Proceedings of the International Symposium on
Spatial Data Handling, 1984, Zurich, Switzerland,
Vol. 1

# COMPUTER-CONTROLLED INPUT AND OUTPUT OF SPEECH: UTILIZING THE ACOUSTIC CHANNEL FOR GEOMETRIC DATA ACQUISITION

Wolf-Dieter Rase
Bundesforschungsanstalt für
Landeskunde und Raumordnung
Postfach 20 01 30
D-5300 Bonn 2
Federal Republic of Germany

## Need for semi-automatic digitizing

During the last years much progress has been made in automated digitizing of lines and line networks on maps. Precise drum and flatbed scanning devices, laser scanners and line followers, in concert with sophisticated pattern recognition software, are used to convert the analog map into digital geometry data, in most cases as orthogonal vectors. The vector data can subsequently be processed and redrawn as a map.

The considerable costs for hardware and software, however, restrict the application of automatic digitization techniques to projects voluminous enough to recover the investments. For institutions with a medium or small volume of geometric data acquisition requirements the semi-automatic digitizer, that means a digitizer with a cursor manually guided by an operator, is still (or again) the appropriate way to get the job done. When I refer to digitizing in the following I mean this technique.

The advance of microelectronics and the general cost decrease for hardware opened up new application areas with a need for medium to high accuracy geometric data bases, for example in Computer Aided Design and Manufacturing (CAD/CAM), electronic circuit layout, and others. In many cases the digitizer is still a necessary tool for the input of geometric data, not only in mapping. The error-free data acquisition is, more than in cartographic applications, an important profitability factor.

To err is human, and every activity where humans are involved is error-prone. Geometric data acquisition is no exception from that rule-of-thumb. There is evidence that the number of errors is highly correlated to the amount of physiological and psychological stress on the operator. To minimize the error rate the system designer must reduce the stress, for example by elaborate operator guidance, plausibility control, and other measures (Jenks, 1981).



## Stress factors in manual digitizing

The human digitizer operator serves three different functions:

- locator: he moves the pen or cursor to a specific location or a series of locations (line), signalling to the computer, when the position is reached; the accuracy is a function of his combined tactile, motorical and visual abilities;

- discriminator: he selects a specific feature from an assortment of features on the map; he transmits a feature code, implicitly or explicitly, to the computer;

- structure builder: he recognizes geometric structures, e. g. the structure of a boundary network; he encodes the structure into a computer-readable format.

At a typical digitizing workstation the informations including the messages from the computer are exchanged through the tactile and visual channels. One or two hands move the cursor; the eye controls the correct location; the feature code is transmitted by pressing a button on a keypad, input of a textstring on a keyboard, or picking a menu item with the cursor.

The frequent switch from one activity to the other, for example from line following (in most cases with a magnifier) to menu picking to keyboard operation, causes fatigue and subsequently pain in different parts of the body.

- eye: lens - frequent focus adaption
  muscles - eyeball movement, stereo alignment
  iris - light intensity variation

- neck and arm muscles: moving the cursor, turning the head

- brain: pattern recognition is interrupted and disturbed; resumption of the recognition process costs time and effort.

Further stress factors such as a noisy or in some other way hostile environment contribute to fatigue and may increase the error rate.

The question is now how to interact with the computer and minimize the aforementioned problems. The answer: unload the tactile and visual channels by utilizing the normally underemployed acoustical channel for output of messages, operator guidance, input of commands and feature codes.

## Acoustic output

In the old days the off-line digitizers indicated the successful coordinate registration by the noise of the usually mechanical output device. When we installed a minicomputer-controlled digitizer more than a dozen years ago, no audible signal was issued by the device. The operator was always uncertain if and when the digitizer was ready to accept input, or if the coordinates had been recorded correctly by the program. Thus the bell of the terminal was used as receipt signal. During development of an interactive system for acquisition of boundary networks (Türke, 1976) it became obvious that it would be better to have more than one signal to indicate the program states, give receipts for inputs, and draw the operator's attention to error messages displayed in full on the terminal. A simple "sound box" was added, with eight different tones selectable by software.

## Fixed vocabulary devices

As usual, the program matured, more features were added, and we got experience how humans perceive and react on acoustical signals. To the surprise of the developer (a gifted amateur musician) most people cannot distinguish eight tones, let alone the ability to memorize their meanings. Just at that time the first solid-state, board level devices appeared on the market which were able to speak like a human and had an reasonable price tag. They provided a limited vocabulary of understanable speech with a slight, but tolerable computer accent. The only board with German vocabulary (from a talking calculator for blind persons) was interfaced to the minicomputer. The device was able to articulate the ten numerals, eleven functions normally found on a four-species calculator ("plus", "minus", "clear", "error", etc.), and two beeps. It was now possible to read to the operator the number of the areal unit to be digitized, and to output commands and error codes as spoken words.

Fixed vocabulary devices, early designs als well as new developments, consist basically of three functional units:

- a memory, in most cases a read-only memory (ROM), where the vocabulary is stored in digitally encoded form;
- a speech processor which fetches the digital code from the memory and converts it into analog output;
- an amplifier supposed to boost the output voltages of the processor to a level sufficient to feed a loudspeaker.

Several techniques to encode and store the spoken words and, vice-versa, their replay have been developped. In the more recent designs the three functions are integrated in one chip. These chips are usually cheaper than the necessary interface to the controlling computer. I will spare the technical details because the fixed vocabulary devices have only a limited significance in this context. Modern chips are able to fetch the speech code from random-access memory (RAM). Their vocabulary is not restricted to the codes cast into a small ROM. But the encoding of speech is a non-trivial process, which requires special development devices and software normally beyond the reach of a cartographer.



## Phoneme synthesizers

When we prepared a large digitization project it became evident that the fixed vocabulary device is not sufficient. We planned to digitize the boundaries of the communities in the Federal Republic of Germany, at this time around 15.000 polygons. The base maps bear the names of the communities. Because the speech output device provided only numerals (not numbers), it would have been necessary to write 8-digit or at least 5-digit area codes into the maps. Besides this error-prone procedure it is nearly impossible to memorize a 5-digit code spoken as a sequence of numerals.

Fortunately the advance of electronics brought us a new class of speech devices, the **phoneme synthesizers**. Fixed vocabulary devices emanate utterances derived from real speech stored in compressed form. The synthesizers, as their name says, use a different approach: they synthesize speech from smaller units called phonemes (1). In most implementations the phonemes are formed by simulating the human vocal tract in electronic circuits by a technique called formant synthesis (Ciarcia, 1981). Theoretically any text in any language can be spoken, but there are some practical restrictions.

**Number of phonemes**: phonemes are language-dependent, and the number of phonemes in just one language is usually larger than the number which can be implemented in the device, for economic as well as for electronic reasons. A widely used synthesizer chip, the Votrax SC-01A, provides 62 phonemes for the English language. This is good enough for understandable speech, but no good enough to satisfy a linguist. The use of English phonemes for a different language, let's say, German, is possible, but the unit speaks German with a heavy American accent. Newer chip contain more phonemes, also a few ones for languages other than English, but still not enough to qualify for a career at the "Züricher Schaupielhaus".

**Phoneme transition**: the phonemes cannot be simply stringed together. They change slightly, depending on the neighbouring phonemes. The transition from one phoneme to the other follows certain rules, and these rules must be implemented, at least partly, in the device.

**Pitch and inflection**: the "typical" computer voice speaks monotonously, that means with uniform pitch and loudness, and no inflection. To make the speech more "human" variations in pitch and loudness must be added. Correct accentuation (for words) and inflection (for sentences) are essential to produce intelligible speech. In most devices pitch (frequency) and loudness can be varied under program control. But accentuation and inflection require more than just hardware; this leads us to the next point.

## Text-to-speech translation

In our digitization project the speech device was supposed to read the names of the areal units to the operator. In contrast to spoken messages and commands where phoneme strings can be defined and iteratively improved in advance, the phonemization of the names must be done in real time. The text-to-speech algorithm analyzes the written text following a set of rules, and converts it to phoneme sequences, including the correct accentuation. In case of ambiguities and for natural-sounding inflection a more or less extended semantic analysis of the words and sentences is necessary. Not all ambiguities can be resolved: for ambitious applications, for example a reading machine for blind persons, an extensive exception word list must be maintained and searched during translation. Reading names is both easier and harder: on one hand context analysis is not necessary, on the other hand the name list is an exception word list for itself.

For the most used romanic languages - French, Italian, Spanish - the conversion from written text to spoken words is not too hard a problem. These languages are written "phonetically", that means, the number of exceptions from the rules is negligible. The germanic languages are more complicated. From the languages I am familiar with Dutch is on the easy, English on the hard side. German lies somewhere in the middle.

A widely applied algorithm for English text to speech conversion is the one developed at the US Naval Research Laboratory (NRL) by Elovitz et al (1976). Already a subset of the NRL rules suffice to obtain intelligible speech; the subset is small and fast enough to be implemented into a 8-bit microcomputer with 8 kbyte ROM (Ciarcia, 1982). For German similiar algorithms have been developped and implemented (Berry-Rogghe, 1976, Breuer et al, 1979, Müller, 1981).

## Available devices

The pace of progress in speech applications is so fast that it is hard to keep track with all the new developments. Many devices on the market will be superceded in a few month's time, therefore I restrict the overview to the ones which seem the most important. The already mentioned SC-01A chip manufactured by VOTRAX (62 phonemes, 4 inflection levels) found entrance in many devices too numerous to be mentioned here. The chip emerged from a series of board level products, some with phonemes for other languages than English. The chip costs now about 50 Dollars (if you only buy one), and reduced the cost for speech output by a factor of ten. Some devices using the SC-01A include a complete microcomputer to control the data transfer, to perform text-to-speech translation and other support functions (Ciarcia, 1982, VOTRAX, 1982).



A rather new chip, the SSI263 from Silicon Systems, is able to pronounce 256 phonemes including a few for German and French. It allows 4096 pitch variations and some other parameter settings to improve intelligibility. I had no opportunity yet to listen to this chip, but a reliable source reports that the SSI263 delivers the most natural sounding voice ever heard from a phoneme synthesizer (Ciarcia, 1984).

Of course the Japanese don't sleep, but they seem to have problems with English. Some speech output products have been announced by Japanese firms, but to my knowlegde failed to be a big success. Mainframe and minicomputer suppliers also started to offer speech output products, for example the "DECtalker" for DEC PDP-11 and VAX computers. The sales figures will show whether the improved speech quality justifies the considerably higher price in comparison to the aforementioned chip-based products.

## Speech input

Our own case history of man-computer interaction with speech ends here with the phoneme synthesizers. We digitize nearly exclusively line networks, mostly boundaries, and only occasionally transportation networks. The code and command input is done on a small keypad integrated into the digitizer cursor. A voice recognition device would be no major improvement, besides the fact that speech input is still approximately ten times more expensive than output. But in all cases where frequent menu picking and/or keyboard input must be done the speech input should be taken into consideration as a noteworthy alternative.

To make it clear from the beginning: we are far from real speech input or even automatic speech recognition. What has been done until now is a more or less sophisticated sound pattern recognition. A sound pattern issued by a human speaker is compared against a table of stored patterns. If a defined degree of similarity is detected a "true" flag and a code number assigned to the recognized pattern is transmitted to the program. The manufacturers of the more expensive products claim a recognition rate of 99%, that means, one out of a hundred trials to input a word fails. The hit rate decreases in noisy environments, in case of similiar words in the vocabulary (3), or poor articulation.

Although a few speaker-independent devices with very small vocabulary have been built the class of devices we discuss here are speaker-dependent. The reference pattern (template) must be established in a "training session" where the speaker has to input the complete vocabulary. Repeated input increases the hit rate because the averaging of the template improves the chances for a successful recognition. Most devices allow uploading of the templates to the computer, and downloading to the device at a later time. Thus the training for a specific speaker may be performed only once, and not every time the speech input is used. The up- and downloading feature can also extend the vocabulary beyond the limits of the device.

The earlier devices had problems in case the speaker's voice changed slightly, when he caught a cold, for example. Improved designs can handle that, and are even able to achieve a limited speaker independence. Nevertheless the devices rely on a fixed vocabulary which must be spoken at least once. An input device with the versatility of the phoneme synthesizer is still far from realization, although much high-powered research is underway due to the commercial implications of continuous-speech recognition, e. g. for voice-actuated typewriters (White, 1984).

Basically the pattern recognition process comprises two steps. First the sound pattern is converted into digital representation. Most devices perform bandpass filtering (2) to obtain a more detailed description of the pattern. In the second step the digital model of the input is compared with the reference pattern. The computations can be done either in the host computer or in the device by a specialized processor. To achieve acceptable response time the latter is preferred. Several methods of pattern correlation have been developed and implemented whereof template matching and dynamic programming gained commercial significance (Poulton, 1983). Again I will not plunge too deep into the details.

The market for speech input products has not yet matured to the same level as with the output devices. The voice recognition chip or chip set as a presupposition for inexpensive devices is "coming very soon now" for several years. The available board level products are intended rather for industrial than for consumer use. Very appealing is an add-on board from Interstate Electronics which converts a DEC VT100 terminal into a speech input device. It is to be hoped that the cost reduction by large-scale integration will act as an incentive for the development of more end-user oriented devices, and the subsequent opening of new application fields.

## Human problems with speech interaction

Computer-controlled input and output of speech has much greater impact on the acceptance by the users than a keyboard: the ability to issue and accept spoken information is considered until now as a human privilege. On the other hand it is always fascinating to me that during demonstrations people not familiar with computers find it usually not astonishing that a machine can speak, whereas the computer buffs get excited.

Some people feel uncomfortable to be addressed vocally by a computer. A psychologist working on the subject told me a story from a different application field, but with the same psychological problem in the background. The cockpit of modern airplanes is equipped with numerous bells and whistles to warn the pilot in case of problems or an emergency. An airplane manufacturer replaced some of the bells, on a trial basis, by a speech output device. To the surprise of the engineers the pilots disliked the verbal outpout so much that they usually switched it off after a while. The manufacturer had no choice but to return to the bells. The highly-trained elite of pilots felt insulted by a machine which reported problems they pretended to have perceived a long time ago. This kind of inferiority complex can be



found with other applications as well, sometimes combined with the "big brother syndrome": the machine as a symbol for evil is watching = controlling you. Psychological treatment would certainly be an overreaction; a practical remedy is to keep the persons away from the talking and listening computer.

Problems of environment seem to be of minor importance in comparison. When more than one digitizer workstation is operated in one room the noise interference can be a problem. A headset would be a simple solution, but the bulky ones used with the early input devices were very uncomfortable. Especially female operators complained that the headset is ruining their hair. Lightweight headsets with integrated noise-cancelling microphones are now available to cure the problem. But keep in mind that technical complaints often reflect unconscious psychological problems: see above.

## Conclusion

Speech output devices have grown up from a technical toy to a routine peripheral available at reasonable costs. The speech recognition devices will follow the same track in the near future. The technical problems of using the acoustical channel are solved. Designers of digitizer workstations should use this technology to improve the interaction between man and computer. The utilization of the acoustical channel will reduce the error rate, and provide a better economic base for computer-assisted mapping in general.

## Notes

1. The term "phoneme" is not used here in the sense linguists understand it. Phonemes cannot be pronounced because they represent a theoretical construct. It has been suggested to use the term "phone" instead (Müller, 1980), but for the sake of compatibility with the existing technical literature, and to avoid misunderstandings I will stay with "phoneme".

2. The frequency spectrum is electronically divided into several bands with defined widths. Each band is digitized and recorded individually.

3. There are persistent rumours that is still rather hard to discriminate between the consonants "b" and "d", even if one dispenses with the real time constraint.

## References


**Berry-Rogghe, G. L. M., 1976**, Ein Programm zur automatischen Phonemisierung deutscher Texte. Abteilung LDV, Mannheim, Institut für Deutsche Sprache.

**Breuer, Brustkern, Thyssen, Willee, 1979**, "PHONTEXT - eine PHONOL-Anwendung zur Erzeugung synthetischer deutscher Sprache". Sprache und Datenverarbeitung 3(1979), 10-17

**Ciarcia, S., 1981**, "Build an unlimited-vocabulary speech synthesizer". BYTE, September 1981, 38-50

**Ciarcia, S., 1982**, "Build the Microvox text-to-speech synthesizer, Part 1: Hardware". BYTE, September 1982, 64-88. "Part 2: Software". BYTE, October 1982, 40-64

**Ciarcia, S., 1984**, "Build a third-generation phonetic speech synthesizer". BYTE, March 1984, 28-42

**Elovitz, H. S., Johnson, R. W., McHugh, A., Shore, J. E., 1976**, Automatic translation of English text to phonetics by means of letter to sound rules. United States Naval Research Laboratory Report 7948

**Jenks, G. F., 1981**, "Lines, computers, and human frailties". Annals of the Association of American Geographers, Vol. 71, No. 1, March 1981, 1-10

**Müller, B. S., 1981**, Regelgesteuerte Umsetzung von deutschen Texten in gesprochene Sprache für das Sprachausgabegerät VOTRAX. Bonn, Gesellschaft für Mathematik und Datenverarbeitung

**Poulton, A. S., 1983**, Microcomputer speech synthesis and recognition. Wilmslow, Cheshire, Sigma Technical Press

**Türke, K., 1976**, "A system for interactive acquisition and administration of geometric data for thematic map production". Computer Graphics, Vol. 10, No. 2, Summer 1976, 154-162

**VOTRAX, 1982**, Personal Speech System. Product Data. Votrax, Troy, Michigan

**White, G. M., 1984**, "Speech recognition: an idea whose time is coming". BYTE, January 1984, 213-222