ABSTRACT

Current vocal command recognition systems suffer from a need for extensive back-end support, communication latency, and network reliability dependence. A disconnected system is superior for security and privacy yet have poor accuracy and versatility. To overcome the limitations of the disconnected system, our implementation utilizes a selection of speech recognition methods, all of which are non language model or speaker dependent. This achieves greater accuracy than any single algorithm may achieve in a given environment. The strengths of each extraction and match algorithm buoy the weaknesses of the others creating a more accurate and robust system. Each algorithm is processed in parallel, allowing greater levels of accuracy with little loss in speed. FPGA technology is utilized, as it is well suited to parallel processing with low latency and near real-time performance.

BACKGROUND THEORY

Linear Predictive Coding (LPC) is a signal analysis technique for extracting coefficients from speech that allow for linearly predicting future time samples and is based on the source-filter model of human speech. The glottis produces an impulse train which mixes with random white noise. This mixed signal then passes through a time varying filter, the vocal tract. This is the primary feature extraction technique used.

SPEECH DETECTOR IMPLEMENTATION

The Speech Detector stage analyzes the sampled input audio signal and discerns between high energy segments and background noise for selectively passing data to feature extraction.

WEIGHTING AND RANKING

The command scores are sorted in order of most probable match for each algorithm and a weight is applied to the ranked results. A weighting is applied to the ranked results and the composite weight for each potential match are again sorted by probability. The most probable match is asserted with a confidence value correlated to the composite weight and ‘perfect’ match score.

MODELING AND TEST RESULTS

MATLAB SYSTEM MODEL & TESTING

Test cases are obtained and tested through a MATLAB model and GUI developed by the team. Over 2500 recorded test utterances are run through 5 million individual qualitative and quantitative tests. Optimal extraction algorithm selection, match algorithms and parameters, as well as final weighting formulae are determined based on these results. Notable results:

- Weighting drastically improves averaged performance, from 53% to 77% total accuracy and enhances robustness to noise.
- Inclusion of the less accurate methods still increases overall system accuracy.

MATCHING

The Matching stage simultaneously computes several values of interest by comparing the set of extracted features from the spoken utterance to the stored features of known commands. Matching methods implemented include cumulative Euclidean error, array covariance, and cumulative delta rate. These match algorithm results are used as the basis for an utterance match score.

SYSTEM ARCHITECTURE

ENGININEERING

MODEL VS. HARDWARE SPEED

System testing using the video game Frogger as the external device. Five command words are used, with overall (speaker dependent) accuracy above 91%.

MODELING AND TEST RESULTS

Sample Speaker Dependent Results

MATCHING

The command scores are sorted in order of most probable match for each algorithm and a weight is applied to the sorted results. A weighting is applied to the ranked results and the composite weight for each potential match are again sorted by probability. The most probable match is asserted with a confidence value correlated to the composite weight and ‘perfect’ match score.

WEIGHTING AND RANKING

The command scores are sorted in order of most probable match for each algorithm and a weight is applied to the sorted results. A weighting is applied to the ranked results and the composite weight for each potential match are again sorted by probability. The most probable match is asserted with a confidence value correlated to the composite weight and ‘perfect’ match score.