University of Pennsylvania
Center for Sensor Technologies
Department of Electrical Engineering
Philadelphia, PA

SUNFEST Report
August 1999

Identifying Key Phoneme Features in Spectrograms

Patrick Lu (Electrical Engineering), Princeton University
NSF Summer Undergraduate Fellowship in Sensor Technologies
Advisors: Ahmed M. Abdelatty Ali, Dr. Jan Van der Spiegel, Dr. Paul Mueller

ABSTRACT

Spectrograms carry all necessary information for reliable human and computer perception of speech. This paper discusses the importance of spectrogram features used by a recognition algorithm developed by Ali et al. as they relate to human perception. Features, including MNSS, burst frequency, formant transitions, voicing onset time, and voicing/unvoicing information are defined and their importance to computer stop consonant recognition described. Confirming many previous findings, burst frequency and formant transitions were found to be most important in the perception of speech synthesized from spectrograms while other features played a secondary role. Software tools developed that should facilitate other similar investigations are described.

Table of Contents

1. Introduction
   - 1.1 Anatomy of Stop Consonants
   - 1.2 Representations of Speech
2. Methods
   - 2.1 Problem Description
   - 2.2 Algorithmic Methods of Automatic Stop Consonant Recognition
     - 2.2.1 Burst Frequency and Vowel Second Formant Frequency
     - 2.2.2 Formant Transitions
     - 2.2.3 Maximized Normalized Spectral Slope
     - 2.2.4 Voicing
   - 2.3 Software Tools
     - 2.3.1 The Speech Synthesis Program and S.A.M. GUI
     - 2.3.2 New Features
   - 2.4 Testing Human Perception
     - 2.4.1 Burst Frequency and Vowel Second Formant Frequency
     - 2.4.2 Formant Transitions
     - 2.4.3 Maximized Normalized Spectral Slope
     - 2.4.4 Voicing
     - 2.4.5 Other Tests
3. Challenges
4. Recommendations
5. Conclusion
6. Acknowledgments
7. References

1. INTRODUCTION

Though it has been extensively studied since the 1950s, computer speech recognition continues to pose a modern engineering problem. Challenges include the difficulties of processing continuous speech, distinguishing between members of a large vocabulary, creating a speaker-independent system, and coping with environmental noise.

Speech recognition has already found many industrial and other applications. Recent uses of this technology include interactive telephone menus, automated credit-card number retrieval systems, and computer dictation. Automatic recognition of speech is more than a gimmick: people communicate most efficiently via speech, with the average person speaking approximately five times faster than he can type and perhaps ten times faster than he can write [1].

However, the various problems mentioned above, among others, have prevented the creation of a system capable of processing unrestricted human speech with an unlimited vocabulary. Instead, the best speech recognition systems to date have been limited to specialized applications, with the vocabulary restricted to a certain set of jargon and lengthy training period often required to adjust a computer to each individual speaker.

In line with the ultimate goal of creating a completely speaker-independent platform capable of discriminating among elements of a vocabulary of arbitrary size, we have turned our efforts to studying the acoustic features of phonemes, the basic, indivisible building blocks of speech. Phonemes, when properly spoken, all exhibit a set of invariant, speaker-independent properties. They also enable computers to represent language in a more compact form. For instance, phonemes can reduce the intractable problem of distinguishing among the 5000+ words of commonly used English to recognizing 50–60 subword units.

Many different methods, with varying success, have been used to implement speech recognition. The most popular and successful ones use Hidden Markov Models (HMMs), or probabilistic models that rate how much a given speech signal deviates from an internal template. However, HMMs are speaker-dependent and require inconvenient training periods. Our approach to speech recognition is to represent phonemes in a graphical form, where key acoustic/phonetic features are made visible. We hope to make computers recognize speech by searching for these features.

1.1 Anatomy of Stop Consonants

The focus of this project has been on a class of phonemes known as stop consonants. Though they have been studied since the birth of speech recognition, even modern automatic speech recognition systems are unreliable at classifying them because of their short duration and context dependence.

This set includes six phonemes, /t/, /k/, and /p/, which are unvoiced, meaning that the vocal chords do not vibrate when they are spoken, and /d/, /g/, and /b/, which are voiced. Stops are formed by the complete obstruction of the vocal tract. The exact location of this obstruction, called the place of articulation, can be the velum (palate), where /k/ and /g/ are formed; the alveolus, where /t/ and /d/ are formed; or the lips, where /p/ and /b/ are formed.

As illustrated in Figure 1, stop consonants begin with a period of silence, called the closure. This part of the consonant is caused by the obstruction of the vocal tract. Next comes a noisy period called the burst, which is typically followed by a vowel. Collectively, these are known as an utterance.

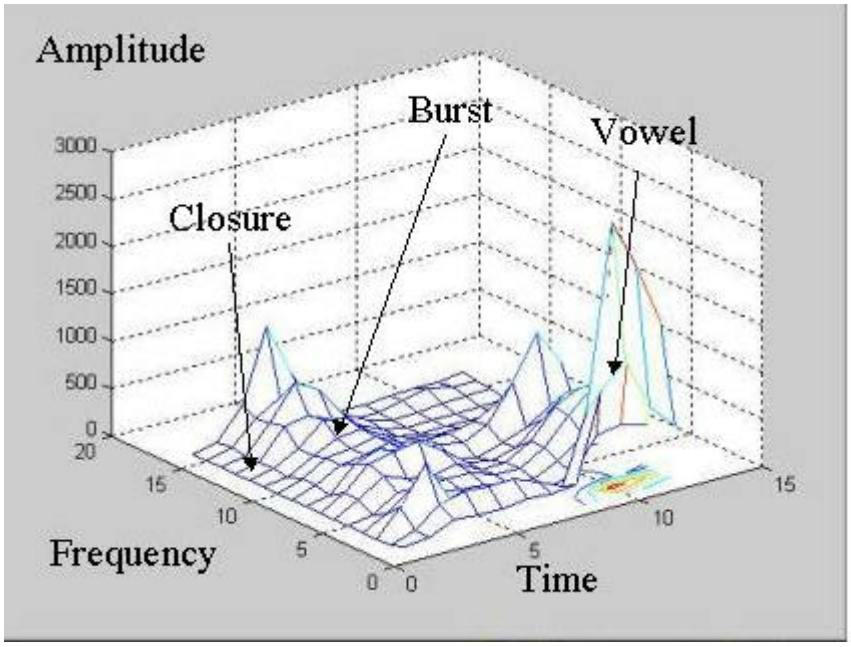

Figure 1: Three dimensional plot of a stop consonant followed by a vowel.

1.2 Representations of Speech

To represent a speech signal graphically, we used spectrograms, or three-dimensional plots, with time on the x-axis, frequency on the y-axis, and a third dimension, intensity, usually depicted in the form of colors.

Spectrograms are usually interpreted in one of two ways. The first is as plots where intensities represent the amplitudes of sine waves of specific frequencies at certain points in time. Speech is generated from these graphs through sine wave synthesis, where amplitude-modulated sine waves are superposed to form a single time signal.

To create these spectrograms, a speech signal is divided into short fragments of equal duration, typically between 10 and 30 ms, and frequency data is extracted from each one. When a linear frequency scale is desired, Fourier transforms are used. Other times, when a logarithmic scale is desired, matched filters are used. This is the same way humans perceive sound. Hairs in the cochlea form logarithmically spaced filter banks. Bark's scale is a commonly used logarithmic scale because it is modeled after the frequency sensitivity of human hearing.

The second way to interpret spectrograms is through cepstral analysis.. This method models speech as comprising a source, corresponding roughly to the human vocal chords, and a filter, which contains information about words being shaped by the vocal tract. To model the time dependence of the filter, linear and time-invariant (LTI) filters, each of which has static response characteristics, are concatenated in time. Again, speech is divided into windows where each one is the convolution of a source with the impulse

response of one of the LTI filters (convolving an input with the impulse response of an LTI system is a standard way of determining the output of the system – in this case the input is the source signal and the system is the filter represented by the vocal tract). Since speech information is primarily encoded in vocal tract positions and not the source (which contains information about inflection and mood), in cepstral analysis, spectrograms represent the frequency data of each of the individual LTI filters, each of which occupies a different position in time.

These spectrograms have an advantage over those used in sine wave analysis because of their speaker and mood independence. Cepstral spectrograms are unaffected by the pitch or cadence of a speaker's voice, which becomes part of the source data.

To separate the source and filter, the cepstrum of a speech signal is taken. This operation is defined to be a Fourier transform followed by a logarithm followed by an inverse Fourier transform. It can be shown that the cepstrum of the convolution of two functions is the sum of the cepstrum of each of the individual functions. In addition, if the two original functions have different frequency characteristics, as the source and filter do, their cepstrums will occupy different positions in time, making separation easy.

2. METHODS

2.1 Problem Description

This project had two parts. The ultimate goal was to evaluate the importance of various acoustic/phonetic features of stops used by automatic speech recognition algorithms as they pertained to human perception. The second was incidental to the first objective: to develop software tools that would aid in such an investigation.

2.2 Algorithmic Methods of Automatic Stop Consonant Recognition

An automatic speech recognition system developed by Ali et al. [2] using a hard decision algorithm has a demonstrated accuracy of 97% for voicing detection, 90% for place of articulation detection, and 86% for overall classification of stops. The following are the main features used for classification by the algorithm.

2.2.1 Burst Frequency and Vowel Second Formant Frequency

Burst frequency (BF) is a parameter aimed at finding the frequency where power is concentrated in a stop. It is defined by A. Ali et al. [2] as:

$$BF = \min_{j:time\_during\_burst} k_j, where: Spec_{kj} = \max_{i:all\_filters} (Spec_{ij})$$

Essentially, at each time position of the stop burst, there exists a position of maximum intensity. Each of these positions has a corresponding frequency. The burst frequency is the minimum value of that set.

A formant is a peaked region in a spectrogram. Formants are numbered from lower frequencies up, so the second formant of a vowel is the peak occupying the second-lowest frequency position.

The speech recognition system implemented by A. Ali et al. heavily relies on the burst freqency of a stop and the second formant frequency of its following vowel for the identification of alveolars and velars.

2.2.2 Formant Transitions

The frequencies of formants often change with time. This is known as formant motion. Liberman et al. [3] noted that that how vowel formants moved was determined by the place of articulation of the preceding stop. These motions occur during transition periods between the onset of a vowel and its nucleus, and are falling for velars, rising for labials, and are sometimes falling and sometimes rising for alveolars, depending on the vowel. The algorithm developed by Ali et al. places formant transitions in a secondary role for stop identification, because of the unreliability of determining the motion of the formants. However, clear and strong formant motions will override all other factors considered by the algorithm.

2.2.3 Maximized Normalized Spectral Slope

Known as MNSS, the Maximized Normalized Spectral Slope is a parameter affected by the steepness of a stop burst's peaks and its energy relative to the following vowel. It is defined as:

$$MNSS = \max_{j:release\_burst}\left(\max_{i:all\_filters} Diff(yenv_{ij})\right) \Big/ \max_{j:all\_utterance}\left(\sum_{i:all\_filters} yenv_{ij}\right)$$

This represents the magnitude of the maximum value, over the stop burst, of the partial derivative of the spectrogram intensity with respect to frequency, divided by the maximum of the set of values created when the intensities over all the frequency channels are summed.

The algorithm employed by A. Ali et al. uses MNSS values to distinguish between labials and other stops, where labials are classified by their flat spectra and, thus, low MNSS values.

2.2.4 Voicing

Two main parameters were used by A. Ali et al. to detect voicing in stops: prevoicing and voicing onset time (VOT).

Prevoicing is voicing during the closure of a stop. Usually, this appears as low - frequency energy before the actual burst of the stop.

Voicing Onset Time (VOT), is the duration of the stop burst. Unvoiced stops were found to have longer VOTs.

2.3. Software Tools

2.3.1 The Speech Synthesis Program and S.A.M. GUI

Written by Gavin Haentjens (EE) of the University of Pennsylvania, the speech synthesis program is a collection of Matlab scripts that generates speech from spectrograms.

The Spectrogram Analyzer & Manipulator, or S.A.M, was first developed by O'Neil Palmer (CSE) and Kelum Pinnaduwage (EE) of the University of Pennsylvania, with the goal of allowing a researcher to edit and create acoustical patterns to better understand speech.

These two packages were connected by their choice of file format for storing spectrograms.

2.3.2 New Features

Chief among the new features were hacks of both the speech synthesis program and S.A.M.(2.0) that permitted an integration of the two programs into one platform, enabling the user to edit and play spectrograms on the fly. This was accomplished through ActiveX automation, with S.A.M. acting as the client and Matlab acting as the server.

The user is given a choice between sine wave synthesis and cepstral synthesis. If he chooses sine wave synthesis, he is given the option of randomizing the phase of the superposed sine waves. When using cepstral synthesis, a source file must be specified.

Also, as the frequency channels of a spectrogram using cepstral analysis are calculated differently from those of a sine-wave analysis, choosing cepstral synthesis will cause S.A.M. to use this formula for calculating the frequencies of each of the channels:

$$Frequency = \frac{Sampling\_Frequency}{2} \times \frac{k}{N},$$

where k = 0 . . . .N – 1 represents the filter number and N represents the total number of filters.

Also, the user is able to see three-dimensional plots of selected portions of the spectrogram. Often, peaks and other features can be more intuitively understood when they are represented as shapes rather than colors.

Another added function was the ability to upload a matrix from Matlab into the copy buffer of S.A.M. In conjunction with the ability to download matrices that was a result of the on-the-fly speech synthesis implementation, this allows the user to take a portion of the spectrogram, process it in an arbitrary manner in Matlab, and to send it back to the GUI to add it on to the main spectrogram.

We found that averaging every 16 channels together in a 256-channel spectrogram to create 16 data-carrying channels and replacing the vacated channels with all zeros leads to dramatically improved speech quality. Therefore, when the user selects cepstral analysis, he is also given the choice to zero-pad the spectrogram before speech synthesis. This feature, illustrated in Figure 2, is primarily used for 16-channel spectrograms created in the above manner.

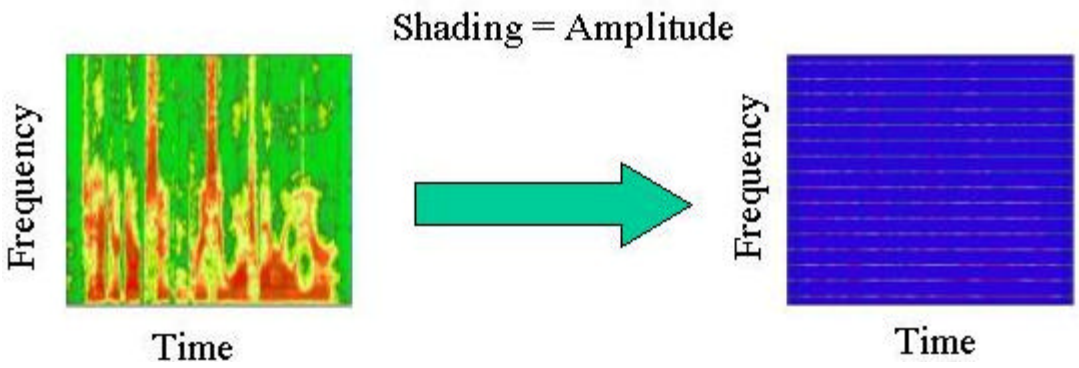

Figure 2: Converting a spectrogram from 256 channels to 16 and adding 0-padding

2.4 Testing Human Perception

Human perception was tested by taking spectrograms that generated intelligible speech when played back and modifying them to see how perception was affected. Because of time constraints, fewer trials were run than desired and many experiments were not performed on spectrograms using sine wave analysis.

The main method of testing human perception was as follows:

- Two different spectrograms, A and B, containing different stop consonants and preferably the same following vowel are concatenated into one spectrogram.
- The burst of B is replaced with that of A.
- A listening test is performed on the modified B. If the utterance sounds as if it begins with the original consonant, then important acoustic information must reside in the vowel. If the utterance sounds as if it begins with the stop from A, then information in the imported burst is more important than that in the vowel.
- If the first condition is true, then the second formant of the vowel is removed to determine whether or not it is the site of important information
- If the second, or neither, condition is true, then the copied stop burst is manipulated to attempt to make the uttered syllable sound as if it begins with the stop consonant that used to occupy its position

With a target stop consonant in mind, the manipulations to the copied stop burst attempt to determine the features that are important for human classification of stops that reside in the stop bursts themselves. Any important information in the vowel can already be considered to be present, since the vowel originally followed the target stop. To achieve the target consonant, the utterance must be made to sound as if it begins with the target stop more than it does any other phoneme.

2.4.1. Burst Frequency and Vowel Second Formant Frequency

We found that perception was strongly linked to burst frequency.

In all 13 trials performed, the target consonant was achieved. In nine of these trials, we had to modify the BF to achieve the target. In seven cases, the utterance, after the copy step, sounded as if it began with the imported burst. In the other two cases, the utterances began with an unclassifiable sound.

Though the target was clearly achieved in these cases, the quality of the utterance was good only in one of the trials.

Burst frequency was modified in eight out of the nine cases by isolating the main peaks of the stop burst and shifting them upwards or downwards. In one case, the main peaks were deleted and the location of secondary peaks determined the location of the burst.

2.4.2. Formant Transitions

Of the 13 trials, four resulted in the target consonant being immediately achieved after the stop consonant transfer step. In these cases, the information for the target consonant was primarily encoded in the formant transitions of the following vowel. Interestingly, in two of the four cases, when the second formant was removed, the target stop was still perceived, indicating that something other than the second formant contained important information. In one, removing the second formant of the vowel resulted in the perceived consonant to be that encoded by the burst, and in the final case, removing the second formant created an unclassifiable sound.

Also, in a separate experiment, we successfully replaced the burst of the stop with random noise without affecting stop perception. This is another instance where clear formant motion can override information in the stop burst.

2.4.3. Maximized Normalized Spectral Slope

This feature was found to be relatively unimportant to human perception. Though previous researchers found that the amplitude of the stop burst relative to the vowel could influence the perception of labials, in none of the 80 samples analyzed did this parameter affect perception meaningfully.

MNSS experiments were conducted mainly by modifying burst amplitude. In order for MNSS to meaningfully distinguish between labials and non-labials, it must be demonstrated that 1) non-labials have large MNSS values and labials have smaller ones, and that 2) a shift in perception can be induced by decreasing non-labial amplitudes so that their MNSS values become comparable to labial ones, or vice-versa. In the observed spectrograms, frequently not even the first condition was achieved. Also, in no case did shifting MNSS cause a change in perception from a labial to a non-labial.

2.4.4. Voicing

In cepstral analysis, voicing is a function of the source and not the spectrogram. Features related to voicing, such as prevoicing and VOT, are correlated statistically to voicing but do not themselves cause it.

Source files are generated by the speech synthesis program from recorded speech. The important information in these source files is when there is voicing and when there is noise. Often, the synthesis program incorrectly identifies when there is voicing, so speech synthesized using these source files can be excessively noisy. We solved this problem by synthesizing speech using a constantly voiced monotone source. This has the additional benefit of being uniform across all spectrograms.

Despite being constantly voiced, this source does not inhibit human distinction between voiced and unvoiced cognates of stops, or any other phonemes, for that matter. We therefore conclude that voicing/unvoicing information is unimportant for human perception of speech.

In none of the 13 trials was the addition of prevoicing or an alteration to VOT necessary to achieve the target stop. In one case, changing VOT lead to a higher-quality target consonant.

Trials conducted separately suggest that VOT can sometimes cause shifts in perception between voiced an unvoiced cognates. Out of ten cases this was true in one.

2.4.5. Other Tests

The other tests run had more to do with speech synthesis quality than phoneme recognition. We experimented with two methods of spectrogram compression: reducing the number of frequency channels, and changing a spectrogram from using a continuous intensity scale to a binary one.

Surprisingly, reducing the number of frequency channels from the default number of 256 in the spectrograms generated by the speech synthesis program to 16 channels dramatically improved speech quality. This can be explained by considering how the spectrograms were compressed. Blocks of 16 channels in the original spectrogram were averaged together to form single channels, canceling out much of the noise that they may have been carrying. Speech synthesis was then performed by first padding the missing channels with zeros such that the reduced spectrogram effectively contained 256 channels.

Other 16-channel spectrograms were generated by applying linearly spaced, notched filter banks to the 256-channel spectrograms. Speech generated from them was virtually identical to those created by the averaging technique.

Speech quality was also tested on channel sizes other than 16. A ranking of their relative ease of comprehension, from best to worst, is as follows: 16, 32, 64, 128, 256, 8, and 4. Smaller spectrograms contained less noise but also less information, causing a sudden loss of quality for spectrograms smaller than 16 channels. The four-channel spectrogram produced speech that was unintelligible to anyone who did not already know what the spectrogram was supposedly saying.

We explored another method of compressing spectrograms in which intensity values beyond a certain threshold were mapped to the maximum intensity value of the

spectrogram, and those below the threshold were given zero intensity. This binary representation of intensity led to comprehensible speech in spectrograms that had at least 16 channels, though this comprehension is possible only with context clues. Individual syllables were sufficiently marred that phonemes could no longer be clearly identified.

3. CHALLENGES

A set of features will define a phoneme if and only if they cause a human to hear it. We are the ultimate standard against which speech recognition systems are judged. Algorithms use parameters for speech recognition that themselves do not define the phoneme; rather, a host of unnamed features that happen to be statistically correlated to those features are what really affect human perception of speech. For instance, BF is related to where energy is concentrated in a stop consonant. However, knowing the formula for BF, a human editor could easily modify a single pixel of a spectrogram to completely change the value of BF, simply by picking a pixel occupying a low frequency position and making it exceptionally intense. Listening tests will not be affected at all by this change, since it affects only one pixel out of the thousands that comprise a spectrogram. A computer algorithm will be completely thrown off, however, because the correlation between BF and energy concentration has been removed.

Computers and humans use different sets of features to classify speech. The set that computers use is much smaller and is connected to the features that humans use through certain correlations. In evaluating the importance of various parameters that computer algorithms use as they relate to human perception, a researcher must be careful not to remove these connections, or he will arrive at the trivial result that those parameters have no importance at all.

Defining a valid change to a spectrogram is difficult. The result of the change must be something that could conceivably arise through an actual recorded speech signal. To preserve the realism of spectrograms, we restricted ourselves to moving and resizing already existing peaks. Creating peaks was to be avoided, if possible.

Of course, a seemingly realistic change may, in fact, never occur in normal speech signals.

Another challenge was recalcitrant data that refused to conform to expectations. Initially, we sought to demonstrate the unimportance of MNSS by showing that non-labials could have their peaks reduced sufficiently to drop their values of MNSS into the characteristic range of labials without impacting human recognition. This would have been especially meaningful because the algorithm developed by Ali et al. considered low MNSS values to be a sufficient condition for labial classification. By violating that condition, we would have demonstrated the unimportance of MNSS to human perception. Unfortunately, MNSS values of labials were not found to be significantly lower than those of non-labials, as previous researchers had found. In fact, many times they were greater. This seems to indicate that the speech samples we worked with were exceptions to a general rule, and that any conclusions extracted from them would therefore be questionable.

Perhaps most difficult of all, though, was objectively rating speech quality. Often, knowledge of the speech contents of a spectrogram dramatically boosted the perceived quality. Also, subjects evaluating the quality of a speech signal frequently gave it

different ratings between different instances of listening to it, despite the signal having not changed. These factors made a quantitative approach to research difficult.

4. RECOMMENDATIONS

We noted the following list of bugs and problems with the GUI. Future researchers may wish to fix them:

- The program does not properly load files into a paste buffer.
- Sometimes the program crashes during Matlab ActiveX automation. The error is difficult to reproduce.
- The ability to change the x- and y-scales of the spectrogram without having to reload it would be convenient
- Spectrograms typically take 15 to 20 seconds to load, much longer than most commercial programs take to load graphics files. The bottleneck in system performance comes from the use of the method Pset to render the pixels in the picture boxes. The setPixel and getPixel APIs are much faster and should be considered.
- When a region of the spectrogram selected to be copied to the clipboard, the box is drawn slightly off.
- In zoom mode, the box dimensions must be even numbers or the program crashes.
- When region is cut out from within a zoom box, if the brush color in zoom is different from that of the main screen, the zoom mode and main screen will disagree over the color of the excised region.
- If Matlab is manually closed, the GUI becomes confused and crashes when it attempts to send control signals to Matlab.
- When a brush is used to draw a pixel in zoom mode, the pixel is drawn slightly off, though this otherwise does not affect program performance.
- When a 256-channel spectrogram is zoomed in on, row one of the spectrogram is never present.
- The program crashes when attempting to load spectrograms beyond a certain size.
- Copy selections are not made correctly when the spectrogram is beyond a certain size.

Another recommendation is to follow up on the curious observation that removing the second formant of a vowel following the burst of a stop consonant still resulted in perception of the target consonant, despite the stop burst belonging to that of a wholly different stop consonant. Something other than the second formant in a vowel apparently contains stop consonant information.

5. CONCLUSION

As in previous experiments, BF and formant transitions were found to be the most important perceptual clues to human classification of stops. Important information in the vowel may reside in places other than the second formant.

Though voicing itself had no affect on perception, VOT was found, in a few cases, to shift perception between voiced and unvoiced cognates.

All results, however, were obtained using synthesized speech. These conclusions may not hold for natural speech.

Many software tools were developed during this project and many previous ones were enhanced, which should assist in further study of human acoustical perception.

6. ACKNOWLEDGEMENTS

I would like to thank my advisors, Ahmed Ali, Dr. Jan Van der Spiegel, and Dr. Paul Mueller, for their extensive help and support, which made this summer research project such a rewarding experience. I would also like to thank the National Science Foundation for making this project possible through their NSF-REU grant.

7. REFERENCES

1. V. Sue, Talking with Your Computer, *Sci. Am.*, http://www.sciam.com, Accessed 8/01/99.
2. A.M.A. Ali, J. Van der Spiegel, and P. Mueller, Acoustic-phonetic Features for the Automatic Classification of Stop Consonants, *IEEE Transaction on Speech and Audio Processing*, (in press, 1999).
3. A.M. Liberman et al., Perception of the Speech Code, *Psychological Review*, 74(6) (1967) 431-461.