

MICROPROCESSOR IMPLEMENTATION OF A LARGE VOCABULARY SPEECH RECOGNIZER AND PHONETIC TYPEWRITER FOR FINNISH AND JAPANESE

Teuvo Kohonen*, Kari Torkkola*, Makoto Shozakai*, Jari Kangas*, Olli Ventä*

ABSTRACT

A flexible and inexpensive real-time speech recognition system is described. It operates in the following modes: recognition of isolated words from a large vocabulary, and orthographic transcription of (eventually continuous) speech. The main parts are the acoustic processor module that transcribes speech into phonemes, a large-vocabulary lexical-access module that recognizes isolated words on the basis of these transcriptions, and a character-string processor module that produces orthographically edited text for Finnish and romanized Japanese from the erroneous transcriptions within unlimited vocabulary.

INTRODUCTION

One of the most common approaches in speech recognition is to consider words as integral acoustic patterns, which are then directly compared with reference patterns. This cannot be a very ambitious goal for speech understanding. Also, when the size of the vocabulary increases, the comparison computations become heavy. It seems to be more advantageous to divide the words into smaller acoustic subunits, e.g., phonemes, and compare strings of their symbolic representations.

Most of the phonemes of Finnish and Japanese are reasonably well distinguishable by stationary spectral properties. Furthermore, the orthographies of Finnish and romanized Japanese are almost completely phonemic. These two facts supported the selection of phonemes as the basic phonological units for our design.

The so called phonetic typewriters for English may be based on the recognition of words from a large vocabulary. The plural and genitive forms, etc., can be regarded as separate words. This is not possible in many other languages, such as Finnish and Japanese, which have numerous inflexions and endings. In the latter case, every phoneme must be recognized separately. Major difficulties are caused by the coarticulation effects. Due to them, it is not sufficient to improve the recognition of individual phonemes alone; the syntactic rules of speech must be taken into account, too.

The above-mentioned facts have motivated us to choose the following two-stage system configuration (fig. 1). First, the acoustic processor module produces a phonemic transcription of uttered speech. Possible errors in the transcriptions (due to coarticulation effects and variations in the speech) are corrected or otherwise taken into account at the postprocessing stage. There are two alternative postprocessing modes: 1) In isolated-word recognition mode, the closest word from a fixed vocabulary is searched using a fast prescreening method combined with a more accurate probabilistic analysis (refs 4,8,9). 2) In phonetic typewriter mode, the raw phonemic transcriptions are corrected by a grammatical method called *Dynamically Expanding Context* (ref 2). This method automatically derives a large number of production rules from samples of natural speech data, which are then used to transform the erroneous strings into orthographically edited text.

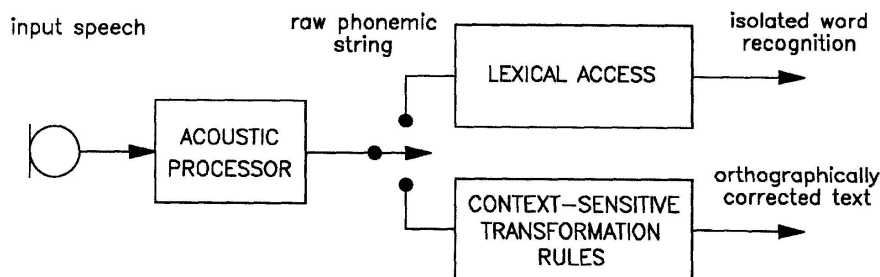


Fig. 1. The system configuration

*Helsinki University of Technology, Faculty of Information Technology, Laboratory of Computer and Information Science, Rakentajanaukio 2C, SF-02150 Espoo, Finland.

ACOUSTIC PROCESSING

Acoustic processing is divided in two separate substages: preprocessing and phonemic recognition. At preprocessing, a 15-component approximation of the short-time power spectrum of the speech waveform is produced every 9.83 ms. The computation is based on the 256-point FFT, and suitable combination of the channels. To improve the analysis of transient sounds, pattern vectors formed of concatenating two 15-component spectral vectors can be used, too.

At phonemic recognition, all spectral vectors are first labeled to make up a quasiphoneme string. We use the so called *Phonotopic Map method* (ref 1) for labeling. The map is a two-dimensional array of processing units each acting as an optimally matched filter to a due phoneme and its variations. Each unit (i.e., its spectral template) is automatically tuned to a class of acoustic spectra, and the units have a distribution which corresponds to the clustering of the phonemic samples. The map is constructed in a *self-organizing process* (ref 6), and such a map is believed to resemble the sensory maps of the biological brain (cf. ref 6). The map is further fine-tuned by a supervised learning scheme named *Learning Vector Quantization* (ref 7) to ultimately improve the recognition accuracy.

The quasiphoneme labeling is accomplished by selecting the best-matching unit in the map. For simplicity, Euclidian distances are used. Notice that this does not mean least-square nearest-neighbour classification, because the reference vectors do not represent original samples; their effective values are optimized carefully.

Segmentation of quasiphoneme strings into phonemic transcriptions is straightforward: if m out of n contiguous quasiphonemes are same, a phoneme is decided to be detected. The parameters m and n vary for different phonemes.

Generally, the spectral properties of consonants behave more dynamically than those of vowels, and also stationary intervals of many consonants are short. On the other hand, the stationary parameters of certain consonants are very similar. It seems advantageous to pay attention to the plosive burst and transient region between a consonant and the subsequent vowel to identify the consonant. In our system transient information is coded in additional phonotopic maps (called *transient maps*) and they are trained by using transient spectral samples alone, to describe the dynamic features with higher resolution.

In the Japanese version, four transient maps have been constructed to distinguish the following cases:

- 1) voiceless stops /k, p, t/ and glottal stop (vowel at the beginning of utterance)
- 2) voiceless stops /k, p, t/ without comparison to the glottal stop
- 3) voiced stops /b, d, g/
- 4) nasals /m, n, ŋ/

Only one transient map has been adopted in the Finnish version, accomplishing the distinction between /k, p, t/ and the glottal stop. (/b/ and /g/ do not exist in Finnish.)

In spectral labeling, the stationary map is used by default to produce a tentative quasiphoneme sequence. Detecting a CV combination in the resulting phonemic transcription activates the corresponding transient map, and the consonant segment is labeled more accurately.

The distinction accuracy of the cases in the due transient maps is as high as about 90 per cent, on the average. Application of the transient maps improves the total recognition accuracy by six to seven per cent (cf. performance below).

WORD RECOGNITION

The word recognition stage selects that word in a fixed vocabulary which is closest to the produced phonemic transcription with respect to some distance measure.

The very quick searching of potential candidates is based on the *Redundant Hash Addressing* method (refs 4,8,9). In this scheme, the phonemic transcription is compared with the reference transcriptions of the stored words by features which are bigrams (or trigrams) of consecutive phonemes. The comparison is very fast due to hash addressing. Only such words which have some similar features need to be compared; the rest of the vocabulary can be ignored.

As stored reference transcriptions of the vocabulary, earlier sample outputs from the acoustic processor module are used. These transcriptions are not stored as such but their highly compressed representation is stored in a special hash table; at each hash address corresponding to a feature, there is then a pointer to the correct dictionary word. During recognition, all the features of the uttered word are converted into hash addresses, and the pointers to the dictionary words are read from the hash table. The majority of the pointers defines the recognized word candidate. If several good candidates have been picked, the final selection among them can be based on a statistically more accurate, but computationally much heavier *maximum a posteriori probability measure*. This measure needs to be evaluated only for a few candidates, however.

We have applied the Redundant Hash Addressing principle to continuous speech recognition, too (ref 5), but no microprocessor implementation exists yet for it.

CORRECTING PHONEMIC TRANSCRIPTIONS

The coarticulation effects cause certain systematic errors to appear in phonemic transcriptions. E.g., the Finnish word "*hauki*" (meaning pike) is almost always recognized as a phoneme string /*haouki*/. This suggests the use of context-sensitive transformation rules for correction.

We have developed a method called *Dynamically Expanding Context* (ref 2), which automatically derives a very large number (typically tens of thousands) of context-sensitive rules from samples of natural speech. The rules are found by aligning correct forms of words or phrases with the corresponding phonemic transcriptions. Context-independent rules are first formed by comparison. If a rule conflicts with an earlier stored rule, the context of both rules is expanded to redistinguish the segments. Expansion is made gradually, symbol by symbol, and only if necessary. In this way, the minimum sufficient context for each rule can be found.

In reality, the system of rules, and its encoding in the memory are made by a rather sophisticated program, and the rules correspond to a tree structure which is traced during the correction process.

In the case that during the correction process the search became unsuccessful, we would have encountered a strange context. In this case we have to take a narrower context and apply majority voting over the conflicting cases. The same is due if the highest contextual level provided in the program is exceeded but the conflicts are not yet resolved.

The set of rules is able to correct strings with the same morphology of errors as the training set. For example, from 5000 pronounced words, about 11000 production rules can be derived. These rules are able to correct up to 70 per cent of the errors from the raw phonemic transcription strings. It seems that the higher the quality of the phonemic transcription strings is, the larger percentage of the remaining errors is corrected by the method.

MICROPROCESSOR IMPLEMENTATION

The described system is implemented on an IBM PC/AT and two processor boards (ref 3). One board is dedicated to preprocessing, and the other to phonemic recognition (fig 2). Currently the postprocessing methods are realized on the personal computer but there is extra capacity on the processor boards for postprocessing, too.

The preprocessor is composed of a switched-capacitor 5.3 kHz low-pass filter, a 12-bit A/D-converter with sampling frequency of 13.02 kHz, a TI TMS32010 signal processor with 8 KB RAM and 512-byte PROM, and a parallel interface to the phonemic recognition board.

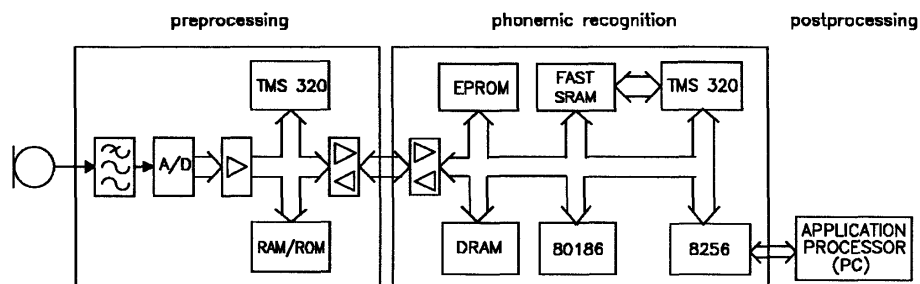


Fig. 2. The structure of the microprocessor implementation.

At every 128 waveform sampling instant (corresponding to spectral sampling every 9.83 ms), the following operations are performed: The digitized speech signal is windowed using a 256-point Hamming window; a 128-point power spectrum is computed by a 256-point FFT; the power spectrum is logarithmized and smoothed, and finally the 15 elements of the pattern vector are picked from the frequency range of 200 Hz - 5 kHz. An approximation of the rms-value of the speech signal is also computed, and this, as well as the 15-component spectrum vector, are transferred to the phonemic recognition board.

The main processor on the phonemic recognition board is an Intel 80186 with 512 KB of RAM. A TMS32010 with 96 KB of fast paged program RAM (384 KB in the Japanese version) acts as a simple slave array processor of the 80186 processor. The 80186 processor has also parallel interfaces both to the preprocessor board and to the

personal computer. The direct memory access technique is used in data transfers between the processors on the board and between the boards.

The map algorithm runs in the TMS32010 of the phonemic recognition board. This processor finds the map unit matching best with the pattern vector given by the 80186 processor. The units of the map are represented in in-line program code of the TMS32010 by the so called immediate multiply instructions (MPYK) with double precision which in this case means 25 bits. The program code is generated automatically by the personal computer and then loaded into the RAM of TMS32010.

The 80186 processor also performs the segmentation based on the quasiphoneme sequence, and the detected phonemes are immediately transferred to the personal computer. Still, only a small fraction of the available computing capacity of the 80186 processor is utilized. Either of the above postprocessing methods could be programmed for the 80186 processor to build a complete stand-alone speech recognition unit.

The memory demands of the word recognition stage are, e.g., for a 1000 word vocabulary with 5 tokens for each word: 160 KB for the hash index table and 10 KB for the dictionary.

A sample set of 5000 words produces about 11000 rules when the Dynamically Expanding Context method is used. These require a memory space of about 300KB.

PERFORMANCE

The system is speaker-dependent. The phonemic maps are computed reasonably fast. Modification of a standard map to a new speaker requires only 100 word samples, and the computation for it takes 25 minutes on the IBM PC/AT. The modification is based on the Learning Vector Quantization algorithm (ref 7). A fast automatic segmentation program has also been developed to pick phoneme prototypes from word samples.

The combined segmentation and labeling accuracy of the acoustic processor for raw phonemes varies between 75% and 90%, depending on the speaker and the difficulty of the vocabulary. Word recognition accuracies using a vocabulary of 1000 words vary between 96% and 98%. The lower accuracy is obtained even with a vocabulary containing a large number of confusable words, i.e., differing only in a single phoneme. The accuracy of the orthographically corrected and edited text varies between 90% and 97%, referring to individual letters. This accuracy depends on the speaker and the amount and the content of speech encoded into rules.

The mean recognition time in isolated word recognition is about 250 ms. When producing orthographically corrected text, the mean correction time for a word is about 300 ms and the recognition process almost completely overlaps the pronunciation of the word. New rules can be easily added on-line, first pronouncing a word or a phrase, and then entering the correct written form via the keyboard.

All output on the personal computer screen is obtained in real time.

REFERENCES

1. T.Kohonen, K.Mäkisara, and T.Saramäki, "Phonotopic Maps - Insightful Representation of Phonological Features for Speech Recognition," *Proc. 7th ICPR*, Montreal, Canada, July 30- Aug. 2, 1984, pp.182-185
2. T.Kohonen, "Dynamically Expanding Context, with Application to the Correction of Symbol Strings in the Recognition of Continuous Speech," *Proc. 8th ICPR*, Paris, France, Oct. 27-31, 1986, pp. 1148-1151
3. K.Torkkola and H.Riittinen, "A Microprocessor-based Word Recognition System for Large Vocabularies," *Proc. IEEE 1986 ICASSP*, Tokyo, Japan, Apr. 7-11, 1986, pp. 333-336
4. T.Kohonen and E.Reuhkala, "A Very Fast Associative Method for the recognition and Correction of Misspelt Words, Based on Redundant Hash Addressing," *Proc. 4th IJCP*, Kyoto, Japan, Nov. 7-10, 1978, pp. 1006-1008
5. O.Ventä, "N-Gram Driven Search for Sentences in a Syntactic Network," *Proc. IEEE 1986 ICASSP*, Tokyo, Japan, Apr. 7-11, 1986, pp. 1145-1148
6. T.Kohonen, *Self-Organization and Associative Memory*. Berlin, Heidelberg, New York, Tokyo: Springer 1984
7. T.Kohonen, "Learning Vector Quantization for Pattern Recognition," Helsinki University of Technology, Report TKK-F-A601, Oct. 1986
8. E.Reuhkala, "Recognition of Strings of Discrete Symbols with Special Application to Isolated Word Recognition," *Acta Polytech. Scand. Math. and Computer Sci. Ser.*, Nr. 38, 1983 (doctoral dissertation, Helsinki University of Technology; available from University Microfilms).
9. T.Kohonen, *Content Addressable Memories*, 2nd Ed., Berlin, Heidelberg, New York, London, Paris, Tokyo: Springer 1987