EMBRY-RIDDLE
Aeronautical University
SCHOLARLY COMMONS

Publications

1986

The Development of Speech Research Tools on MIT's Lisp Machine-based Workstations

D. Scott Cyphers
Massachusetts Institute of Technology

Robert H. Kassel
Massachusetts Institute of Technology

David H. Kaufman
Massachusetts Institute of Technology

Hong C. Leung
Massachusetts Institute of Technology

Mark A. Randolph
Massachusetts Institute of Technology

See next page for additional authors

Follow this and additional works at: https://commons.erau.edu/publication

 Part of the Computer and Systems Architecture Commons, and the Phonetics and Phonology Commons

Scholarly Commons Citation

Cyphers, D. S., Kassel, R. H., Kaufman, D. H., Leung, H. C., Randolph, M. A., Seneff, S., Unverferth, J. E., Wilson, T., & Zue, V. W. (1986). The Development of Speech Research Tools on MIT's Lisp Machine-based Workstations. , (). Retrieved from https://commons.erau.edu/publication/146

This Conference Proceeding is brought to you for free and open access by Scholarly Commons. It has been accepted for inclusion in Publications by an authorized administrator of Scholarly Commons. For more information, please contact commons@erau.edu.

Authors

D. Scott Cyphers, Robert H. Kassel, David H. Kaufman, Hong C. Leung, Mark A. Randolph, Stephanie Seneff, John E. Unverferth III, Timothy Wilson, and Victor W. Zue

This conference proceeding is available at Scholarly Commons: https://commons.erau.edu/publication/146

THE DEVELOPMENT OF SPEECH RESEARCH TOOLS ON MIT'S LISP MACHINE-BASED WORKSTATIONS*

D. Scott Cyphers, Robert H. Kassel, David H. Kaufman,
Hong C. Leung, Mark A. Randolph, Stephanie Seneff,
John E. Unverferth, III, Timothy Wilson, and Victor W. Zue

Research Laboratory of Electronics, and
Department of Electrical Engineering and Computer Science
Massachusetts Institute of Technology
Cambridge, MA 02139

ABSTRACT

In recent years, a number of useful speech- and language-related research tools have been under development at MIT. These tools are aids for efficiently analyzing the acoustic characteristics of speech and the phonological properties of a language. They are playing a valuable role in our own research, as well as in research conducted elsewhere. This paper describes several of the systems being developed for use on our Lisp Machine workstations.

INTRODUCTION

In many areas of speech research, ranging from speech analysis to synthesis to recognition, researchers often follow a common set of analysis procedures. Specifically, there is frequently a need to:

- record and digitize utterances, and define and compute various attributes of the speech signal,
- display and perform interactive measurements of these attributes,
- obtain statistical descriptions of the interrelation between acoustic and phonetic events by examining a large speech database,
- investigate the phonological properties of the language at the symbolic level, using large lexicons and/or printed text, and
- interactively synthesize speech in order to study the relative merits of acoustic cues for phonetic contrasts.

The ability to perform these tasks with ease will greatly facilitate the gathering of information and the corresponding improvement of our speech knowledge. One of our ongoing activities at MIT is the development of a speech research tools to satisfy these needs. This facility has already proven so useful that the necessary hardware and software have been acquired by many laboratories around the world. This paper is intended to provide a progress report on the development of these tools. It begins with a discussion of hardware requirements and then focuses on software tools.

*This research was supported by DARPA under contract N00039-85-C-0290, monitored through Naval Electronics Systems Command.

HARDWARE REQUIREMENTS

The speech workstation that we are refining centers around a Symbolics 3600 series Lisp Machine with 4 Mbytes of main memory, a 474 Mbyte disk, a floating-point accelerator, and a FEP or Generic-Bus Unibus interface. The Lisp Machine's high-resolution graphics console and mouse provide extremely convenient user interfaces.

The Lisp Machine may be augmented with a Floating Point Systems FPS-100E or FPS-5100 array processor for speed-up in numerical computations, and with a Digital Sound Corporation DSC-200/240 A/D and D/A converter for data input/output. The workstation also has a shared Versatec V-80 electrostatic printer/plotter, and assorted audio equipment such as a microphone, a set of headphones, and a tape recorder. The Lisp Machine workstations are connected to one another and to central file servers via a packet-switched local area network.

SOFTWARE SYSTEMS

Several interactive speech research tools are under development on the Lisp Machine workstation. In this section we will describe four such systems: *Spire*, *Search*, *ALexiS*, and *Synth*.

Spire

Spire (Speech and Phonetics Interactive Research Environment) is a software package that enables users to digitize, transcribe, process, and examine speech signals. A variety of different computations can be performed on the signal, and the results can be conveniently displayed and measured.

Spire organizes an utterance as a collection of *attributes*. The attributes may be either symbolic (e.g., phonetic transcription) or numeric (e.g., RMS amplitude). Some of the attributes are one-dimensional (e.g., speech waveform), while others are multidimensional (e.g., a series of short-time spectra). *Spire* has knowledge of the properties and

parameters of the attributes. As a result it is convenient to redefine parameter values (such as LPC order) and to define an attribute that depends upon another attribute.

Displays in *Spire* are organized in the form of *layouts*. The recording and transcription layouts are provided by *Spire*, since these are almost always needed by users. Other frequently used layouts can also be predefined. Many of the commands in *Spire* are given with the hand-held mouse pointer. The mouse can be used to configure a layout, play a section of the utterance, edit waveforms, examine data values, alter display options, and perform other functions.

Spire has been designed with two general goals in mind. First, it is intended to provide an extremely interactive environment and a basic set of capabilities such that speech scientists, even those with little or no programming experience, are able to collect and analyse speech data. Second, *Spire* is designed for easy customizing: users can readily add new attributes to suit their research needs. Currently the core of *Spire* defines default computations for approximately 40 attributes of the speech signal, whereas a customized version can define any number of attributes. Some of the customized *Spire* systems in our research group have as many as 300 attributes. The remainder of this section gives some examples of the operation and capabilities of *Spire*. For a detailed description of *Spire*, see Cyphers [2].

Collecting Speech Speech can be sampled at any rate up to 75 kHz, with appropriate anti-aliasing filters selected by the user. Up to 60 seconds of speech (the maximum is controlled by a parameter) can be digitized at a time. Currently the speech samples are transferred to virtual memory. In the near future, speech will be transferred directly to disk so as to permit the digitization of much longer samples of speech. An automatic end-point detector attempts to locate each utterance. The user can listen to the located utterance, modify the endpoints, and accept the utterance into the database, all with several clicks of a mouse button.

Information about the talker, sampling rate, file name, and orthographic transcription can be changed easily with a click of the mouse. Alternatively, an agenda file can be set up to sequentially change these parameters automatically. This latter option is particularly useful for bulk data input when a list of the recorded utterances already exists on-line.

Signal Processing Since most signal processing is computationally expensive, *Spire* provides mechanisms for reducing unnecessary processing. One way this is done is by ensuring that nothing is computed until it is needed. Once something has been computed, *Spire* remembers the value for the duration of the user's session, and will reuse that value if it is needed again, unless the user specifically forces it to be recomputed. All of this happens in a way that is virtually transparent to the user. Whenever the amount of overhead involved in data transfer is justified, computations are done on the FPS. Most of the basic analysis procedures are also written to run (albeit more slowly) in Lisp when the machine is without an FPS.

Attributes may also be saved in parameter files and reloaded during future sessions. When there are multiple parameter files associated with a particular utterance, *Spire* reassociates the parameters with the appropriate utterance as they are loaded. Several attributes can be precomputed on a large database (which may take many hours or days), and retrieved later as they are needed.

Looking at Data The Lisp Machine provides a high-resolution monitor that is useful for displaying data. A *layout* is a collection of *displays* of information that occupy an entire screen. Associated with each display are a number of *overlays*, which may be graphs, scales, or other attributes, such as a phonetic transcription overlaid on a spectrogram.

Spire allows users to compose their own layouts for specific research needs. Figure 1 shows an example of such a layout. The figure displays the wide-band spectrogram of the utterance, the zero-crossing rate, the original waveform, the orthographic and phonetic transcriptions, the narrow-band spectrum, and the LPC spectrum. This layout illustrates some of the interactive features of *Spire*. First, the user has direct control over all relevant display parameters. Thus, for example, the zero-crossing rate is displayed on the same time scale as the wide-band spectrogram. Second, all displays are time-synchronized. Moving a cursor in one display causes the other displays to change accordingly. Third, displays can be overlaid, and the display parameters of the overlay can be changed as well. For example, the LPC spectrum, overlaid on the narrow-band

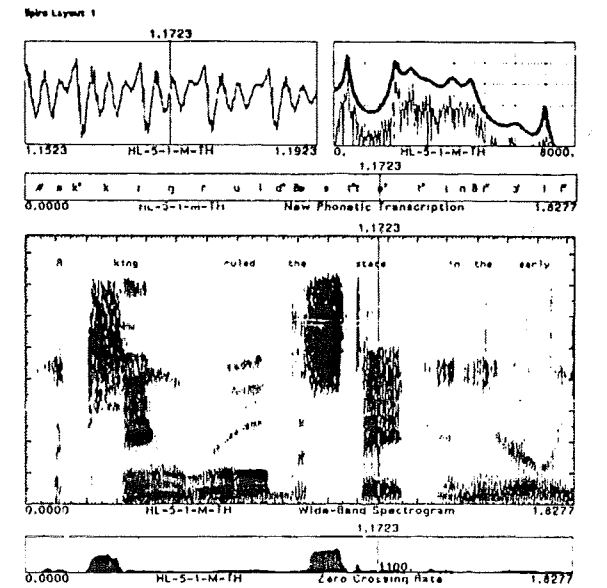

Figure 1: A *Spire* layout.

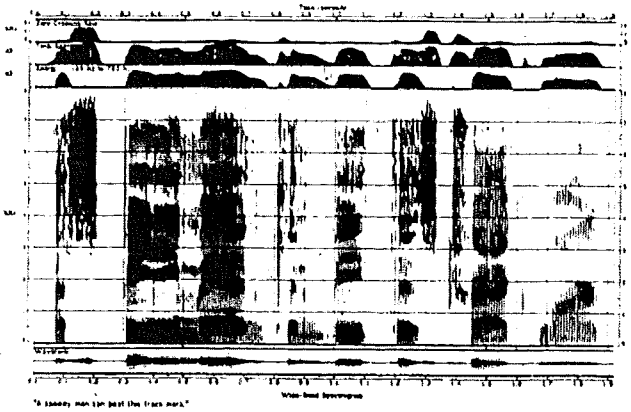

Figure 2: A high-quality spectrogram produced by *Spire*.

spectrum, is distinguished from the latter by a difference in line thickness.

Hard copies of displays can be obtained by a screen hardcopy command. In addition, a special higher-resolution hardcopy spectrogram format is available. Standard Lisp Machine hardcopy protocols are used, so the printer may be of any type. MIT has a Versatec V-80 printer/plotter with a resolution of 200 pixels per inch. Figure 2 is an example of a Versatec spectrogram.

Labeling Data *Spire* provides a convenient mechanism for users to enter an orthographic or phonetic transcription and time-align it with the speech waveform. The mouse is used to mark specific regions of the waveform and to associate each region with a label. An experienced acoustic phonetician can align a two-second utterance using *Spire* in about five minutes.

While manual time-alignment using *Spire* is quite efficient, it still requires the knowledge of a small group of experts. As a result, the amount of data that can be collected and aligned is greatly limited. In addition, phonetic alignment is often subjective, leading to inconsistencies among transcribers. The tedious nature of the task also tends to introduce human errors. We have recently extended *Spire*'s basic capabilities by developing a semi automatic system to perform the time alignment. The results of our preliminary evaluation are encouraging. For a description of the alignment system, see Leung and Zue [4] and Leung [3].

Search

In the search for relationships between the acoustic properties of speech and the underlying linguistic forms, speech researchers typically begin by examining a small number of utterances using tools such as *Spire*. Guided by their knowledge and intuition, they may propose an acoustic measurement that is potentially useful for identifying a particular phonetic event. Once such a hypothesis has been established, they must then apply the measure to a large body of data, either to validate assumptions derived from the limited data set, or simply to characterize the performance of the features. At this point, a relatively sophisticated battery of statistical techniques is most useful.

Search (Structured Environment for Assimilating the Regularities in speeCH) is an interactive tool designed for exploratory analysis of speech data. *Search* consists of a set of statistical tools that operate on data generated by *Spire*. This software allows the user to gather statistics on thousands of tokens taken from hundreds of utterances. It has extensive graphics capabilities for displaying the data in various forms, including histograms, scatter-plots, and a bar-like display that allows users to view univariate distributions of data as a function of categorical variables (e.g., speaker sex or phonetic environment). *Search* also features a set of extensible data structures that form a convenient workbench for the design, implementation, and testing of various statistical algorithms.

The basis of *Search* is a set of algorithms for automatically designing classification trees from a learning sample. The primary method is the *CART* (Classification And Regression Trees) algorithm [1], a supervised classification algorithm that generates a binary decision tree using a maximum-entropy reduction criterion. An alternative method is an *ISODATA* cluster analysis routine that features a k-Means clustering algorithm. Both of these algorithms are optimization techniques that attempt to organize speech knowledge automatically. They reflect our emerging philosophy that researchers should approach the speech analysis problem with as much intuition as they can develop, and then allow the data and statistics to fill in any gaps in knowledge.

Search can also be used simply as a program for partitioning the data in terms of a set of criteria, and then displaying the data in a variety of ways. Figure 3 compares the distribution of duration for voiced and voiceless fricatives. The mean, the standard deviation, and the sample size for each class are indicated next to the bar-like displays. The data confirm the fact that voiced fricatives are generally shorter than voiceless ones, although there is substantial overlap due to the phonetic environments

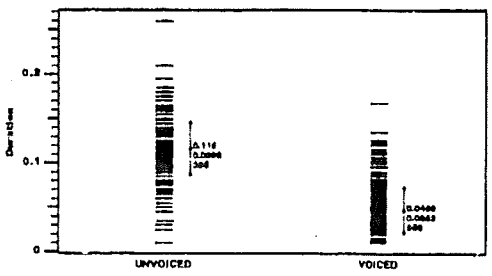

Figure 3: Duration distribution produced by *Search*.

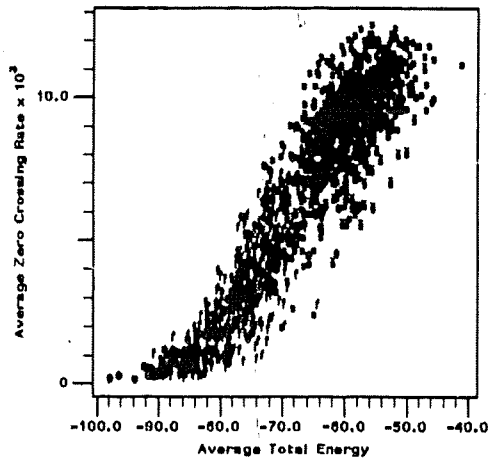

Figure 4: Zero-crossing rate/energy scatter plot produced by *Search*.

in which the fricatives appear. Figure 4 shows a scatter plot of zero-crossing rate versus total energy for the voiceless fricatives. We see that these two parameters are highly correlated; the strident fricatives /s/ and /š/ appear mostly in the upper right quadrant, and weak fricatives /f/ and /θ/ appear mostly in the lower left quadrant.

ALexiS

A language is limited not only by the inventory of basic sound units, but also by the frequency of usage and the allowable combinations of these sounds. With the availability of large and powerful computers, it is now possible to discover and quantify such distributional and sequential constraints using a large body of speech data.

ALexiS is an interactive system that provides many options for studying and displaying the constraints of a lexicon. *ALexiS* enables users to determine the frequency with which sound patterns occur, to study the phonotactic constraints imposed by the language, and to test the effectiveness of phonetic and phonological rules. In addition, users can define new operations and integrate them into the program.

ALexiS operates on a corpus consisting of a list of words or a set of sentences. Words in the lexicon are usually represented in terms of spelling, pronunciation (including syllable and stress markers), and other corpus-specific features such as the frequency count based on the Brown Corpus. User-specified constraints can be applied for each of these features, leading to a list of all words in the corpus matching the constraints.

Once the lexicon is specified, *ALexiS* can analyze the corpus in a number of ways. For example, users can generate a frequency distribution of words in the corpus in terms of the number of syllables, the stress patterns, or a particular sound pattern. As an example, Figure 5 shows

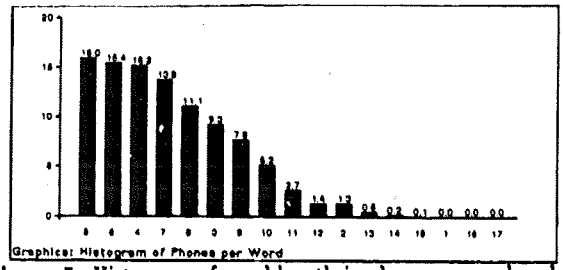

Figure 5: Histogram of word length in phonemes produced by *ALexiS*.

a histogram of the number of phonemes per word for the Merriam-Webster Pocket Dictionary.

Synth

Synth is the Klatt cascade/parallel speech synthesiser implemented as a *Spire* subsystem. Parameters that can be manipulated include fundamental frequency; formant frequencies, bandwidths, and amplitudes; amplitudes of voicing, frication, and aspiration; and frequencies and bandwidths of nasal and glottal poles and zeros.

The computations are done mainly in the FPS, and therefore the synthesis process is reasonably fast (approximately 15 times real time). The user interface is a special *Spire* layout, as shown in Figure 6. The user specifies time-value pairs for the parameter tracks, either with the mouse or from text. For example, to enter a track for F_2, the user selects F_2 from the menu on the left, and a display of the three formant tracks appears. The user can then enter a new track for F_2 or modify the existing track. Default parameter track layouts satisfy most users' requirements; however, users are also free to design their own special-purpose layouts. Once a set of tracks has been completed, the user selects "Synthesize" on the synthesiser menu, and a speech waveform is generated from the track data. This waveform is now available on the *Spire* utterance list and can be treated like any other *Spire* utterance. Thus, for example, a wide-band spectrogram and LPC spectrum of the synthetic utterance can be displayed, along with the same attributes for a similar natural utterance.

STATUS REPORT

Spire

Spire has been carefully evaluated and refined over the last several months. A number of improvements have been made to the system, including the following:

- It is now possible to save computed parameters separately from permanent parameters such as the waveform. This reduces the chance that "permanent data" will be accidentally destroyed (as has happened) and also reduces name conflicts among different users.
- Each Unibus interface now has better support. The FEP interface code has been modified to make it more stable in shared Unibus environments (such as a Lisp Machine or a Vax). The G-Bus interface

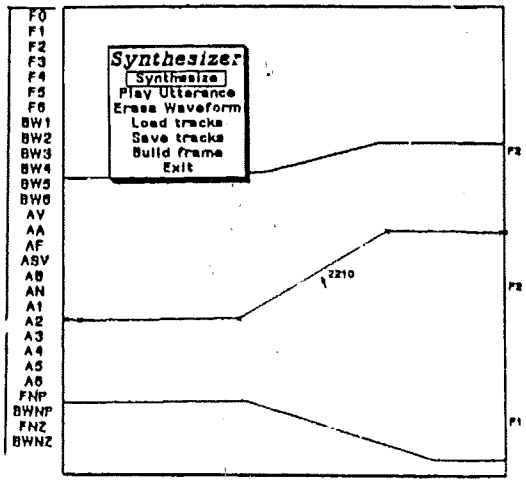

Figure 6: Typical Synthesizer Layout

seems to transfer data twice as quickly as the FEP interface.

- Utterances can now be played through the console when the Lisp Machine has audio capabilities, although the method used does not provide accurate reproduction.
- Lisp-based FFT and LPC are now available. With the availability of these programs, it is now possible to run *Spire* on a stand-alone Lisp machine with no array processor.
- A simple waveform editor has been added.
- The hand-alignment of transcriptions has been greatly simplified.
- The user interface is more consistent, making it easier for users to extend *Spire* by adding commands. The same command processor used by the Lisp Machine is now used by *Spire*.

Another task we have completed is evaluation of the FPA. Although for straight signal processing the FPS is still much faster than any other alternative, an FPA does cut times roughly in half. Table 1 gives some illustration of the relative timing for a Lisp Machine with different configurations. The energy computation is based on the computation of the energy in the range from 0 to 5,000 Hz, computed once every 5 ms. from speech sampled at 16,000 Hz. The spectrogram was computed from the speech waveform with a 383-Hz analysis rate. The numbers in Table 1 represent processing time (in seconds) per second of speech.

Finally, two documentation efforts for *Spire* are under way. A user's guide to *Spire*, aimed at the beginning user, has been written and is currently being reviewed. The final version should be available within the next month.

Table 1: Timing Comparison for Lisp Machine with Different Configurations

Configuration	Energy	Spectrogram
LM alone	27.6	63.0
LM + IFU	24.0	53.1
LM + FPA	11.8	31.5
LM + FPA + IFU	8.3	21.8
LM + FPS	1.6	3.5

A reference manual has also been prepared; a copy of this document, along with the source codes, is to be distributed with the current release of *Spire*.

Search

Search has undergone extensive redevelopment in recent months. Although the exterior aspects of the program (i.e., the graphical displays and the user interface) have remained more or less unchanged, we have redesigned a major portion of the system internals, including a complete overhaul of basic data structures. In instituting these changes, we have made every effort to keep major differences from affecting the user. For example, our basic data structure is the sample, which contains a collection of tokens. Although the sample has been reimplemented, most previous code written to extract data from samples and to manipulate samples should still work.

At present, the system is considered experimental and still somewhat fragile. We are now making *Search* available to other users in our own laboratory, so that we may get feedback and correct problems before beginning wider distribution. We expect that in two to three months, *Search* will be ready for release to other sites.

We are, however, only now starting to create documentation for the software. As this is expected to be a somewhat lengthy process, it is likely that early releases of *Search* will not have accompanying documentation.

ALexiS

Although *ALexiS* is still in the developmental stage, it has been used for a number of tasks. In addition to use with Japanese lexicons and in teaching, it provided the framework for the design and analysis aids used in the acoustic-phonetic database project. These applications helped test the design of *ALexiS* for extensibility and uncovered a number of weaknesses.

In the near future these design flaws will be corrected and the implementation will continue. By using the program in its uncompleted state, we hope to provide a better, more flexible system while avoiding untested design hypotheses.

Synth

The *Synth* system is nearing its final stage of development. It is currently being evaluated by a number of users at our laboratory and should be ready for release by mid-

to late spring. Documentation and a user's guide are being developed concurrently, but may not be available at the time of the initial release. However, user interaction with the system is essentially the same as user interaction with *Spire* and should require little new instruction.

While *Synth* is limited at present to systems whose hardware includes an FPS, future revisions should remove this constraint. A further change in *Synth* that is now under discussion is improvement of the system's modularity, i.e., allowing user-designed modules to replace those provided.

SUMMARY

This paper describes a set of research tools being developed in the Speech Group at MIT. Together these systems provide a unified environment that enables speech scientists to move from one task to another. For example, users can explore the statistical properties of a large body of data using *Search*, and then directly enter *Spire* to examine specific outlier tokens. An integral part of the system is the maintenance of a large database of more than three hours of digitized speech. Most of the utterances have been transcribed, and the transcriptions have been time-aligned with the corresponding waveform.

The development of these research tools is an ongoing process. Our goal is to create a research environment that is easy to use, thereby increasing the amount of data that speech scientists can examine and, as a consequence, extending our knowledge about speech. The Lisp Machine workstation and related software systems are playing an important role in advancing our understanding of the acoustic properties of speech sounds.

While these systems are still being actively improved, and not all the software is widely available, members of the Speech Subsystem of the DARPA Strategic Computing Program may contact us directly to obtain further information on acquiring the software.

ACKNOWLEDGMENT

We gratefully acknowledge the contribution of David W. Shipman, who designed and implemented the original *Spire* and provided many helpful ideas for developing the software tools.

REFERENCES

[1] Breiman, L., J. H. Friedman, R. A. Olshen, and C. J. Stone, *Classification and Regression Trees*, Belmont, CA: Wadsworth International Group, 1984.

[2] Cyphers, D. Scott, "Spire: A Speech Research Tool," S.M. thesis, Massachusetts Institute of Technology, May 1985.

[3] Leung, Hong C., "A Procedure for Automatic Alignment of Phonetic Transcriptions with Continuous Speech," S.M. thesis, Massachusetts Institute of Technology, January 1985

[4] Leung, H. C., and V. W. Zue, "A Procedure for Automatic Alignment of Phonetic Transcription with Continuous Speech," *Proc. ICASSP 84: IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1984, pp. 2.7.1-2.7.4.