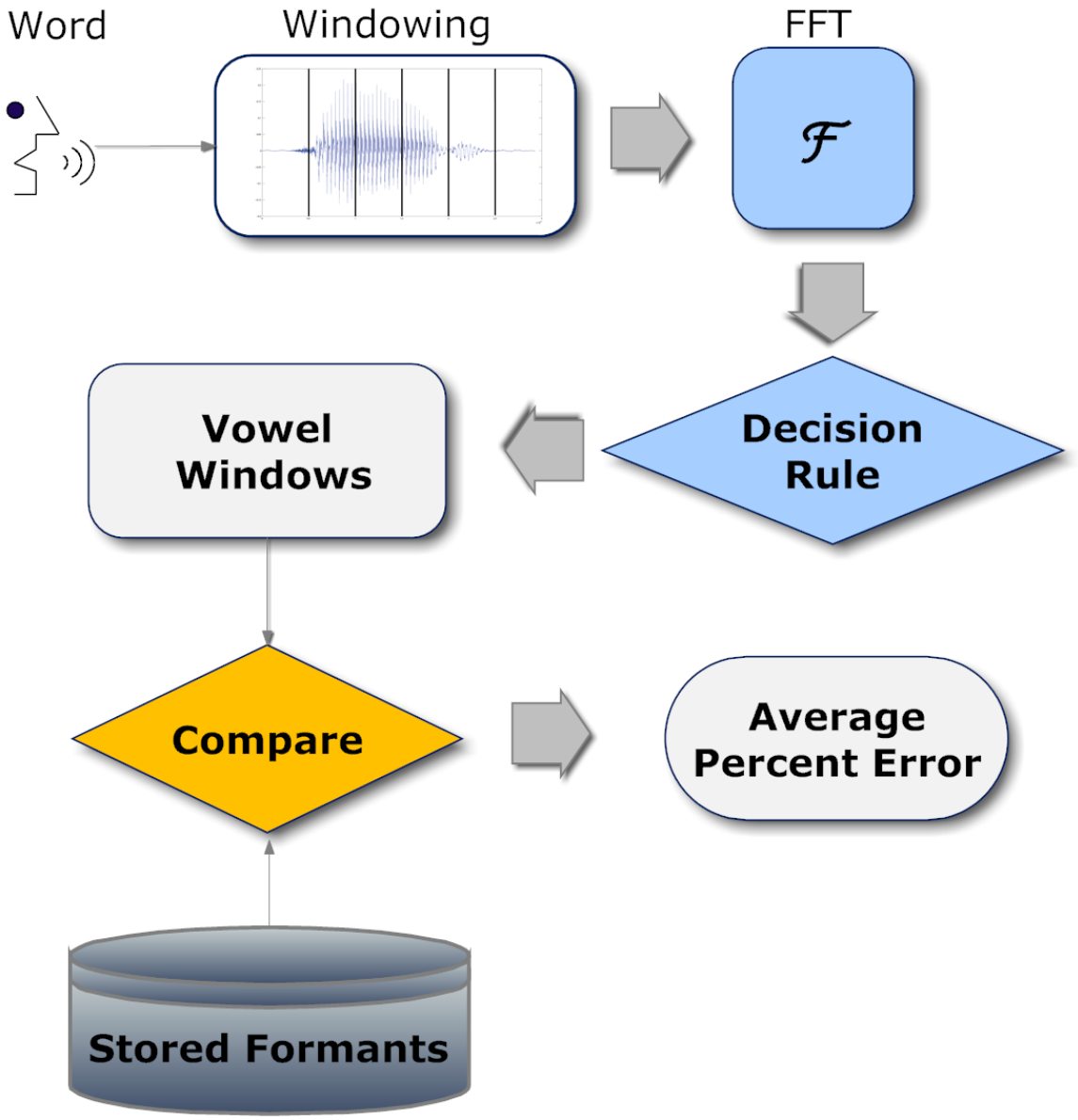

Figure 3.1: Block diagram of our system's processing of a single-word input. Explained in detail below.

3.1.3 Overview

The first step in doing Fourier analysis on the speaker's voice is to first split the signal into windows(see background for more info) such that the FFT(Fast Fourier Transform) algorithm can be used to find the

Fourier transform of the signal. This usually just requires that the number of windows is a power of 2, so we chose a window size of 1024 which translates to about .025 seconds in the time domain. This choice allowed us to capture substantial but not excessive information in the time domain. Moreover, we chose to use the hamming window, which has a slower cutoff, to avoid the various negative effects(see background section) that a window with a sharp cutoff can cause. After this process is done on every window we are essentially left with an Array of Fourier transforms, with each transform representing the speaker voicing at different moments in time.

To filter out any excessive noise in frequency ranges where we know vowel information cannot exist (as no human could have a vocal tract that produces vowels with formants in these ranges), we next apply a bandpass filter to our signal in the ranges of 0-100Hz and frequencies above 3500Hz.

Once this is done, the formants and their magnitudes are found in each window by finding the peaks (see **Finding Peaks**) within two frequency ranges i.e. where the two formants are known to lie from **Tuning.** Then a **Decision Rule** (explained below in more detail) is used to remove all the windows that are sure not to contain a vowel. Using those windows, we calculate a percent error based on where the speaker's formants lie with respect to the template speaker's formants, which the system is tuned to (See **Tuning**). This process is repeated for every vowel in the English language (each subject wishing to access the system must input in sequence the set of "hVd" words described in the Background section for our trials); once that is complete an average percent error is calculated based on the subject's performance across all vowels and a final decision is made based on this percent error.

3.1.3.1 Decision Rule

To determine which windows possible contain vowels, the decision rule essentially removes any windows whose formant one or formant two magnitudes differ by more than 2 standard deviations away from the formant one and formant two maximum magnitudes (across windows). It also removes any false positives i.e formants that were detected outside of the possible formant range(above 100Hz and below 3500Hz). What is left are all the windows in which the vowel is being voiced. This essentially filters out all non-vowel windows and to remove any additional noise that wasn't removed from the band pass filter.

3.1.3.2 Tuning

To 'tune' our system to the template speaker, we first had to find where the speaker's frequencies generally lie. This would form the basis for our percent error calculation when checking against another speaker's formants. To do this we had the user voice every vowel multiple times in which we recorded where we found the first and second formants. These trials were then used to find the range in which the first and second vowel formants lie and the average first and second formant frequencies for each vowel. These were hardcoded into our code for comparison against the formants of the speaker.

3.1.3.3 Finding Peaks

Finding peaks involved first finding the maximum magnitude in the window and then checking that it's a peak(increasing on the lefts side and decreasing on the right side of the maximum). This prevented false positives, for example detecting a magnitude that may be the highest in the window but is not a peak. To do this we simply have have a sieve that is centered around the maximum magnitude. We then check that values to the left and the right of the maximum magnitude are lower than the maximum magnitude. If this is not the case the sieve moves to the right and a new maximum is found in the window and repeats the process until a peak is found.

3.1.4 Code

Download our code here[2]

[2] http://dl.dropbox.com/u/13097644/code.zip

Chapter 4

Experiments, Results, Conclusions, Future Work[1]

Adrian A. Galindo Experimental Methodology, Conclusions, and Future Work

4.1 Experimental Methodology, Conclusions, and Future Work

We now turn our attention to the work that went into testing our model to ensure it worked as designed.

4.1.1 Experimental Methodology

In order to test if our system works we had to ensure that the algorithm would produce the correct result (allow, deny) regardless of which recording of the template speaker or the two intruders we used.

1. The Template User records a series of just vowels from the English language in order to train the algorithm. This recording is done on a separate day to better simulate the common variability in vocal resonance found in day to day speech. See the results for our template speaker in Figure 4.2.
2. Each of the 3 users (template, intruder 1, intruder 2) records each of the 12 HVD words(see Figure 4.1 which encapsulate the entirety of the English vowel space.
3. We run the algorithm, comparing each of the 3 recordings against the originally stored vowels of the template user. A percent error is computed for each user (even the template speaker) and a decision of allow or deny is made based on the minimum error.
4. Steps 2 through 3 are repeated a total of 3 times to ensure some measure of repeatability.

[1] This content is available online at <http://cnx.org/content/m41764/1.4/>.

#	Vowel	Word	#	Vowel	Word
1	[i]	“heed”	7	[ɔ]	“hawed”
2	[ɪ]	“hid”	8	[ʊ]	“hood”
3	[e]	“hayed”	9	[o]	“hoed”
4	[ɛ]	“head”	10	[u]	“who’d”
5	[æ]	“had”	11	[ʌ]	“hudd”
6	[a]	“hod”	12	[ɝ]	“heard”

Figure 4.1: HVD Words

		AVERAGE		REP1		REP2		REP3	
Word	**Vowel**	**F1**	**F2**	**F1**	**F2**	**F1**	**F2**	**F1**	**F2**
heed	i	266.9	2437	270.3	2473	262.7	2468	267.8	2370
hid	ɪ	443	2028	447.1	2055	434	2020	448	2010
hayed	e	467.8	2207	471.3	2228	465.3	2219	466.7	2174
head	ɛ	578.9	1914	608.7	1930	577.3	1906	550.9	1907
had	æ	779.3	1758	782.4	1739	786.4	1768	769.2	1768
hod	a	749.9	1090	745.1	1085	758.1	1088	746.6	1097
hawed	ɔ	692.5	968.2	684.1	958.9	696.8	954.9	696.7	991
hood	ʊ	449.1	1207	433.7	1182	453.9	1200	459.8	1237
hoed	o	472.9	1020	471.3	1015	473.2	1007	474.1	1039
who'd	u	301.7	849.1	278.8	811.4	338.3	867.4	288	868.4
Hudd	ʌ	615.7	1139	617.6	1119	611.6	1115	617.9	1185
heard	ɝ	460.1	1292	466.7	1288	456	1280	457.5	1306

Figure 4.2: Our Template Speaker’s Average Formant Frequencies

4.1.2 Results

When we ran our experiments we found that while the template speaker was occasionally not the person with the lowest percent error on a per-vowel basis he had the lowest percent error across the entire vowel space. We found this result to be consistent across several different trials. In each a trial a new recording of the same three people was made - often on different days, in order to allow for maximum day to day variability in voice and recording positioning. We wanted to make sure that it was not just a particular set of data that looked good against the template speaker. In addition we wanted the template speaker to make several recordings to ensure that the system could cope with differences in both the template speaker and in the potential intruders.

Subject	Average % Error Trial 1	Average % Error Trial 2	Average % Error Trial 3
Template Speaker	7.673	8.902	7.747
Speaker A	11.787	11.301	11.102
Speaker B	20.676	16.933	12.389

Table 4.1: Experimental Results

More detail on each of the trials can be found in Figures Figure 4.3, Figure 4.4, Figure 4.5

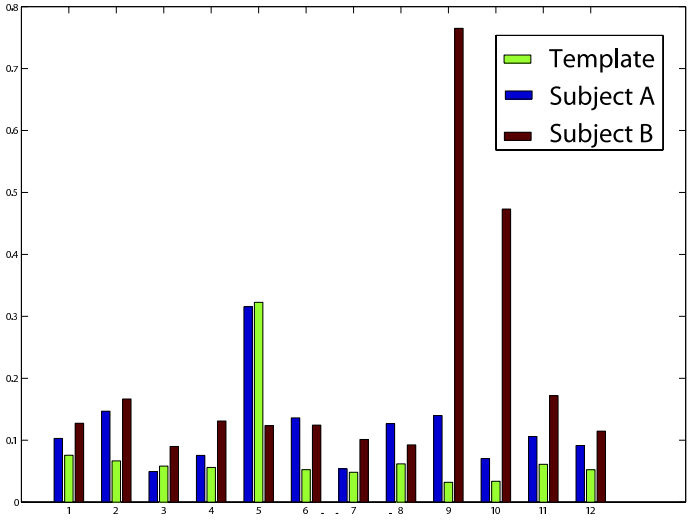

Figure 4.3: Trial 1 Note: the X-Axis all 3 trials is labeled according to the HVD listings in Figure 4.1

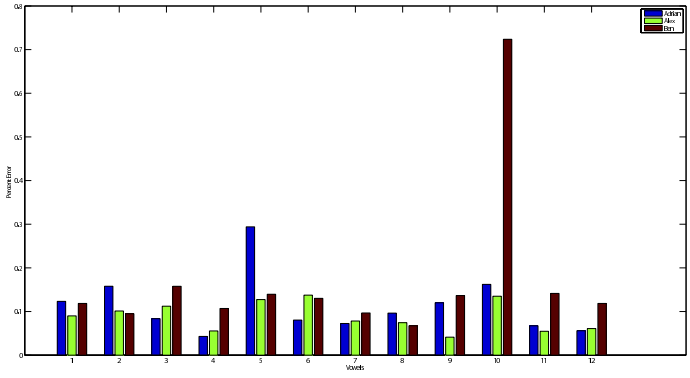

Figure 4.4: Trial 2

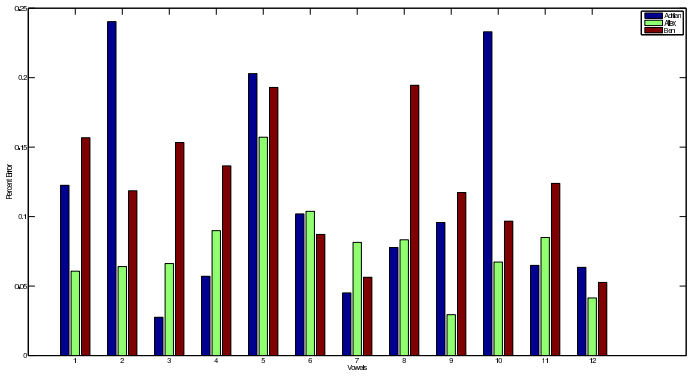

Figure 4.5: Trial 3

4.1.3 Conclusions

In conclusion we found that the system we built does exactly as we intended: it is able to tell the difference between the user it was tuned to and everybody else. There were a few caveats to the system we built. In particular we noticed that when the template speaker got a little sick the system was no longer able to grant him access. Due to the newfound resonant characteristics of the template speaker's vocal tract caused no doubt by the addition of mucous and swelling, it was quite difficult for the system to match it to the stored template formants. This is a bit of a sticky problem because we want to tune the system as sharply as we can to the template speaker's resonant characteristics so that even a slight change would cause the system to revoke the user. These slight changes seen through a security lens are an added opportunity to take advantage of the entire variability space to increase "passphrase" entropy and make the entire system more secure. The same slight changes seen through a usability perspective however are seen as a rather formidable annoyance for the sick template user attempting to gain access to his environment. This balance between lenience and security is one that most biometric security systems must weigh. Anyone who has attempted to use a fingerprint reader after a long shower (raisined fingers) knows which way the company that made that biometric system choose to lean towards. In the same manner we figured that while the balance could definitely use some fine tuning we figured it would be best to produce a false negative than to produce a false positive and allow a rouge user access to the secure environment. Of interesting note we noticed that when a user changed the pitch of his voice in order to try to match the template speaker the position of the formants in the frequency domain changed very little. This result shows that the system is indeed tuned to a user's resonant characteristics and not just the pitch of the voice. This result also makes it extremely difficult for a would be attacker to gain access to the system even if he did know the exact passphrase and what the secured template user sounds like.

4.1.4 Future Work

In its current state the formant detection system makes use of several very convenient MATLAB features such as the filter design tool to rapidly create a computable filter. It also leverages the rather large amount of computational power available to modern computer platforms to make some rather sharp filters and decisions. While it is true that we did not set out to make the most efficient mechanism by which to identify a user based on formants, we would truly like to see this system implemented in an embedded hardware environment. This would require that we trim the program and its requisite filters considerably just to

get the algorithm to fit within the confines of the restricted memory space available in most embedded environments. In addition this type of embedded security system is of little use if it cannot be run at real or near-real time speeds. Use of a powerful embedded environment such as an FPGA could see this type of implementation without a drastic reduction in the filter sharpness; however such an implementation would also require considerable effort in order to port the filter and surrounding decision rules into the hardware reconfigurable languages of Verilog or VHDL.

4.1.4.1 Hidden Feature: Vowel Recognition

In order to accurately guess whether the right vowel is even being spoken before attempting to compute a percent error from a potentially wrong vowel to the template speaker we had to build in some sort of vowel recognition feature. The recognition implementation is rather crude but we found it to be quite accurate in our tests. We never tested the vowel recognition alone but rather saw the results of it in the streaming debug statements our program can output. Based solely on this we believe that the subsystem could be expanded and refined to both aide in the voice recognition process and to increase security by fully checking the phrase spoken.

Index of Keywords and Terms

Keywords are listed by the section with that keyword (page numbers are in parentheses). Keywords do not necessarily appear in the text of the page. They are merely associated with that section. *Ex.* apples, § 1.1 (1) **Terms** are referenced by the page they appear on. *Ex.* apples, 1

Attributions

Voice Recognition
ELEC 301 Voice Recognition Project

About Connexions
Since 1999, Connexions has been pioneering a global system where anyone can create course materials and make them fully accessible and easily reusable free of charge. We are a Web-based authoring, teaching and learning environment open to anyone interested in education, including students, teachers, professors and lifelong learners. We connect ideas and facilitate educational communities.

Connexions's modular, interactive courses are in use worldwide by universities, community colleges, K-12 schools, distance learners, and lifelong learners. Connexions materials are in many languages, including English, Spanish, Chinese, Japanese, Italian, Vietnamese, French, Portuguese, and Thai. Connexions is part of an exciting new information distribution system that allows for **Print on Demand Books**. Connexions has partnered with innovative on-demand publisher QOOP to accelerate the delivery of printed course materials and textbooks into classrooms worldwide at lower prices than traditional academic publishers.