

## PERFORMING SPEECH RECOGNITION RESEARCH WITH HYPERCARD

Chip Shepherd  
Lockheed Engineering and Sciences Company

*Research directed by Barbara Woolford, Flight  
Crew Support Division, NASA JSC.*

### PURPOSE

The purpose of this paper is to describe a HyperCard-based system for performing speech recognition research and to instruct Human Factors professionals on how to use the system to obtain detailed data about the user interface of a prototype speech recognition application.

### BACKGROUND

The development of the first Macintosh-based speech recognizer (Voice Navigator by Articulate Systems, Inc.) has enabled engineers at the NASA Johnson Space Center (JSC) to develop rapid-prototype speech recognition interfaces for space applications with HyperCard, an information management software package. A layout of the required hardware is presented in Figure 1.

Just like most speech recognizers, the Voice Navigator will allow a user to define a unique vocabulary, assign the computer action(s) to be associated with each vocabulary word, and record a personalized voice pattern for each word. The Navigator goes a step farther than most recognizers, however, because it allows access to the various recognition parameters generated within the machine while it is in an operating mode.

To obtain data on speech recognition parameters while the unit is being operated, engineers have taken advantage of the fact that HyperCard's command language may be expanded through the implementation of executable commands (XCMDs), which are user-created subroutines written in assembly or C languages.

In JSC's speech recognition research, three types of recognition data were required each time a command was spoken: the recognized word, the loudness of the speech, and the "confidence score," which is an internally calculated measure of how closely the spoken word compares to a pre-recorded sample; the score is expressed as a number from 1 to 100, with 100 being a perfect match. Using specifications from JSC contractor engineers, special HyperCard XCMDs were developed by Articulate Systems to obtain speech recognition data and assist in JSC's research applications. By integrating the XCMDs into experiments, project engineers have learned how to use the XCMDs to perform quantitative speech recognition research.

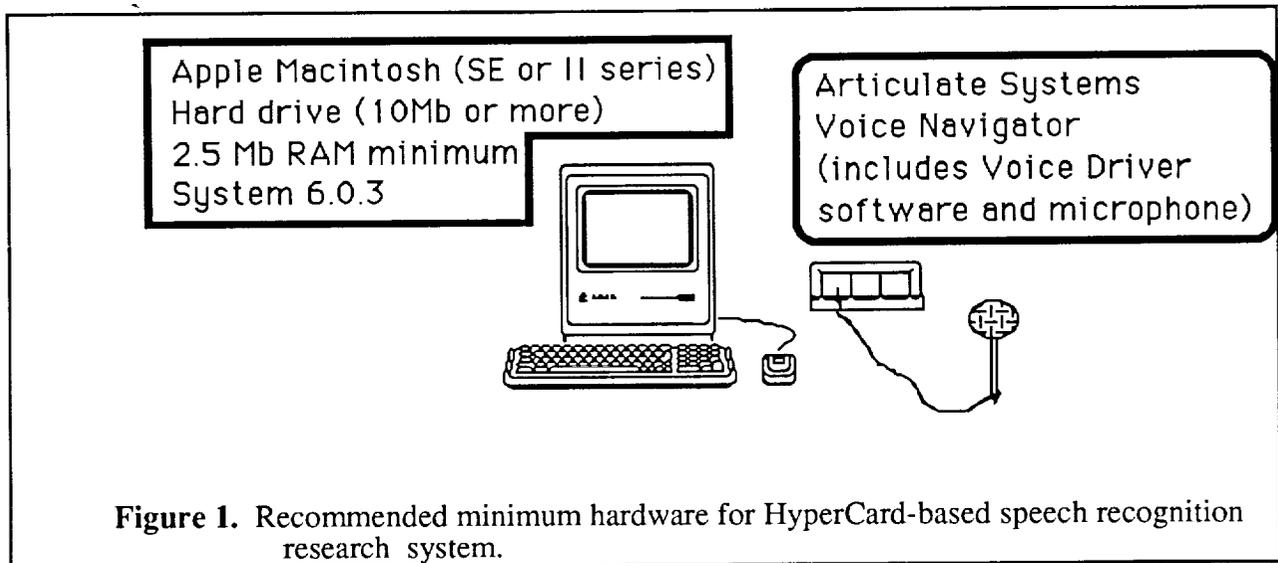
### XCMD DESCRIPTIONS

Articulate Systems included twelve XCMDs in a HyperCard stack called VoiceTalk, created to assist with the completion of the speech recognition research. Seven of the twelve XCMDs were designed to control necessary file management tasks. The other five have been applied in conducting research and are described below.

"Vocabulary" (name of vocabulary file) - Writes the active vocabulary list into the local HyperCard variable 'it.' The subject and/or researcher may be informed at all times which vocabulary set the recognizer is listening for.

"Collect" (vocabulary word) - Collects new voice samples of the specified vocabulary word. Thus, the researcher may record new voice samples just before, or even during, an experiment session.

"Macro" (vocabulary word) - This feature returns the command, or macro, associated with the specified vocabulary word. The macro is



usually a string of text, but can also be a mouse click or a series of HyperTalk commands, complete with returns, tabs, and linefeeds.

"Listen" - Prepares the recognizer to accept speech input.

"Recognize" - When an utterance is detected, this command returns the name of the recognized word. In addition, several associated speech recognition parameters are stored in reserved variables. The most valuable of these associated parameters is the confidence score, which is stored in the reserved variable "Confidence." The loudness of the speech is reported in decibels in the reserved variable "Amplitude."

### **GETTING THE SYSTEM TO WORK FOR YOU**

The present Voice Navigator, which is commercially available, does not include the HyperCard XCMDs as part of its standard package. This is because the XCMDs were created while the Voice Navigator was a prototype unit. However, the XCMDs may be easily obtained from Articulate Systems and will operate with available hardware after a few minor software adjustments are made by the user.

To utilize the XCMDs in HyperCard applications, the researcher should follow the procedures outlined below. These procedures will work with Voice Navigator System 1.0.1:

(1) Obtain a Voice Navigator from Articulate Systems, specifically requesting VoiceTalk.

(2) After you have connected the Voice Navigator and installed its software system files into the Macintosh as described in the Voice Navigator User's Manual, use the "Duplicate" command to make copies of the "Voice Control" and "Voice Driver" files that reside in the System folder. You should now have two new files named "Copy of Voice Control" and "Copy of Voice Driver."

(3) Change the name of the file "Copy of Voice Control" to "Dragon" and "Copy of Voice Driver" to "Dragon.LOD." You must do this to get the XCMDs to operate because, as stated before, the XCMDs were created when the Voice Navigator was in the prototype stage and, at that stage, the driver files were named "Dragon" and "Dragon.LOD." The Macintosh must be rebooted before the new files can be accessed, so you may as well do so before continuing to the next step.

(4) Install the HyperCard XCMDs onto whichever HyperCard stack(s) you will be using for speech recognition applications. You

may accomplish this with either the ResEdit program or the "install" feature provided with the VoiceTalk stack.

(5) Now that the XCMDs are installed, you may include them into any HyperTalk script on the stack. Examples of how to integrate the commands into HyperTalk programs are included with the VoiceTalk package.

Articulate Systems reports that improved versions of the HyperCard stacks will be available in the near future, but the procedures just described should enable presently available components to function.

### DISCUSSION

Using the XCMDs created by Articulate Systems, engineers at JSC have been able to extract important information about the speech recognition application unobtrusively in real time as the user operates the application. Information about the recognized word, its confidence score, the loudness of the speech, and the elapsed time may be recorded in an invisible background data field that is stored and analyzed after the user has completed the session. With this system, the collection of the speech parameter data has not affected the user interface and has not added noticeable time delays to the execution of the interface programming.

### EXAMPLE APPLICATION

The described system has been used extensively at JSC in examining speech as a means of controlling a computer display in the extravehicular environment. Future space suits may be equipped with a Helmet-Mounted Display (HMD), capable of providing substantial amounts of text, graphics, and video information. Controlled by spoken commands, this system would provide hands-free access to information for the suited astronaut.

A test was conducted with a prototype HMD system at NASA-JSC by having subjects use the HMD-based information system to construct an electronic circuit (Shepherd,

1989). The speech recognizer was active at all times and four types of data were collected each time a word was recognized: the word, the confidence score, the amplitude, and the elapsed time for the session (see Figure 2). A segment of the HyperTalk script used to perform these functions is included at the end of this paper (see EXAMPLE STACK SCRIPT).

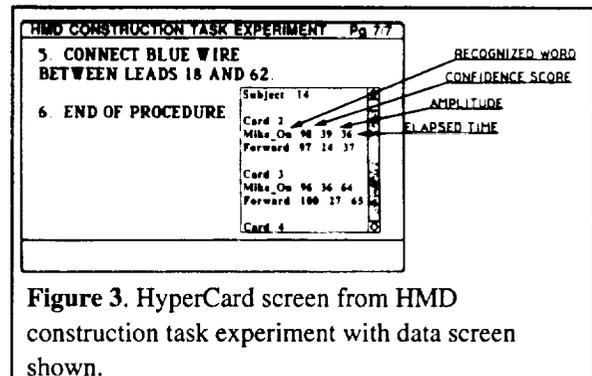


Figure 3. HyperCard screen from HMD construction task experiment with data screen shown.

A list of the recognized words was correlated with the elapsed times and compared to the videotape of the experiment, enabling researchers to easily classify each entry as a correct recognition or an error. Upon further analysis of the errors, it was found that the most common type of error occurred when the recognizer had inexplicably registered a command from the subject's conversational speech.

The confidence scores of the errors were analyzed and compared to those of the correct recognitions. The patterns were very different. While almost all of the correct recognitions had scores above 70, the errors were scattered throughout the range of scores, indicating that the errors had occurred randomly.

The amplitudes of the errors were analyzed and compared to those of the correct recognitions to see if the errors were said more loudly or more quietly than commands. No significant differences were found, which indicated that the errors were said just as loudly as the commands.

It should be noted that the XCMDs have not only been useful in collecting the data but are also helping to improve the interface itself. Because of the difference in confidence score patterns between correct recognitions and errors, system performance has been increased by setting a confidence score threshold in the system (i.e., each time a word is recognized, the corresponding confidence score, which is generated by the XCMD, is checked and, if it is not 70 or above, the command is ignored). With this threshold in place, many errors are screened out while almost no correct recognitions are affected. A similar threshold for the amplitude could have been put into place if a difference in the patterns had been found.

The described system has been beneficial in studying speech control for space applications, but it can also be employed in evaluating prototype interfaces in any of the leading speech recognition fields, including medicine, defense, products for the handicapped, and consumer systems.

### EXAMPLE STACK SCRIPT

*Explanation:* The first script basically instructs the recognizer to keep listening until it hears a vocabulary word. Once a vocabulary word registers, a second script would be triggered. The script for "Mike\_On" is provided as a sample. This script "activated" the microphone to accept a page forward/backward command. However, if the confidence score was not 30 or more, the script instructs the recognizer to ignore the command it just heard and to start listening for another.

```
On OpenCard
  global RecognizedWord, StartTime
  put the seconds into StartTime
  put empty into RecognizedWord
  listen
  repeat until the mouseclick
    put recognize() into RecognizedWord
    if RecognizedWord is not empty then
      get macro(RecognizedWord)
    do it
  end if
end repeat
end OpenCard
```

```
on Mike_On
  global RecognizedWord, StartTime,
  Confidence
  global Amplitude
  if Confidence > 30 then
    put RecognizedWord &&
  Confidence&&Amplitude→
    && ((the seconds) - StartTime) & return→
    after bkgnd field "Recording"
  else
    put 3 into dsd_state :(i.e., ignore the
  command)
  end if
end Mike_On
```

### ACKNOWLEDGEMENTS

The author would like to thank Barbara Woolford and José Marmolejo of NASA-JSC and Tim Morgan of Articulate Systems for their support in this research.

### References

1. Articulate Systems, "Voice Navigator User's Manual," 1989.
2. Cohen, A. D. "Rapid Prototype Development with HyperCard," Workshop at the 33rd Annual Meeting of the Human Factors Society, October 1989.
3. Shepherd, C. K., Jr. "Human Factors Investigation of a Speech-Controlled Bench-Model Extra-vehicular Mobility Unit Information System in a Construction Task," NASA document no. JSC-23632, Lockheed Engineering and Sciences Company document no. LESC-26799, May 1989.
4. Shepherd, C. K., Jr. "The Helmet-Mounted Display as a Tool to Increase Productivity During Space Station Extravehicular Activity," Proceedings of the 32nd Annual Meeting of the Human Factors Society, October 1988.
5. Weimer, J. "HyperCard for Human Factors Applications," Workshop at the 32nd Annual Meeting of the Human Factors Society, October 1988.