Performance Trade-offs in Search Techniques for Isolated Word Speech Recognition

R. Bisiani, A. Waibel

Computer Science Department
Carnegie-Mellon University
Pittsburgh, PA 15213

Abstract

The cost effectiveness of various search methods used in experimental and practical discrete utterance speech recognition systems is a very critical factor for the usefulness of such systems. The advantages of some cost effective search techniques, e.g. branch and bound search, branch and bound search with pruning and beam search, have been previoulsy reported. In this paper we analyze the properties that affect the practical usefulness of these algorithms when task characteristics and machine architecture are considered.

1. Introduction

One of the problems that hinder the implementation of isolated word recognition systems based on dynamic time warping (DTW) is the amount of computation and memory required when vocabularies larger than a few hundred words are used. Although the basic algorithm [4] has linear complexity in the number of vocabulary words, the large number of instructions required to warp even one single word make it impractical to implement large vocabulary systems on general purpose architectures.

Many variations of the basic DTW algorithm have been suggested e.g. constraining the warp search space around the diagonal [8, 10]. These changes substantially improve the behavior of a DTW algorithm, independently of most of the characteristics of the implementation (e.g. the architecture; the speech features). A different kind of improvements reduces the search space by aborting a warp that fails to meet certain criteria, e.g. the rejection threshold presented in [9, 7]

This paper is concerned with evaluating the characteristics of two algorithms that reject unpromising candidates based on a locally computed heuristic function. DTW algorithms of this sort might not be necessary or even useful when highly parallel architectural solutions [1, 3] are available. On the other hand, parallel "brute force" solutions are still a few years away since "real world application" prototypes have yet to be disclosed. Therefore, the class of algorithms to which the two here described belong will still have a considerable importance in the next few years.

This research was sponsored in part by the National Science Foundation, Grant MCS-7825824 and in part by the Defense Advanced Research Projects Agency (DOD), ARPA Order No. 3597, monitored by the Air Force Avionics Laboratory Under Contract F33615-78-C-1551.

The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the US Government.

2. Description of the Algorithms

In this section we give a brief description of the two search methods evaluated in this paper. The first (BB) can be loosely referred to as a best-first or a modified branch-and-bound search strategy [12]. The second (BEAM) is a beam-search technique. Both search techniques are applied to dynamic time warping for an isolated word recognition system.

Both methods assume that time warping for recognition is done in parallel, i.e., that all references are available in memory and each frame in the test utterance is being matched against all reference tokens. Thus search proceeds in a breadth-first manner rather than in a depth-first manner as it is currently being done by most isolated word recognizers. The advantage of this is that a number of heuristics can be applied to improve the efficiency of the search.

The underlying idea of the first method, (BB) is to only expand the so far least expensive path, i.e., the warp which has the smallest cumulative distance score. This means that warping paths are grown depending on the "likelihood" of a particular match and badly matching tokens will naturally fall behind. When, during this process a particular warping path reaches the end the optimal path is found and the recognition is completed. We have reported elsewhere [11] that this method can save more than 30% of the run time while guaranteeing optimal recognition accuracy. The addition of a pruning threshold to prune off unpromising candidates (paths that fall behind the best path by a certain number of time frames) has also been shown to yield additional computational savings (>60%) at no loss in accuracy. This method is illustrated in Fig. 2-1. Fig. 2-1 depicts 5 warping planes viewed from the top. The heavy solid lines represent the warping paths in each warping plane as expanded so far. For the branch and bound method it can be seen that paths "grow" unevenly depending on the likelihood of the match. In addition to terminating the recognition run as soon as the best path arrives at the end (frame N of the test token) pruning provides increased efficiency, by removing from consideration references whose warping paths have fallen behind by a certain amount of frames. In this example, for instance, search for reference token $k = 2$ could be aborted since its path has substantially fallen behind the best matching candidate $k = 1$. (A more detailed discussion can be found in Waibel et al. [11]).

One alternate search method that uses a substantially different pruning technique is the beam search strategy (BEAM) that has succesfully been demonstrated to yield drastically improved run time performance for the HARPY-system. The beam search expands all the paths in parallel frame by frame (unlike BB that expands only the most promising path) and at the end of each

CH 1746-7/82/0000 · 0570 $ 00.75 © 1982 IEEE

expansion discards (prunes) all the paths that are worse than an heuristically computed "threshold". A detailed description of the general beam search strategy can be found in the literature [6]. The crucial factor for a beam search algorithm is the heuristic used in computing the threshold. The threshold used in our experiments defines the range ("beam") of allowable cumulative distances as a function of the difference between the best and worst partial scores. More elaborate thresholds have been evaluated but they showed no significant improvement in performance.

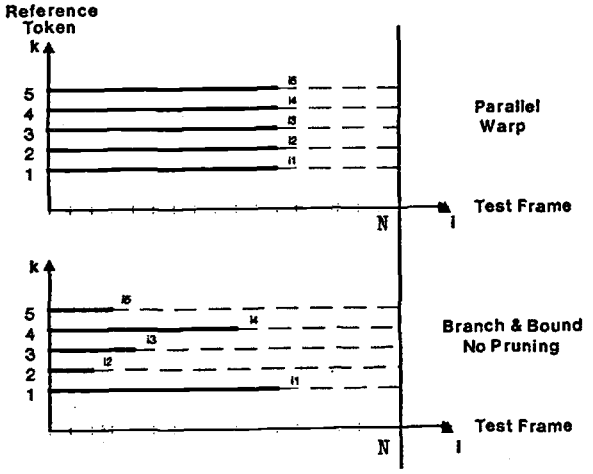

Figure 2-1: Expanding search paths in the modified branch and bound algorithm (BB).

As a test data base eight speakers (4 male, 4 female) have read the alpha-digit vocabulary (36 utterances) ten times. Five of these readings of each speaker were used to automatically select a reference template set [5]. The signal was sampled at 10kHz and parametrized into 15 noise-subtracted, differenced, log-dB spectral coefficients. All error rates were obtained for each speaker by running recognitions on the remaining 5 readings using an isolated word recognition system.

3. Anatomy of the Algorithms

The recognition error rate is influenced by all the techniques that increase performance by decreasing the number and the lengths of the paths examined. On the other hand, error rate is the most crucial performance measure for a recognition system and it should be kept as low as possible. We constrained our experiments on BB and BEAM so that the error rate would remain the same or diminish when compared with the "optimal" (from the point of view of accuracy) exhaustive research, i.e. the complete Itakura warp of all references with the unknown utterance.

In the following the performance of the two algorithms with respect to machine architecture, memory requirements and ruggedness will be reported.

3.1. Basic Performance

As a first measure of performance we plot in Fig. 3-1 the total number of points in the warping space (grid points) that are examined for the exhaustive search, BB and BEAM under the task conditions described in Section 2. Fig. 3-2 shows the run time (normalized to BB performance) as measured on a VAX-780 programmed in C with all the reference templates kept in core during the recognition. It is clear from the two curves that the decrease in number of grid points does not directly translate into an equal decrease in running time. As the number of grid points becomes smaller, the search management overhead cannot be ignored. Therefore, the exact relationship between search and search management (e.g. distance calculations and pruning) must be understood.

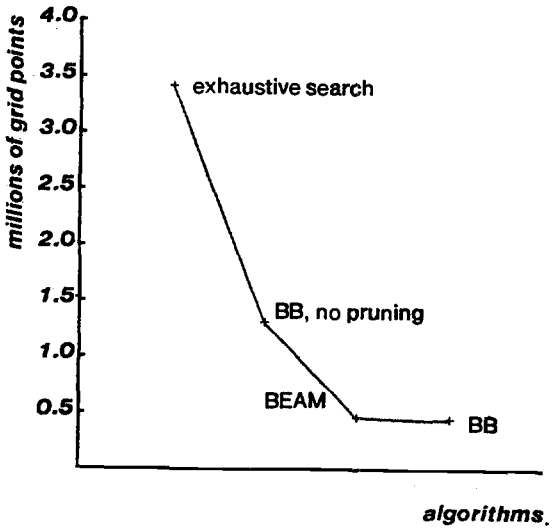

Figure 3-1: Number of grid points as a function of the algorithm, 36 word vocabulary.

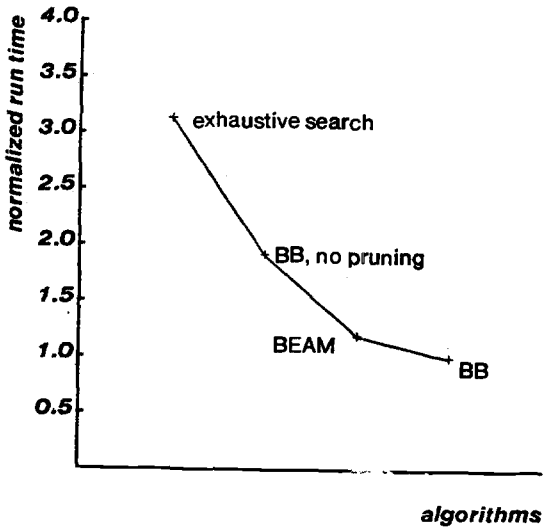

Figure 3-2: Run time as a function of the algorithm, 36 word vocabulary.

3.2. Computer Architecture Related Characteristics

Both the amount of computation and the data access patterns have to be considered when examining the behavior of an algorithm. First we will examine the data access pattern.

BB always computes the next score of the reference that has the currently best warping score. After a frame is processed (i.e. distances and scores are computed) the algorithm selects the next reference to be expanded. The selected reference depends on the characteristics of the input and cannot be predicted at the time of the implementation. Hence, an efficient implementation of

BB must have instantaneous access to all the references or, at least, to the "next frame to be expanded" in all the references. This is one of the major limitations of BB if one tries to apply it to large vocabularies. Assuming an average word length of 50 frames, 9 numbers per frame and a 40 word vocabulary the memory will have to hold 1800 numbers. This amount of memory is today very inexpensive and uses very little area on a card. On the other hand, larger vocabularies that are still within the capabilities of DTW algorithms (e.g. 500 words) would make the implementation of BB too expensive for a production system. Slow secondary memories as disks are not appropriate for BB in the present form because of the randomness of the access pattern.

BEAM also examines all the references in parallel but its access pattern is fixed because for each given frame all references are examined (unless they have been pruned). Therefore, slower memories like disks can be used more easily by prefetching the required frames of data.

Internal storage for the warping computation is not a problem with either algorithm: assuming that an adjustment window of 100 milliseconds is used to constrain the allowable warping paths, the memory needed to perform the Itakura warp has to contain two distance arrays of 10 entries each for each reference (i.e., 800 bytes for a 40 word vocabulary).

Assuming that data is available at the right time, let us now take a look at the characteristics of the computation. We can divide both algorithms into three subtasks that are different from the point of view of the "architectural resources" they need. These subtasks are distance calculation, score calculation and management of the search.

The distance calculation applies the distance metric to pairs of frames. This subtask has a very simple control structure and requires (depending on the metric used) the efficient execution of arithmetic operations. The score calculation computes the score for each grid point as a function of previous legal scores and distances. This subtask needs a balanced amount of control and arithmetic statements. BB and BEAM have a different search management. BB examines the status of the search of all references and decides which path will be expanded next. BEAM examines all paths and prunes the paths that seem unlikely to become the optimal path, then it expands all the remaining paths by one frame.

Both algorithms take about the same amount of resources to execute the search management subtask. BB executes this subtask after each expansion of a reference, while BEAM does it only after expanding all the active references by one frame. Therefore, BEAM unlike BB, executes this subtask as many times as there are frames in the test utterance, independently of the number of words in the vocabulary or the behavior of the search.

Table 1 shows the percentage of computation (when measured in instructions per second of speech) for the three subtasks and the two algorithms with vocabulary size 36 under the conditions explained in Section 2.

Table 1	Distance Calculations	Score Calculations	Search Management
BB	43%	42%	15%
BEAM	49%	49%	2%

We see that BEAM spends a smaller percentage of time in doing search management and this partially counterbalances the higher number of grid points it has to search. As the vocabulary size increases, the effect of search managements gets bigger and bigger because the number of grid points the algorithms go through also increases. Let us ignore for a moment the effect that a larger vocabulary might have on the behavior of the search, e.g. a smaller percentage of references might be pruned because the confusability of the vocabulary increases. If we simply assume that the amount of grid points will increase linearly with the vocabulary size, the amount of computation required by the two algorithms can be computed by analyzing and instrumenting the code as described in [2].

Fig. 3-3 shows the relationship between the number of words and the amount of computation (number of instructions) required. Although the absolute values are not reported because they are a function of the architecture, the relative performance of the algorithms is indicative of how any general purpose architecture would behave. The slope of the two curves indicates that BEAM will be more and more profitable when vocabulary size increases. Slight variations in the slopes of the curves in Fig.4 can be expected as vocabularies of different confusability are used.

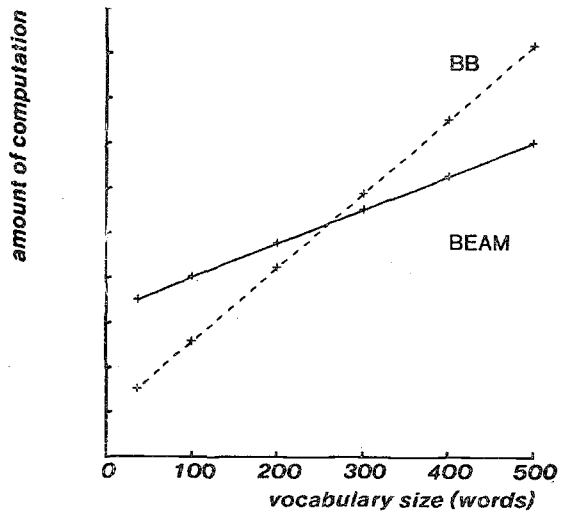

Figure 3-3: Amount of computation as a function of the vocabulary size

3.3. Task Related Characteristics

Speed and memory size characteristics can be sometime less important than other characteristics that make one or the other algorithm impractical. First of all we should consider the problem of tuning the search on the characteristics of the features and distance measures used. BB has the advantage of being rather insensitive to modifications of speech parameterization since the distance scores are not directly involved in directing the search. In contrast, the performance of BEAM is dependent on the characteristics of the score computation since pruning thresholds are directly related to the numerical values of the scores. Similar behavior is exhibited by other algorithms in the literature e.g. the rejection threshold evaluated in [7, 9]. It is doubtful that any algorithm that does pruning should be used when investigating feature extraction procedures and metric, since pruning can have unpredictable effects on accuracy that in turn can mask the effects of changing other parts of the system. BB can be useful in

this context because BB without pruning is still faster than the exhaustive search while retaining the capability of always finding the best matching reference template.

Finally, the experiments we have performed on these two algorithms showed that BB could very efficiently cope with localized errors in the input data. We have observed high localized errors in all systems that have been developed at Carnegie-Mellon University, independently of the kind of front end processing done. Localized errors, due to begin-end detection problems, DTW alignment problems, noise, etc. can cause the correct reference template to have very poor partial score for several consecutive frames. This makes any algorithm that prunes on the basis of partial scores, e.g. BEAM, difficult to tune and less effective when conservative thresholds have to be used to account for these variations.

4. Conclusions

In this paper we have analyzed the performance trade-offs for two search techniques as applied to isolated word recognition: beam search and a modified branch and bound technique. They have in our experience proven to yield good results with respect to recognition accuracy and run time efficiency. Due to the trade-offs between the cost of searching and search management, beam search is better when many alternatives need to be investigated (large vocabularies). On the other side, for vocabularies in the order of about 100 isolated words, the BB method offers better performance in addition to being more robust to system modifications.

References

1. B. Ackland, N. Weste and D.J.Burr. An Integrated Multiprocessing Array for Time Warp Pattern Matching. The 8th Annual Symposium on Computer Architecture, ACM SIGARCH, April, 1981, pp. 193-203.

2. R. Bisiani. The Harpy Machine: A Data Structure Architecture. 5th Workshop on Computer Architecture for Non-Numeric Processing, ACM SIGARCH, SIGIR and SIGMOD, 1980.

3. D.J.Burr, B. Ackland, N.Weste. A High Speed Array computer for Dynamic Time Warping. 1981 IEEE International Conference on Acoustics, Speech and Signal Processing, IEEE ASSP, April, 1981, pp. 471-474.

4. F.Itakura. "Minimum Prediction Residual Principle Applied to Speech Recognition." *IEEE Transactions on Acoustics, Speech, Signal Processing ASSP-23*, 1 (February 1975), 67-72.

5. Z.Li, F.Alleva, R.Reddy. Frame Compression in Isolated Word Recognition. Tech. Rept. CMU-CS-81-134, Carnegie-Mellon University, Department of Computer Science, 1981.

6. B.T. Lowerre and R.D. Reddy. The Harpy Speech Understanding System. In Wayne A. Lea, Ed., *Trends in Speech Recognition*, Prentice-Hall, 1979.

7. C.Myers, L.R.Rabiner, A.E.Rosenberg. "Performance Tradeoffs in Dynamic Time Warping Algorithms for Isolated Word Recognition." *TASSP ASSP-28*, 6 (December 1980).

8. L.R.Rabiner, A.E.Rosenberg, S.E.Levinson. "Considerations in Dynamic Time Warping Algorithms for Discrete Word Recognition." *IEEE Transactions on Acoustics, Speech, Signal Processing ASSP-26*, 6 (December 1978), 575-582.

9. L.R.Rabiner,S.E.Levinson,A.E.Rosenberg,J.G.Wilpon. "Speaker-Independent Recognition of Isolated Words Using Clustering Techniques." *IEEE Transactions on Acoustics, Speech, Signal Processing ASSP-27*, 4 (August 1979), 336-349.

10. A.Waibel, B.Yegnanarayana. Optimization of Nonlinear Time Warping Techniques in Isolated Word Recognition Systems. Tech. Rept. CMU-CS-81-125, Carnegie-Mellon University, Department of Computer Science, 1981.

11. A. Waibel, N.Krishnana, R.Reddy. Minimizing Computational Cost for Dynamic Programming Algorithms. Tech. Rept. CMU-CS-81-124, Computer Science Dept. Carnegie- Mellon University, June 1981.

12. P. H. Winston. *Artificial Intelligence*. Addison-Wesley Publishing Company, 1977.