SPEECH RECOGNITION CIRCUIT USING PARALLEL PROCESSORS

The present invention generally relates to a speech recognition circuit which uses parallel processors for processing the input speech data in parallel.

Conventional large vocabulary speech recognition can be divided into two processes: front end processing to generate processed speech parameters such as feature vectors, followed by a search process which attempts to find the most likely set of words spoken from a given vocabulary (lexicon).

The front end processing generally represents no problem for current processing systems. However, for large vocabulary, speaker independent speech recognition, it is the search process that presents the biggest challenge. An article by Deshmukh et al entitled "Hierarchical Search for Large-Vocabulary Conversational Speech Recognition" (IEEE Signal Processing Magazine, September 1999, pages 84 to 107), the content of which is hereby incorporated by reference, discusses the general concepts of large vocabulary speech recognition. As discussed in this paper, one algorithm for performing the search is the Viterbi algorithm. The Viterbi algorithm is a parallel or breadth first search through a transition network of states of Hidden Markov Models. An acoustic model for words in a lexicon are represented as states of Hidden Markov Models. These states represent phones or n phones in a phone model of the words. The search requires the evaluation of possible word matches. It is known that such a search is computationally intensive.

In order to speed up the processing performed during such a search in a speech recognition system, parallel processing has been explored. In an article by M K Ravishankar entitled "Parallel Implementation of Fast Beam Search for Speaker-Independent Continuous Speech Recognition" (Indian Institute of Science, Bangalor, India, July 16, 1993) a multi-threaded implementation of a fast beam search algorithm is disclosed. The multi-threading implementation requires a significant amount of communication and synchronization among threads. In an MSC project report by

R Dujari entitled "Parallel Viterbi Search Algorithm for Speech Recognition" (MIT, February 1992) the parallel processing of input speech parameters is disclosed in which a lexical network is split statically among processors.

It is an object of the present invention to provide an improved circuit which can perform parallel processing of speech parameters.

In accordance with a first embodiment of the present invention, a speech recognition circuit comprises an input port such as input buffer for receiving parameterized speech data such as feature vectors. A lexical memory arrangement is provided which contains lexicon data for word recognition. The lexical data comprises a plurality of lexical tree data structures representing a plurality of lexical trees. Each lexical tree data structure comprises a model of words having common prefix components and an initial component which is unique as an initial component for lexical trees. A plurality of lexical tree processors are connected in parallel to the input port and perform parallel lexical tree processing for word recognition by accessing the lexical data in the lexical memory arrangement. A results memory arrangement is connected to the lexical tree processors for storing processing results from the lexical tree processors and lexical tree identifiers to identify lexical trees to be processed by the lexical tree processors. A controller controls the lexical tree processors to process lexical trees identified in the results memory arrangement by performing parallel processing of a plurality of lexical tree data structures.

Thus in accordance with this embodiment of the present invention, the processing in order to perform word recognition is distributed across the processors by controlling the processors to perform processing on different lexical trees. The controller controls the processor by the processes to provide for efficient process management by distributing lexical processing to appropriate processors.

The lexical tree data structure can comprise a phone model of words, wherein the components comprise phones. For reduced storage, the lexical tree data structure can comprise a mono phone lexical tree. The mono phone lexical tree can be used to generate context dependent phone models dynamically. This enables the use of context dependent phone models for matching and hence increased accuracy whilst not

increasing memory requirements. Alternatively, the lexical tree data structure can comprise context dependent phone models.

The processing performed by each processor in one embodiment comprises the comparison of the speech parameters with the lexical data, e.g. phone models or data derived from the lexical data (e.g. dynamically generated context dependent phone models) to identify words as a word recognition event and to send information identifying the identified words to the results memory as the processing results. In this embodiment a language model processor arrangement can be provided for providing a language model output for modifying the processing results at a word recognition event by a lexical tree processor. The modification can either take place at each lexical tree processor, or at the language model processing arrangement.

In one embodiment each lexical tree processor determines an output score for words in the processing results at word recognition events. Thus in this embodiment the language model processing arrangement can modify the score using a score for a language model for n preceding words, where n is an integer.

In one embodiment the controller instructs a lexical tree processor to process a lexical tree by passing a lexical tree identifier for the lexical tree and history data for a recognition path associated with the lexical tree from the results memory. The history data preferably includes an accumulated score for the recognition path. This enables a score to be determined based on the score for the recognition path to accumulate a new score during recognition carried out using the lexical tree data structure. The scores can be output in the processing results to the results memory during the processing of the speech parameters so that the scores can be used for pruning.

In one embodiment of the present invention, each lexical tree processor operates on more than one lexical tree at the same time, e.g. two lexical trees represented by two different lexical tree data structures, or two lexical trees represented by the same data structure but displaced in time (which can be termed to instances of the same lexical tree).

At word recognition events, the controller determines new lexical tree identifiers for storing in the results memory for words identified in the results memory for respective word events. In order to reduce the processing, the controller can prune the new lexical tree identifiers to reduce the number of lexical trees which are required to be processed. This pruning can be achieved using context dependant n phones to reduce the number of possible next phones. The number can be further reduced by using a language model look ahead technique.

In one embodiment of the present invention, the lexical tree processors are arranged in groups or clusters. The lexical memory arrangement comprises a plurality of partial lexical memories. Each partial lexical memory is connected to one of the groups of lexical tree processors and contains part of the lexical data. Thus a group of lexical tree processors and a partial lexical memory form a cluster. Each lexical tree processor is operative to process the speech parameters using a partial lexical memory and the controller controls each lexical tree processor to process a lexical tree corresponding to partial lexical data in a corresponding partial lexical memory.

In another embodiment of the present invention the lexical memory arrangement comprises a plurality of partial lexical memories. Each partial lexical memory being connected to one of the lexical tree processors and containing part of the lexical data. Each lexical tree processor processes the speech parameters using a corresponding partial lexical memory and the controller is operative to control each lexical tree processor to process a lexical tree corresponding to partial lexical data in a corresponding partial lexical memory.

In one embodiment of the present invention the lexical memory arrangement stores the lexical tree data structures as Hidden Markov Models and the lexical tree processors are operative to perform the Viterbi search algorithm using each respective lexical tree data structure. Thus in this way, this embodiment of the present invention provides a parallel Viterbi lexical tree search process for speech recognition.

The first aspect of the present invention is a special purpose circuit built for performing the speech recognition search process in which there are a plurality of processors for performing parallel lexical tree processing on individual lexical tree processors.

In another aspect of the present invention a speech recognition circuit comprises an input port such as an input buffer for receiving parameterized speech data such as feature vectors. A plurality of lexical memories are provided which contain in combination complete lexical data for word recognition. Each lexical memory contains part of the complete lexical data. A plurality of processors are provided connected in parallel to the input port for processing the speech parameters in parallel. The processors are arranged in groups in which each group is connected to a corresponding lexical memory to form a cluster. A controller controls each processor to process the speech parameters using partial lexical data read from a corresponding lexical memory. The results of processing the speech parameters are output from the processors as recognition data.

Thus this aspect of the present invention provides a circuit in which speech recognition processing is performed in parallel by groups of processors operating in parallel in which each group accesses a common memory of lexical data. This aspect of the present invention provides the advantage of parallel processing of speech parameters and benefits from a limited segmentation of the lexical data. By providing a plurality of processors in a group with a common memory, flexibility in the processing is provided without being bandwidth limited by the interface to the memory that would occur if only a single memory were used for all processors. The arrangement is more flexible than the parallel processing arrangement in which each processor only has access to its own local memory and requires fewer memory interfaces (i.e. chip pins). Each processor within a group can access the same lexical data as any other processor in the group. The controller can thus control the parallel processing of input speech parameters in a more flexible manner. For example, it allows more than one processor to process input speech parameters using the same lexical data in a lexical memory. This is because the lexical data is segmented into domains which are accessible by multiple processors.

In a preferred embodiment this aspect of the present invention is used in combination with the first aspect of the present invention. In such an arrangement each processor performs lexical tree processing and the lexical data stored in each lexical memory

DOCKET ALARM

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS
Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS
Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS
Sync your system to PACER to automate legal marketing.